HIERARCHICAL MULTI-VLAD FOR IMAGE RETRIEVAL

Yitong Wang[⋆†], *Ling-Yu Duan*[⋆†*], *Jie Lin*[‡], *Zhe Wang*[⋆†], *Tiejun Huang*[⋆†]

[⋆] Institute of Digital Media, School of EE&CS, Peking University, Beijing, China
[†] Cooperative Medianet Innovation Center, Shanghai, China
[‡] Institute for Infocomm Research, Singapore
e-mail: [⋆†]{wangyitong, lingyu, zwang, tjhuang}@pku.edu.cn, [‡]lin-j@i2r.a-star.edu.sg

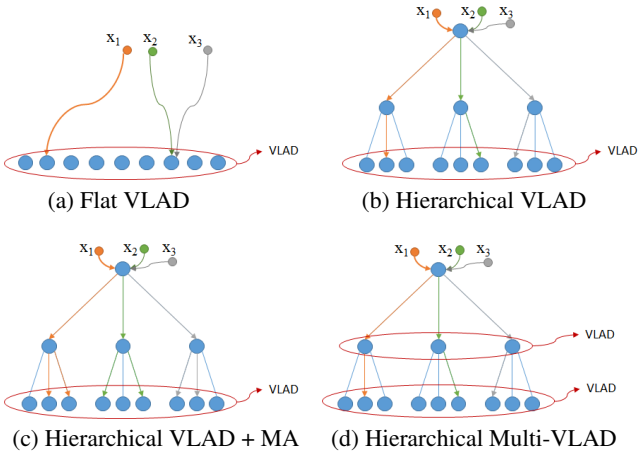
ABSTRACT

Constructing discriminative feature descriptors is crucial towards effective image retrieval. The state-of-the-art powerful global descriptor for this purpose is Vector of Locally Aggregated Descriptors (VLAD). Given a set of local features (say, SIFT) extracted from an image, the VLAD is generated by quantizing local features with a small visual vocabulary (64 to 512 centroids), aggregating the residual statistics of quantized features for each centroid and concatenating the aggregated residual vectors from each centroid. One can increase the search accuracy by increasing the size of vocabulary (from hundreds to hundreds of thousands), which, however, it leads to heavy computation cost with flat quantization. In this paper, we propose a hierarchical multi-VLAD to seek the tradeoff between descriptor discriminability and computation complexity. We build up a tree-structured hierarchical quantization (TSHQ) to accelerate the VLAD computation with a large vocabulary. As quantization error may propagate from root to leaf node (centroid) with TSHQ, we introduce multi-VLAD, which constructing a VLAD descriptor for each level of the vocabulary tree, so as to compensate for the quantization error at that level. Extensive evaluation over benchmark datasets has shown that the proposed approach outperforms state-of-the-art in terms of retrieval accuracy, fast extraction, as well as light memory cost.

Index Terms— Image Retrieval, Hierarchical Quantization, Multi-VLAD

1. INTRODUCTION

Image retrieval regards the discovery of images contained within a large database that depict the same objects/scenes as those depicted by query images. In general, state-of-the-art image search systems are built upon a visual vocabulary with an inverted indexing structure, which quantizes local features (e.g., SIFT [1] or SURF [2]) of query and database images into centroids. Each database image is then represented as a Bag-of-Words (BoW) histogram [3] and is invert indexed by quantized centroids of local features in the image. Recently, the Vector of Locally Aggregated Descriptors (VLAD) [4][5] and Fisher vectors [6] have extended the BoW by aggregating higher-order statistics of the distribution of local features. The VLAD is generated by quantizing the set of local features with a small vocabulary (64 to 512 centroids), aggregating the residual statistics of features quantized to each centroid and concatenating the residual vector from each centroid. Compared to the BoW with a large vocabulary (e.g., 1 million centroids), the VLAD have achieved the state-of-the-art search performance at a much smaller vocabulary [4][5][7].



(a) Flat VLAD (b) Hierarchical VLAD

(c) Hierarchical VLAD + MA (d) Hierarchical Multi-VLAD

Fig. 1. Several VLAD schemes. (a) Flat VLAD. (b) Hierarchical VLAD. (c) Hierarchical VLAD with Multi-assignment. (d) Hierarchical Multi-VLAD. x_1, x_2, x_3 refer to local features quantized to centroids.

