INVITED PAPER

Efficient Visual Search for Objects in Videos

Visual search using text-retrieval methods can rapidly and accurately locate objects in videos despite changes in camera viewpoint, lighting, and partial occlusions.

By Josef Sivic and Andrew Zisserman

ABSTRACT | We describe an approach to generalize the concept of text-based search to nontextual information. In particular, we elaborate on the possibilities of retrieving objects or scenes in a movie with the ease, speed, and accuracy with which Google [9] retrieves web pages containing particular words, by specifying the query as an image of the object or scene. In our approach, each frame of the video is represented by a set of viewpoint invariant region descriptors. These descriptors enable recognition to proceed successfully despite changes in viewpoint, illumination, and partial occlusion. Vector quantizing these region descriptors provides a visual analogy of a word, which we term a "visual word." Efficient retrieval is then achieved by employing methods from statistical text retrieval, including inverted file systems, and text and document frequency weightings. The final ranking also depends on the spatial layout of the regions. Object retrieval results are reported on the full length feature films "Groundhog Day," "Charade," and "Pretty Woman," including searches from within the movie and also searches specified by external images downloaded from the Internet. We discuss three research directions for the presented video retrieval approach and review some recent work addressing them: 1) building visual vocabularies for very large-scale retrieval; 2) retrieval of 3-D objects; and 3) more thorough verification and ranking using the spatial structure of objects.

KEYWORDS | Object recognition; text retrieval; viewpoint and scale invariance

I. INTRODUCTION

The aim of this research is to retrieve those key frames and shots of a video containing a particular object with the ease, speed, and accuracy with which web search engines such as Google [9] retrieve text documents (web pages) containing particular words. An example visual object query and retrieved results are shown in Fig. 1. This paper investigates whether a text retrieval approach can be successfully employed for this task.

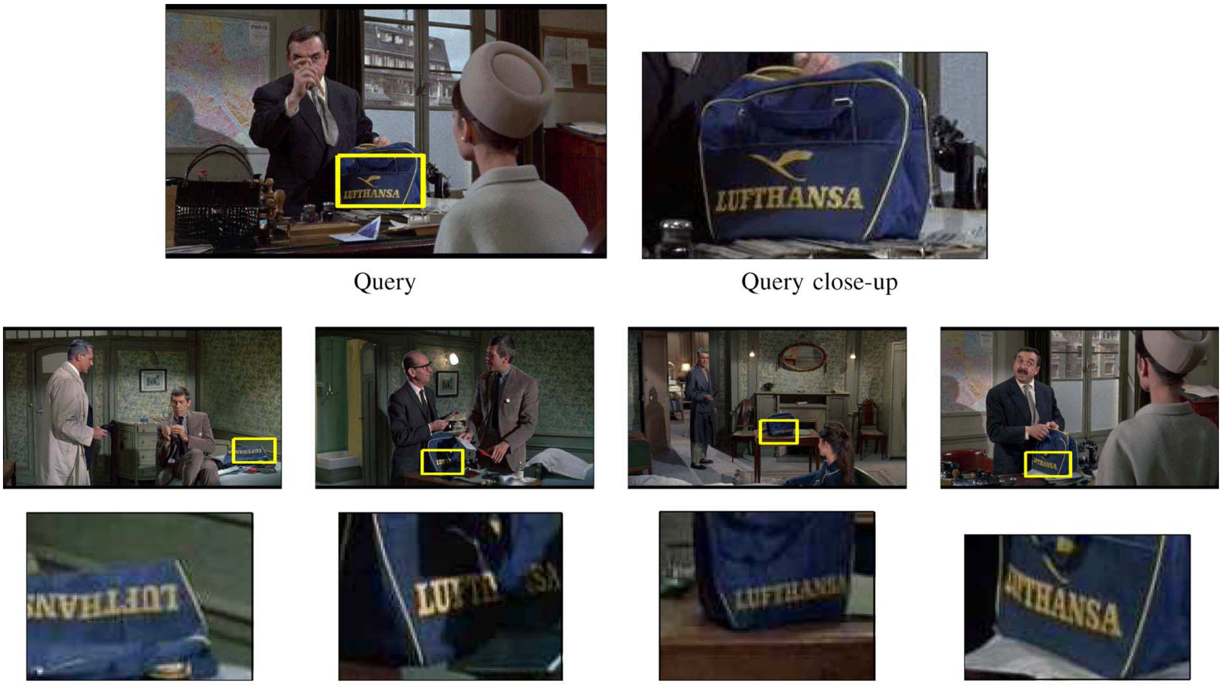
Identifying an (identical) object in a database of images is a challenging problem because the object can have a different size and pose in the target and query images, and also the target image may contain other objects ("clutter") that can partially occlude the object of interest. However, successful methods now exist which can match an object's visual appearance despite differences in viewpoint, lighting, and partial occlusion [22]–[24], [27], [32], [38], [39], [41], [49], [50]. Typically, an object is represented by a set of overlapping regions each represented by a vector computed from the region's appearance. The region extraction and descriptors are built with a controlled degree of invariance to viewpoint and illumination conditions. Similar descriptors are computed for all images in the database. Recognition of a particular object proceeds by nearest neighbor matching of the descriptor vectors, followed by disambiguating or voting using the spatial consistency of the matched regions, for example by computing an affine transformation between the query and target image [19], [22]. The result is that objects can be recognized despite significant changes in viewpoint, some amount of illumination variation and, due to multiple local regions, despite partial occlusion since some of the regions will be visible in such cases. Examples of extracted regions and matches are shown in Figs. 2 and 5.

In this paper, we cast this approach as one of text retrieval. In essence, this requires a visual analogy of a word, and here we provide this by vector quantizing the

Manuscript received June 10, 2007; revised November 25, 2007. This work was supported in part by the Mathematical and Physical Sciences Division, University of Oxford and in part by EC Project Vibes.

The authors are with the Department of Engineering Science, University of Oxford, OX1 3PJ Oxford, U.K. (e-mail: josef@robots.ox.ac.uk; az@robots.ox.ac.uk).

Digital Object Identifier: 10.1109/JPROC.2008.916343



Query

Query close-up

Four retrieved shots shown by a single frame (top) and object close-up (bottom)

Fig. 1. ***Object query example I. Frames from top six ranked shots retrieved from a search of the movie "Charade" for visual query shown. Querying the entire movie took 0.84 s on a 2-GHz Pentium. A live demonstration of object retrieval system is available online [3].***

descriptor vectors. The benefit of the text retrieval approach is that matches are effectively precomputed so that at run time frames and shots containing any particular object can be retrieved with no delay. This means that any object occurring in the video (and conjunctions of objects) can be retrieved even though there was no explicit interest in these objects when descriptors were built for the video.

Note that the goal of this research is to retrieve instances of a specific object, e.g., a specific bag or a building with a particular logo (Figs. 1 and 2). This is in contrast to retrieval and recognition of "object/scene categories" [8], [11], [13], [14], [35], [44], sometimes also called "high-level features" or "concepts" [4], [47] such as "bags," "buildings," or "cars," where the goal is to find any bag, building, or car, irrespective of its shape, color, appearance, or any particular markings/logos.

We describe the steps by which we are able to use text retrieval methods for object retrieval in Section II. Then in Section III, we evaluate the proposed approach on a ground truth set of six object queries. Object retrieval results, including searches from within the movie and specified by external images, are shown on feature films: "Groundhog Day" [Ramis, 1993], "Charade" [Donen, 1963] and "Pretty Woman" [Marshall, 1990]. Finally, in Section IV we discuss three challenges for the presented video retrieval approach and review some recent work addressing them.

