Int J Multimed Info Retr (2013) 2:213–225
DOI 10.1007/s13735-012-0023-3

REGULAR PAPER

Searching for images by video

Linjun Yang · Yang Cai · Alan Hanjalic · Xian-Sheng Hua · Shipeng Li

Received: 7 December 2011 / Revised: 9 August 2012 / Accepted: 16 October 2012 / Published online: 11 November 2012

Abstract Image retrieval based on the query-by-example (QBE) principle is still not reliable enough, largely because of the likely variations in the capture conditions (e.g. light, blur, scale, occlusion) and viewpoint between the query image and the images in the collection. In this paper, we propose a framework in which this problem is explicitly addressed to improve the reliability of QBE-based image retrieval. We aim at the use scenario involving the user capturing the query object by his/her mobile device and requesting information augmenting the query from the database. Reliability improvement is achieved by allowing the user to submit not a single image but a short video clip as a query. Since a video clip may combine object or scene appearances captured from different viewpoints and under different conditions, the rich information contained therein can be exploited to discover the proper query representation and to improve the relevance of the retrieved results. The experimental results show that video-based image retrieval (VBIR) is significantly more reliable than the retrieval using a single image as query. Furthermore, to make the proposed framework deployable in a practical mobile image retrieval system, where real-time query response is required, we also propose the priority queue-based feature description scheme and cache-based bi-quantization algorithm for an efficient parallel implementation of the VBIR concept.

Keywords CBIR · Video Search · Image Search · Image Search by Video

L. Yang (✉) · X.-S. Hua
Microsoft Corporation, Redmond, USA
e-mail: linjuny@microsoft.com

X.-S. Hua
e-mail: xshua@microsoft.com

Y. Cai
Zhejiang University, Zhejiang, China
e-mail: yangcai1988@gmail.com

A. Hanjalic
Delft University of Technology, Delft, The Netherlands
e-mail: a.hanjalic@tudelft.nl

S. Li
Microsoft Research Asia, Beijing, China
e-mail: spli@microsoft.com

1 Introduction

Image retrieval based on the query-by-example (QBE) principle has recently been revived and gained increasing attention from both the research community and industry. A probable reason lies in the success of the applications like Google Goggles[1], TinEye[2], and "Find more sizes" of Bing Image Search[3] that have become popular tools for retrieving images or other information related to the visual example serving as query. In particular, in the mobile use scenario, where the user can easily capture the visual query using the camera on his/her mobile device, QBE-based image retrieval appears as a highly convenient retrieval concept, as opposed to the one requiring textual keywords as queries.

Despite extensive research efforts in the past, QBE-based image retrieval is still insufficiently reliable, largely because of the likely variations in the capture conditions (e.g. light, blur, scale, occlusion) and viewpoint between the query image and the images in the collection. This query-collection mismatch has been difficult to resolve due to the still imperfect visual features used to represent the query

[1] http://www.google.com/mobile/goggles/.
[2] http://www.tineye.com/.
[3] http://www.bing.com/images.

and the collection images. While, for instance, the SIFT features [8] are effective in general, they are still insufficiently capable of handling the variations in blur and occlusion. Furthermore, in a typical SIFT-based image representation using visual words [15], the visual word quantization degrades the retrieval reliability to trade off for the scalability of the retrieval system. However, even if the problems related to the varying capture conditions can be avoided, the likely mismatch between the query and collection images in terms of the viewpoint from which an object or a scene are captured still remains the main obstacle for the successful practical adoption of QBE-based image retrieval. This obstacle is particularly critical since it makes the retrieval performance inconsistent with a user's expectations. For example, a user may expect a good retrieval result given a query image of a high quality. However, a high-quality query may perform worse than a low-quality one if the object in the high-quality query is captured from a very different viewpoint from that for the collection images. This problem is illustrated in Fig. 2 using a set of queries which are visually similar. The frames extracted from a video clip about a landmark of the Oxford University are used to query the images in the Oxford building dataset [1]. As shown in Fig. 2, the retrieval performance varies greatly if different video frames are taken individually as queries, although they all show the same object and are visually similar.

While the example in Fig. 2 is used to illustrate the query-collection mismatch problem as the main reason for unreliable QBE-based image retrieval, this example also reveals a potential effective solution to this problem. Multiple images of the same object that are characterized by different capture conditions and viewpoints could, namely, be aggregated together to extract the information for creating a more robust representation of the query object, a representation that is more complete than if any of the individual images are used as query alone. Although multiple images of the same object can be collected in various ways, a video capturing the object provides the most intuitive way to generate such a complex query, as it removes the need for the user to decide about the type and number of images to take for the same object. We therefore refer to this promising solution to the query-collection mismatch problem further as video-based image retrieval (VBIR).

Video-based image retrieval is also regarded as a useful extension to QBE video retrieval [20], which uses video query to retrieve videos in the collection. Compared with QBE-based video retrieval, VBIR can provide an alternative way to satisfy users in many application scenarios. First, although a video contains more information than a single image, it may be more convenient for users to browse image search results than video search results in a hand-held small screen device. Furthermore, video browsing suffers from adaptation problem in small screen devices. Second, the metadata accompanied with or the web pages containing an image are usually more descriptive and informative for users to understand the contained object than that for a video.

In this paper, we investigate the potential of the VBIR concept for improving QBE-based image retrieval. Due to the convenience of video capture in a mobile search scenario and the high practical importance of successfully realizing QBE-based image retrieval there, we focus on this particular scenario. As a consequence, we not only propose a method for improving the quality of retrieval results using the VBIR concept, but also a method for improving the retrieval efficiency under this retrieval concept.

The paper is organized as follows. After reviewing the related work and positioning our contribution with respect to it in Sect. 2, we provide in Sect. 3 an overview of our proposed VBIR framework. Then, in Sect. 4, we describe the key components of the framework in more detail, which is followed in Sect. 5 by the description of the proposed algorithms for improving the retrieval efficiency. The experimental evaluation of the proposed approach is presented in Sect. 6, followed by the conclusions and prospectives for future work in Sect. 7.

2 Related work

QBE-based image retrieval is one of the first retrieval paradigms introduced in the field of multimedia information retrieval, and has been extensively studied already for two decades [5,16]. Since recently, it has gained increasing attention due to a number of successful commercial applications built on this retrieval paradigm. For example, TinEye released a reverse image search engine to retrieve web pages containing the near-duplicates of the query image. Bing Image Search released a new feature called "Find more sizes", which allows users to retrieve different sizes of images that are near-duplicates of the query image. Particularly addressing the challenge of image retrieval in a mobile use scenario, Google Goggles was developed to allow search for information using an image captured by a mobile phone. The retrieval mechanisms underlying these applications are mostly based on image representation and matching using SIFT features [8] and the concept of bag-of-visual-words [13,15] derived from these features.