Recent works have proposed to improve the VLAD representation by enhancing the residual statistics in the aggregation stage after quantizing local features. For example, Tolias et al. [8] introduced an aggregation approach that achieves orientation covariance of residual statistics. Jégou et al. [9] presented democratic aggregation to limit the interaction of unrelated local features in generating the residual vectors. Arandjelovic et al. [10] proposed intra normalization of residual vectors to suppress bursty visual elements. These improved aggregation strategies have shown promising results. However, the descriptor discriminability of VLAD is yet limited by the coarse quantization due to a small vocabulary.

An alternative solution is to directly increase the vocabulary size, as a large vocabulary usually provides fine-grained partition of feature space and improves the discriminability of centroids. For instance, Tolias et al. [11] formulated image retrieval as a match kernel framework and used a large vocabulary trained with flat k-means, leading to state-of-the-art search accuracy. Unfortunately, the computation cost with flat quantization is linearly increased with the vocabulary size. This usually leads to slower VLAD extraction.

In this paper, we propose a hierarchical multi-VLAD to address the problem of image retrieval using the state-of-the-art VLAD descriptor with large vocabulary (see Fig. 1). Firstly, we adopt tree-

*Corresponding author

structured hierarchical quantization (TSHQ) by hierarchical k-means (HKM) [12] to greatly speed up the VLAD computation with a large vocabulary (e.g., hundreds of thousands). One drawback of TSHQ is that it brings about larger quantization error than flat quantization, as TSHQ only visit a small number of nodes (centroids) to find the nearest neighbor when propagating local feature from root to leaf node. Thus, we propose multi-VLAD by constructing a VLAD for each level of the vocabulary tree, to compensate for the quantization error at that level. We show that the multi-VLAD performs better than multi-assignment [13] in both retrieval accuracy and descriptor storage saving. Extensive evaluation over benchmark datasets shows that the proposed approach obtains superior retrieval accuracy than various VLAD schemes and the state of the art, and provides over 100 times speedup than flat quantization.

The rest of the paper is organized as follows. In Section 2, we first give a brief review of the flat VLAD (known as the original VLAD). We present the proposed hierarchical multi-VLAD in Section 3. Experimental results are presented in Section 4. We conclude in Section 5.

2. FLAT VLAD

Jégou et al. [4] proposed the VLAD descriptor to produce a global vector representation for image. The extraction pipeline of VLAD descriptor is summarized as follows. Firstly, like BoW, a vocabulary with K centroids $C = \{c_i | i = 1, 2, ..., K\}$ is trained off-line by k-means clustering. Given an image with N local features, denoted by $X = \{x_n | n = 1, 2, ..., N\}$, each local feature x_n is quantized to the nearest centroid by flat quantization:

$$x_n \mapsto q(x_n) = \arg\min_{c_i \in C} \|x_n - c_i\|^2 \tag{1}$$

For each centroid c_i, a sub vector v_i is obtained by accumulating the residual vector between the centroid and local features quantized to this centroid:

$$v_i = \sum_{x:q(x)=c_i} x - c_i \tag{2}$$

Finally, the VLAD is formed by concatenating the residual vectors from all centroids $V = [v_i]$, resulting in a $d \times K$ dimensional descriptor, where d denotes the length of local features. It is shown in [10] that intra L2 normalization on residual vector from each centroid followed by L2 normalization on the whole VLAD descriptor can suppress the visual "burstness" problem [13] and significantly improve the retrieval accuracy.

Referring to Eq.(2), a residual vector v_i is a zero vector in case that there is no local feature quantized to the i_{th} centroid. This means that the i_{th} centroid is not selected for the aggregation step. Given a set of local features detected from an image, the number of selected centroids is relatively small, especially for large vocabulary size K. For storage saving, we only need to store the residual vectors from the selected centroids. Besides, an overhead of K bits is needed to keep track of the centroids, each bit indicates if the corresponding centroid is selected (1 means selected, otherwise, 0).

