Compact and Distinctive Visual Vocabularies for Efficient Multimedia Data Indexing

Dimitris Kastrinakis[1], Symeon Papadopoulos[2] and Athena Vakali[1]

[1] Department of Informatics, Aristotle University,
54124, Thessaloniki, Greece
{kastrind,avakali}@csd.auth.gr,
[2] Information Technologies Institute, CERTH-ITI
57001, Thessaloniki, Greece
papadop@iti.gr

Abstract. Multimedia data indexing for content-based retrieval has attracted significant attention in recent years due to the commoditization of multimedia capturing equipment and the widespread adoption of social networking platforms as means for sharing media content online. Due to the very large amounts of multimedia content, notably images, produced and shared online by people, a very important requirement for multimedia indexing approaches pertains to their efficiency both in terms of computation and memory usage. A common approach to support query-by-example image search is based on the extraction of *visual words* from images and their indexing by means of inverted indices, a method proposed and popularized in the field of text retrieval.
The main challenge that visual word indexing systems currently face arises from the fact that it is necessary to build very large visual vocabularies (hundreds of thousands or even millions of words) to support sufficiently precise search. However, when the visual vocabulary is large, the image indexing process becomes computationally expensive due to the fact that the local image descriptors (e.g. SIFT) need to be quantized to the nearest visual words.
To this end, this paper proposes a novel method that significantly decreases the time required for the above quantization process. Instead of using hundreds of thousands of visual words for quantization, the proposed method manages to preserve retrieval quality by using a much smaller number of words for indexing. This is achieved by the concept of *composite words*, i.e. assigning multiple words to a local descriptor in ascending order of distance. We evaluate the proposed method in the Oxford and Paris buildings datasets to demonstrate the validity of the proposed approach.

Key words: multimedia data indexing, local descriptors, visual word, composite visual word

1 Introduction

Multimedia content is produced at unprecedented rates and is extensively used in both personal (e.g. holiday albums) and professional (e.g. stock image collec-

tions) settings. As the size of media collections increases, the need for efficient content retrieval becomes more pronounced. One of the prevalent image search paradigms pertains to Content Based Image Retrieval (CBIR), which enables searching large image collections using their visual content to establish relevance. Typically, CBIR is implemented by means of a query-by-example application, which given an input query image returns the top N most relevant results from the collection, assessing relevance on the basis of visual content alone.

In recent years, the performance of CBIR systems has substantially improved thanks to the development of rich image representations based on local descriptors (e.g. SIFT [8], SURF [2], etc.) and the use of full-text search technologies that formulate query-by-example as a text retrieval problem [11, 15]. According to those, a set of local descriptors is extracted from each image and are subsequently quantized into *visual words*, leading to the so-called Bag of Words (BoW)[1] representation. The BoW representation is amenable to inverted indexing techniques, thus enabling indexing and efficient querying of very large image collections by use of robust full-text indexing implementations such as Lucene[2] and ImageTerrier[3].

Despite its success, the application of the BoW image indexing scheme is still considered a very challenging and computationally demanding task due to the fact that visual words are not natural entities (as are terms/tokens in the case of text documents). In fact, visual words are the result of a training process, whereby the local descriptors from a large collection of images are clustered around k centres, and the corresponding centroids are considered as the words of the visual vocabulary. Having built such a vocabulary, new images are indexed by mapping their local descrptors to words of the vocabulary, i.e. for each image descriptor the most similar centroid (among the k words of the vocabulary) is selected for use in the BoW representation. As the number of local descriptors per image typically lies in the range of some hundreds to a couple of thousands, it becomes obvious that deriving the BoW representation for an image may incur significant computational cost.