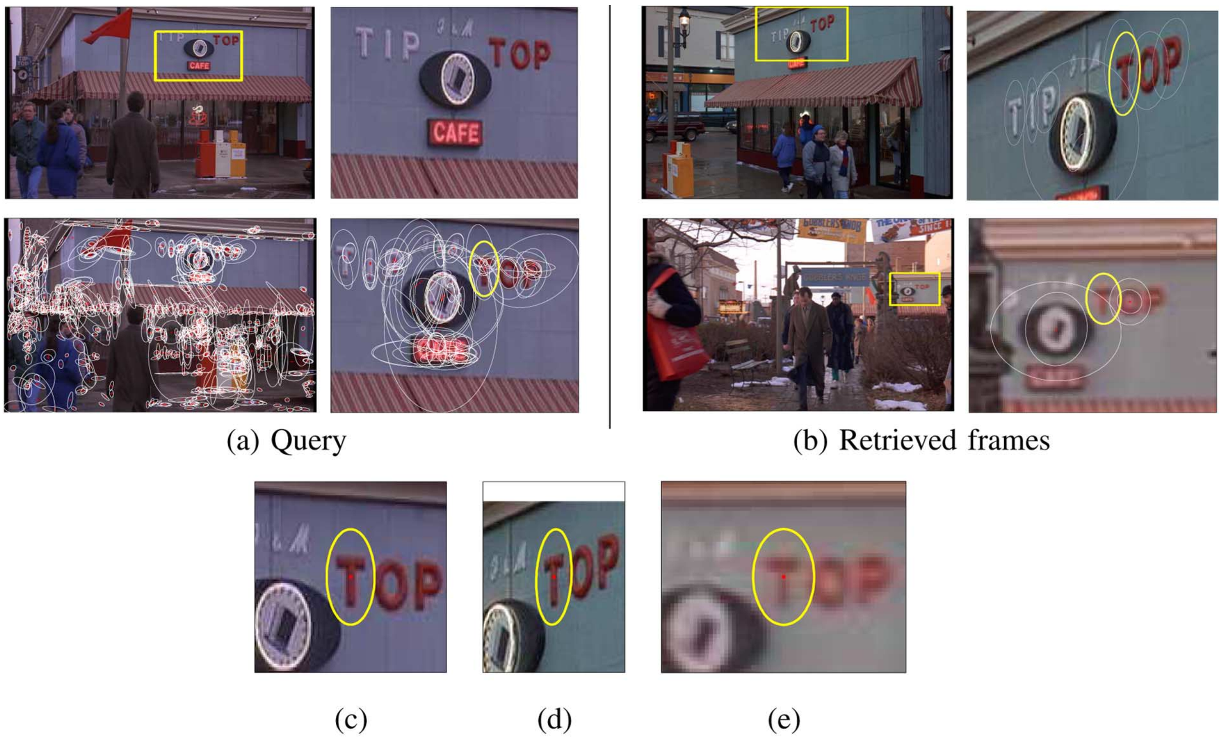
II. TEXT RETRIEVAL APPROACH TO OBJECT MATCHING

This section outlines the steps in building an object retrieval system by combining methods from computer vision and text retrieval.

Each frame of the video is represented by a set of overlapping (local) regions with each region represented by a visual word computed from its appearance. Section II-A describes the visual regions and descriptors used. Section II-B then describes their vector quantization into visual "words." Sections II-C and II-D then show how text retrieval techniques are applied to this visual word representation. We will use the film "Groundhog Day" as our running example, though the same method is applied to all the feature films used in this paper.

A. Viewpoint Invariant Description

The goal is to extract a description of an object from an image which will be largely unaffected by a change in camera viewpoint, object's scale, and scene illumination, and also will be robust to some amount of partial occlusion. To achieve this we employ the technology of viewpoint covariant segmentation developed for wide baseline matching [27], [32], [39], [49], [50], object recognition [22], [32], and image/video retrieval [41], [46]. The idea is that regions are detected in a viewpoint



(a) Query

(b) Retrieved frames

(c)

(d)

(e)

Fig. 2. ***Object query example II. (a) Top row: (left) frame from the movie "Groundhog Day" with outlined query region and (right) close-up of query region delineating object of interest. Bottom row: (left) all 1039 detected affine covariant regions superimposed and (right) close-up of query region. (b) (Left) two retrieved frames with detected regions of interest and (right) close-up of images with affine covariant regions superimposed. These regions match to a subset of the regions shown in (a). Note significant change in foreshortening and scale between query image of object and object in retrieved frames. (c)–(e) Close-ups of one of the affine covariant regions matched between query (c) and retrieved frames (d), (e). Note that regions are detected independently in each frame, yet cover the same surface area on building facade (the letter "T").***

covariant manner—so that for images of the same scene, the pre-image of the region covers the same scene portion. This is illustrated in Fig. 2. It is important to note that the regions are detected *independently* in each frame. A "region" simply refers to a set of pixels, i.e., any subset of the image. These methods differ from classical detection and segmentation since the region boundaries do not have to correspond to changes in image appearance such as color or texture. A comprehensive review of viewpoint covariant (also called affine covariant) region detectors, and a comparison of their performance, can be found in [29].

In this paper, two types of affine covariant regions are computed for each frame of the video. The first is constructed by elliptical shape adaptation about a Harris [18] interest point. The implementation details are given in [27] and [39]. This region type is referred to as shape adapted (SA). The second type of region is constructed by selecting areas from an intensity watershed image segmentation. The regions are those for which the area is approximately stationary as the intensity threshold is varied. The implementation details are given in [26]. This region type is referred to as maximally stable (MS).

Two types of regions are employed because they detect different image areas and thus provide complementary representations of a frame. The SA regions tend to be centered on corner-like features, and the MS regions correspond to blobs of high contrast with respect to their surroundings such as a dark window on a grey wall. Both types of regions are represented by ellipses. These are computed at twice the originally detected region size in order for the image appearance to be more discriminating. For a 720 × 576 pixel video frame the number of regions computed is typically 1200. An example is shown in Fig. 2.

Each elliptical affine covariant region is represented by a 128-dimensional vector using the SIFT descriptor developed by Lowe [22]. This descriptor measures orientation of image intensity gradients, which makes it robust to some amount of lighting variations. In [28], the SIFT descriptor was shown to be superior to others used in the literature, such as the response of a set of steerable filters [27] or orthogonal filters [39]. One reason for this superior performance is that SIFT, unlike the other descriptors, is designed to be invariant to a shift of a few pixels in the region position, and this localization error is one that often occurs. Combining the SIFT descriptor with

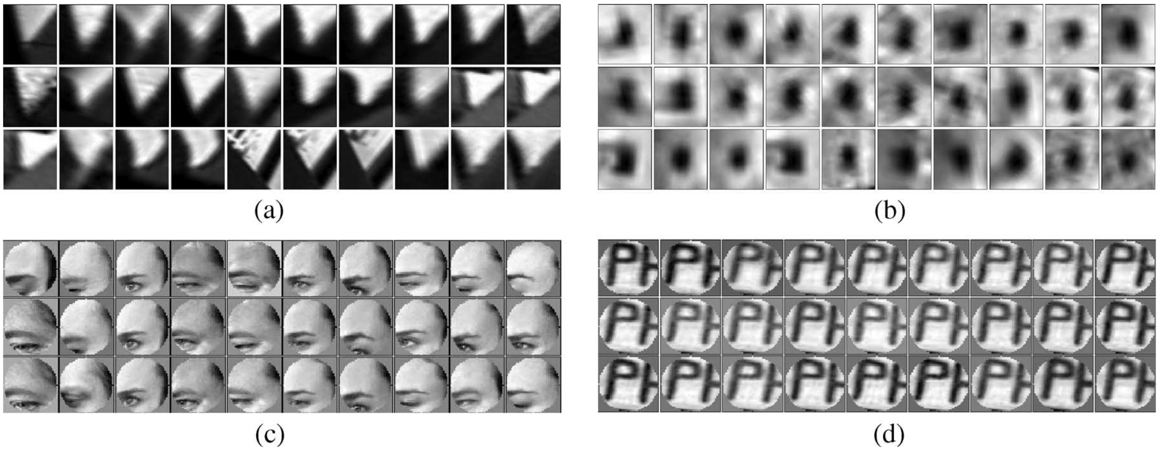


Fig. 3. *Samples of normalized affine covariant regions from clusters corresponding to a single visual word: (a), (c), (d) shape adapted regions and (b) maximally stable regions. Note that some visual words represent generic image structures, e.g., corners (a) or blobs (b), and some visual words are rather specific, e.g., eye (c) or a letter (d).*

affine covariant regions gives region description vectors which are invariant to affine transformations of the image. Note, both region detection and the description is computed on monochrome versions of the frames, color information is not currently used in this work.

To reduce noise and reject unstable regions, information is aggregated over a sequence of frames. The regions detected in each frame of the video are tracked using a simple constant velocity dynamical model and correlation. The implementation details are given in [40] and [45]. Any region which does not survive for more than three frames is rejected. This "stability check" significantly reduces the number of regions to about 600 per frame.

B. Building a Visual Vocabulary

The objective here is to vector quantize the descriptors into clusters which will be the visual "words" for text retrieval. The vocabulary is constructed from a subpart of the movie, and its matching accuracy and expressive power are evaluated on the entire movie, as described in the following sections. The vector quantization is carried out by k-means clustering. Two alternative vocabulary building methods are discussed in Section IV-A.

Each descriptor is a 128-vector, and to simultaneously cluster all the descriptors of the movie would be a gargantuan task. Instead, a random subset of 474 frames is selected. Even with this reduction there still remains around 300 K descriptors that must be clustered. A total of 6000 clusters is used for SA regions and 10 000 clusters for MS regions. The ratio of the number of clusters for each type is chosen to be approximately the same as the ratio of detected descriptors of each type. The k-means algorithm is run several times with random initial assignments of points as cluster centers and the lowest cost result used. The number of clusters K is an important parameter, which can significantly affect the retrieval performance. Typically, it is chosen empirically to maximize retrieval performance on a manually labelled object or scene ground truth data. Empirically, the relatively high number of cluster centers (with respect to the number of clustered data points) is important for good retrieval performance [42], [46]. We will return to the issue of choosing the number of cluster centers in Section IV-A.

Fig. 3 shows examples of the regions which belong to particular clusters, i.e., which will be treated as the same visual word. The clustered regions reflect the properties of the SIFT descriptors—the clustering is on the spatial distribution of the image gradient *orientations*, rather than the intensities across the region.

The reason that SA and MS regions are clustered separately is that they cover different and largely independent regions of the scene. Consequently, they may be thought of as different vocabularies for describing the same scene, and thus should have their own word sets, in the same way as one vocabulary might describe architectural features and another the material quality (e.g., defects, weathering) of a building.

