Efficient Bag of Words Based Concept Extraction for Visual Object Retrieval

Hilal Ergun and Mustafa Sert

Abstract Recent burst of multimedia content available on Internet is pushing expectations on multimedia retrieval systems to even higher grounds. Multimedia retrieval systems should offer better performance both in terms of speed and memory consumption while maintaining good accuracy compared to state-of-the-art implementations. In this paper, we discuss alternative implementations of visual object retrieval systems based on popular bag of words model and show optimal selection of processing steps. We demonstrate our offering using both keyword and example-based retrieval queries on three frequently used benchmark databases, namely Oxford, Paris and Pascal VOC 2007. Additionally, we investigate effect of different distance comparison metrics on retrieval accuracy. Results show that, relatively simple but efficient vector quantization can compete with more sophisticated feature encoding schemes together with the adapted inverted index structure.

Keywords Bag of words · Visual Object Retrieval · Distance metrics · SIFT · SVM

1 Introduction

Searching multimedia content and retrieving useful information for the user is becoming a trending research area due to availability of vast amounts of video and image data. Thousands, if not millions, of video content is created every day and uploaded to on-line or cloud communities. In addition to on-line content, much more is kept in local databases. All this data is waiting to be indexed for search and retrieval applications.

Among different indexing approaches, bag-of-visual-words model, which is well known by the information retrieval (IR) community, is the one being used most

H. Ergun · M. Sert(✉)
Department of Computer Engineering, Baskent University, Ankara, Turkey
e-mail: 21020005@mail.baskent.edu.tr, msert@baskent.edu.tr

T. Andreasen et al. (eds.), *Flexible Query Answering Systems 2015*,
Advances in Intelligent Systems and Computing 400,
DOI: 10.1007/978-3-319-26154-6_30

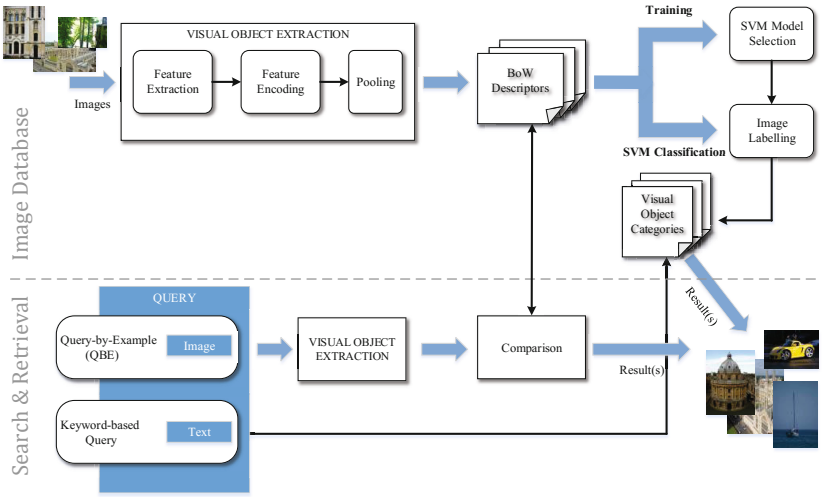

Fig. 1 Overall view of the proposed search and retrieval system

frequently and the one offering most successful results both in terms of precision and query running time performance in visual object retrieval applications. Bag of words or bag of visual words, will be referred as BoW from now on, finds its roots in the document retrieval domain and applied to image domain by Sivic et.al [20]. In very simple terms, BoW approach counts occurrence of local image features and tries to represent higher level image categories using this information. In their very simple form, bag-of-features methods discard all spatial information present in the image and retain only the visual words' visibility frequencies [27]. Lazebnik et al. introduce a novel method of spatial pyramids showing how spatial information can be integrated in BoW pipeline to further improve classification accuracy [11]. Philbin et al. explored image retrieval from a large dataset and showed how complimentary spatial re-ranking can be used to improve retrieval accuracy. [6] increased performance of object retrieval with the introduction of automatic query expansion. Perronnin and Dance applied Fisher kernel encoding to area of image category detection [16]. Jegou et al. introduced Hamming embedding for representing images with binary encodings and for efficient image retrieval against a user query from 1 million images [8]. In order to improve quantization step of BoW, Philbin et al. introduce a method that uses soft assignment of image features to visual codewords [19]. [26] showed how sparse coding can be used in-place of vector quantization to further improve classification accuracy. [24] improved sparse coding with feature-space locality constraints. In [1], the importance of query expansion is proved and also the most important steps which should be taken into consideration for improving object retrieval systems are

outlined. Yan et. al. improved tf-idf scheme by learning a similarity matrix from labeled data [25].