While the development of SIFT-based image representation solutions has been remarkable over the past several years, it is unrealistic to expect that this development could lead to a perfect image representation for any retrieval use case. Therefore, the idea behind the VBIR concept proposed in this paper is not to work towards an improved feature-based image representation, but rather to put the currently available and imperfect features into a good use, by incorporating relevant auxiliary information. Working in this direction, Yang

et al. [21] proposed to incorporate the visual context of the object captured by the query image to enrich the visual query representation and in this way improve the relevance of the retrieved images. In this paper, we enrich the query representation by drawing benefit from the information contained in the multiple frames of the query video to compensate for the deficiencies of a single-image query.

The proposed VBIR approach is partially related to several recent works in the field. In [14] Sivic et al. proposed an application to retrieve the shots in a given video similar to the query shot in terms of the object of interest captured in the query shot. There, feature tracking is used to identify the object of interest in the frames of the query shot. Then, the search is performed using different frames in the query shot individually as query images, after which the partial results are aggregated into the final search result. Compared to [14], we also use a video clip as query, but target the general (unconstrained) image search problem. Furthermore, we explicitly address the problem of improving the query representation by searching for the most stable feature points and by constructing query expansion using synonyms. Finally, we propose a comprehensive solution to VBIR including the postprocessing of the search results using reranking and taking into account the issues related to the implementation efficiency in view of the targeted mobile image search scenario.

Query expansion using an automatically constructed synonym is a well-known technique in information retrieval [11] and has been utilized in [4,18] for improving image retrieval performance. While in [4,10,18] the synonym is learned from the database, in our approach the synonym is learned from the query video, which is more effective and also more adapted to the user's current search intent. Although in both [17] and our approach video is used as the context to improve image retrieval, our proposed approach is different from [17] in that we use the video directly as query, while in [17] the video context is used to learn the parameters offline for a domain-specific image feature representation. As a result, the developed approaches are entirely different.

The research on efficient implementation of image retrieval systems has mainly focused on searching for efficient image representation features [2] and on efficient implementation of existing features [6]. Wagner et al. [19] proposed to utilize the feature tracking results to reduce unnecessary detection of feature points for an efficient implementation of image search on mobile phones. Our proposed priority queue- based feature description addresses this efficiency problem in a different way, namely by optimally using the limited time budget. The visual word quantization is often realized using fast approximate nearest neighbor search [12,13]. However, the feature points are mostly quantized independently. To exploit the redundancy across the frames in a video, we propose the cache-based bi-quantization to quantize the feature points jointly to further reduce the time cost.

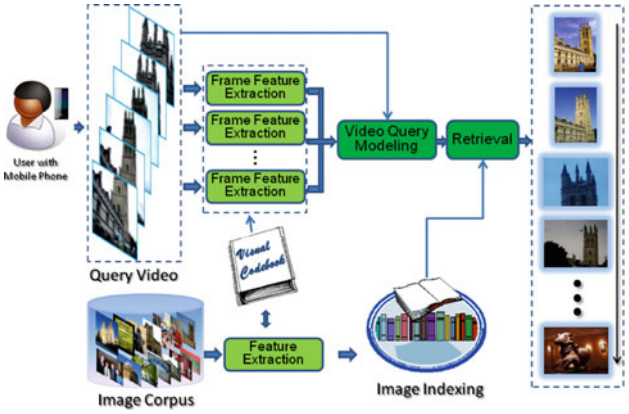

Fig. 1 System overview of the proposed video-based image retrieval (VBIR) framework

3 Video-based image retrieval

A system overview of our proposed VBIR framework is illustrated in Fig. 1. The considered use scenario is that users first capture a video clip on the object of interest using their mobile devices and then submit it to the VBIR system to retrieve images or other information relevant to the object of interest. The retrieved images can be browsed by users, or the metadata associated with these images can be presented to help users understand the observed object.

When a query video is submitted to the system it is decoded into a frame sequence and the local features, such as SIFT [8], are extracted from each frame. The features are then quantized into visual words based on already built visual codebook. The codebook is built in the same way as in a typical QBE-based image retrieval system, namely using Approximate Kmeans algorithm [13]. Then the SIFT features are mapped onto visual words using hierarchical Kmeans tree algorithm [12]. The generated visual words for all the images in the collection are indexed using the inverted file structure [11]. Furthermore, frame-level visual words aggregated over video frames are used to derive an improved query representation. Finally, we retrieve the images from the collection based on the improved query representation and present the results to users. In the following section, we focus on the core of our system, where the improved query representation is derived from multiple video frames and used to improve the retrieval results.

4 The proposed approach

Given the visual words extracted from all video frames and the temporal structure information in the query video, we need to appropriately process the video query and design a retrieval model, to draw maximum benefit from the rich

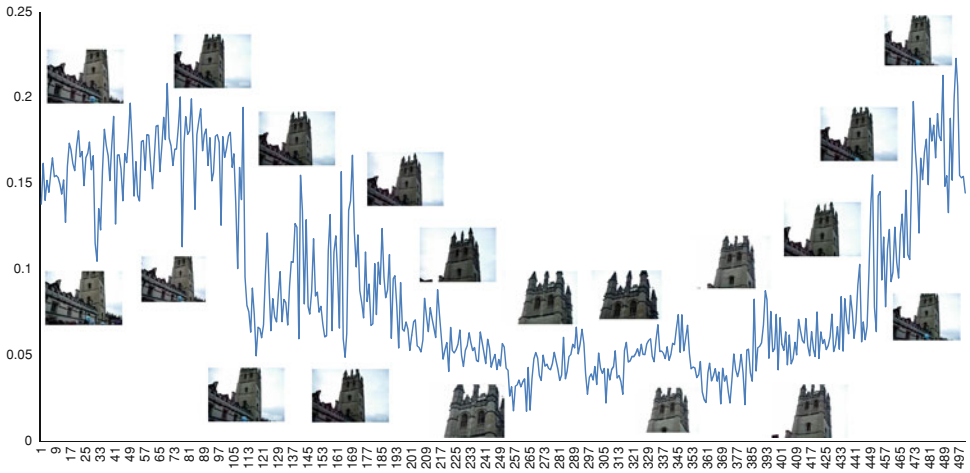

Fig. 2 Illustration of the variance in the retrieval performance using different captures of the same visual object as query. The query images are extracted from the segment beginning at 7 s and ending at 27 s of the video found at http://www.youtube.com/watch?v=ehPaPXaxQio

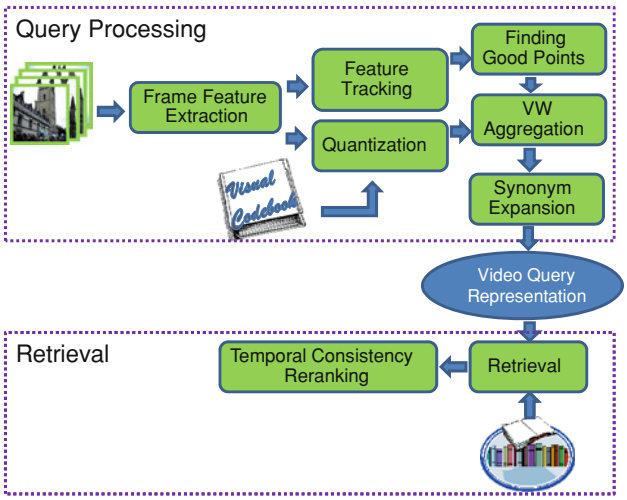

Fig. 3 The flowchart of the proposed VBIR approach zooming in on the query processing and retrieval steps

information contained in the query video. Figure 3 illustrates the flowchart of the proposed VBIR approach zooming in on the query processing and retrieval steps. In the query processing step, we first perform feature tracking among the detected SIFT feature points over adjacent video frames and then find "good" points, which are stable and therefore able to represent the query well. After that, the good points are aggregated into a histogram to obtain a first improved query representation. This representation is then further expanded based on the mined synonyms. In the retrieval step, temporal consistency reranking is introduced to further refine the search result obtained by a general image retrieval model based on the expanded query. In the following, we will describe the elements of the flowchart in Fig. 3 in more detail.