In fact, typical sizes for visual vocabularies range from hundreds of thousands to even millions of visual words (i.e. $k \sim 10^5 - 10^6$) according to related studies [12]. This creates two computational problems: (a) creating vocabularies of such sizes by means of clustering techniques becomes extremely expensive, (b) the indexing time for new images increases significantly due to the need for mapping each local descriptor of the image to its most similar visual word as explained above. While the first of these problems appears only at offline settings, and thus does not affect retrieval efficiency, the second problem incurs substantial overhead at indexing time. To this end, this paper proposes a new approach for visual word indexing that significantly reduces the number of visual words that are necessary to achieve satisfactory retrieval accuracy. This is achieved by considering *composite visual words*, i.e. permutations of multiple visual words or-

[1] In several works, the preferred abbreviation is BOV (Bag of Visual words).
[2] `http://lucene.apache.org/`
[3] `http://www.imageterrier.org/`

dered according to their distance from the corresponding local descriptors. Even with a small visual vocabulary, our approach leads to a much more distinctive BoW representation. Our experimental study on two standard datasets, Oxford [12] and Paris buildings [13]) reveals that **as few as** $k = 200$ **visual words** can be utilized to match the retrieval performance of existing approaches using **two to three orders of magnitude more visual words**.

The paper is structured as follows: Section 2 offers the necessary background on the problem of BoW indexing, also covering important contributions in the area. Section 3 provides a description of the proposed approach. Next, we present an evaluation of the approach in Section 4 and we summarize our findings and discuss future steps in Section 5.

2 Background

2.1 Image indexing using Bag of Words representations

We consider a collection of images $P = \{p_i\}$ to be indexed. For each image, we extract a set of local descriptors $F_i = \{f_{i,1}, ..., f_{i,|F_i|}\}$ where each descriptor is a feature vector, which is real-valued ($f_{i,x} \in \mathbb{R}^D$), e.g. in the case of SURF [2], or integer-valued ($f_{i,x} \in \mathbb{Z}^D$), e.g. in the case of SIFT [8]. Typical values for $|F_i|$, i.e. number of descriptors per image, range from a few hundreds to few thousands, while typical values for the dimensionality of the descriptor vectors are $D = \{64, 128\}$. To derive a BoW representation for an image, we need to discretize the set of local descriptors F_i to end up with the BoW representation denoted as $W_i = \{w_{i,1}, ..., w_{i,k}\}$, where $w_{i,x}$ is the weight that visual word $x \in [1, k]$ has in the representation of image p_i, a process that is often called *feature quantization*. This presumes a visual vocabulary $V = \{v_1, ..., v_k\}$ where $v_x \in \mathbb{R}^D$ or $v_x \in \mathbb{Z}^D$ depending on the local descriptor of choice. Having derived the BoW vector for each image of the collection, indexing is typically implemented by means of inverted index structures and relevance is assessed on the basis of classic text retrieval schemes such as $tf * idf$ [15]. Table 1 summarizes the described notation.

The arising issue is the need of a rich and distinctive vocabulary V. To this end, a clustering process, e.g. k-means, must be carried out on a large number of images that act as the training or learning set for V. Since a satisfactory vocabulary may need to contain $> 10^6$ words, as evidenced by recent studies [12], it becomes clear that the feature quantization process may become a significant computational hurdle at both indexing and query time.

2.2 Related Work

The first popular attempt towards CBIR using a text retrieval approach was proposed by Sivic and Zisserman [15], who proposed the BoW representation for retrieving objects in video content. The descriptor vectors, which are computed for each frame, are quantized into visual words using k-means clustering. For

Symbol	Description
$P = \{p_i\}$	Collection of images, an image being denoted as p_i.
$n = \|P\|$	Number of images in the collection to be indexed.
$f \in \mathbb{Z}^D$	A D-dimensional local descriptor feature vector (integer-valued in the case of SIFT).
$F_i = \{f_{i,1}, ..., f_{i,\|F_i\|}\}$	Set of local descriptor feature vectors for image p_i.
$\|\bigcup_i F_i\|$	The set of all local descriptors in a collection (e.g. used to learn a vocabulary).
$V = \{v_1, ..., v_k\}$	Visual vocabulary used for indexing.
$k = \|V\|$	Size of visual vocabulary (number of visual words).