C. Visual Indexing Using Text Retrieval Methods

Text retrieval systems generally employ a number of standard steps [5]: the documents are first parsed into words, and the words are represented by their stems, for example "walk," "walking," and "walks" would be represented by the stem "walk." A stop list is then used to reject very common words, such as "the" and "an," which occur in most documents and are therefore not discriminating for a particular document. The remaining words are then assigned a unique identifier, and each document is represented by a vector with components given by the frequency of occurrence of the words the document contains. In addition, the components are weighted in various ways (such as inverse document frequency weighting, see the following). All of the above steps are carried out in advance of actual retrieval, and the set of vectors representing all the documents in a corpus are organized as an *inverted file* [52] to facilitate efficient retrieval. An inverted file is structured like an ideal book index. It has an entry for each word in the corpus followed by a list of all the documents (and position in that document) in which the word occurs.

A query text is treated in a similar manner: its vector of weighted word frequencies is computed. Matching documents are obtained by measuring the similarity between the query and document vectors using the angle between the vectors. In addition, the degree of match on the ordering and separation of words may be used to rank the returned documents.

We now describe how these standard steps are employed in the visual domain where a document is replaced by an image/frame.

1) Stop List: Using a stop list analogy the most frequent visual words that occur in almost all images are suppressed. These might correspond to small specularities (highlights), for example, which occur in many frames. Typically, 5%–10% of the most common visual words are stopped. This amounts to stopping the 800–1600 most frequent visual words out of the vocabulary of 16 000. Fig. 5 shows the benefit of imposing a stop list—very common visual words occur in many places in an image and can be responsible for mismatches. Most of these are removed once the stop list is applied.

2) tf-idf Weighting: The standard weighting [5] is known as "term frequency-inverse document frequency" (*tf-idf*) and is computed as follows. Suppose there is a vocabulary of V words, then each document is represented by a vector

$$\mathbf{v}_d = (t_1, \ldots, t_i, \ldots, t_V)^\top \qquad (1)$$

of weighted word frequencies with components

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{N_i} \qquad (2)$$

where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in the document d, N_i is the number of documents containing term i, and N is the number of documents in the whole database. The weighting is a product of two terms: the *word frequency*, n_{id}/n_d, and the *inverse document frequency*, $\log N/N_i$. The intuition is that the word frequency weights words occurring more often in a particular document higher (compared to word present/absent), and thus describes it well, while the inverse document frequency downweights words that appear often in the database, and therefore do not help to discriminate between different documents.

At the retrieval stage documents are ranked by the normalized scalar product (cosine of angle)

$$\mathrm{sim}(\mathbf{v}_q, \mathbf{v}_d) = \frac{\mathbf{v}_q^\top \mathbf{v}_d}{\|\mathbf{v}_q\|_2 \|\mathbf{v}_d\|_2} \qquad (3)$$

between the query vector $\mathbf{v}_q$ and all document vectors $\mathbf{v}_d$ in the database, where $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^\top \mathbf{v}}$ is the L_2 norm of $\mathbf{v}$.

In our case, the query vector is given by the frequencies of visual words contained in a user specified subpart of an image, weighted by the inverse document frequencies computed on the entire database of frames. Retrieved frames are ranked according to the similarity of their weighted vectors to this query vector.

3) Spatial Consistency: Web search engines such as Google [9] increase the ranking for documents where the searched for words appear close together in the retrieved texts (measured by word order). This analogy is especially relevant for querying objects by an image, where matched covariant regions in the retrieved frames should have a similar spatial arrangement [38], [41] to those of the outlined region in the query image. The idea is implemented here by first retrieving frames using the weighted frequency vector alone and then reranking them based on a measure of spatial consistency.

A search area is defined by the 15 nearest spatial neighbors of each match, and each region which also matches within this area casts a vote for that frame. Matches with no support are rejected. The object bounding box in the retrieved frame is determined as the rectangular bounding box of the matched regions after the spatial consistency test. The spatial consistency voting is illus-

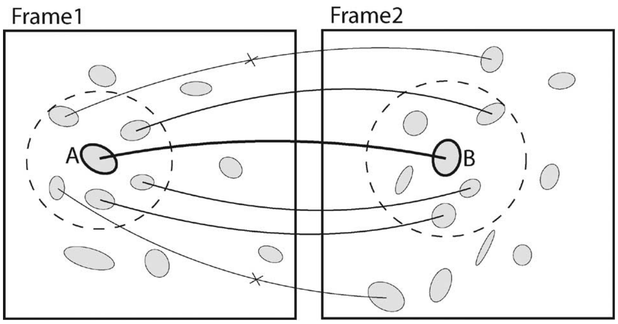


Fig. 4. ***Spatial consistency voting. To verify a pair of matching regions (A, B) a circular search area is defined by k (= 5 in this example) spatial nearest neighbors in both frames. Each match which lies within the search areas in both frames casts a vote in support of match (A, B). In this example, three supporting matches are found. Matches with no support are rejected.***

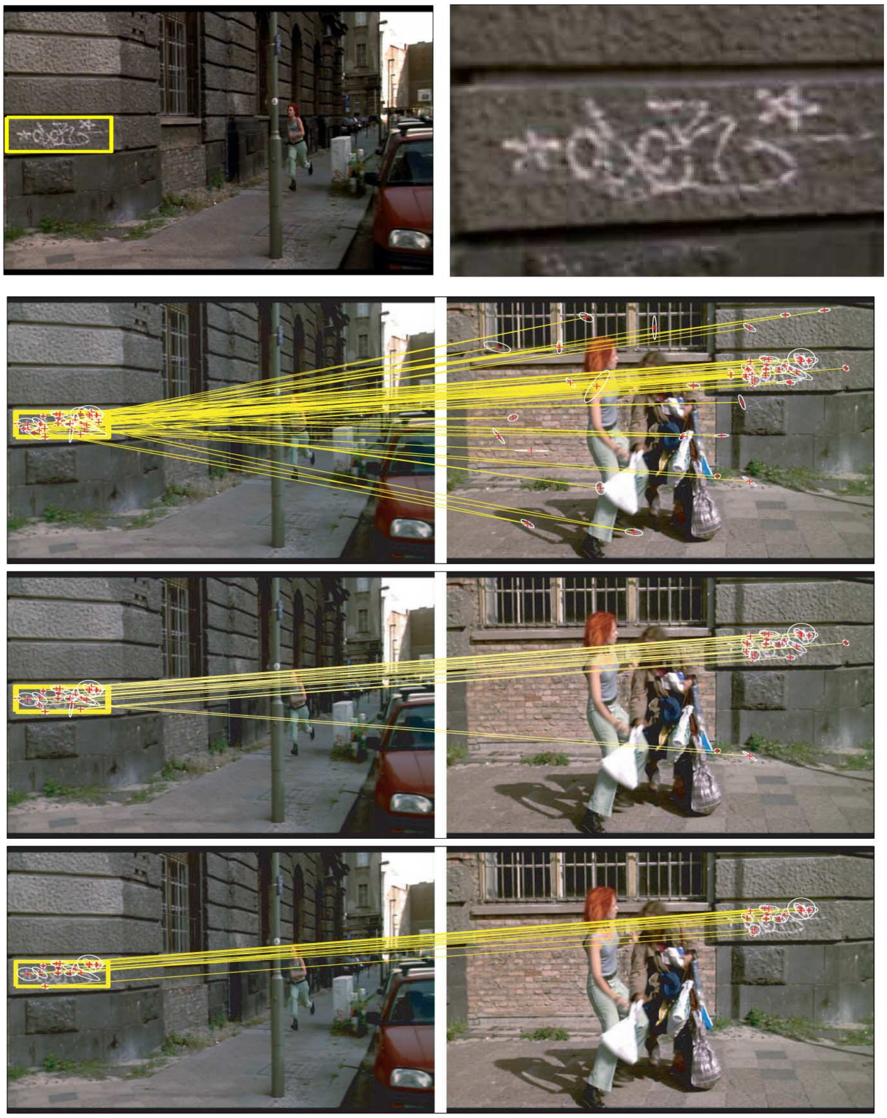


Fig. 5. *Matching stages. Top row: (left) query region and (right) its close-up. Second row: original matches based on visual words. Third row: matches after using stop-list. Last row: final set of matches after filtering on spatial consistency.*

trated in Fig. 4. This works well as is demonstrated in the last row of Fig. 5, which shows the spatial consistency rejection of incorrect matches. The object retrieval examples presented in this paper employ this ranking measure and amply demonstrate its usefulness.

Other measures which take account of, e.g., the affine mapping between images may be required in some situations, but this involves a greater computational expense. We return to this point in Section IV-C.

D. Algorithm for Object Retrieval Using Visual Words

We now describe how the components of the previous sections are assembled into an algorithm for object retrieval given a visual query.

We first describe the offline processing. A feature length film typically has 100–150 K frames. To reduce complexity, roughly one keyframe is used per second of video, which results in 4–6 K keyframes. Shot boundaries are obtained by simple thresholding of the sum of absolute differences between normalized color histograms of consecutive frames of video [21]. Descriptors are computed for stable regions in each keyframe (stability is determined by tracking as described in Section II-A). The descriptors are vector quantized using the centers clustered from the training set, i.e., each descriptor is assigned to a visual word. The visual words over all keyframes are assembled into an inverted file structure where for each word, all occurrences and the position of the word in all keyframes are stored.