4.1 Corresponding SIFT points among frames

First, we track the SIFT points over all video frames to construct the corresponded point sequences. Here, the corresponded point sequence is defined as a sequence composed of the SIFT points in temporarily adjacent frames, which can be corresponded by tracking. The construction of corresponded point sequences is performed as follows. For each pair of temporally adjacent frames in the query video, we first track the SIFT points detected in the previous frame using Lucas and Kanade optical flow algorithm [9] implemented in OpenCV and modified using image pyramids [3]. Then, the tracked positions in the subsequent frame are further aligned to the detected SIFT points. Specifically, we find all the SIFT points which are not more than one pixel far from the tracked positions as the tracked SIFT points. The process is repeated for each pair of adjacent frames in the query video to produce many corresponded point sequences, each of which comprises a sequence of tracked SIFT points across video frames.

All the corresponded point sequences obtained using the procedure described above comprise the set $\mathcal{S}$, where $\mathbf{S}_k = \{p_i^j\}$ is the kth element of $\mathcal{S}$ and represents the kth point sequence comprising several SIFT points and p_i^j represents the jth SIFT point in the ith frame. For the convenience of implementation, those SIFT points which cannot be corresponded are also added into the point sequence set. Consequently, each of these sequences comprises only one SIFT point.

4.2 Finding good points

We assume that a good point that is reliable for retrieval should have the following properties. First, it can be tracked and corresponded in multiple adjacent frames, which states that it is stable and clearly identifiable. Second, it should gravitate towards the center of the frame, which is due to our observation that people usually tend to put the object of interest in the center of the frames when capturing a video, so the central points are more likely to be related to a users' search intent.

Based on the above assumptions, we design a set of criteria to evaluate the goodness of points. For each point p_i^j, its corresponded point sequence is denoted as $\mathbf{S}(p_i^j)$. Then, the goodness of p_i^j is defined by Eq. (1) as a combination of two terms, the *stableness* term and the *center-awareness* term,

$$G(p_i^j) = \alpha \times \frac{\text{Len}(\mathbf{S}(\mathbf{p_i^j}))}{\text{FrameCount}} + (1-\alpha) \times \text{Cent}(p_i^j). \quad (1)$$

Here, α is a parameter to control the respective contributions from the two terms, and FrameCount is the number of frames in the query video, which is used for normalization. $\text{Len}(\mathbf{S}(p_i^j))$ denotes the number of frames being tracked in the point sequence $\mathbf{S}(p_i^j)$ to represent the stableness of the point. The center-awareness term $\text{Cent}(\mathbf{S})$ is defined to reflect the assumption that the object near the center of the image is of more importance. Considering the occasional departures of intended objects from the central image area, we use the average distance of all the points in the tracked sequence to represent the center-awareness of each point in the sequence. The center-awareness of point p_i^j is defined as,

$$\text{Cent}(p_i^j) = -\frac{\sum_{p \in \mathbf{S}(\mathbf{p_i^j})} d(p,c)}{\text{Len}(\mathbf{S}(\mathbf{p_i^j})) \times d(0,c)}. \quad (2)$$

Here, d denotes the distance from point p to the frame center c, and $d(0, c)$ represents the distance from the origin of the frame to the center.

After the goodness of the points has been computed, we select those points with a goodness value larger than a threshold as good points, which will be used to construct the query model, as will be explained in the following sections.

4.3 Aggregating visual words

Given the good points selected in all the frames in the query video, we now aggregate them to construct an improved query model as a bag of corresponded visual words. For efficiency reasons, the temporal information of the points is not used in the query representation. However, we noticed that the temporal information may be important to further improve the retrieval result. Hence, we incorporate the temporal information into the reranking process described in Sect. 4.5 for a trade-off between the retrieval effectiveness and efficiency.

The query video is represented as a histogram, denoted as $\mathbf{q}$, where each bin q_i corresponds to a visual word w_i in the vocabulary. Then, for each visual word, we aggregate its occurrence in all frames, divided by the number of frames in the query video, as the value in the corresponding bin of the query histogram. Representing the query as an aggregated histogram is a convenient way to take into account all the appearances of the query object in different frames with variations including scales, viewpoints, and lighting. It utilizes the redundancy in the video to achieve a comprehensive representation of the object of interest captured by the query video. In addition, compared with that of fusing the retrieval results using different video frames as query, which requires multiple scan of the database [14], the aggregation of visual words into a single query representation makes the retrieval process more efficient.

Even though the aggregated visual words already contain rich information that should be sufficient to enable a more reliable retrieval compared to a single-image query case, we will show in the following that reliability could be improved even further, by mining the video for query synonyms to further expand the query representation.

4.4 Synonym expansion

While SIFT features are generally effective for image retrieval, different SIFT descriptors can still be extracted for the same object patch in different images, due to which similar images with large variations cannot be matched well. The visual word quantization, which is used to improve the retrieval efficiency and scalability, makes this problem even more severe since the quantization error brings additional obstacle for matching the image patches.

One of the advantages of a video compared to a single image is that it may contain a wide range of different appearances of the same object. This redundancy provides useful information for deriving the relations between the features extracted in different frames. Stavens et al. [17] used such information to learn the parameters for feature description. In this paper, this information is utilized to construct the synonym relations among visual words to partially address the imperfections due to visual word quantization.

For each visual word w_i, its term count in all frames of the query video is denoted as tc_i, and the number of points in a corresponded point sequence $\mathbf{S}_k$ being quantized as w_i is denoted as $tc_i(\mathbf{S}_k)$. Then we can construct an affinity matrix M with the element m_{ij} defined as follows,

$$m_{ij} = \frac{\sum_k \min(tc_i(\mathbf{S}_k), tc_j(\mathbf{S}_k))}{tc_i}, \quad (3)$$

with the diagonal elements set to zero.

The affinity matrix is then used to generate a contextual histogram from the aggregated query histogram so that the term counts of synonymous visual words can boost each other to alleviate the problem of quantizing similar feature descriptors into different visual words.

The contextual histogram is generated as,

$$cq = M \cdot q. \quad (4)$$

This histogram is then combined with the aggregated query histogram into the new query representation,

$$q_{\text{new}} = \beta q + (1-\beta) M \cdot q. \quad (5)$$

Using the new query representation, we can construct the vector space model based on the standard *tf-idf* scoring function known from text information retrieval to compute the similarity between the query video and images in the collection:

$$q_v = q_{new} . * idf. \tag{6}$$

Here the operator $.*$ stands for element-wise vector multiplication, while idf is a vector where idf_i represents the *idf* (inverted document frequency) of the visual word w_i.