Table 1. Notation used in the paper.

each visual word, an entry is added to the index that stores all its occurrences in the video frames of the collection, thus building an inverted file. Text retrieval systems often promote documents where query keywords appear close together. This analogy is also adopted for BoW-based visual indexing [12, 15], where it is required that matched regions in the retrieved frames of a video should have similar spatial arrangement to the regions of the query image.

To speed up the vocabulary construction step and the image query process, Nistér and Stewénius introduce a scheme in which local descriptors are hierarchically quantized in a vocabulary tree [11]. In particular, this tree is built by use of hierarchical k-means clustering (HKM) relying on a set of descriptor vectors for the unsupervised creation of the tree. An initial k-means process clusters the training data to groups, where each group consists of the descriptor vectors closest to its center. This process is recursively applied to each group, forming a specified maximum number of levels L. A descriptor vector is propagated down the tree by comparing the vector to the cluster centres that reside at each level.

Another clustering method often used for feature space quantization is the approximate k-means clustering (AKM) [12]. Typical k-means implementations fail to scale to large volumes of data, since the time complexity of each iteration is linear to the number of data, dimensionality of the data and the number of desired clusters. Instead of calculating the exact nearest neighbours between data points and cluster centres, an approximate nearest neighbour method can be applied to increase speed. In [12], a forest of eight randomized k-d trees [7] is used, which is built over the cluster centres at the beginning of each iteration. A forest of randomized k-d trees prevents points lying close to partition boundaries from being assigned to an incorrect nearest neighbour. This is especially important for the quantization of high dimensional features such as SIFT ($D = 128$).

In [13], Philbin et al. introduce an approach called *soft assignment*, where a high dimensional descriptor is mapped to a weighted combination of visual words, rather than a single visual word. The weight assigned to neighbouring clusters depends on the distance between the descriptor and the cluster centres. It was shown that soft assignment can boost the recall of a retrieval system and,

if combined with spatial verification, precision could be increased too. However, this technique requires more space for the inverted index.

A similar work to the one proposed here introduces the concept of *visual phrases* [20]. The authors propose a method for mining visual word collocation patterns from large image collections and then use those for indexing. Compared to our approach, the concept of visual phrases does not take into account word ordering, which could harm precision, and in addition it requires an expensive phrase mining process to derive the phrase dictionary. A more sophisticated approach for deriving more descriptive visual vocabularies is presented in [21], in which a very large corpus of images is processed to extract a descriptive visual vocabulary consisting of both words and phrases. Though the reported retrieval accuracy is substantially improved, the approach of [21] presumes the existence of a very large image collection for training and the extraction of a vocabulary of considerable size ($k \sim 10^4 - 10^5$) that makes the approach considerably more demanding compared to ours.

In [22], an alternative visual vocabulary generation mechanism is proposed, wherein groups of visual words are extracted making sure that the spatial relations among the words of the same group are maintained. This could be considered as a generalization of the method proposed here since our compositve visual words maintain a Euclidean distance-induced ordering, but lose the spatial layout information. However, the vocabulary generation process is significantly more complex and carries the risk of leading to an incomplete vocabulary, i.e. not all possible spatial visual word configurations can be adequately represented in the vocabulary index. A similar approach, facing similar complications, is described in [19], where groups of spatially consistent local descriptors are called *bundled features* and are considered as the unit of visual indexing.

3 Proposed Framework

Our framework is described in two steps: (a) creating the visual vocabulary, (b) using composite visual words to index new image collections.

3.1 Visual vocabulary creation

Having stored the features of the collection, quantization of the feature space can take place, in order to produce the initial visual vocabulary. At a later step, this vocabulary will be enriched with composite visual words to improve retrieval.

Clustering large numbers of descriptor vectors. Feature discretization is performed by applying a clustering algorithm on the descriptors of the learning set. We used the VLFeat [17] implementation of Lloyd's k-means algorithm. To cluster the data, they need to be loaded in main memory, which is impossible in the case of millions of multidimensional features. Despite the fact that Lloyd's

algorithm does not perform triangular inequality checks[4], it cannot run in systems with limited main memory for large volumes of data, since it would require $n \cdot d$ space, n being the number of vectors and D their dimensionality.