In this paper, we are focusing on efficient retrieval of image queries from a mid-to-large scale database with a trade-off between speed and accuracy. We propose a processing pipeline which can be used to issue two different query types to a local database. Block diagram of our proposed pipeline architecture is depicted in Figure 1. One other contribution of this paper is that we evaluated different distance comparison metrics used in literature on 2 different benchmark datasets for the BoW based visual object retrieval systems and we show that improper choice of distance metric and normalization selection can degrade retrieval performance.

The rest of the paper is organized as follows. Section 2 describes our search and retrieval framework. In Section 3, we introduce our keyword- and example-based retrieval architecture by adapting inverted index structure. Comprehensive experimental results and evaluations on three datasets are given in Section 4. Finally we conclude the paper in Section 5.

2 Proposed Search And Retrieval System

In this study, we target two types of querying methodologies against a video/image database, namely keyword based querying and querying by example(QBE). We extract visual objects from all images and create a global representation for the image based on these extracted objects. In the context of QBE, given image representation is directly compared to ones present in the database. This permits user to execute queries like "Find all images which are similar to this one". For keyword querying, we utilize machine learning techniques to learn a single textual representation (also referred to as visual concept) of previously created global image representations. This allows a user to retrieve images from database with a query like "Find all images which is mainly related to *X*" where *X* may be any visual object, event description, or a general concept.

2.1 Visual Object Extraction

As depicted in Figure 1, our visual object extraction scheme consists of three stages. First, we calculate image features for target images. Different local image features can be used at this step, SIFT [13] being one of the most popular choices. Furthermore, more than one type of image features can be extracted. [9] showed that multiple features can be combined to further increase effectiveness of BoW. Color information present in images can be included as well [21].

After image feature extraction comes the encoding step. In this step, we quantize image features into different BoW dictionary bins. Hard or soft assignment may be employed in this step. Vector quantization is the mostly employed hard assignment technique. It can be expanded to include soft-assignment though[18]. Fisher kernel encoding and sparse coding can be used to further increase softness of quantization.

However, in the context of large-scale image retrieval, vector quantization is the mostly adopted technique due to superior run-time performance when compared to more complicated encoding schemes. [5] provides detailed experimental analysis of various encoding schemes as well as their run-time complexity. Next comes the pooling stage. In this step, quantized image features are pooled to create global image representation which constitutes BoW descriptor for the given image. Different encoding schemes may perform better with different pooling techniques. Summation, or average, pooling mostly used in vector quantization type of encodings. Sparse coding performs better with maximum pooling operators. Applying spatial pyramids yields finer-grained pooling regions which boosts classification accuracy [11]. Here we use average pooling technique since we chose vector quantization without any soft-assignment.

2.2 Classifier Design

Different machine learning approaches may be used in this step, support vector machines (SVMs) being most frequently used and successful classifiers in the literature and therefore selected in our study. We make use of the classifier to enable keyword based queries; example based queries (QBE), on the other hand need a slightly modified approach. For QBE, when a query is desired to be executed, BoW descriptor of query image is compared to the ones in database. Instead of SVM based classification, more simple yet powerful distance metrics, such as Euclidean or Manhattan distances, are used to compare images. Then best matches are returned to the user.