1) **Pre-processing (off-line)**
   - Detect affine covariant regions in each frame of the video. Represent each region by a SIFT descriptor.
   - Track the regions through the video and reject unstable regions.
   - Build a visual vocabulary by clustering stable regions from a subset of the video. Assign each region descriptor in each keyframe to the nearest cluster centre.
   - Remove stop-listed visual words.
   - Compute tf–idf weighted document frequency vectors.
   - Build the inverted file indexing structure.

2) **At run-time (given a user selected query region)**
   - Determine the set of visual words within the query region.
   - Retrieve keyframes based on visual word frequencies.
   - Re-rank the top $N_s(= 500)$ retrieved keyframes using spatial consistency.

Fig. 6. ***The object retrieval algorithm. Example retrieval results are shown in Fig. 7.***

At run time a user selects a query region, which specifies a set of visual words and their spatial layout. Retrieval then proceeds in two steps: Firstly, a short list of $N_s = 500$ keyframes are retrieved based on their tf-idf weighted frequency vectors (the bag-of-words model), and those are then reranked using spatial consistency voting. The entire process is summarized in Fig. 6 and an example is shown in Fig. 7.

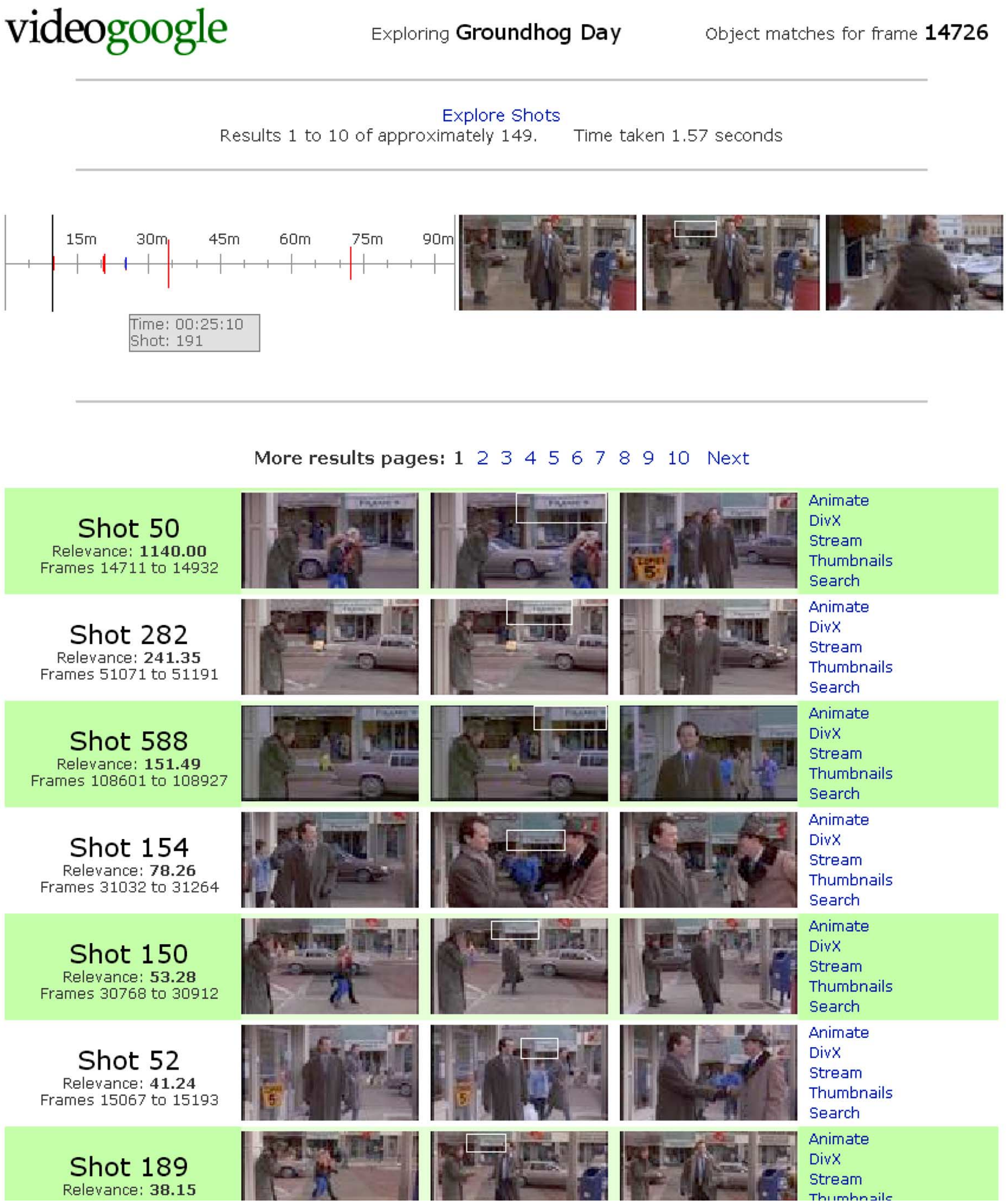
1) Processing Requirements: Optimized implementations of the region detection, description, and visual word assignment currently run at 5 Hz [31]; this is an offline cost. The average query time for the six ground truth queries on the database of 5640 keyframes is 0.82 s with a Matlab implementation on a 2-GHz Pentium. This includes the frequency ranking and spatial consistency reranking. The spatial consistency re-ranking is applied only to the top $N_s = 500$ keyframes ranked by the frequency based score. This restriction results in no loss of performance (measured on the set of ground truth queries in Section III).

In terms of memory requirements, the inverted file for the movie "Groundhog Day" takes about 66 MB and stores about 2 million visual word occurrences (this is with the 10% most frequent words removed).

III. PERFORMANCE AND RETRIEVAL EXAMPLES

In this section, we first evaluate the object retrieval performance over the entire movie on a ground truth test set of six object queries. In part, this retrieval performance assesses the expressiveness of the visual vocabulary, since only about 12% of ground truth keyframes (and the invariant descriptors they contain) were included when clustering to form the vocabulary. In the following sections we: 1) examine in detail the retrieval power of individual visual words; 2) show examples of searches specified by external images downloaded from the Internet; and finally 3) discuss and qualitatively assess the retrieval performance.

The performance of the proposed method is evaluated on six object queries in the movie "Groundhog Day." Fig. 8 shows the query frames and corresponding query regions. Ground truth occurrences were manually labelled in all the 5640 keyframes (752 shots). Retrieval is performed on keyframes as outlined in Section II-D and each shot of the video is scored by its best scoring keyframe. Similar to [4] and [47], the performance is evaluated on the level of shots rather than keyframes. We found video shots better suited than keyframes for browsing and searching movies as a particular shot may contain several similar keyframes, which usually have similar rank and clutter the returned results. However, the suitable granularity (frames, keyframes, shots) of returned results might depend on a particular application. Performance is measured using a precision-recall plot for each query. Precision is the number of retrieved ground truth shots relative to the total number of shots retrieved. Recall is the number of retrieved ground truth shots relative to the total number of ground truth shots in the movie. Precision-recall plots are shown in Fig. 9. The results are summarized using the average precision (AP) in Fig. 9. AP is a scalar valued measure computed as the area under the precision-recall graph and reflects performance over all recall levels. An ideal precision-recall curve has precision 1 over all recall


videogoogle

Exploring **Groundhog Day**

Object matches for frame **14726**

Explore Shots

Results 1 to 10 of approximately 149. Time taken 1.57 seconds

More results pages: 1 2 3 4 5 6 7 8 9 10 Next

Shot 50
Relevance: **1140.00**
Frames 14711 to 14932
Animate
DivX
Stream
Thumbnails
Search

Shot 282
Relevance: **241.35**
Frames 51071 to 51191
Animate
DivX
Stream
Thumbnails
Search

Shot 588
Relevance: **151.49**
Frames 108601 to 108927
Animate
DivX
Stream
Thumbnails
Search

Shot 154
Relevance: **78.26**
Frames 31032 to 31264
Animate
DivX
Stream
Thumbnails
Search

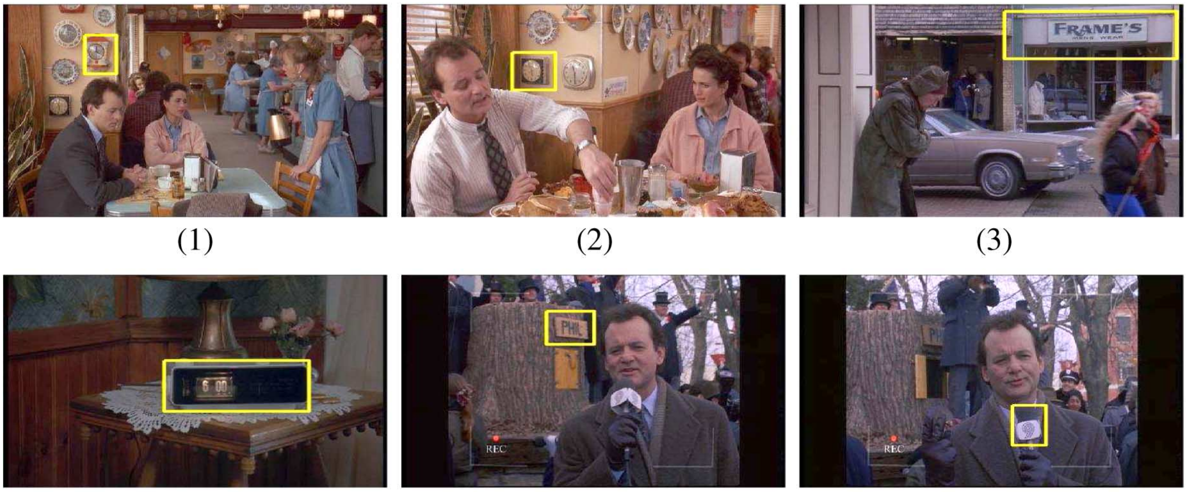
Shot 150
Relevance: **53.28**
Frames 30768 to 30912
Animate
DivX
Stream
Thumbnails
Search