Multiple Assignment. The hard-assignment of local feature to the nearest neighbor centroid brings about the quantization error problem. Therefore, multi-assignment [13] or soft-assignment [14] have been proposed to reduce the quantization error, as local feature is represented as a linear combination of multiple centroids. Combining with multi-assignment, we can rewrite Eq.(2) as:

$$v_i = \sum_{x:c_i \in N_\gamma(x)} x - c_i, \tag{3}$$

where $N_\gamma(.)$ denotes γ nearest centroids of x. As quantization error reduced with multi-assignment, the retrieval accuracy is usually improved [13][11]. However, multi-assignment increases the number of selected centroids for aggregation, bringing about extra storage cost (as shown in Section 4).

3. HIERARCHICAL MULTI-VLAD

In this section, we firstly introduce the tree-structured hierarchical quantization to boost the VLAD extraction speed, especially for large vocabulary. Then we present the multi-VLAD representation to compensate for quantization error. We finally discuss the relationship between multi-VLAD and multi-assignment.

3.1. Tree-Structured Hierarchical Quantization (TSHQ)

To work with large vocabulary, we employ TSHQ to speed up feature quantization. A hierarchical vocabulary tree T_K^N, with depth N and branches K, is trained off-line by hierarchical k-means clustering. We denote the nodes at i_{th} level as $C^i = \{c_j^i | j = 1, 2, ..., K^i\}$, each node is considered as a centroid. During quantization, each local feature walks from root node to leaf node of the tree T_K^N. At i_{th} level, a local feature is assigned to the nearest neighbor node c_j^i from C^i, then the feature is propagated down to the next level started from the node c_j^i. Finally, the VLAD is extracted from the leaf nodes (centroids). We refer to the VLAD extracted with TSHQ as Hierarchical VLAD (HVLAD) from here on.

Descriptor Matching. When the number of local features is far less than the vocabulary size, the HVLAD exhibits the so called sparsity. Following [6][15], the descriptor matching between HVLADs V_q and V_r is computed as a normalized similarity on their overlapped centroids:

$$Sim(V_q, V_r) = \frac{\sum_{i=1}^{K^N} f_i^q f_i^r \sigma(v_i^q, v_i^r)}{\sqrt{\sum_{i=1}^{K^N} f_i^q \sum_{i=1}^{K^N} f_i^r}}, \tag{4}$$

where $\sigma(.,.)$ is match kernel [11] to measure the similarity between residual vectors of HVLAD (e.g., L2 distance, cosine distance). $f_i^q = 1$ means the i_{th} centroid is selected for image q, 0 otherwise. f_i^r has the same meaning for image r. According to the similarity equation, similarity is only computed based on the overlapping centroids between V_q and V_r.

3.2. Multi-VLAD Representation

There exists quantization error when assigning local feature to the nearest centroid with flat quantization. This issue becomes worse for hierarchical quantization, as the number of visited centroids are largely reduced comparing with flat quantization (i.e., the path traced by a local feature contains only a small number of nodes for TSHQ). One can simply apply multi-assignment at each level to alleviate the quantization error at that level, but as aforementioned, it leads to extra storage cost. Thus, we propose a multi-VLAD representation to take both quantization error compensation and descriptor storage into consideration. The basic idea is to extract a VLAD descriptor at each level independently, each VLAD can compensate for the loss from the corresponding layer.

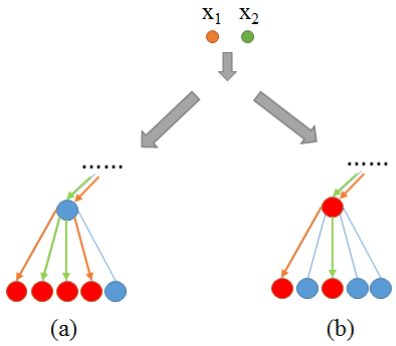


Fig. 2. A sample of generating VLAD with (a) Multi-assignment and (b) Multi-VLAD representation. For Multi-VLAD, the number of selected centroids (marked as red) is smaller than multi-assignment scheme, resulting in light descriptor storage.