SVMs are kernel based classifiers, different non-linear kernels may be employed for classification of different representations. Among non-linear kernels we found χ^2 and histogram intersection kernels to be most useful[5] [11]. One other choice frequently used in the literature is the Earth's mover distance kernel, namely EMD kernel. However, previous work showed that performance of EMD is comparable with χ^2 [27] so we don't use EMD at all. BoW descriptors are nothing but histograms of visual words in a given dictionary; this explains the success of χ^2 and other histogram comparison kernels on BoW data classification. Major drawback of non-linear kernels are their big performance hit. Non-linear SVMs are known to have higher complexity than linear SVMs [26] and they are not preferred for large image databases. On the other hand, when using vector quantization, linear kernels deliver substantially worse results [11] [2] Yang et al. states this is due to high quantization error in encoding step. One alternative is to use of an efficient suitable feature mapping for the data and using linear SVMs in place of non-linear ones. [23] provides a mathematically complete alternative for three of the mostly used histogram kernels and we used their implementation in our work. We investigated different SVM kernels relevant to image retrieval in a previous study and we found χ^2 as the most successful one [19].

2.3 Distance Metrics

Distance metrics are used in two different steps of image classification. During BoW image descriptor creation, local image features are compared to dictionary words using a suitable distance metric. While comparing different image BoW descriptors, again a suitable distance comparison is employed. Metric selection for the former is tightly coupled to feature extraction steps used. For SIFT, L2 distance comparison is suggested by the original author[13]. On the other hand, different distance metrics can be employed for the comparison of the latter. We investigate and evaluate the distance metrics, namely L1, L2, Histogram intersection, Hellinger distance, χ^2 and cosine distance, in our study for comparing BoW descriptors.

2.4 Visual Dictionary Creation

Vector quantization uses a pre-computed visual words called visual dictionary or codewords. Visual dictionaries can be created using different techniques, however, many papers use simple but effective K-means clustering algorithm and its variants. Target dataset or one another dataset may be used for visual dictionary creation. In this paper, we evaluated visual dictionary creation on the same dataset only.

For creating visual dictionary, all images are processed for local feature extraction and extracted features are clustered into K clusters using K-means. In an average dataset, one might extract 10 millions of image features and clustering of this amount of features may be not tractable. Both processing power and memory resources may be scarce at this step. Hierarchal clustering or approximate k-means clustering algorithms may be employed then. One another technique used is to randomly sub-sample available features before performing clustering.

3 Querying Schemes

While there can be different types of querying, image/video retrieval can be classified into two alternatives: text-based approaches and content-based approaches [12]. Text-based approaches associate each video shot or image frame with single or multiple keywords which permits user to query image or video database with a selection of keywords. Content-based approaches, mostly abbreviated as CBIR, allows user to search media database by supplying an example item.

Text-based approaches require the underlying data to be classified and/or every object contained within is detected prior to keyword query. Mostly attributed to semantic gap phenomenon, classification of images or scenes by computer programs into meaningful categories which can be resolved by humans natively is a non-trivial problem [27]. We describe our methods in the following subsections.

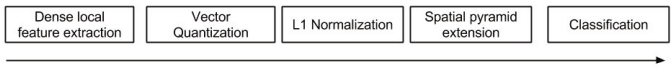

Fig. 2 Classification pipeline overview

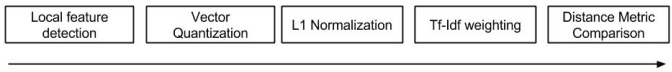

Fig. 3 Retrieval pipeline overview

3.1 Keyword Based Querying

Our image classification pipeline steps employed in keyword based queries are depicted in Figure 2. As a first step, we extract local SIFT features for every image. Local features can be extracted from a given image in one of two ways. One can use an interest point detector, or image can be densely sampled. We use both methods for different type of queries. Many papers show that dense feature extraction outperforms interest-point based extraction on visual classification tasks [11], [10]. For this reason we use dense sampling strategy for keyword type queries. We only extract one type of image feature, namely SIFT features.

After local feature extraction, local features are vector quantized to create image feature histograms. During vector quantization, a previously generated visual word dictionary, codebook, is used for comparison as is described in section 2.4. After vector quantization, created histogram is L1 normalized so that effect of unequal feature cardinalities in different images are neutralized. Then spatial pyramid extension is applied so that spatial layout of images are taken into consideration. At this step, we have the desired BoW descriptor for our query image and we use SVM classifiers to classify image category. Classifier design is detailed in section 2.2.