Shot 52
Relevance: **41.24**
Frames 15067 to 15193
Animate
DivX
Stream
Thumbnails
Search

Shot 189
Relevance: **38.15**
Animate
DivX
Stream
Thumbnails

Fig. 7. ***Object query example III: "Groundhog Day." Screenshot of running object retrieval system showing results of object query 3 from query set of Fig. 8. Top part of screenshot shows an interactive timeline, which allows user to browse through retrieved results on that page in a chronological order. Bottom part of screenshot shows the first seven ranked shots from the first page of retrieved shots. Each shot is displayed by three thumbnails showing (from left to right) the first frame, matched keyframe with identified region of interest shown in white, and last frame of the shot. Precision-recall curve for this query is shown in Fig. 9.***

levels, which corresponds to AP of 1. In other words, the prefect performance is obtained when all relevant shots are ranked higher than nonrelevant shots. Note that a precision-recall curve does not have to be monotonically decreasing. To illustrate this, say there are three correct shots out of the first four retrieved, which corresponds to precision $3/4 = 0.75$. Then, if the next retrieved shot is correct the precision increases to $4/5 = 0.8$.


(1) (2) (3)

(4) (5) (6)

Object	# of keyframes	# of shots	# of query regions
1 Red Clock	138	15	31
2 Black Clock	120	13	29
3 Frames sign	92	14	123
4 Digital clock	208	23	97
5 Phil sign	153	29	26
6 Microphone	118	15	19

Fig. 8. ***Query frames with outlined query regions for six test queries with manually obtained ground truth occurrences in the movie "Groundhog Day." Table shows number of ground truth occurrences (keyframes and shots) and number of affine covariant regions lying within query rectangle for each query.***

It is evident that for all queries the AP of the proposed method exceeds that of using frequency vectors alone—showing the benefits of using the spatial consistency to improve the ranking. In [42], we further show that the proposed visual word matching method does not result in a loss of retrieval performance compared to a standard frame-to-frame descriptor matching used by, e.g., [22] and [41].

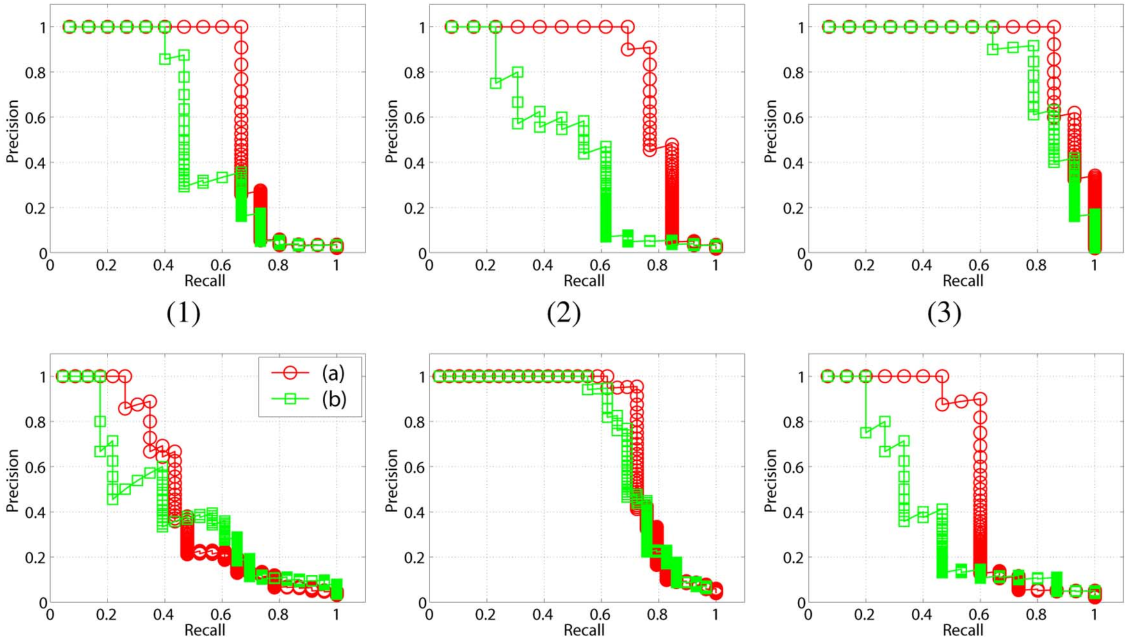
Examining the precision-recall curves in Fig. 9 we note that the performance is biased towards high precision at lower recall levels. In practice, this might be acceptable for some applications: for example a visual search of videos/images on the Internet, where the first few correctly retrieved videos/images (and their corresponding web pages) might contain the relevant information. We note, however, that for some other applications, where finding all instances of an object is important (e.g., surveillance), higher precision at higher recall levels might be preferable.

Examples of frames from low ranked shots are shown in Fig. 10. Appearance changes due to extreme viewing angles, large scale changes, and significant motion blur affect the process of extracting and matching affine covariant regions. The examples shown represent a significant challenge to the current object matching method.

Figs. 2 and 7 show example retrieval results for two object queries for the movie "Groundhog Day," Figs. 1, 11, and 12 show retrieval examples for the film "Charade," and Fig. 13 shows a retrieval example for the movie "Pretty Woman." Movies "Charade" and "Pretty Woman" are represented by 6503 and 6641 keyframes, respectively, and a new visual vocabulary was built for each of the two movies, as described in Section II-B.

A. Quality of Individual Visual Words

It is also interesting to inspect the "quality" of individual query visual words. The goal here is to examine retrieval performance if only a single visual word is used as a query. Visual words with good retrieval performance: 1) should occur mostly on the object of interest (high precision) and 2) should retrieve all the object occurrences in the database (high recall). In particular, for an individual visual word, the retrieved keyframes are all keyframes where the visual word occurs. Note that here there is no ranking among the retrieved keyframes as all occurrences of a single visual word are treated with an equal weight. As a result, a single visual word produces a single point on the precision-recall curve. Precision is the number of retrieved ground truth keyframes relative to the



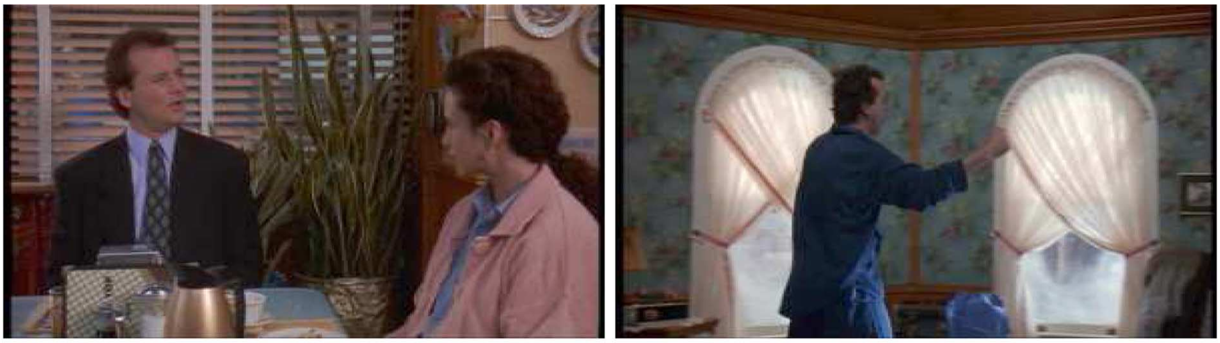
(1) (2) (3)

(4) (5) (6)

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6	Average
AP freq+spat (a)	0.70	0.81	0.93	0.48	0.77	0.62	**0.72**
AP freq only (b)	0.55	0.49	0.86	0.43	0.73	0.41	**0.58**

Average precision (AP) for the six object queries.