4.5 Temporal consistency reranking

While many frames in the query video have been utilized to achieve a robust query representation, the noisy information spread in the frames may also get aggregated to produce an amplified negative effect on the query quality. Hence, to suppress this negative effect while keeping the advantages of visual word aggregation, we propose a reranking approach to adjust the search result achieved in the above steps by taking the temporal consistency of the visual content into consideration.

The reranking approach is based on our assumption that the false matches between the query video frames and the database images should not be consistent among temporally adjacent video frames. In other words, since temporally adjacent frames usually do not exhibit great changes in their appearance and all contain the object of interest, the similarity scores computed between a relevant image in the collection and adjacent frames in a video should not change greatly. However, for a mismatch, a high similarity score obtained on one video frame, e.g. due to noise in feature representation and capture conditions, will most likely be followed by a low score on the next video frame. In view of this, we choose to rerank the images in the top of the results list based on the temporal consistency information.

For each image I_i in the top of the results list, we compute the similarity scores between that image and all frames in the query video based on the vector space model with *tf-idf* weighting, denoted as $v(I_i, F_k)$, where F_k represents the kth frame in the query video. Then by regarding $v(I_i, F_k)$ as a function of k, we can compute the gradient of the function as

$$g_i^k = v(I_i, F_{k+1}) - v(I_i, F_k). \tag{7}$$

The absolute values of the gradients are then averaged to reflect the temporal consistency of the matching scores for temporally adjacent frames:

$$\tilde{g}_i = \frac{\sum |g_i^k|}{\text{FrameCount}}. \tag{8}$$

The average gradient is then combined with the similarity score computed in Sect. 4.4 to obtain a new reranking score for the top-ranked results,

$$r_i = -\tilde{g}_i + \gamma \bar{r}_i, \tag{9}$$

where $\bar{r}$ is the initial ranking score.

We noticed that some of the query videos are highly dynamic due to camera shake. For such a query video, all the images in the database may have a high average gradient, which implicitly increases the impact of temporal consistency on reranking. Actually, for the highly dynamic videos we want to decrease the contribution of temporal consistency to reranking, since even for a positive image it cannot achieve a low-average gradient in such cases. We use the mean of average gradients of the top-ranked images as the measure of the dynamics degree of the query video, which is then used to weight the average gradient term to achieve a new reranking function. In this way, the expression in Eq. (9) can be modified as

$$r_i = -\frac{\tilde{g}_i}{\frac{1}{N}\sum_{i=1}^{N} \tilde{g}_i} + \gamma \bar{r}_i, \tag{10}$$

where N is the number of top-ranked images to be considered in reranking.

5 Efficient implementation

A naïve implementation of the proposed VBIR approach may be inefficient, leading to a slow query response. Hence, to make the proposed approach applicable in a real-life mobile use scenario, where the realtime query response is required, we propose a pipeline described in this section to further improve the efficiency of the VBIR framework introduced before.

The proposed implementation is based on the client-server architecture. Users capture a video on the client computer or a mobile device and then upload it to the server, which is responsible to process the video and retrieve relevant images. The first issue to be considered is the network transferring. Based on our experiments, transferring a 6 s 10 fps 320×240 video clip over a 3G network will cost about 1.1 s. In other words, it costs on average 18 ms to transfer one video frame. Hence, by adopting the progressive uploading or streaming, the video uploading may become realtime, which means that the whole query video can be transferred to the server in a short time after the video capturing is completed.

To identify the computational bottleneck, we analyze the computational cost of the components of the proposed VBIR framework.[4] The entire query processing part costs about 650.82 ms for processing a video frame. There, the most time-consuming component is the SIFT feature extraction

[4] The experiments about the computational cost in this paper are performed on a workstation with two dual-core Intel Xeon 2.67 GHz CPUs and 12 GB memory.

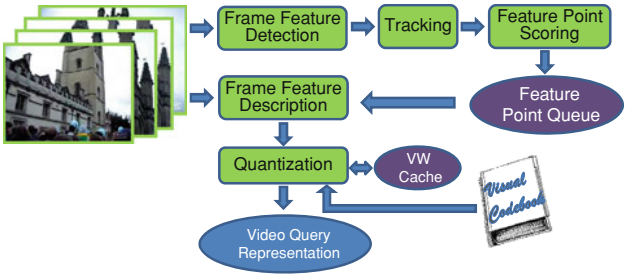

Fig. 4 The flowchart of the proposed efficient VBIR implementation

and quantization, which costs about 345.23 and 255.43 ms, respectively. The SIFT feature extraction contains two separate processes, interest point detection and description, and they cost about 99.11 and 243.01 ms, respectively. From these results, it can be observed that the feature description and quantization cost in total about 76.59 % of the computation of query processing, and therefore jointly form the bottleneck of query processing in our VBIR framework, The retrieval step, on the other hand, is efficient. It only takes 1.28 s to handle one query.

Based on the above analysis, we propose an efficient VBIR implementation, as illustrated in Fig. 4. Since the processing of different frames is independent of each other, the video query processing can easily be parallelized. We maintain a thread pool comprising three threads, and for each input video frame, a feature point detection thread is created and added into the thread pool. Since the feature description and quantization are time-consuming, we develop a priority queue-based mechanism, so that the most important feature points can be processed in a limited time budget, as will be introduced in Sect. 5.1. To further reduce the quantization time, we rely on the fact that the feature points in adjacent frames are often similar and correspond to the same visual words. Hence, we develop a cache-based bi-quantization algorithm for speed-up, as presented in Sect. 5.2.

5.1 Priority queue-based feature description and quantization

In a realtime image search system, the search results should be returned quickly after the user has finished uploading the query video. Hence, there will be only a limited time budget available for the query video processing. In such a limited time, it may be difficult to process all the detected feature points in the video. Hence, we will maintain a priority queue to keep all the detected feature points for which the description has not been extracted. For each frame, after tracking, the newly detected feature points will be enqueued and the priority of points in the former frames will be updated based on Eq. (1). The feature description thread will continuously fetch the feature points from the queue for processing, until the queue is empty or the time budget has been used up.

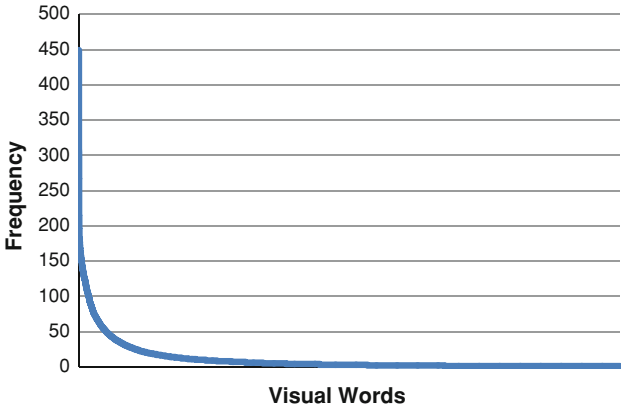

Fig. 5 The frequency of visual words in the query video All_Souls_v1

5.2 Cache-based bi-quantization

The proposed cache-based bi-quantization algorithm is motivated by the fact that there is a *local consistency* in the visual word quantization for adjacent frames in a query video. Since the adjacent frames are normally very similar to each other, the descriptions of the feature points in adjacent frames also tend to be similar to each other and the quantized words would be identical. Figure 5 shows the occurrence times of each visual word into which the feature points in a query video are quantized. We can see that 6.60 % visual words occur more than 40 times in the query video All_Souls_v1, which corresponds to 50 % feature points.