3.2 Query By Example

Example based retrieval is performed with a slightly modified version of previously described keyword based algorithm. Figure 3 shows a flowchart of this new algorithm. In contrast to classification pipeline, interest point based feature detection works better for exemplar based queries and that's what we have used in our evaluation while performing example based queries. After features are detected and their descriptors are extracted, we apply vector quantization as in classification pipeline. L1 normalization is used once again to get rid of different feature cardinalities. Spatial pyramid extension is skipped in this type of queries due to use of interest point detector. We rely on feature detection methodologies here so that relevant spatial information is represented by detected features. Next step is to insert inverse document frequencies (idf) into created feature histograms so that frequently used visual words are suppressed. After this step, BoW descriptors are ready to be compared.

We evaluate distance metrics described in section 2.3 and provide retrieval accuracies on two different datasets using the metrics.

3.3 *Inverted Index*

In order to reduce time complexity of underlying retrieval operations, we build inverted index in a way to benefit sparse structure of image descriptors. It keeps a look-up table of all images in database which contains specific codeword. i.e. dictionary entry. Figure 4 shows a graphical interpretation of our inverted index structure.

Let we have a dictionary of size K and a database containing total of N images. Furthermore, to describe sparsity of our descriptors let's also assume average number of SIFT features per image is M. Descriptor of one image consists of K numbers, one for each dictionary word. It can be represented with a K-sized vector as follows:

$$D(w) = (w_1, w_2, w_3, w_4, w_5 ... w_K) \tag{1}$$

In the worst case, at most M elements of D is non-zero. In case more than one SIFT feature of image is quantized into the same dictionary word, which is the case in practice, number of non-zero elements will be much smaller which further increases sparsity. If we were going to keep image BoW descriptors as is in our database, both our storage size and image query execution time will be linear both in dictionary size and number of images in our database. Taking into account that K is in range of millions and number of images are tens of thousands, storage requirements tents to increase very quickly, as well retrieval times. For Oxford database, N is 9000 and K is 1 million; so for each query image this results in K * N = 9 billions of basic mathematical operations. In case of more complex distance metrics containing square rooting or natural logarithm retrieval run times will increase dramatically.

On the other hand, inverted index only keeps non-zero elements of a BoW descriptor in its database, as depicted in Figure 4. Since M $\ll$ K, this reduces number of basic mathematical operations needed for calculation of query distances to M opera-

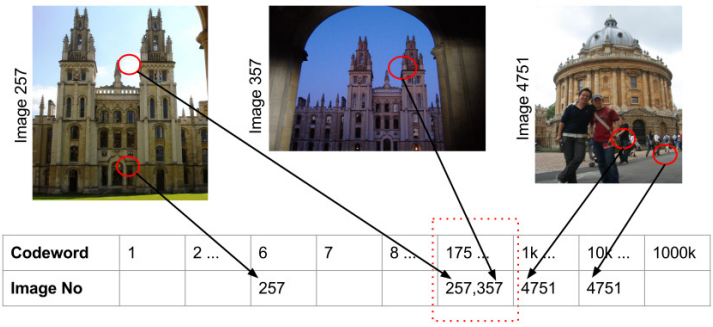

Fig. 4 Structure of the inverted index

Table 1 Summarization of Concepts for Oxford Dataset

Concept	No of Perfect Positives	No of Partial Positives	No of Total Images
All Souls	120	270	390
Ashmolean	60	65	125
Balliol	25	35	60
Bodleian	65	55	120
Christ Church	255	135	390
Cornmarket	25	20	45
Hertford	175	95	270
Keble	30	5	35
Magdalen	65	205	270
Pitt Rivers	15	15	30
Radcliffe	525	580	1105

tions per image. For Oxford database, N is 9000 and M is 3000, which results M*N = 30 million operations. There is a dramatic decrease from the case of no inverting index is used.

One difficulty of using an inverted index is that it is non-trivial to compute distance metrics since all elements of database vectors are not present in the query time. Fortunately, most of the distance metrics, if not the all, can be decomposed into at least two summation terms, one depending on query vector and the other depending on both query and database vectors. Then one can apply distance metrics for each of the query vector and only the non-zero elements of database vector. This results in almost constant time distance comparison for even heavier distance comparison metrics.

