Light-Weight Spatial Distribution Embedding of Adjacent Features for Image Search

Yan Zhang[1,2], Yao Zhao[1,2], Shikui Wei[3(✉)], and Zhenfeng Zhu[1,2]

[1] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[2] Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China
[3] Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, Hubei, China
shkwei@bjtu.edu.cn

Abstract. Binary code embedding methods can effectively compensate the quantization error of bag-of-words (BoW) model and remarkably improve the image search performance. However, the existing embedding schemes commonly generate binary code by projecting local feature from original feature space into a compact binary space. The spatial relationship between the local feature and its neighbors are ignored. In this paper, we proposed two light-weight binary code embedding schemes, named content similarity embedding (CSE) and scale similarity embedding (SSE), to better balance the image search performance and resource cost. Specially, the spatial distribution information for any local feature and its nearest neighbors are encoded into only several bits, which are used to verify the asserted matches of local features. The experimental results show that the proposed image search scheme achieves a better balance between image search performance and resource usage (i.e., time cost and memory usage).

Keywords: Image search · Product quantization · Embedding · Bow

1 Introduction

Content-based image search is the core technique for many real-world visual applications, such as frame fusion based video copy detection [1], logo detection [2], visual content recognition [3]. However, image search remains a challenge due to the deviation of semantic understanding between human and computer, and the appearance variations in scale, orientation, illuminations, etc. [4]. In consideration of the robustness and effectiveness of local visual features, the image searching frameworks based on local features are commonly employed in both research and industrial areas. Local features like SIFT [5], SURF [6], etc., are originally proposed for image matching, which are generally invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion. Nevertheless, original matching schemes between

H. Zha et al. (Eds.): CCCV 2015, Part I, CCIS 546, pp. 387–397, 2015.
DOI: 10.1007/978-3-662-48558-3_39

two images are generally based on the similarity measurement of local feature sets, which requires large cost in both computation and storage.

To facilitate the image search with large scale image datasets, pioneering scheme, named Bag-of-Words (BoWs) model [7], is proposed for significantly simplifying the matching process. The key idea of BoW is to quantize each local feature into one or several so-called visual words, and represent each image as a collection of orderless visual words. After mapping local features into visual words, lots of excellent techniques in text retrieval area can be directly employed, which makes it possible to represent, index, and retrieve images like text documents. Although the BoW model shows impressive performance in both image search accuracy and time cost, it suffers greatly from visual word ambiguity and quantization error, i.e., representing a high dimensional descriptor with a visual word results in large information loss. It is possible for BoW model to quantize totally different local features into the same visual word when the visual dictionary is small. This will unavoidably cause false matches and decrease image search accuracy. A straightforward solution to this problem is to build a large scale visual dictionary [8]. However, it is not easy for the traditional methods like k-mean clustering to build a large scale visual dictionary due to their high computing cost. In addition, the performance improvement will trend smoothing when dictionary size is large enough. Another solution is to build a compact binary code for each local feature [4]. In this way, each local feature is associated with one visual word and one binary code. Since the binary code can greatly filter out false matches, the image search accuracy is remarkably improved. More significantly, the searching time cost is greatly reduced. However, this kind of embedding method separately build a binary code for each local feature, yet the spatial relationships among local features are not taken into account. Therefore, these methods severely limit the discriminative power of binary codes. To address this issue, multiple visual phrase (MVP) [9] is proposed in recent years. Instead of visual word, a multiple visual phrase is used to represent each local feature by exploring the spatial distribution of local features. In fact, the existing MVP scheme will result in large computing cost and memory usage since the MVP of each local feature is associated with several adjacent local features and their correlation information.

In this paper, our key goal is to design a light-weight image search framework, which can achieves a better balance between search performance and resource cost. This framework consists two key components, i.e., large-scale visual dictionary construction and light-weight binary code embedding. To constructing a large-scale visual dictionary, the optimal product quantization (OPQ) [10] is employed. Since a large-scale visual dictionary is built by Cartesian product of a set of small sub-dictionaries, the visual word assignment is extremely efficient and the memory usage for storing the dictionary is much less than traditional ones. To fully explore the spatial information among local features but reduce computational cost and memory usage, we propose two light-weight binary code embedding schemes, named content similarity embedding(CSE) and scale similarity embedding(SSE). Since spatial clues among local features are encoded into only several bits, the memory usage is much less than the existing schemes. The experimental results show the proposed image search schemes achieve a better balance between image search performance and resource usage (i.e., time cost and memory usage).