Fig. 9. ***Precision-recall graphs (at shot level) for six ground truth queries on the movie "Groundhog Day." Each graph shows two curves corresponding to (a) frequency ranking (tf-idf) followed by spatial consistency reranking (circles) and (b) frequency ranking (tf-idf) only (squares). Note the significantly improved precision at lower recalls after spatial consistency reranking (a) is applied to the frequency based ranking (b). Table shows average precision (AP) for each ground truth object query for two methods. Last column shows mean average precision over all six queries.***



(1,2) (4)

Fig. 10. ***Examples of missed (low ranked) detections for objects 1, 2, and 4 from Fig. 8. In left image, two clocks (objects 1 and 2) are imaged from an extreme viewing angle and are barely visible—red clock (object 2) is partially out of view. In right image, digital clock (object 4) is imaged at a small scale and significantly motion blurred.***

total number of keyframes retrieved. Recall is the number of retrieved ground truth keyframes relative to the total number of ground truth keyframes in the movie. The precision/recall graph, shown in Fig. 14, indicates that individual visual words are "noisy," i.e., occur on multiple objects or do not cover all occurrences of the object in the

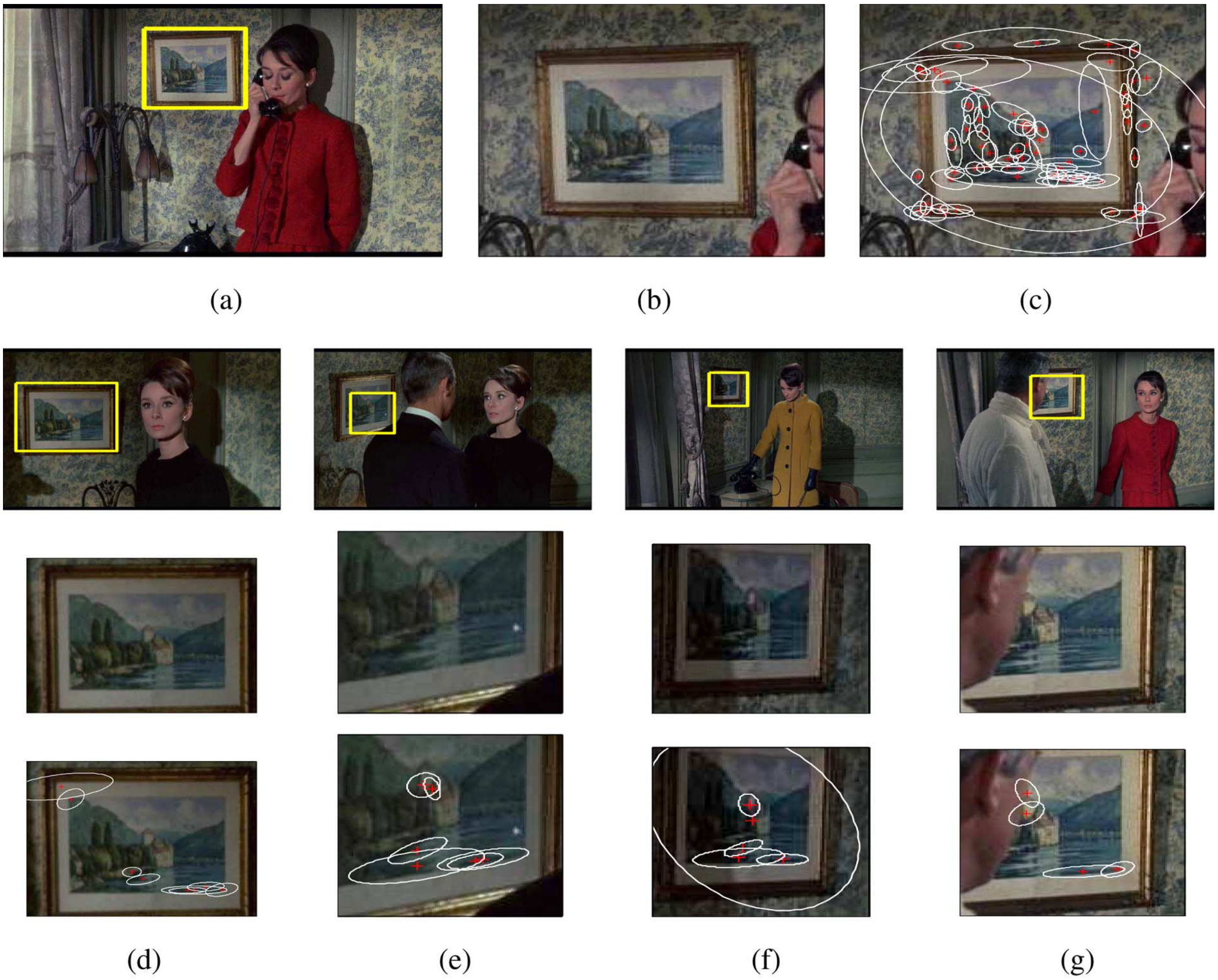


Fig. 11. *Object query example IV: "Charade," (a) Keyframe with user specified query region (a picture), (b) close-up of query region, and (c) close-up with affine covariant regions superimposed. (d)–(g) (First row) keyframes from 5th, 8th, 12th, and 18th retrieved shots with identified region of interest, (second row) close-up of image, and (third row) close-up of image with matched elliptical regions superimposed. First false positive is ranked 20th. Querying 6503 keyframes took 1.28 s on a 2-GHz Pentium.*

database. It should be noted here that the requirement that each visual word occurs on only one object (high precision) might be unrealistic in a real world situation as it implies that the vocabulary would grow linearly with the number of objects. A more realistic situation is that visual words are shared across objects and that an object is represented by a conjunction of several visual words. Also, perfect recall might not be attained simply because the region is occluded in some of the target keyframes.

B. Searching for Objects From Outside the Movie

Fig. 15 shows an example of a search for an object specified by a query image outside the "closed world" of the film. The object (a Sony logo) is specified by a region of an image downloaded from the Internet. The image was preprocessed as outlined in Section II-A. Fig. 16 shows two more examples of external searches on feature length movies "Pretty Woman" and "Charade."

To evaluate the external search performance we manually labelled all occurrences of the external query objects in the respective movies. Both the "Hollywood" sign and the "Notre Dame" cathedral appear in only one shot in the respective movies and in both cases the correct shot is ranked first, which corresponds to a perfect AP of one. The "Sony" logo appears on two objects—a monitor and a TV camera, which are both found and shown in Fig. 15. The TV camera, however, appears in several other shots throughout the movie. To measure the performance, we manually selected shots where the "Sony" logo is readable by a human (usually larger than roughly 20 pixels across in a 720 × 576 keyframe), which resulted in a ground truth set of three shots. These are all retrieved, ranked 1st, 4th, and 35th (AP of 0.53).

Searching for images from other sources opens up the possibility for product placement queries, or searching

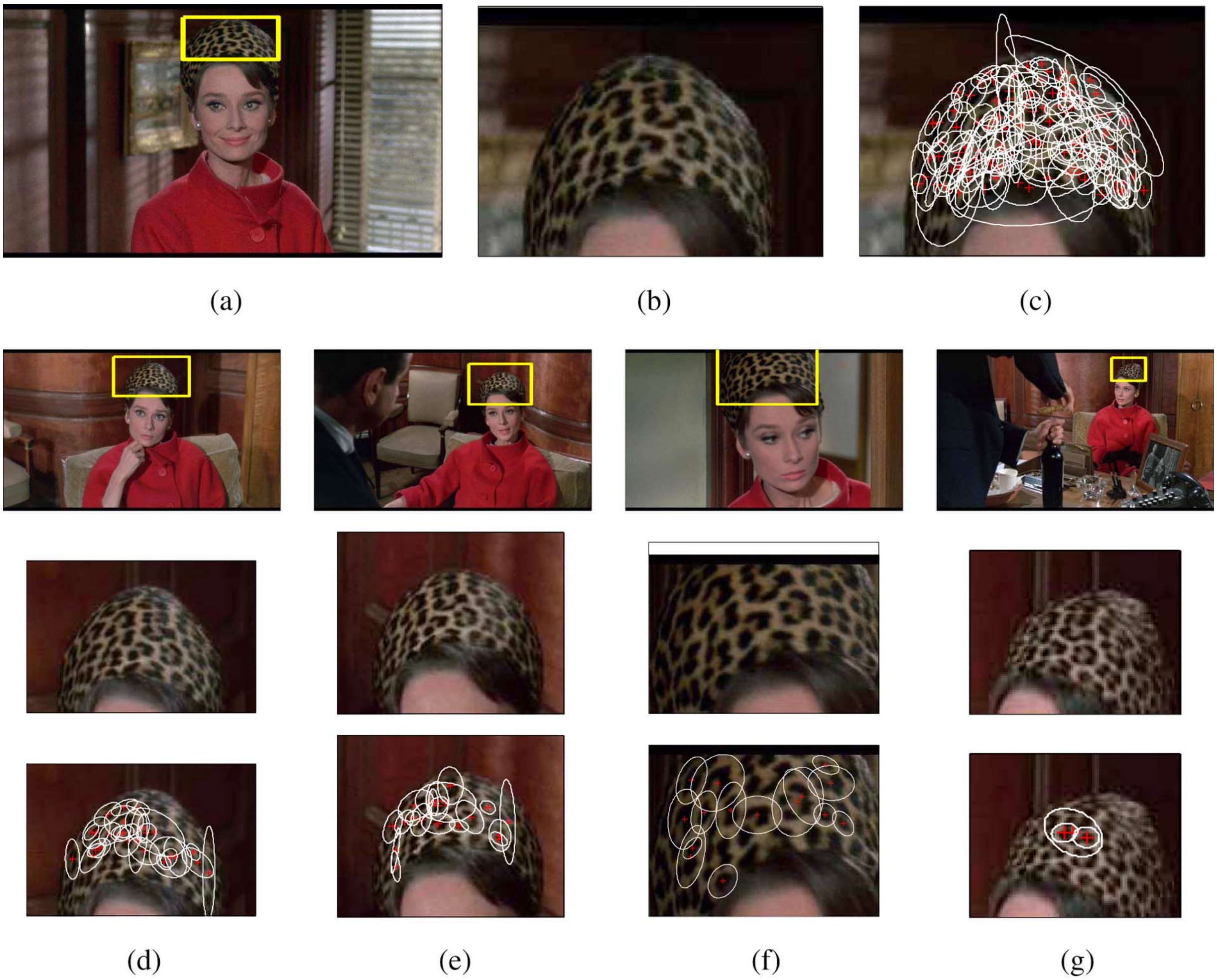


Fig. 12. *Object query example V: "Charade." (a) Keyframe with user specified query region (a hat), (b) close-up of query region, and (c) close-up with affine covariant regions superimposed. (d)–(g) (First row) keyframes from 5th, 17th, 22nd, and 28th retrieved shots with identified region of interest, (second row) close-up of image, and (third row) close-up of image with matched elliptical regions superimposed. First false positive is ranked 30th. Querying 6503 keyframes took 2.06 s on 2-GHz Pentium.*

movies for company logos, or particular buildings or locations.