4 Experiments and Analysis

In this section, we present our experimental results for QBE and keyword based queries. We used three mostly adopted datasets by researchers in our comparisons. We evaluated our QBE queries on Oxford[17] and Paris[18] datasets, while using Pascal VOC 2007 challenge dataset[7] for keyword based queries.

4.1 Dataset Information

Oxford Buildings Dataset: Oxford dataset is composed of 5062 images containing several pictures of buildings in Oxford, along with false positives. Dataset contains 55 different queries for 11 different buildings. Table 1 summarizes the number of positive examples from each building (concept) type present in the database. In addition to those, there are also 2222 unrelated pictures containing none of the buildings.

Table 2 Summarization of Concepts for Paris Dataset

Concept	Positives	Concept	Positives	Concept	Positives	Concept	Positives
Defense	585	Musee Dorsay	360	Eiffel	1445	Notre Dame	595
Invalides	990	Pantheon	630	Louvre	760	Pompidou	255
Moulin Rouge	1158	SacreCoeur	745	Triomphe	1405		

Table 3 Summarization of Concepts for Pascal Dataset

Concept	Positives	Concept	Positives	Concept	Positives		
aeroplane	238	bus	186	dining table	200	potted plant	245
bicycle	243	car	713	dog	421	sheep	96
bird	330	cat	337	horse	287	sofa	229
bottle	244	chair	445	motorbike	245	train	261
boat	181	cow	141	person	2008	tvmonitor	256

Paris Buildings Dataset: Paris dataset is very similar to Oxford, it contains 6412 images of Paris instead of Oxford. It includes 55 different queries on 12 different Paris buildings. Concepts present in the dataset is shown in Table 2.

Pascal VOC 2007 Challenge Dataset: This dataset is used for keyword-type queries. Pascal dataset contains more than 9000 images for 20 different categories. It is one of the mostly adapted benchmark databases used by image classification community. It is well suited to retrieval tasks because it evaluates query results using retrieval metrics. Dataset contains images of 20 different objects or concepts. For each concept total of 5011 images are selected as either positive or negatives. For each concept, varying number of positives and negatives are provided. Table 3 summarizes numbers for each concept.

During implementation of our algorithms we used various publicly available software packages. OpenCV [3] is used for image processing, feature detection and K-means clustering. [15] is used for Hessian-Affine feature detection and SIFT descriptor creation. VLFeat framework available at [22] is used for homogeneous kernel mapping for SVM classification. LIBSVM [4] library is used for SVM classification. Publicly available software package of [5] is utilized for Pascal VOC 2007 experiments.

4.2 Experimental Setup

We evaluate CBIR on Oxford and Paris datasets. We run experiments with aforementioned distance comparison metrics on both datasets. Since different metrics yields different results with different descriptor normalizations, we evaluated all metrics on two types of normalization, L1 and L2. We evaluated our query results using query evaluation software packages by each dataset. Results are presented in mean average

precisions. We picked this scheme so that our results are comparable with other studies. In all of our evaluations, we created our own image features, visual dictionaries and BoW descriptors and didn't use any samples supplied by the datasets. To be comparable with other studies we evaluated 1 million words dictionaries for Oxford and Paris datasets. Local image features are detected using Hessian affine local feature detector, on top of that we extracted SIFT feature descriptors. Euclidean, L2, distance is used to compare SIFT features. We didn't apply any query expansion or spatial re-ranking techniques so that raw performance of different distance metrics can be visualized.

In contrast to QBE queries we used 5000 words for Pascal VOC 2007 challenge dataset as increasing dictionary size beyond certain limit does not increase classification precision while increasing computation time. Results are presented using the mAP score like the QBE case. It should be noted that Pascal 2007 dataset was using a slightly different average precision calculation method than Oxford and Paris datasets. As previously noted, we used dense sampling on Pascal dataset instead of using feature detectors. SIFT features are extracted at each 2 pixels. We applied 2 levels of spatial pyramids while creating final image descriptions. χ^2 homogeneous kernel mapping was applied to each image descriptor before using SVM classifiers. This greatly increased classification accuracy compared to linear SVM classification while keeping training and testing times comparable with the linear case. We apply χ^2 expansion with a gamma parameter 1.0 as lower values between [0.1, 1.0] did not yield better results. We cross-validated in training set for best SVM cost parameter and running a grid-search algorithm for cost parameter gave us a cost parameter of 20.