To this end, we partition the input vectors and provide each partition to a streaming implementation of the k-means algorithm [10]. According to it, the set $F = F_i \cup ... \cup F_n$ of the extracted descriptors is divided into an appropriate number of subsets, to which k-means++ [1] is applied (such that an arbitrarily poor approximate solution of k-means is avoided). The union of the resulting cluster centres is then provided to a second execution of the algorithm, producing the set of visual words V.

The requirement for a compact vocabulary. During the clustering step, Lloyd's k-means algorithm introduces significant time (and space) complexity requirements due to the large number of vectors to cluster and cluster centres to calculate. Simple k-means has $O(n \cdot D \cdot k \cdot I)$ complexity, where n is the number of vectors to cluster, D the dimensionality, k the number of centres to find and I the number of iterations of the algorithm. The above fact renders the application of k-means impractical in this context, when the size of the visual vocabulary becomes very large. As shall be shown below, this problem can be addressed at indexing time with the application of a technique that makes unnecessary the creation of a vocabulary of large size.

3.2 Indexing the Collection with Composite Visual Words

After the vocabulary creation step, a set k of visual words is available for indexing. Then, each descriptor vector $f \in F_i$ of a new image p_i is compared to the vocabulary visual words using the L2 distance. According to the BoW representation [15], in order to index the collection, each vector f is assigned to the nearest visual word v using the selected distance measure d:

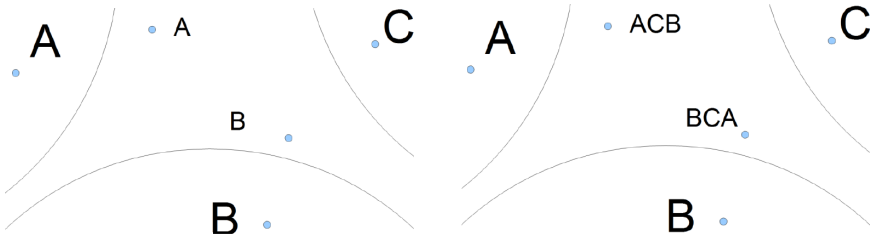
$$v_f = \arg\min_{v \in V} d(f, v) \tag{1}$$

For each image in the collection, a document is created that contains the assigned visual words. Eventually, each such document is indexed using an inverted index of terms (visual words) pointing to documents.

Composite Visual Words. If the initial visual vocabulary is limited (due to the excessive computational requirements incurred by the clustering process when k is too large), its distinctive capability will be limited, thus inflicting a decrease on the performance of the retrieval system. To this end, for each feature vector f, instead of indexing the nearest visual word only, we index the concatenation of the B nearest words in ascending order using L2 distance.

[4] k-means with triangular equality checks requires $\frac{k \cdot (k-1)}{2}$ extra space to store the distances between centres [4].

Figure 1 depicts the above idea: composite visual word ACB corresponds to a feature vector lying nearest to center A, then C and B. Similarly, visual word BCA is created by a feature vector that lies nearest to B, C and A.



(a) Standard BoW indexing (b) Composite visual word indexing

Fig. 1. Illustration of composite visual words.

This approach implicitly exploits the inter-word relationships around neighbouring words so that the resulting composite visual word, as a permutation of relevant visual words, describes the corresponding descriptor f more distinctively, even with a small number of initial visual words. Making composite visual words is somewhat similar to soft-assignment [13]; however, there is a key difference between the two techniques: in soft-assignment, a feature is assigned to several visual words (cluster centers) separately, whereas in making composite visual words a feature is assigned to their concatenation, thus effectively enriching the resulting visual vocabulary.

At this point, we need to point out the distinction between the set of visual words V in the original visual vocabulary and the *effective visual vocabulary* resulting from the use of composite visual words. Since a composite word w' is formed by a permutation of words in V, this method creates an extended vocabulary V'. The maximum number of possible words in V' is: $|V|'_{max} = P(|V|, B)$, where B is the user-defined number of words that will form a composite word, and $P(|V|, B)$ the number of B-permutations of $|V|$, i.e. $P(|V|, B) = \frac{|V|!}{(|V|-B)!}$.