To utilize the local consistency, we propose a cache-based bi-quantization algorithm, as shown in Algorithm 1. Since the visual word quantization is normally performed using approximate nearest neighbor search, such as *k–d* trees [12], we can adjust the search parameters to trade-off the precision and the time cost. In our approach, we built two quantization methods. One is Q_{h}, slow but with high precision, and the

Algorithm 1 Cache-based bi-quantization

Require: a high-precision quantizer Q_h, a low-precision quantizer Q_l, and N_l which is the frame interval for cache refreshing.

Initialization Set cache $\mathcal{C} = \varnothing$.
for Frame $F_i = F_1$ to F_N **do**
  for all Feature point P_j in F_i **do**
    Quantize P_j into visual word W_j using Q_l: $W_j = Q_l(P_j)$;
    if $W_j \in \mathcal{C}$ **then**
      continue;
    else
      $W_j = Q_h(P_j)$;
      $\mathcal{C} = \mathcal{C} \cup \{W_j\}$;
    end if
  end for
  if i%N_l==0 **then**
    $\mathcal{C} = \varnothing$;
  end if
end for

Table 1 Summary of query video clips used in the experiments

Query id	Video url	Begin time (s)	End time (s)
All_Souls_v1	http://www.youtube.com/watch?v=C1hwL-QHiec	0	6
All_Souls_v2	http://www.youtube.com/watch?v=V-sn0vkVYXo	77	86
All_Souls_v3	http://www.youtube.com/watch?v=V-sn0vkVYXo	152	159
All_Souls_v4	http://www.youtube.com/watch?v=V-sn0vkVYXo	260	280
Ashmolean_v1	http://www.youtube.com/watch?v=2g8G2XDJZZ4	6	11
Bodleian_Library_v1	http://www.youtube.com/watch?v=oGkHvCa1hRQ	6	13
Bodleian_Library_v2	http://www.youtube.com/watch?v=Mxjue1nf6oE	3	8
Bodleian_Library_v3	http://www.youtube.com/watch?v=XxNhfgL0nUk	28	33
Christ_Church_v1	http://www.youtube.com/watch?v=L3mvKQorVRY	16	18
Christ_Church_v2	http://www.youtube.com/watch?v=CCOMJ3boZIY	18	21
Christ_Church_v3	http://www.youtube.com/watch?v=o4ywV2cQ0Q4	7	10
Christ_Church_v4	http://www.youtube.com/watch?v=o4ywV2cQ0Q4	15	18
Christ_Church_v5	http://www.youtube.com/watch?v=o4ywV2cQ0Q4	195	199
HertFord_v1	http://www.youtube.com/watch?v=jtgRA9Abxs4	9	17
HertFord_v2	http://www.youtube.com/watch?v=OwxYkWwsgLE	143	144
HertFord_v3	http://www.youtube.com/watch?v=OwxYkWwsgLE	158	160
HertFord_v4	http://www.youtube.com/watch?v=OwxYkWwsgLE	168	171
Kebel_College_v1	http://www.youtube.com/watch?v=KpuC-yj_uc0	11	14
Magdalen_College_v1	http://www.youtube.com/watch?v=ehPaPXaxQio	7	27
Radcliffe_Camera_v1	http://www.youtube.com/watch?v=C1hwL-QHiec	52	60
Radcliffe_Camera_v2	http://www.youtube.com/watch?v=jtgRA9Abxs4	64	69
Radcliffe_Camera_v3	http://www.youtube.com/watch?v=qhAVFlSwQ3c	16	18
Radcliffe_Camera_v4	http://www.youtube.com/watch?v=Pf6JHXhUgtg	52	59
Radcliffe_Camera_v5	http://www.youtube.com/watch?v=Pf6JHXhUgtg	67	90
Radcliffe_Camera_v6	http://www.youtube.com/watch?v=OwxYkWwsgLE	210	216

other one is Q_l, which is fast but with a low precision. For each feature point, we firstly use Q_l to get a rough quantization with a small time cost. To verify the reliability we check whether the quantized word has appeared in the cache. If so, it should be a reliable quantization. If not we further achieve a reliable quantization using the high-precision quantizer Q_h. For every N_l frames, the cache will be cleared and refreshed to maintain the locality. In this paper, we simply set $N_l = 20$.

6 Experiments

6.1 Experimental setup

To set a benchmark for VBIR and allow for comparison of other methods with the approach proposed in this paper, we first chose the publicly available Oxford building dataset [1] as the image collection. Then, as explained in Sect. 6.5, we also expanded our investigation to a larger, web-scale image collection for the purpose of a more comprehensive evaluation of the algorithm performance. The Oxford building dataset comprises 5,062 images crawled from Flickr[5] using 11 landmarks of Oxford University as queries. To collect the query videos for our experiments, we used the 11 landmark names as query to search for suitable videos in YouTube. Finally, we obtained 15 videos for 8 landmarks, while for the other 3 landmarks we were not able to find any relevant videos. Since not all the parts in these videos are about the corresponding landmarks, we selected those segments exactly describing the landmarks as query video clips in our experiments. Consequently, 25 video clips were selected as queries, which are summarized in Table 1. The ground-truth corresponding to each landmark in the Oxford building dataset is still used as the ground-truth for the video-based image retrieval experiments. The key frames in the query videos and the images in the database were down-sampled to 300 × 300 by preserving the aspect ratio and then SIFT features were extracted. A 100 K visual vocabulary was constructed using Approximate Kmeans [7].

The average precision (AP) is used to evaluate the retrieval performance. AP is defined as the average of the precisions computed at all recall levels. The mean average precision (MAP) is the average of the APs across all queries.

6.2 Performance comparison

We implemented QBE-based image retrieval (QBEIR) as a baseline to be compared with the proposed VBIR concept. Specifically, we used each frame in the query videos individually to query the image database to simulate QBE-based image retrieval. The average performance (QBEIR_E)

[5] http://www.flickr.com.