2 Related Work

Local descriptors like SIFT provide robust matching across a substantial range of affine distortion. But matching local descriptors is time-consuming because each image may contain thousands of high-dimensional descriptors. An efficient solution is to quantize local descriptors into visual words [7]. However, due to the quantization error, many false matches occur when matching two images, which unavoidably decrease the image search accuracy. To improve the discriminative power of BoW model, one solution is to build a compact binary code for each local feature [4]. Although embedding methods can filter out false matches, it has been illustrated that single visual word cannot preserve the spatial information in images, which has been proven important for visual matching.

To introduce spatial information into BoW model, lots of works are conducted to combine multiple visual words with spatial information. For example, descriptive visual phrase (DVP) is generated in [12] by selecting two nearby visual words. Generally, considering visual words in groups rather than single visual word captures stronger discriminative power. However, in the existing visual phrase methods, two visual phrases are matched only if they have the same number of visual words, which reduces the match flexibility. In [13], Geometry preserving Visual Phrase(GVP) is proposed to encode the spatial information of local features, including both co-occurrences and the local and long-range spatial layouts of visual words. With little increase in memory usage and computational time, the improvement of search accuracy is witnessed. However, GVP only captures the translation invariance. Although its extension to scale and rotation invariance can be achieved by increasing dimension of the offset space, more memory usage and time cost will be needed. Spatial coding [14] is proposed to efficiently check spatial consistency globally. It uses spatial maps to record the spatial relationship of all matched feature pairs. Nevertheless, spatial coding is very sensitive to rotation due to the intrinsic limitation of the spatial map. Zhang etc. in [9] proposed a multi-order visual phrase (MVP) which contains two complementary clues: center visual word quantized from the local descriptor and the visual and spatial clues of multiple nearby local features. This method shows an impressive performance improvement in image search accuracy. However, it needs large memory to store the MVP information and more computational time in online searching phase. In [15], a novel geometric relation which computes a binary signature leveraging existence and nonexistence of interest points in the neighborhood area was proposed. But it only consider the adjacent area instead of adjacent features which will results in a lot of mismatch.

To address abovementioned issues, we design a light-weight image search framework and propose two simple but effective binary embedding schemes. By encoding spatial distribution information among local features into several bits, the proposed image search framework can better balance the image search accuracy and resource cost.

3 The Proposed Approach

In this section, the proposed light-weight image search scheme is discussed in details. The overall framework includes four key components, i.e., large-scale visual dictionary construction, light-weight binary code embedding, image database indexing, and similarity querying. In this paper, we focus on the first two tasks. To give a complete discussion about image search scheme, the implementation details about indexing and querying are also presented in this section.

3.1 Large-Scale Visual Dictionary Construction

Our final goal is to reduce the time complexity and memory usage while remaining comparable image search accuracy. To this end, we firstly need to generate a large-scale visual dictionary for reducing quantization error. In fact, most of existing methods generate visual dictionary by clustering a large training set of local features in their original feature space. In this way, it will lead to an intractable computation cost when training a large-scale visual dictionary. More importantly, the memory usage for storing the dictionary itself is not trivial. To avoid these issues, we employ partitioned k-means clustering (or Product Quantization) to build a large-scale visual dictionary. For further reducing the quantization error, an optimal step is used to improve the product quantization method. Here, a short review about product quantization is presented as follows:

Product Quantization

Product quantization is an extremely efficient vector quantization approach, which can compactly encode high dimensional vectors for fast approximate nearest neighbor (ANN) search. Product quantization involves two key steps: (1) decomposing the *D*-dimensional vector space into *S* subspaces; (2) computing a sub-dictionary for each subspace.

For the original *D*-dimensional representation space of local features X^D, it is first divided into *S* subspaces with dimensions *D/S.* In each subspace $X^m, s \in \{1,2,\cdots S\}$, a small sub-dictionary D^m is built. The objective function of product quantization is as follows:

$$\min_{D^1,D^2,\ldots\ldots D^S} \sum_X ||x - d(i(x))||^2 \tag{1}$$

$$d \in D = D^1 \times D^2 \times \ldots\ldots \times D^S$$

Here, $x = [x^1, x^2, \ldots\ldots, x^S]$ is any training sample, the function $i(\cdot)$ is called an *encoder*, and function $d(\cdot)$ is called a *decoder*, $d(i(x))$ is the visual word of x. $D = D^1 \times D^2 \times \ldots\ldots \times D^S$ is the final visual dictionary constructed by Cartesian product of M sub-dictionaries $\{D^1, D^2, \cdots D^S\}$.