4.3 QBE Retrieval Results

All QBE results are summarized in Table 4. Among all metrics, Hellinger distance performed the best on all configurations. This is consistent with the results presented at [1]. Although Hellinger distance is heavier to compute compared to most of the other distances it was shown that it can be computed very efficiently with an additional normalization step to SIFT descriptor calculation[1]. This normalization step doesn't even need a modification in SIFT extraction routines, it is possible to apply normalization during quantization of image features with dictionary codewords.

Histogram intersection based distance comparison is consistently second after Hellinger distance. This is not a surprising result, it has been shown to be superior for object classification tasks by Lazebnik et al [11]. It should be noted that histogram intersection performs better with L1 on some datasets while it is performing better with L2 normalization on some other datasets. Still, with both normalization techniques are better than other distance metrics excluding Hellinger distance.

L1 and L2 distances should be used with properly normalized descriptors. Cosine distance is normalization agnostic and yields comparable results for both type of normalizations.

Table 4 Comparison of different distance metrics on Oxford and Paris datasets

Metric	Oxford5k	Paris6k	Oxford5k(No Idf)	Oxford5k	Paris6k	Oxford5k(No Idf)
	L1 Normalized			L2 Normalized		
L1	0.6176762	0.62068862	0.60776925	0.038044896	0.033984952	0.038047113
L2	0.075195491	0.043052912	0.075260796	0.61359107	0.60580742	0.59819621
Min	0.61762261	0.6501981	0.60809761	**0.62426698**	**0.64156407**	**0.61962712**
Cos	0.61361635	0.6333003	0.59808475	0.61361593	0.63330024	0.59807926
Hellinger	**0.6378966**	**0.65200478**	**0.6314373**	0.60204697	0.60970277	0.59328997
χ^2	0.59236705	0.62142205	0.58517522	0.55918133	0.5897671	0.55282593

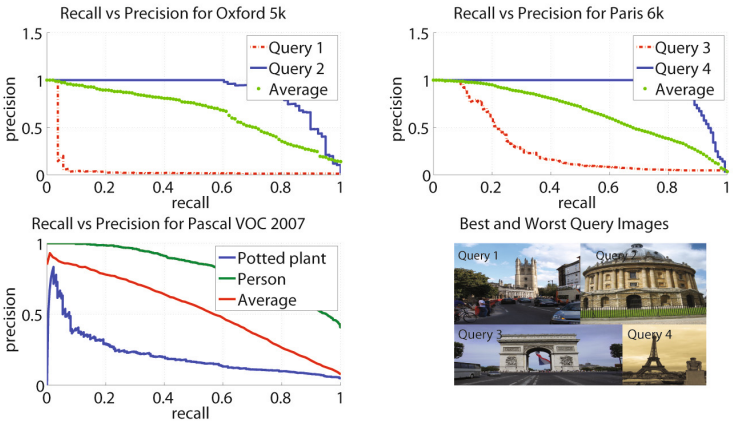

Fig. 5 Precision and recall (PR) curves. The best, average, and worst queries/concepts are considered for each dataset using the proposed scheme: (a) Oxford, (b) Paris, (c) Pascal VoC 2007.

Table 5 Keyword based retrieval results on Pascal VOC 2007 dataset

mAP	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
0.5336	0.6918	0.5594	0.3822	0.6296	0.2409	0.5955	0.7333	0.5548	0.4886	0.3893
dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tvmonitor	
0.4980	0.3627	**0.7507**	0.6346	**0.8109**	0.2372	0.4392	0.4516	0.7381	0.4842	

We also evaluated different metrics on Oxford dataset without IDF weighting scheme applied. Applying IDF weighting adds approximately 2 percent to average precision.

In addition to mAP scores in Table 4, we depicted precision-recall curves for three of the datasets we used in Figure 5. For each dataset, we included best and worst queries/concepts as well average results of each queries. By looking at the PR curves; we can conclude that while some queries achieving more than 95% AP score, some suffer from very low AP scores. Most of these under achievers belong to concepts which relatively have low number of positive samples in our database.