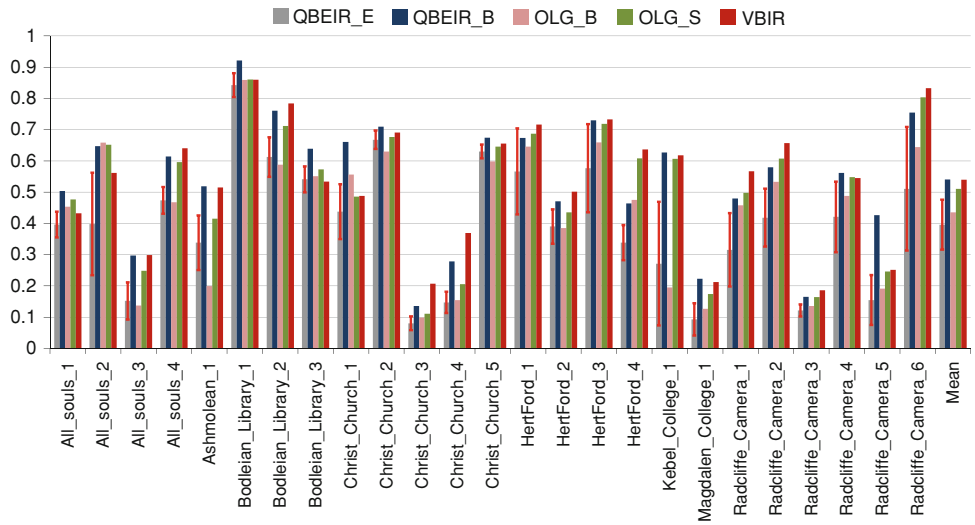

Fig. 6 The performance comparison of QBE based image retrieval and VBIR on the Oxford building dataset. QBEIR_E shows the mean and the standard deviation of the MAP of image retrieval using single frames in the query video, QBEIR_B is the best possible performance of retrieval using a single frame, and VBIR shows the MAP of video-based image retrieval

with standard deviation and the best possible performance (QBEIR_B) for each query video using different frames as query are illustrated in Fig. 6. The methods in [14], which fuse the retrieval scores using each frame individually as query are also implemented and used for comparisons, including fusion by summing all scores (OLG_S) and fusion by taking the maximum (OLG_B).

We can see from Fig. 6 that QBE-based image retrieval suffers from a dramatic performance variation, which demonstrates its insufficient reliability. Intensive camera motion, e.g., zoom in/out in HertFord_v1, and large changes of light conditions caused by different shooting angles in Christ_Church_v1 and Radcliffe_Camera_v1 cause that the object of interest is described at a broad range of capture conditions, which can only in part match the conditions at which collection images have been captured. This causes large variation in the retrieval performance if video frames are used individually as query. However, such information can be put into a good use to improve the retrieval performance by using the whole video clip as query, as proposed in this paper.

The performances of VBIR and QBE-based image retrieval are compared in Fig. 6. We can see that the performance of VBIR is significantly better than the expected performance of QBE-based image retrieval. The improvement was computed as 36.37 % in terms of MAP. Furthermore, for 24 among 25 query videos VBIR can achieve a performance boost compared to the average performance of QBE-based image retrieval. In particular, the MAP of VBIR is even larger than the expected performance of QBEIR by a margin of the standard deviation for 19 queries. This demonstrates that the incorporation of the information contained in all video frames has a clear potential for improving the reliability of the retrieval performance. Finally, we can see that VBIR achieves a comparable result with the best possible performance of QBEIR. This means that, using a video as query, we can achieve a result similar to the best one achievable using an arbitrary single image as query.

The proposed VBIR approach further outperforms OLG_B and OLG_S by 23.95 and 5.81 % and in 84 and 80 % queries, respectively. While VBIR achieves a better performance compared to its competitors, it also exhibits a better efficiency in terms of the retrieval part. While OLG_B and OLG_S cost 5.2 s to complete the retrieval part for one query, VBIR only needs 1.2 s for the same task.

For those videos that exhibit significant camera motion while introducing new information such as the object at multiple scales or viewpoints, e.g., Magdalen_College_1 and All_Souls_2, incorporating multiple frames into the query representation significantly improves the retrieval performance. However, for those videos in which all frames have the same scale and viewpoint, e.g. Christ_Church_2, VBIR cannot provide a large benefit since a video in that case hardly contains more useful information than a single image. We believe, however, that when users search for something using videos, it is realistic to expect that camera will move and zoom in/out will be deployed to capture more information. In this sense, VBIR is expected to boost the retrieval performance in most cases, compared to QBE-based image search.

6.3 Analysis of the proposed approach

To analyze the effects of each step of our proposed approach, we compare the intermediate results of the three steps, including filter and aggregation, synonyms mining, and temporal consistency reranking in Fig. 8. Since the result after temporal consistency reranking is just the result of the complete approach, it is denoted as VBIR in the figure.

It can be observed that by introducing the filter and aggregation step the performance is improved by 0.130 over QBE-based image retrieval and that the performance is boosted

Fig. 7 The examples of "good" feature points. The *green* points are good points and the *red* ones are those being filtered out.

for 24 queries. We argue that the reasons for this effect are twofold. First, in the "finding good points" step, the noisy SIFT points are filtered out so that they do not have a negative effect on the retrieval. For example, as shown in Fig. 7, the trees in the background are filtered out by our approach. Second, aggregating visual words over all frames collects the appearances of the object taken under different conditions, so that a more comprehensive representation of the query object is constructed to improve the retrieval performance. This is especially useful when a single image can only capture a partial view of the object of interest, which is likely to happen if the user stands near the object with a common camera without ultra-wide-angle lens. For instance, each frame in query Radcliffe_Camera_v2 only contains a part of the building while aggregation brings us a full view of the object and therefore a better retrieval performance.

By incorporating the synonyms mining, the performance is further improved by 0.006, as shown in Fig. 8. Moreover, we can see that for a large majority of queries, introducing the synonyms boosts the retrieval performance. Among them, we observe that a video with drastic camera motion, e.g., zoom in/out, such as HertFord_v1, and light condition changes like Radcliffe_Camera_v1 and Christ_Church_v1, tends to achieve a larger improvement. Since such videos contain different views of the object of interest due to the camera motion and light condition changes, the synonyms mining can discover the correlation between the visual words under different views or scales. By including the visual word correlation into the retrieval process, the system reliability is further improved.

The temporal consistency reranking step further contributes 0.008 to the overall performance. As illustrated in Fig. 9, the temporal consistency assumption is verified to discriminate the positive from negative images. In such query videos, the incorporation of temporal consistency reranking improves the performance significantly. However, we also notice that for some videos the temporal consistency reranking even degrades the performance. For example, on the query HertFord_v1, the performance is degraded by 0.001 after reranking. By observing the video clip, we found that this query video contains a significant shot (dissolve) change, which breaks the temporal consistency assumption. However, we note that in a real-life VBIR system, a user-captured short video clip is not expected to contain shot changes.