For example, if $|V| = 100$ and $B = 3$, then the maximum number of words that V' can possibly contain is $100 \cdot 99 \cdot 98 = 970{,}200$. For $B = 4$, we get $|V'| = 94,109,400$[5]. Of course, a vocabulary of such size might have a negative impact on retrieval, because a lot of terms would appear only once in the collection. For this reason, we devise a thresholding strategy based on the distances of features from candidate words, as explained below.

[5] In practice, we expect somewhat smaller vocabularies than the aforementioned ones due to the fact that some word combinations would be highly improbable (depending also on the distribution of the local descriptor vectors of the images to be indexed).

Thresholding strategy. When assigning visual words to features, a restriction on distance would disqualify a word v from the indexing of feature f if:

$$d(v, f) > \beta \cdot \max_{v' \in V} d(v', f) \tag{2}$$

where $\beta \in (0, 1)$, and $\max_{v' \in V} d(v', f)$ is the maximum distance between feature f and a visual word. We extend the above condition to include the B nearest candidate words, instead of the single nearest: instead of a fixed condition for all top B words, we introduce a constraint that becomes progressively stricter for the next nearest word. The i-th nearest word, $1 \leq i \leq B$, is disqualified if:

$$d(v, f) > e^{-ai} \cdot \max_{v' \in V} d(v', f) \tag{3}$$

Increasing a makes the condition stricter. In this case, notice that we do not consider a fixed number of words in the composite visual word; such a word may consist of many words as long as they satisfy the above condition. On the other hand, given a distance threshold and a user-defined maximum number of words B that will form a composite word, there may be composite words formed by less than B words. Apparently, constants a, B and $|V|$ strongly affect the size of the composite vocabulary V'. In fact, the maximum number of words in V' can be analytically determined as $|V|'_{max} = \sum_{i=1}^{B} P(|V|, B)$. This carries the risk of creating an even larger vocabulary V' compared to the original one. However, this is highly unlikely if an appropriate condition is set, as will be shown in the experiments section. Apart from that, collections typically contain images that share a lot of visual patterns; therefore, we expect the vocabularies resulting from this process to not contain too many unique words.

Algorithm 1 specifies the process for extracting a composite visual word v^+ given a feature vector f of an image, a set V of visual words, a constant a used in the distance condition and the number of words B that can form v^+. At first, the algorithms calculates the distances of the visual words from the given vector and sorts them in ascending order. Then it concatenates the labels of the nearest B words, as long as their distances are less or equal to the threshold. If a word v_i, $1 \leq i \leq B$ is disqualified, the process does not continue for v_{i+1} as it would not satisfy the distance condition, since $d(v_{i+1}, f) \geq D(v_i, f)$ (words are already sorted with respect to f).

4 Experimental Evaluation

To test the validity of our approach, we developed a prototype implementation of the proposed framework and used the Oxford [12] and the Paris Buildings [13] datasets to assess the quality of retrieved results. As local descriptors, we made use of SIFT using one of the most popular implementations [16]. For building and maintaining the index, we use the Apache Solr full-text indexing framework. In the first dataset, we limited the extracted features per image to 2,000, whereas in the second, a maximum of 1,000 descriptors per image were extracted. Table 2 summarizes these basic statistics.

Algorithm 1 Indexing with composite visual words

Input: f, V, a, B
Output: v^+ (composite visual words)

```
for all v ∈ V do
    v.distance ← computeDistance(v, f)
end for
sort(V)
d_max ← V.last().distance
i ← 1
v+ ← {}
while i ≤ B do
    d_i ← V[i].distance
    threshold ← e^{-ai} d_max
    if d_i ≤ threshold then
        v+ ← v+.concat(V[i])
    else
        break
    end if
    i ← i + 1
end while
```

Dataset	Oxford	Paris
$n = \|P\|$	5,063	6,412
$\|\bigcup_i F_i\|$	9,731,989	6,314,776
$avg(\|F_i\|)$	1,923	985

Table 2. Basic dataset statistics.