Optimized Product Quantization

To further reduce the quantization error, T. Ge etc. in [10] introduce an iterative optimization step into product quantization. Specially, an orthonormal matrix R is introduced into the space decomposition process. For each iteration, the D-dimensional vector space is first transformed by R before decomposing it into S subspaces. Therefore, the visual dictionary constructing process can be split into two iterative steps. First, an orthogonal matrix R is initialized and fixed, and sub-dictionaries $\{D^s\}_{s=1}^{S}$ are calculated by following the objective function (1). Then, the currently generated sub-dictionaries $\{D^s\}_{s=1}^{S}$ are fixed, and the orthogonal matrix R is optimized. The optimized product quantization can be formulated as follows:

$$min_R \sum_X \left\| RX - c\left(i(X)\right) \right\|^2 \tag{2}$$

where X is the set of training samples, and $c\left(i(X)\right)$ is the visual word of X.

For the product quantization scheme, the problem of building a large-scale visual dictionary is transferred into build a series of small visual sub-dictionaries. Since the memory usage for storing these visual sub-dictionaries is trivial, the memory usage is less than traditional ones. In our work, the value of S is set to 2, and the value of k is set to 1000. Therefore, we finally get a large-scale dictionary with 1M visual words.

3.2 Light-Weight Binary Code Embedding

Although a large-scale visual dictionary can remarkably alleviate the problem of quantization error, additional binary code can still improve the image search accuracy furthermore. Therefore, this paper focuses mainly on the design of binary code embedding schemes. Two light-weight binary code embedding schemes, named content similarity embedding and scale similarity embedding, are proposed. The proposed scheme can be treated as light-weight MVP. For the original MVP scheme, both the nearest neighbors of current local features and their spatial relationship are recorded when indexing images, which will result in a big indexing structure. Instead, the proposed schemes encode the spatial relationship into light-weight binary codes. In this way, the memory usage will be greatly reduced. In addition, the proposed method is also different from traditional embedding schemes. For the existing embedding schemes like Hamming embedding, the binary codes are generated by projecting local features from the original representation space into a binary space. Generally, the generated code is treated as a compact version of original local feature. In contrast, the proposed schemes only encode spatial distribution information surrounding the local feature. The implementation details are discussed as follows:

Content Similarity Embedding (CSE)

The content similarity embedding is based on an underlying assumption that the spatial distribution with the nearest N neighbors in content is similar for two matched local features. Therefore, we can verify the asserted matches of local features by comparing their spatial clues with the nearest N neighbors in content similarity.

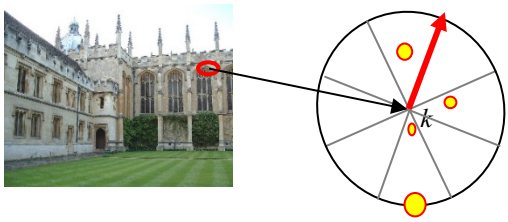

Fig. 1. Illustration of CSE scheme, where red line indicates the dominant orientation of *k*

To encode the spatial clue for any local feature k in image, we first find out the nearest N local features surrounding k as shown in Fig.1. Yellow dots indicates the neighborhood interest points with different scales. Then, the region containing the nearest N neighbors are divided into 8 equal portions $\{P^i, i = 1 \dots \dots 8\}$ started from the dominant orientation of current local feature. Finally, a binary code $b_k = (b_k^1, \cdots, b_k^i, \cdots, b_k^8)$ is generated by encoding occurrence of neighbors in each portion in counterclockwise, which is formulated as follows:

$$b_k^i = \begin{cases} 1, & if\ one\ of\ \ N\ neighbors\ falls\ into\ P^i \\ 0, & othewise \end{cases} \tag{3}$$

In Fig.1, the binary code for local feature *k* is $b_k = 10001010$.

Scale Similarity Embedding (SSE)

For the content similarity embedding, the nearest *N* neighbors are selected by computing the content similarity of local features. To capture the scale invariance surrounding each local feature, we propose a scale similarity embedding scheme. The key assumption is that the spatial distribution with the nearest *N* neighbors in scale is similar for two matched local features.