For the hierarchical vocabulary tree T_K^N with depth N and branches K, we construct N VLAD descriptors $V = \{V^i | i = 1, 2, ..., N\}$, where the i_{th} VLAD $V^i = [v_j^i]$ is extracted at the i_{th} level of the tree $C^i = \{c_j^i | j = 1, 2, ..., K^i\}$, following the same procedure introduced in Section 2. Note that the multi-VLAD scheme can be combined with the multi-assignment. Thus, we can rewrite Eq.(2) and Eq.(3) as:

$$v_j^i = \begin{cases} \sum\limits_{x:q^i(x)=c_j^i} x - c_j^i, if\ HA \\ \\ \sum\limits_{x:c_j^i \in N_\gamma^i(x)} x - c_j^i, if\ MA \end{cases}, \quad (5)$$

where HA and MA denotes hard-assignment and multi-assignment, respectively. $q^i(x)$ refers to the quantizer at i_{th} level of the vocabulary tree and $N_\gamma^i(x)$ the top γ nearest neighbor centroids (nodes) for local feature x.

Descriptor Matching. For the descriptor matching between multi-VLADs, we firstly compute the normalized similarity between the VLAD at each level, then power weight the similarity scores and fuse them by multiplication. Specifically, given the multi-VLAD descriptors $V_q = \{V_q^i | i = 1, 2, ..., N\}$ from image q and $V_r = \{V_r^i | i = 1, 2, ..., N\}$ from image r, the similarity $Sim_m(V_q, V_r)$ is defined as:

$$Sim_m(V_q, V_r) = \prod_{i=1}^{N} Sim(V_q^i, V_r^i)^{\alpha_i}, \quad (6)$$

where $Sim(.,.)$ is the normalized similarity equation defined in Eq.(4) and α_i is the weighting factor to control the impact of the i_{th} VLAD. In general, the VLAD at high level of the tree is more discriminative than low level. Thus, we empirically set the weights as $\alpha_i = i$ in this paper.

Multi-VLAD vs. Multi-assignment. Both multi-VLAD and multi-assignment aim to reduce the quantization error at each level of the tree. One drawback of multi-assignment is that it is hard to determine the optimal number of assignments (i.e., γ). Smaller γ leads to insufficient quantization error reduction, while larger γ brings noise into quantization. Compared to multi-assignment, multi-VLAD has two advantages: (1) multi-VLAD is parameter free and shows superior retrieval accuracy, as shown in Section 4; (2) multi-VLAD offers light storage. We observe that the storage of multi-assigned VLAD increases with the parameter γ, while multi-VLAD is stable. On the other hand, as the vocabulary size exponentially increases with depth, for multi-VLAD, the number of selected centroids at intermediate level is much smaller than high level. This means that the storage of VLAD extracted from intermediate level is fewer than multi-assignment (see Fig. 2).

4. EXPERIMENTS

4.1. Datasets and Baselines

We evaluate the proposed method on two benchmark datasets, INRIA Holidays [16] and UKBench [12]. The Holidays dataset consists of 1491 images and 500 of them are used as query images. Each query has about 1∼12 relevant images undergoing various changes. The UKBench dataset contains 2550 objects, each containing 4 images taken from differen viewpoints. Mean Average Precsion (mAP) is used to evaluate retrieval performance. We compare several VLAD schemes, including: (1) **FVLAD.** The flat VLAD introduced in Section 2. (2) **HVLAD.** The VLAD extracted from leaf nodes with TSHQ. (3) **HMVLAD.** The proposed multi-VLAD representations extracted from all levels with TSHQ. For all schemes, multiple assignment (denoted as MA) is adopted for performance comparison. We implement MA as follows: MA = 4 nearest neighbors on all query sides and MA = 2 on database side if stated.

We use a variant of SIFT descriptor, RootSIFT [17], which simply applies L2-normalizaiton and square root to each component of SIFT. Following [4][5][6] closely, the dimension of RootSIFT is reduced from 128 to 32 by applying PCA dimension reduction. We evaluate the performance of all VLAD schemes from a small vocabulary (i.e., 128 centroids) to large ones (i.e., $40^3 = 64k$, $50^3 = 125k$). The vocabulary is trained by HKM (with depth 3 branches 40, and depth 3 branches 50). Correspondingly, for HMVLAD, there are three VLAD descriptors with vocabulary size {40, 1600, 64k} and {50, 2500, 125k}. Finally, L2-normalization [10] is applied to the residual vector computed from each centroid. Aggregated selective match kernel [11] is used as the match kernel.