Each dataset contains manually created ground truth for 55 queries around 11 different landmarks, for which relevant and non-relevant images are known. Thus, each image in the dataset can be characterized by one of four possible states with respect to a given query:

- GOOD: a clear picture of the query object/building.
- OK: more than 25% of the query object is clearly visible.
- JUNK: less than 25% of the query object is visible, or there are very high levels of occlusion or distortion.
- BAD: the query object is not present.

The calculation of mean Average Precision (mAP) considers as correct the images that fall under 'GOOD' or 'OK' states using the formula:

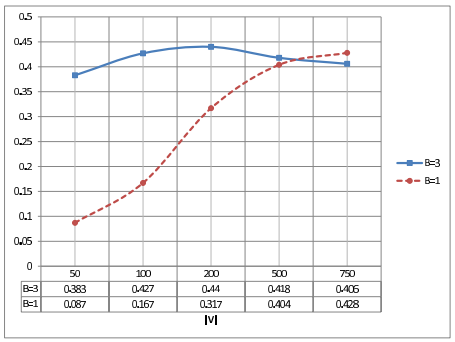
$$mAP = \frac{\sum_{q \in Q} AP(q)}{|Q|} \tag{4}$$

where Q is the set of queries, q is an individual query and $AP(q)$ is the Average Precision for a specific query computed as:

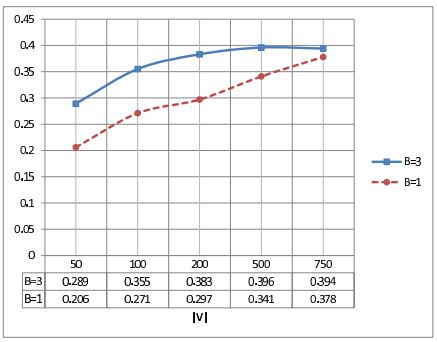
$$AP(q) = \sum_{i=1}^{N} pr(i) \cdot \Delta rec(i) \tag{5}$$

where i is the rank in the list of top N retrieved images, $pr(i)$ denotes the average precision at the i- and $(i-1)$-th position and $\Delta rec(i)$ the difference of recall between the i- and $(i-1)$-th position. We calculated the mAP for two values of B (maximum number of words that may form a composite visual word). Obviously, a $B = 1$ setting does not generate any composite words, which allows us to compare the mAP of the proposed method against the standard BoW indexing scheme. We applied the distance condition of Equation 3 for an empirically selected parameter $a = 0.2$.

Figures 2(a) and 2(b) illustrate the mAP results for increasing size of the initial vocabulary V, i.e. increasing number of cluster centres (visual words). At this point, it should be noted that no supplementary mechanism for enhancing the quality of retrieved results was utilized (e.g. soft-assignment [13] or spatial verification [12]). It is noteworthy that the proposed approach achieves a mAP score of 0.383 for as few as 50 centres, whereas the baseline is limited to only 0.087 (Figure 2(a)). Performance appears to level off as $|V|$ rises, but this is natural since $|V|$ affects the size of the composite vocabulary $|V'|$: if V' is too rich then probably different words are assigned to similar feature vectors, mitigating the performance benefits induced by the increased distinctive capability. Nevertheless, we can still build an image retrieval system with acceptable performance in less time, because we would have to quantize the feature space among 50 clusters only. For instance, in the case of k-means, with $O(n \cdot D \cdot k \cdot l)$ time complexity and $k = 50$, we quantize the feature space 10 times faster compared to the case of $k = 500$, incurring negligible loss in mAP (Figure 2(a)).



(a) Oxford



(b) Paris

Fig. 2. Comparing standard BoW indexing ($B = 1$) with indexing based on composite visual words ($B = 3$). Retrieval accuracy is expressed with mAP.