C. Qualitative Assessment of Performance

Currently, the search is biased towards (lightly) textured regions which are detectable by the applied region detectors (corner-like features, blob-like regions). Examples of challenging object searches include textureless objects (bottles, mugs), thin and wiry objects (bicycles, chairs), or highly deformable objects such people's clothing. The range of searchable objects can be extended by adding other covariant regions (they will define an extended visual vocabulary), for example those of [50]. Including shape and contour-based descriptors [7], [30] might enable matching textureless or wiry [10] objects. Another interesting direction is developing specialized visual vocabulary for retrieving faces of a particular person in video [43].

IV. DIRECTIONS FOR FUTURE RESEARCH IN VISUAL OBJECT RETRIEVAL

In this section, we discuss three research directions and review some recent work addressing them. In particular, we focus on: 1) building visual vocabularies for very large-scale retrieval; 2) retrieval of 3-D objects; and 3) more thorough verification and ranking using spatial structure of objects.

A. Challenge I: Visual Vocabularies for Very Large Scale Retrieval

In this paper, we have shown object retrieval results within an entire feature length movie, essentially searching through 150 000 frames indexed by more than 6000 keyframes. One direction of future work is scaling-up, with the ultimate goal of indexing the billions of images available

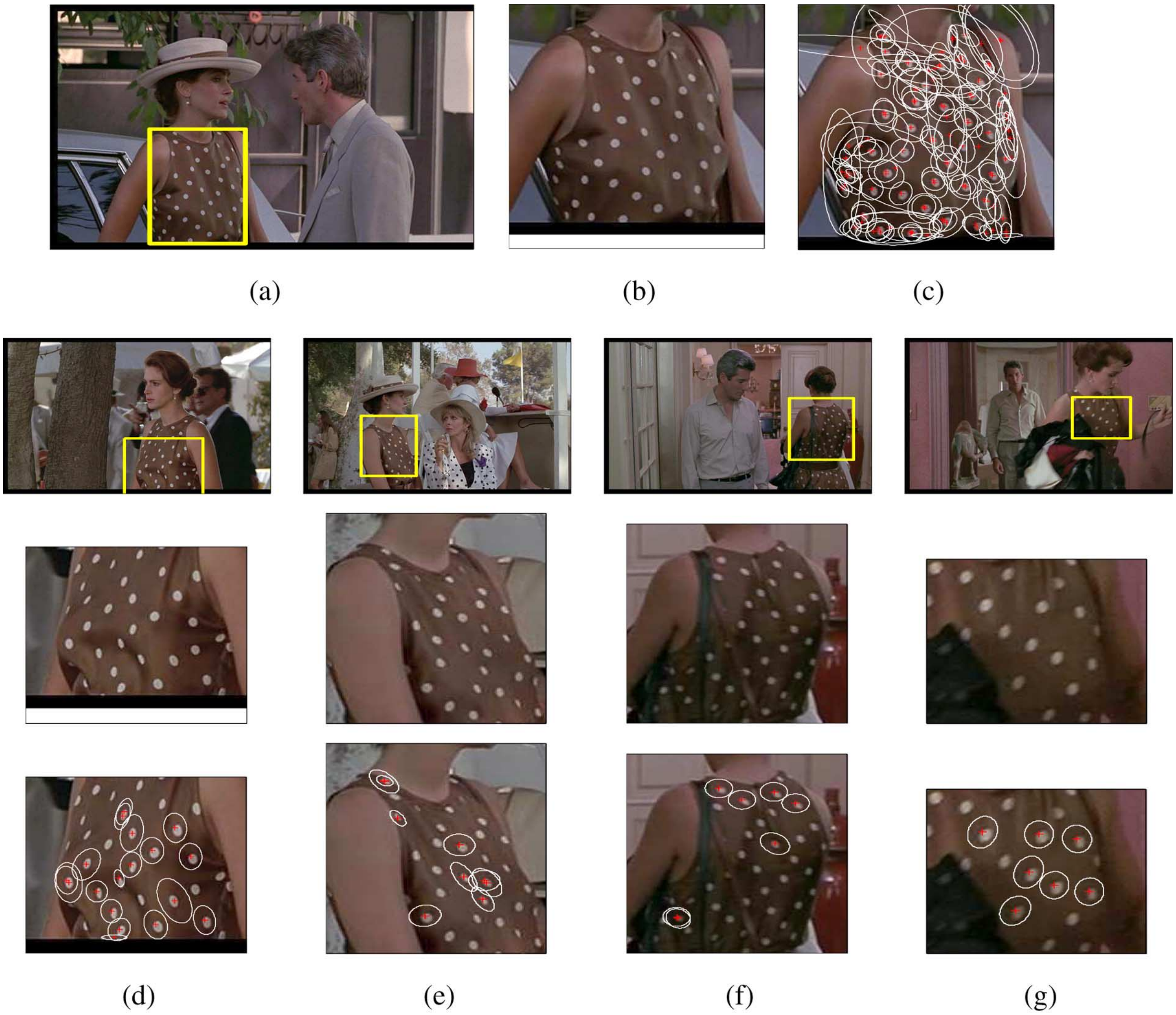


Fig. 13. ***Object query example VI: "Pretty Woman." (a) Keyframe with user specified query region (a polka dot dress), (b) close-up of query region, and (c) close-up with affine covariant regions superimposed. (d)–(g) (First row) keyframes from 2nd, 6th, 8th, and 13th retrieved shots with identified region of interest, (second row) close-up of image, and (third row) close-up of image with matched elliptical regions superimposed. First false positive is ranked 16th. Querying 6641 keyframes took 1.19 s on 2-GHz Pentium.***

online. Issues are the cost of building a visual vocabulary from such large databases (if descriptors from all images are used) and the size of the vocabulary—should the vocabulary increase in size as the number of images grows? How discriminative should visual words be? Recent papers by Nister and Stewenius [31] and Philbin *et al.* [33] have addressed these issues.

Nister and Stewenius use a hierarchical k-means clustering (also called tree structured vector quantization [17, p. 410]) to obtain a vocabulary organized in a tree. The benefit of this approach is a reduced algorithmic complexity of the vocabulary building stage to $O(N\log(K))$ compared to $O(NK)$ of standard k-means used in our approach. Here, N is the number of descriptors being clustered and K is the number of cluster centers. As a result, building larger vocabularies (with up to 1 million visual words reported in [31]) from more training descriptors became feasible. Another benefit is the reduced cost ($O(\log(K)$ compared to $O(K)$ used in this paper) of assigning visual word labels to novel unseen descriptors. This allows fast insertion of new images into the database. Nister and Stewenius show successful object retrieval on a database of 50 000 CD covers. Experiments reported in [31] also suggest that the tree structured vocabulary might overcome to some extent the difficulty of choosing a particular number of cluster centers.

Philbin *et al.* [33] replace exact k-means by an approximate k-means also reducing the algorithmic complexity of vocabulary building to $O(N\log(K))$ and showing impressive retrieval results on a database of up to 1 million images downloaded from the photo sharing site Flickr [2]. Furthermore, vocabularies built using the