4.2. Performance Comparison

Table 1 compares the mAP of various VLAD schemes over both Holidays and UKBench datasets. Firstly, it is shown that the retrieval accuracy of VLAD is significantly improved by increasing the vocabulary size, for example, +20% in mAP on Holidays from 128 to 40^3 centroids. Secondly, the proposed HMVLAD significantly outperforms HVLAD and HVLAD combined with multi-assignment (MA). For instance, HMVLAD yields the best retrieval accuracy, e.g., mAP 83.16% on Holidays dataset vs. 81.74% for HVLAD vs. 82.29% for HVLAD+MA, with vacabulary size 40^3. The performance gap on UKbench is larger between HMVLAD and the baselines. When combine HMVLAD with MA, the retrieval accuracy is slightly better than HMVLAD. It seems that multi-VLAD and MA are not complementary to each other. Thirdly, HMVLAD obtains a comparable mAP with FVLAD and FVLAD+MA. For example, 83.89% for FVLAD+MA vs. 83.55% for HMVLAD on Holidays with vacabulary size 50^3. Finally, the HMVLAD achieves competitive retrieval accuracy compared to state-of-the-art methods (82.2% on Holidays in [11] and 90% on UKBench in [13]).

Table 2 shows the comparison of VLAD extraction time and average number of selected centroids between various VLAD schemes. One can see that HVLAD and HMVLAD is over 100 times faster than FVLAD. This demonstrates that our HMVLAD can largely re-

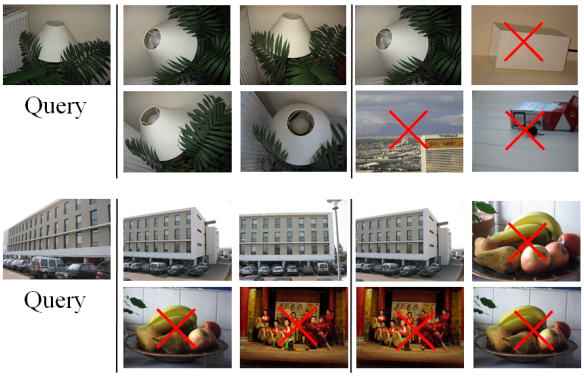


Fig. 3. Two sample queries (Left) from the Holidays dataset and their retrieval results obtained by HMVLAD (Middle) and HVLAD (Right).

duces the extraction time, while without incurring retrieval accracy loss. In addition, the number of selected centroids of HMVLAD is significantly fewer than HVLAD+MA. This leads to smaller storage as we only have to store the sub VLAD vectors of selected centroids. Fig. 3 shows sample results on the Holidays dataset obtained by HVLAD and HMVLAD.

Dataset	Holidays			UKBench		
Vocabulary Size	128	40^3	50^3	128	40^3	50^3
FVLAD	63.75	83.24	83.06	79.68	86.84	86.15
FVLAD+MA	-	83.66	83.89	-	87.62	86.82
HVLAD	-	81.74	82.01	-	86.23	84.97
HVLAD+MA	-	82.29	82.5	-	86.64	85.57
HMVLAD	-	83.16	83.55	-	88.53	87.9
HMVLAD+MA	-	83.41	83.83	-	89.04	88.46

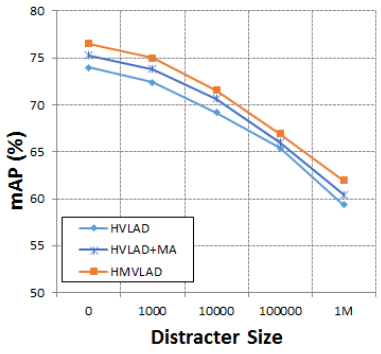
Table 1. Performance comparison on Holidays and UKBench between various VLAD schemes.