As illustrated in 2(b), for the Paris Buildings collection, there is still a considerable increase in mAP with our method, with a small decrease for 750 centers, as the baseline tends to outperform the $B = 3$ setting for larger initial vocabularies. Since the size of the composite vocabulary V' depends on that of the initial vocabulary V and constants a and B, V' could have been smaller for $B = 2$ or for larger values of a (i.e. a stricter distance condition). In this way, constants a and B offer control over $|V'|$, given any $|V|$. Table 3 depicts the effective vocabulary V' size dependence on the size of the original vocabulary, and for fixed values of a and B in the two datasets.

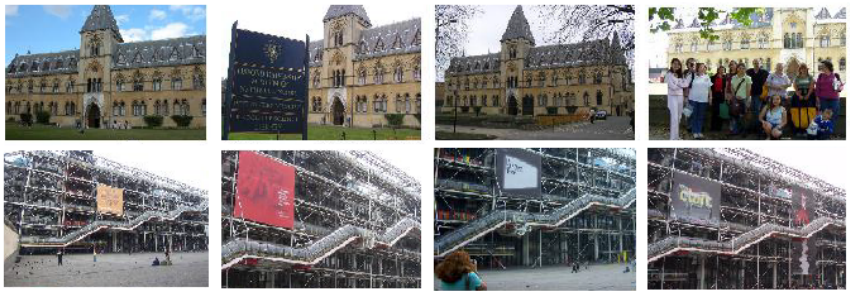


Fig. 3. Top-3 results of the highest ranked queries for the Oxford and Paris buildings datasets (the query image is the first on the left).

$\|V\|$	$\|V'\|$	
	Oxford	**Paris**
50	3,688	3,220
100	26,978	20,613
200	120,950	104,810
500	599,683	477,239
750	980,644	778,499

Table 3. Size of extended vocabulary V', given $|V|$, $B = 3$, and $a = 0.2$.

Figure 3 illustrates the results of the proposed approach for two example queries (using $|V| = 200$). In addition, Table 4 presents a comparison between the proposed approach and three other indexing schemes based on the standard BoW model, as they were reported in [12]. It is remarkable that our approach attains a maximum mAP score of 0.44 for just $|V| = 200$ ($|V'| \approx 120K$), while Philbin et al. attained 0.355 mAP with $|V| = 10K$ and only slightly outperform our approach (mAP=0.464) when they use a vocabulary of significantly larger size ($|V| = 1M$). However, it should be noted that the two approaches are

not directly comparable due to the fact that they create the vocabulary using 5 millions descriptors. It is also noteworthy that our approach slightly outperforms hierarchical k-means (HKM), which attained 0.439 mAP with as many as 1M visual words generated from 16.7M descriptors. Approximate k-means (AKM) reaches 0.618 mAP with 1M cluster centers using a forest of eight randomized k-d trees, but this is not directly comparable to our approach as the authors submit only a subset of the original image as query taking into account only the actual object of interest.

Approach	$\lvert\bigcup_i F_i\rvert$	$\lvert V\rvert$	**mAP**
k-means	5M	50K	0.434
HKM	16.7M	1M	0.469
AKM	16.7M	1M	0.618
CVW	9.7M	$\lvert V\rvert = 200$, $(\lvert V'\rvert = 120K)$	0.44

Table 4. Comparison between the proposed method, denoted as CVW (Composite Visual Words), and exact k-means, hierarchical (HKM) and approximate (AKM) k-means, as reported in [12].

5 Conclusions and Future Work

The paper proposed a novel approach for BoW-based indexing of images using a very compact visual vocabulary. The approach is based on the concept of composite visual words, i.e. sequences of visual words ordered based on their distance from the local descriptors of the images to be indexed. An appropriate thresholding strategy was devised to make the proposed indexing scheme effective by eliminating a large number of potentially spurious composite visual words, and an algorithm was described for extracting the composite words to be indexed by an incoming image.