6.4 Efficiency

While the above experiments demonstrate that the proposed VBIR approach is effective, it will cost 650.82 ms to process one frame in a query video and therefore it is difficult to apply in a search system where realtime query response is required. In this subsection, we will show that after adopting the proposed efficient implementation, the time can be reduced to less than 300 ms, which makes the system able to

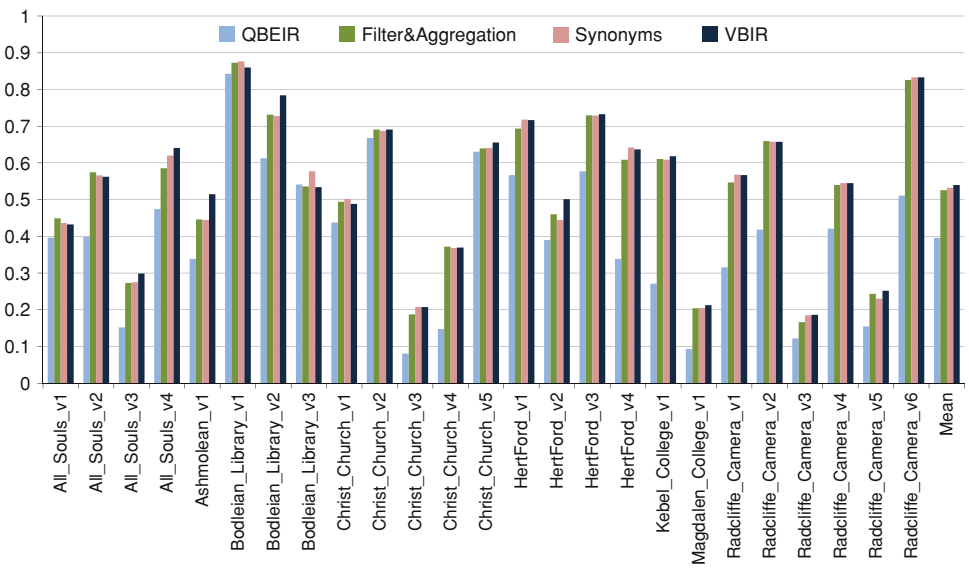

Fig. 8 The performance of each step in our proposed approach

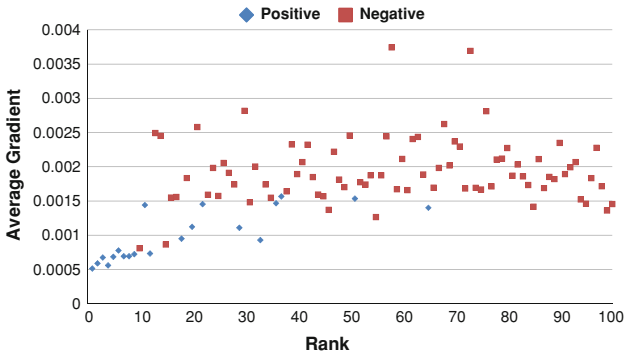

Fig. 9 The distribution of the average gradients for top 100 images of the query All_Souls_v2.

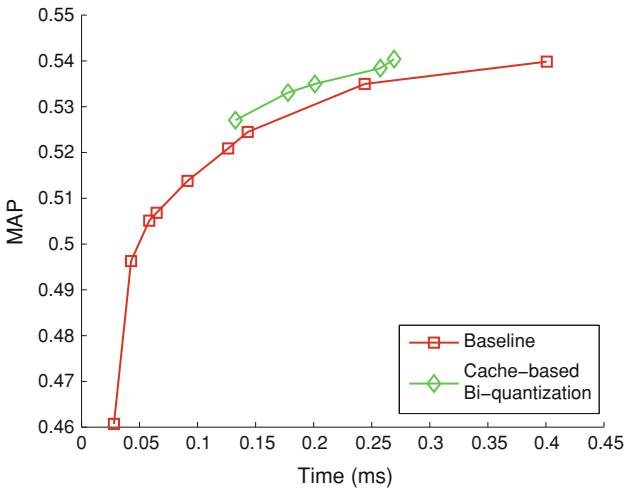

Fig. 10 The performance comparison between the cache-based bi-quantization and the baseline quantization method.

process a 10 fps query video in real time with three threads on a normal computer.

Figure 10 shows a comparison between the proposed cache-based bi-quantization and the baseline quantization approach that quantizes each feature points independently. It can be observed that under the same time budget, the cache-based bi-quantization approach can achieve a better MAP than the baseline. In other words, to achieve the same effectiveness, the cache-based bi-quantization can perform faster. Further, the MAP of the cache-based bi-quantization with Q_h being a 0.9 precision quantizer and Q_l a 0.5 precision quantizer is 0.5404 and better than that of the baseline quantizer with precision 0.9. However, the quantization time is reduced by 32.80 %, from 0.4 to 0.27 ms for one feature point. Hence, in our experiment, we used the cache-based bi-quantization with 0.9 precision Q_h and 0.5 precision Q_l.

From Fig. 11 we can see that the cache-based bi-quantization without cache refresh achieves the highest speed but the lowest MAP, which validates the locality of the visual words consistency and demonstrates the necessity for cache refresh. Based on the result, we can set the refresh interval

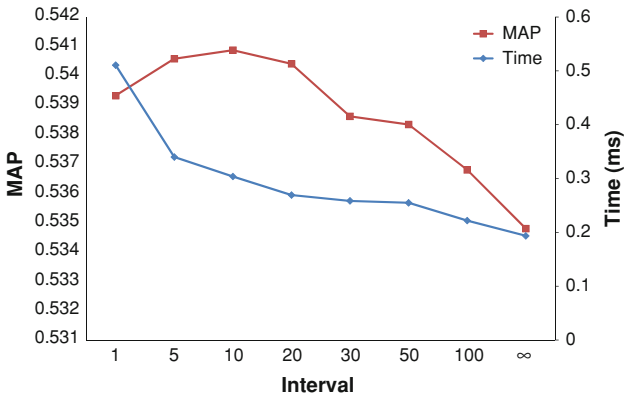

Fig. 11 MAP and time for cache-based bi-quantization with different intervals for refreshing the cache. ∞ means no cache refresh.

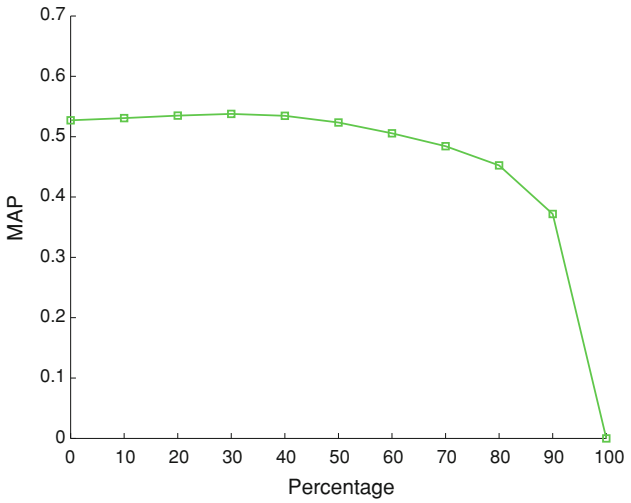

Fig. 12 MAP for different percentages of visual words to be filtered out.

to a moderate size, e.g., 20 frames, to achieve a trade-off between the effectiveness and efficiency.

To study the relationship between the MAP and the time cost of the priority queue-based feature description and quantization, we illustrate in Fig. 12 the respective MAP of filtering out different percentages of feature points to reduce the time cost. We can see that by filtering a small amount of points (less than 30 %) the performance even improves over that using all feature points. This demonstrates the effectiveness of the proposed feature filtering step described in Sect. 4.2. By filtering 70 % feature points, we can still achieve 0.4841 MAP, which improves 22.3 % over QBEIR. But the total time for query processing is reduced to 283.12 ms, which can be completed in realtime using three threads.