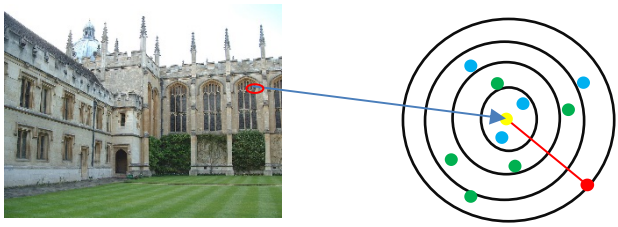

Fig. 2. Illustration of SSE scheme. Red dot is the M^{th} nearest neighbor in content similarity, and blue dots indicate 4 nearest neighbors in scale similarity.

Similarly, for any local feature *k* in image, we first select *M* nearest neighbors in content for it. As shown in Fig. 2, we select *M* nearest neighbors, which include both green and blue points. Then, 4 local features (blue points in Fig.2) whose scales are

closest to the scale of k are selected from the M nearest neighbors. According to the distance between k and its M^{th} neighbor, we make a circular region as k's neighborhood region, which is partitioned into a set of patches $\{P^i, i = 1, \cdots, 4\}$. By encoding occurrence of scale-similar neighbors in patches, we can generate a compact binary code, which is formulated as follows:

$$b_k^i = \begin{cases} 1, \text{if any nerghbor keypoint exiet in } P^i \\ 0, \qquad\qquad\qquad\quad \text{othewise} \end{cases} \tag{3}$$

In Fig.2, the binary code for local feature k is $b_k = 1011$.

3.3 Indexing and Retrieval

Indexing

To speed up the image searching process, we also build an inverted table for the image database against the pre-trained dictionary **D**. For each local feature in database image, it is first quantized into the nearest visual word w_k, and then an item is inserted into the list associated with the visual word w_k. As shown in Fig.3, both ID of image and spatial embedding code (SEC) are contained in the item.

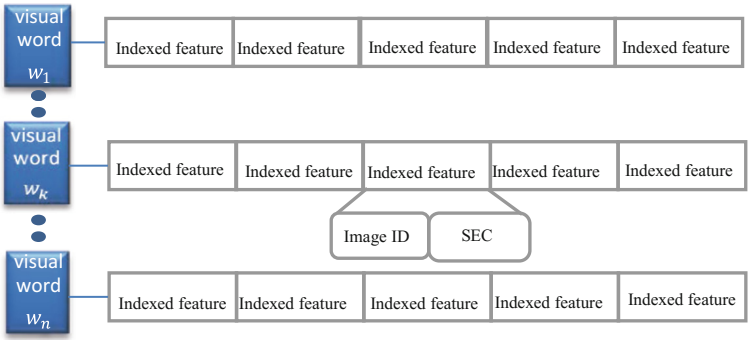

Fig. 3. Illustration of the inverted indexing structure with spatial embedding codes

Retrieval

Given a query image Q, we also quantize the local features extracted Q into visual words against the pre-trained dictionary D, and generate the corresponding binary codes of these local features. For each visual word in Q, the corresponding lists in inverted table are returned. By a voting procedure, we can get a similarity score for each potentially matched image. In particular, we use match order computed by spatial verification to measure the importance of matched local features to image similarity ,i.e., high order matches between query Q and a database image D are more important for image similarity. The image similarity can be formulated as follows:

$$score(Q,D) = \frac{\sum_{w_q=w_d} idf(w_q)\times(1+\alpha)^{order(b_q,b_d)}}{norm(D)} \quad (4)$$

where, w_q and w_d denote the visual words in query image Q and database image D, respectively; b_q and b_d are corresponding special binary codes associated with w_q and w_d; $idf(w_q)$ is the inverse document frequency of visual word w_q, and $norm(D)$ is L2-norm of term frequency vector of database image *D*. $order(b_q, b_d)$ is the order between b_q and b_d, which is defined as follows:

$$order(b_q, b_d) = L - hd(b_q, b_d) \quad (5)$$

where, *L* is the length of the binary code, *L* is set to 8 and 4 in CSE and SSE, respectively, *hd* represents the Hamming distance of b_q and b_d.