Vocabulary Size	40^3		50^3	
Methods	Time (ms)	Centroids	Time (ms)	Centroids
FVLAD+MA	~4500	1833	~10400	1883
HVLAD+MA	46	1819	63	1870
HMVLAD	47	1586	63	1693

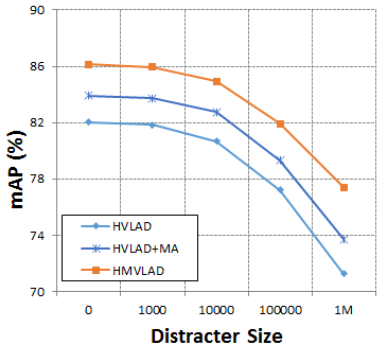
Table 2. Comparison of VLAD extraction time (excluding local feature detection and description) and average number of selected centroids (related to descriptor storage). Experimental results are measured on a server with 2.6GHz CPU and 32G RAM.

4.3. Large Scale Experiments

To evaluate the scalability of our proposed approach in large scale image retrieval, a FLICKR1M dataset containing 1 million images is used as distracters, combined with the Holidays and UKbench database images. Considering the slow floating-point computation and heavy storage with very high-dimensional VLAD descriptors over large scale dataset, local features are selected to a maximum of 500 [18] and sign binarization is applied to compress VLAD into binary codes, followed by inverted file structure to index the codes. The vocabulary with size 40^3 is used to construct descriptors. For



(a) Holidays



(b) UKBench

Fig. 4. Performance comparison in terms of mAP between HVLAD, HVLAD+MA and our HMVLAD, as distracter size changes.

HMVLAD, the 3_{rd} level's VLAD is indexed by inverted file and used for a first stage retrieval to generate 1000 candidate images, the 1_{st} and 2_{nd} levels' VLAD are subsequently combined to re-rank the candidate images. Fig. 4 illustrates the performance of various VLAD schemes at varied distracter size. We can see that HMVLAD obtains better retrieval performance than other schemes in all different distracter sizes, which proves that our approach is superior.

5. CONCLUSION

In this paper, we propose to generate VLAD descriptor by tree-structured hierarchical quantization (TSHQ). A high discriminative representation can be extracted in milliseconds even with a very large vocabulary. Furthermore, we propose to generate a VLAD at each level of the vocabulary tree and form multi-VLAD for reducing the negative effect of quantization error caused by TSHQ. Experimental results show that our approach achieves the state-of-the-art performance. Meanwhile, low extraction time and light storage are offered. More research work on how to incorporate feature selection [18][19] and rate-adaptive descriptor coding [15] will be included in our future work.

6. ACKNOWLEDGMENT

This work was supported by the Chinese Natural Science Foundation under Contracts No. 61271311, No. 61390515, No. 61421062 and by the National Hightech R&D Program of China (863 Program) under Grant No. 2015AA016302.

7. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *ECCV*, pp. 404–417. Springer, 2006.

[3] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2003, pp. 1470–1477.

[4] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3304–3311.

[5] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[6] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 3384–3391.

[7] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez, "Revisiting the vlad image representation," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 653–656.

[8] Giorgos Tolias, Teddy Furon, and Hervé Jégou, "Orientation covariant aggregation of local descriptors with embeddings," in *ECCV*, pp. 382–397. Springer, 2014.

[9] Hervé Jégou and Andrew Zisserman, "Triangulation embedding and democratic aggregation for image search," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

[10] Relja Arandjelovic and Andrew Zisserman, "All about vlad," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 1578–1585.

[11] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 1401–1408.

[12] David Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2006, vol. 2, pp. 2161–2168.

[13] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, "On the burstiness of visual elements," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1169–1176.

[14] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[15] Jie Lin, Ling-Yu Duan, Yaping Huang, Siwei Luo, Tiejun Huang, and Wen Gao, "Rate-adaptive compact fisher codes for mobile visual search," *Signal Processing Letters, IEEE*, vol. 21, no. 2, pp. 195–198, 2014.

[16] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, pp. 304–317. Springer, 2008.

[17] Relja Arandjelovic and Andrew Zisserman, "Three things everyone should know to improve object retrieval," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2911–2918.

[18] Gianluca Francini, Skjalg Lepsøy, and Massimo Balestri, "Selection of local features for visual search," *Signal Processing: Image Communication*, vol. 28, no. 4, pp. 311–322, 2013.

[19] Jie Lin, Ling-Yu Duan, Tiejun Huang, and Wen Gao, "Robust fisher codes for large scale image retrieval," in *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 1513–1517.