Experiments on two standard datasets revealed that the proposed approach can accomplish decent retrieval results (as expressed by use of mean Average Precision). In particular, a setting of $B = 3$ nearest initial words from an initial vocabulary of 200 words attained 0.44 mAP, outperforming the standard BoW indexing scheme. Moreover, such a small initial vocabulary introduces significant performance gains on quantization and indexing; a vocabulary of 500 words with the standard method reaches comparable retrieval quality to an extended vocabulary (using composite visual words of maximum size $B = 3$) from an initial vocabulary of as few as 50 words, which offers a speedup of 10 for feature discretization. The parsimony of the produced vocabularies makes the proposal ideal for use in contexts with restricted computational resources, e.g. mobile applications.

Future Work. In the future, we consider experimenting on additional datasets of much larger scale ($n \sim 10^5 - 10^6$)[6] to investigate the scalability properties of the approach, both in terms of indexing time and in terms of robustness with respect to retrieval accuracy. In addition, we are interested in devising novel vocabulary learning methods that enable further gains in retrieval performance for even larger initial vocabulary sizes. More specifically, we will consider appropriate composite visual word ranking strategies to restrict the part of extended vocabulary V' used for indexing.

As a more specific plan, it will be interesting to further investigate the retrieval performance of our approach in relation to parameters a and B. We believe that these values are highly correlated with the size of the initial vocabulary V: a large vocabulary requires a strict distance condition, (i.e. a high a) and a small B, to avoid building an ineffective extended vocabulary V'. On the other hand, it would be wise to have a small a and a high B given a small initial vocabulary in order to produce as many useful words as possible. Developing a method for automatically adjusting these two parameters given the size of V appears to offer considerable opportunities for future work.

Acknowledgments. Symeon Papadopoulos was supported by the SocialSensor project, partially funded by the European Commission, under contract number FP7-287975.

References

1. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics (2007)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Proceedings of ECCV 2006, 404–417 (2006)
3. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. ECCV International Workshop on Statistical Learning in Computer Vision, Prague (2004)
4. Elkan, C.: Using the triangle inequality to accelerate k-means. Proceedings of the 20th international conference on Machine Learning (2003)
5. Harris, C., Stephens, M.: A combined corner and edge detector. Proceedings of the Alvey Vision Conference, 147–151 (1988)
6. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. Tenth IEEE International Conference on Computer Vision, ICCV Vol. 1. IEEE (2005)
7. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. IEEE (2005)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60: 91–110 (2004)

[6] Or equivalently, we are going to use the ground truth of the two datasets used here with a large number of distractor images.

9. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. Advances in Neural Information Processing Systems 19: 985–992 (2007)
10. Nir, A. Jaiswal, R., Monteleoni, C.: Streaming k-means approximation. Advances in Neural Information Processing Systems 22, 10–18 (2009)
11. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. (2006)
12. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07 (2007)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
14. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2008)
15. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. Ninth IEEE International Conference on Computer Vision, ICCV (2003)
16. Van De Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32.9: 1582–1596 (2010)
17. Vedaldi, A., Fulkerson, B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms. Available at: `http://www.vlfeat.org/` (2008)
18. Wang, C., Zhang, L., Zhang, H.: Learning to reduce the semantic gap in web image retrieval and annotation. Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 355–362 (2008)
19. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009)
20. Yuan, J., Wu, Y., Yang, M.: Discovery of collocation patterns: from visual words to visual phrases. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07, 1–8, IEEE (2007)
21. Zhang, S., Tian, Q., Hua, G., Huang, Q., Li, S.: Descriptive visual words and visual phrases for image applications. In Proceedings of the 17th ACM international conference on Multimedia, 75–84, ACM (2009)
22. Zhang, S., Huang, Q., Hua, G., Jiang, S., Gao, W., Tian, Q: Building contextual visual vocabulary for large-scale image applications. In Proceedings of the international conference on Multimedia, 501–510, ACM (2010)
23. Zhao, R., Grosky, W.: Bridging the semantic gap in image retrieval. Distributed multimedia databases: Techniques and applications, 14–36 (2002)