4.4 Keyword Based Retrieval Results

We summarized our results for Pascal VOC 2007 dataset in Table 5. Our mAP score of 53% is a compatible result to Pascal challenge best performers given at [14]. Best performers achieved 59% while average success rate is below our 53% rate. Further taking multiple feature types used by competitors into account this result is a good trade-off between classification run-time performance and accuracy. We should emphasize that it is possible to obtain up to 5% higher accuracies using different encoding schemes than vector quantization. However, using sophisticated encoding schemes greatly increases running time of algorithms. For instance, it is possible to encode one image descriptor under 1 second using vector quantization wheras sparse coding needs 30 seconds and Fisher encodings requires 12 seconds for the same image on a decent CPU[5]. We believe sacrificing query run-time performance for a relatively small increase in accuracy is not optimal for real-world applications.

5 Conclusions

BoW based image classification and object extraction techniques are known to be very successful and efficient on multimedia retrieval tasks. In this paper we represent various components of a multimedia retrieval system which can execute different types of queries on a relatively big image database by adapting inverted index structure to BoW representation. We outlined principal differences between keyword and example based queries with respect to processing pipeline implementation. We showed that with proper choice of implementation parameters, relatively simple but efficient vector quantization can compete with more sophisticated encoding schemes, such as sparse coding or Fisher kernel encoding, in retrieval accuracy maintaining a low processing overhead for database system. Another contribution of our paper was comparison of different distance metric performances while issuing QBE queries. We believe fusion of different distance metrics is a research area which may worth spending some extra time on. We believe integrating not so used correlation information between visual words into distance metric fusion frameworks may further increase of vector quantization retrieval accuracy with a little or at no cost on multimedia database side.

Acknowledgement This work is supported in part by a research grant from The Scientific and Technological Research Council of Turkey (TUBITAK EEEAG) with grant number 109E014. The authors also would like to thank Caglar Akyuz for his valuable supports.

References

1. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2911–2918, June 2012
2. Boureau, Y.-L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2559–2566, June 2010
3. Bradski, G.: Dr. Dobb's Journal of Software Tools (2000)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011)
5. Chatfield, K., Lempitsky, V.S., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC, vol. 2, p. 8 (2011)
6. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: IEEE 11th Int'l Conf. on Computer Vision, ICCV 2007, pp. 1–8. IEEE (2007)
7. Everingham, M. Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results
8. 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)
9. Jiang, Y.-G., Ngo, C.-W., Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 494–501. ACM (2007)
10. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: IEEE Int'l Conf. on Computer Vision (ICCV 2005), vol. 1, pp. 604–610, October 2005
11. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178 (2006)
12. Liu, J.: Image retrieval based on bag-of-words model (2013). CoRR, abs/1304.5168
13. Lowe, D.G.: Object recognition from local scale-invariant features. In: The proc. of the 7th IEEE Int'l Conf. on Computer Vision, 1999, vol. 2, pp. 1150–1157. IEEE (1999)
14. Marszałek, M., Schmid, C., Harzallah, H., Van De Weijer, J.: Learning object representations for visual object class recognition. In: Visual Recognition Challange Workshop, in Conjunction with ICCV (2007)
15. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 9–16. IEEE (2009)
16. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
17. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
18. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2008, pp. 1–8. IEEE (2008)
19. Sert, M., Ergun, H.: Video scene classification using spatial pyramid based features. In: 2014 22nd Signal Processing and Communications Applications Conference (SIU), pp. 1946–1949, April 2014

20. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, vol. 2, pp. 1470–1477, October 2003
21. Van De Sande, K.E., Gevers, T., Snoek, C.G.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1582–1596 (2010)
22. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008)
23. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(3), 480–492 (2012)
24. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367, June 2010
25. Yan, Z., Yu, Y.: Sparse similarity matrix learning for visual object retrieval. In: The 2013 Int'l Joint Conf. on Neural Networks (IJCNN), pp. 1–8. IEEE (2013)
26. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1794–1801, June 2009
27. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)