4 Experiments

4.1 Datasets

We test the proposed methods on two commonly used datasets: UKbench dataset [16] and Oxford5K dataset [17]. All experiments are conducted under the same configuration conditions on a PC with a 2-core 3.2Ghz processor and 8GB memory.

UKbench. UKbench dataset, contains 2,550 objects, each of which has 4 images under 4 different viewpoints for object search. So the total number of images in this dataset is 10200. In this experiment, all of the 10200 images are used as database images and queries either. And we measured the retrieval performance by the top-four candidates, which means how many similar images are returned at the first four images, called N-S score.

Oxford5K. For landmark search, we use the Oxford5K dataset, which contains 5,062 annotated landmark images. The collection has 11 different landmark categories and 5 queries for each category. The performance is measured by mean Average Precision (MAP).

4.2 Evaluation on Image Search Performance

In our experiments, the popular SIFT descriptor is employed as local feature. We decompose the 128-dimensional SIFT vector space into 2 subspaces, and then train 2 sub-dictionaries with 1000 visual sub-words in two subspaces. Therefore, we obtain the final large-scale visual dictionary with 1M visual words.

Effect of Key Parameters

For the content similarity embedding (CSE) method, we need extract the nearest *N* neighbors for each local feature as spatial clues. Likewise, for scale similarity embedding

(SSE) method, the radius r_k is the distance between the current local feature k and its M^{th} nearest neighbor. Therefore, the selection of N and M will affect the experimental results. In addition, the selection of α in Eq.4 will affect both CSE and SSE schemes. To evaluate the effect of different parameters, we carry out some experiments on UKbench dataset by employing different value combinations of N、M and α. The experimental results are illustrated in Fig.4. Clearly, for both CSE and SSE schemes, the number of nearest neighbors have remarkable effect on image search accuracy. When the number of N and M increase from zero to a certain value, the search accuracy trend to the best. However, this conclusion is reasonable. When the value N and M is too small, the nearest neighbors cannot provide enough spatial clues. At other extreme, when the value N and M is too large, the reliability of spatial clues will lose. Therefore, both extreme cases will lead to bad performance. Similar conclusion can be obtained for the parameter α, yet the reason is different. As indicated in Eq.4, larger α puts larger weight to spatial verification. There is also two extreme cases. When α is too small, the effect of spatial verification is trivial on image similarity calculation. On the contrary, if α is too large, spatial codes will dominate the image similarity calculation. Both extreme cases will result in bad performance.

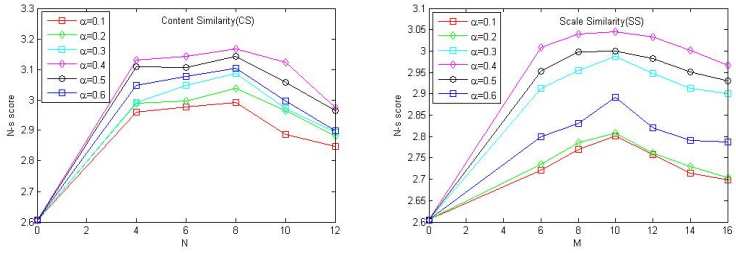

Fig. 4. The influence of N、M and α on Ukbench dataset

In the following experiments, the values of N、M and α are fixed to 8, 10 and 0.4 respectively.

Evaluation of Image Search Performance

Our final goal is to design a light-weight image search framework so as to better balance the search performance and resource cost. In this section, the proposed methods are compared with existing schemes under the same conditions. The experimental results are demonstrated in Table.1 and Table.2. Clearly, introducing embedding codes into the original OPQ BoW model remarkably improves the image search performance in terms of effectiveness. For the different BoW+ Embedding schemes, the proposed OPQ+CSE scheme achieves the best search accuracy on Ukbench dataset and comparable accuracy on Oxford5K dataset. For the time cost and memory usage in querying phrase, the proposed schemes, i.e., OPQ+CSE, OPQ+SSE, OPQ+CSE+SSE, outperform the OPQ+MVP scheme. That's because that MVP needs 64 bits to

preserve the spatial and visual clues for each local features [9], meanwhile the proposed schemes only need at most 12bits (CSE+SSE). And to verify the neighbor keypoints of two MVPs, at most conduct $4 \times 4 = 16$ times of verification, at the same time we only need to calculate the hamming distance for one time. It means that the proposed methods indeed better balance the image search accuracy and resource cost. Notice that the image search accuracy have a remarkable degeneration when CSE and SSE are combined. The possible reason is that a long binary code possibly leads to large order in Eq.5. Therefore, spatial codes will dominate the image similarity calculation which result in bad image performance.