6.5 Experiment on a large-scale dataset

To further demonstrate the effectiveness of the proposed VBIR approach, we performed another experiment on a

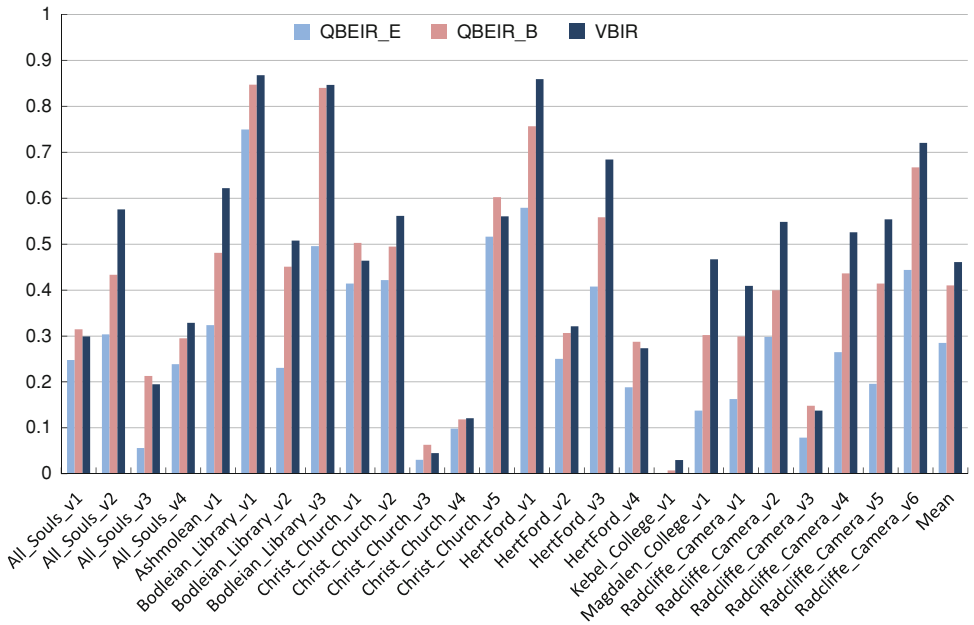

Fig. 13 The performance of VBIR on a large-scale dataset comprising 1M images

large-scale dataset. The videos of Oxford university buildings we crawled from YouTube were still employed as queries, but the database was composed of not only the images in Oxford building dataset, but also one million images collected from Flickr. Finally the database comprises totally 1,004,834 images. The performance of our proposed VBIR approach, and the expected and the best performance of QBE are shown in Fig. 13.

By comparing Figs. 6 and 13 we can see that when the database scales up, the retrieval performance of QBE drops significantly. Specifically, the MAP of QBEIR_E decreases from 0.396 to 0.285 when the scale of the database increases from 5 K to 1 M. This shows that the QBE-based approach is less robust and less scalable than VBIR, which still achieves 0.461 MAP for 1 M database. Moreover, we found that on the large-scale dataset, VBIR even outperforms the best possible performance of QBE (QBEIR_B), which further demonstrates the effectiveness and reliability of VBIR.

7 Conclusion and future work

We proposed in this paper a new image search framework that we refer to as the VBIR framework. VBIR makes it possible to search for images and related information using a short video clip taken on the object of interest as query. The approach underlying the proposed framework includes mining of the useful information from all frames of the query video and using this information to refine the query representation, and in this way improve the retrieval performance. The experimental results show that VBIR significantly improves the retrieval reliability over that of using a single image as query.

Since VBIR as a paradigm is particularly of importance for the mobile use scenario, where visual queries are captured using a mobile device, we also addressed the efficiency of VBIR framework implementation to make it deployable in a practical (mobile) use case.

We envision four main directions for future work building on the insights presented in this paper. First, we will focus on utility aspects of the VBIR concept and further investigate the expected properties of the acquired query videos and their relation to a users' search intent based on the typical user behavior when capturing videos for VBIR. Domain-related insights collected here will help us to further improve the framework from the design and implementation perspective. Second, we will expand the VBIR concept to investigate other possibilities for drawing benefit from the rich information contained in a video to improve the effectiveness and efficiency of query representation. One possibility is to generate a 3D object model from the query video clip and then use that to retrieve images in the database. The other is to discover useful information from the aspects typical for the video nature of the query, like motion patterns, to efficiently process the video query and prepare it for search. Third, the efficiency of the VBIR implementation could be further improved by relying on more efficient features, e.g. SURF [2]. To identify the fourth direction for future research, we note again that the goal of this paper was to investigate the potential of VBIR to improve the image search performance relative to the conventional search using a single image as query. However, to achieve significant improvement in the search performance in the absolute sense, a broader investigation

is required involving the criteria related to the quality of the video query, and in particular, the cases where the query does not optimally capture the object of interest, e.g. due to occlusion or insufficient focus. Construction of the representative sets of video queries for this purpose and identifying the possibilities to effectively cope with sub-optimal video queries are therefore important future steps in bringing VBIR to the following development stage.

References

1. http://www.robots.ox.ac.uk/~vgg/data/oxbuildings
2. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: ECCV
3. Bradski G, Kaehler A (2008) Learning openCV: computer vision with the openCV library. O'Reilly, Cambridge
4. Chum O, Philbin J, Sivic J, Isard M, Zisserman A (2007) Total recall: automatic query expansion with a generative feature model for object retrieval. In: CVPR
5. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):5:1–5:60
6. Heymann S, Muller K, Smolic A, Frohlich B, Wiegand T (2007) SIFT implementation and optimization for general-purpose GPU. In: Proceedings of the international conference in Central Europe on computer graphics, visualization and computer vision
7. Li D, Yang L, Hua XS, Zhang HJ (2010) Large-scale robust visual codebook construction. In: ACM multimedia
8. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. doi:10.1023/B:VISI.0000029664.99615.94
9. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of the 1981 DARPA imaging understanding, workshop
10. Makadia A (2010) Feature tracking for wide-baseline image retrieval. In: ECCV
11. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval, 1 edn. Cambridge University Press, Cambridge
12. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: VISSAPP
13. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: CVPR
14. Sivic J, Schaffalitzky F, Zisserman A (2006) Object level grouping for video shots. Int J Comput Vis 67(2):189–210
15. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. In: ICCV
16. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Patt Anal Mach Intell 22:1349–1380
17. Stavens D, Thrun S (2010) Unsupervised learning of invariant features using video. In: CVPR
18. Turcot P, Lowe D (2009) Better matching with fewer features: the selection of useful features in large database recognition problems. In: ICCV workshop (WS-LAVD)
19. Wagner D, Schmalstieg D, Bischof H (2009) Multiple target detection and tracking with guaranteed framerates on mobile phones. In: ISMAR
20. Wu X, Hauptmann AG, Ngo CW (2007) Practical elimination of near-duplicates from web video search. In: ACM multimedia
21. Yang L, Geng B, Hanjalic A, Hua XS (2010) Contextual image retrieval model. In: CIVR