Table 1. Retrieval performance on Ukbench for 10200 query images

	OPQ	OPQ+MVP	OPQ+CSE	OPQ+SSE	OPQ+CSE+SSE
Performance(N-S)	2.6050	3.1463	3.1657	3.0443	3.0749
Index time(s)	1056.5	3081.8	3453.1	1964.4	4151.8
Search time(s)	0.1432	0.7612	0.2369	0.2123	0.3068
Index storage(MB)	28.5	217.0	53	42.8	61.6

Table 2. Retrieval performance on Oxford5K for 55 query images

	OPQ	OPQ+MVP	OPQ+CSE	OPQ+SSE	OPQ+CSE+SSE
Performance(MAP)	0.2475	0.5172	0.4921	0.4683	0.4027
Index time(s)	901.1	4986.7	4625.6	2821.4	6555.7
Search time(s)	0.2279	1.6751	0.7650	0.7434	0.8139
Index storage(MB)	33	306.0	55.9	50.2	61.6

5 Conclusion

In this paper, we design a light-weight image search framework to better balance the image search performance and resource cost. Instead of extracting a compact binary code from local feature itself, the proposed binary code embedding schemes only encode the spatial distribution information of nearest neighbors surrounding the current local feature. Besides content similarity, the scale similarity is also employed to select neighbors. Since spatial distributions among local features are encoded into only several bits, both the memory usage and time cost are much less than existing schemes. The experimental results show the proposed image search scheme achieves a better balance between image search performance and resource usage (i.e., time cost and memory usage).

Acknowledgements. This work was supported in part by National Basic Research Program of China (No.2012CB316400), National Natural Science Foundation of China (No.61202241, No.61210006), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT201206), Fundamental Research Funds for the Central Universities (No.2015JBM028), and Joint Fund of Ministry of Education of China and China Mobile (No.MCM20130421).

References

1. Wei, S.K., Zhao, Y., Zhu, C., Xu, C.S., Zhu, Z.F.: Frame Fusion for Video Copy Detection. IEEE Transactions on Circuits and Systems for Video Technology **21**(1), January 2011
2. Yan, W.Q., Wang, J., Kankanhalli, M.S.: Automatic Video Logo Detection and Removal. Multimedia Systems **10**(5), 379–391 (2005)
3. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)
4. Wei, S.K., Xu, D., Li, X., Zhao, Y.: Joint Optimization Toward Effective and Efficient Image Search. IEEE Transactions on Cybernetics **43**(6), December 2013
5. Lowe, D.G.: Distinctive Image Features from Scale Invariant Keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
6. Bay, H., Tuytelaars, T., Gool, L.V.: Speeded-up Robust Features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
7. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: Proc. IEEE Int. Conf. Comput. Vision, vol. 2, pp. 1470–1477 (2003)
8. Jegou, H., Douze, M., Schmid, C.: Product Quantization for Nearest Neighbor Search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 117–128 (2011)
9. Zhang, S.L., Tian, Q., Huang, Q.M., Gao, W., Rui, Y.: Multi-order visual phrase for scalable image search. In: ICIMCS 2013, August 17–19, 2013
10. Ge, T.Z., He, K.M., Ke, Q.F., Sun, J.: Optimized Product Quantization. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2014)
11. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
12. Zhang, S., Tian, Q., Hua, G., Huang, Q., Li, S.: Descriptive visual words and visual phrases for image applications. In: ACM Multimedia, pp. 75–84 (2009)
13. Zhang, Y., Jia, Z., Chen, T.: Image retrieval with geometry-preserving visual phrases. In: Proc. IEEE Conf. Compute. Vis. Pattern Recognit., pp. 809–816 (2011)
14. Zhou, W., Lu, Y., Li, H., Song, Y., Tian, Q.: Spatial coding for large scale partial-duplicate web image search. In: Proceedings of the ACM International Conference on Multimedia, pp. 511–520 (2010)
15. Ozkan, S., Esen, E., Akar, G.B.: Visual group binary signature for video copy detection. In: International Conference on Pattern Recognition (ICPR), August 2014
16. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
17. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)