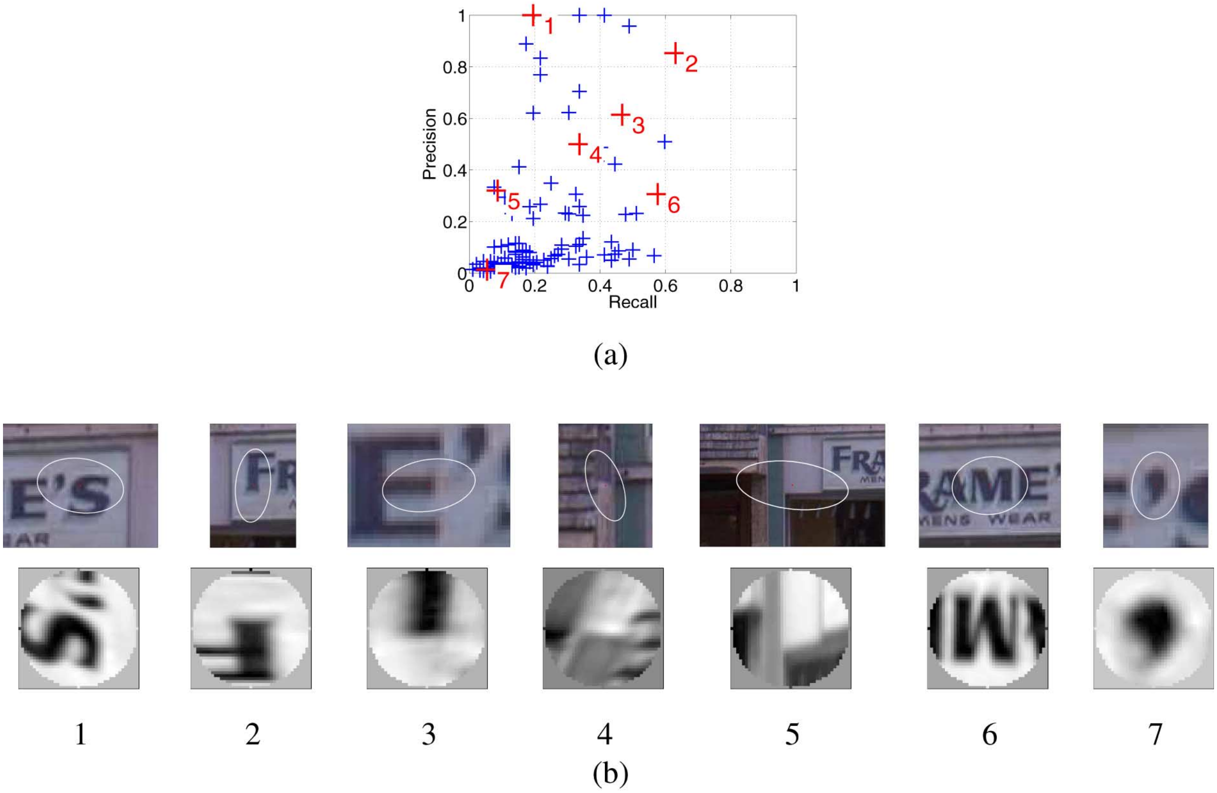


Fig. 14. *"Quality" of single visual words. (a) Precision-recall graph shows "quality" of each individual query visual word for ground truth object (3) of Fig. 8. Each precision-recall point in graph represents quality/performance of a single query visual word. Note that many visual words are quite weak individually with low recall and precision. Some visual words are more indicative for presence of the object, but none of them achieves perfect results, which would be the top-right corner of graph. (b) Examples of visual words describing object (3)—"Frames sign." Top row: scale normalized close-ups of elliptical regions overlaid over query image. Bottom row: corresponding normalized regions. Visual words are numbered and their precision and recall values are shown in precision-recall graph (a).*

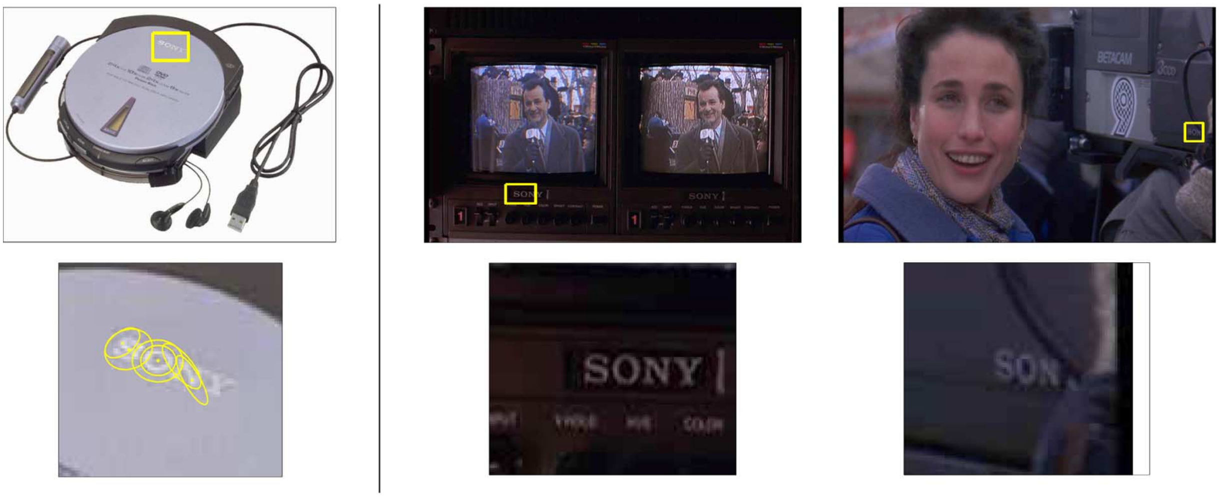
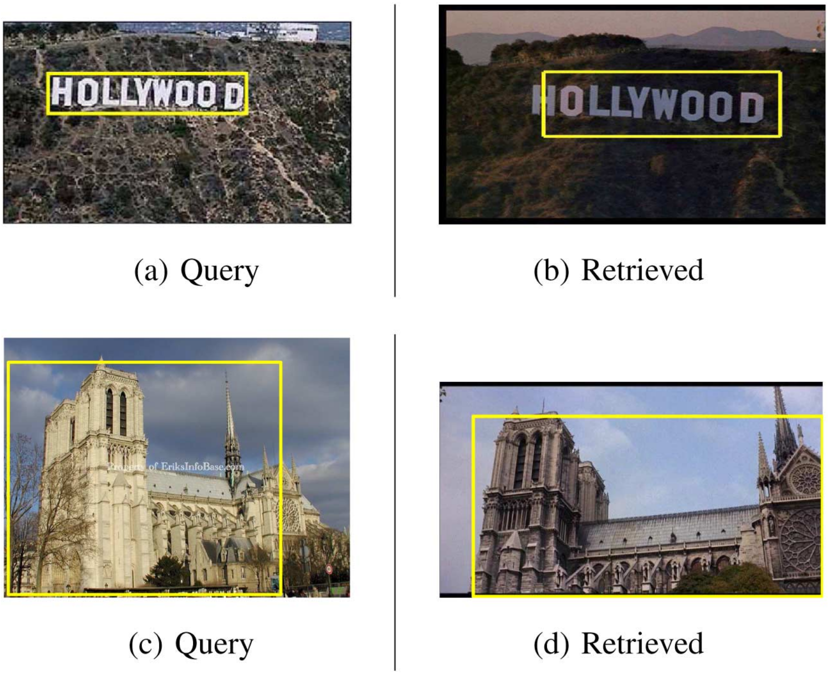


Fig. 15. *Searching for a Sony logo. First column: (top) Sony Discman image (640 × 422 pixels) with user outlined query region and (bottom) close-up with detected elliptical regions superimposed. Second and third column: (top) retrieved frames from two different shots of "Groundhog Day" with detected Sony logo outlined in yellow and (bottom) close-up of image. Retrieved shots were ranked 1 and 4.*



(a) Query

(b) Retrieved

(c) Query

(d) Retrieved

Fig. 16. *Searching for locations using external images downloaded from the Internet. (a) Query frame. (b) Frame from first shot retrieved (correctly) from "Pretty Woman." (c) Query frame (Notre Dame in Paris). (d) Frame from first shot retrieved (correctly) from "Charade." Note viewpoint change between query and retrieved frame.*

approximate k-means method outperform the vocabulary tree [31] on the standard image retrieval benchmark [1].

In Section III-B, we show that a vocabulary built from one movie can generalize to images from outside the closed world of the movie, e.g., downloaded from the Internet. Another issue is the ability of a vocabulary to generalize to new objects and scenes, not seen at the vocabulary building stage. To this end, experiments performed in [42] indicate a drop in retrieval performance when a vocabulary built from one movie is used for retrieval in another movie. However, retrieval performance can be restored using a vocabulary built jointly from both movies. Learning a universal visual vocabulary, with improved generalization to unseen objects and scenes, remains a current research challenge. Alternatively, a visual vocabulary might not be static but instead evolve over time when new images are added to the database.

B. Challenge II: Retrieval of 3-D Objects

In the video retrieval application described so far, a query is specified by an image of the object of interest. Such queries enable retrieval of objects with some degree of generalization over viewpoint and deformation—but specifying the front of a car as a query will not retrieve shots of the rear of the car. In general, there is a problem of not retrieving a visual *aspect* that differs from that of the query. We mention here two approaches to address this problem.

The first approach builds on the idea that within a video there are often several visual aspects that can be associated automatically using temporal information—for example the front, side, and back of a car as illustrated in Fig. 17. Grouping aspects by tracking can be performed on the query side (the query frame is associated with other frames in the query shot) or/and on the entire stored and indexed video database. On the query side, selecting one aspect of the object then automatically also issues searches on all the associated aspects. An example of the resulting retrievals following a multiple aspect query is shown in Fig. 17. The exemplars (regions of the frames) are associated automatically from video shots despite background clutter. More details can be found in [45]. Grouping on the database side (rather than the query) means that querying on a single aspect can then return all of the pregrouped frames in a particular video shot. Other approaches to building appearance models from video include that of [25], where optic-flow-based motion segmentation is used to extract objects from video, and that of [51], where an object is modelled by selecting keyframes (using interest point tracking) from video sequences of single objects (some of which are artificial).

Note that in the approach above, the 3-D structure of the object (a van in Fig. 17) is represented implicitly by a set of exemplar images. An alternative approach was developed by Rothganger *et al.* [37], where tracks of affine

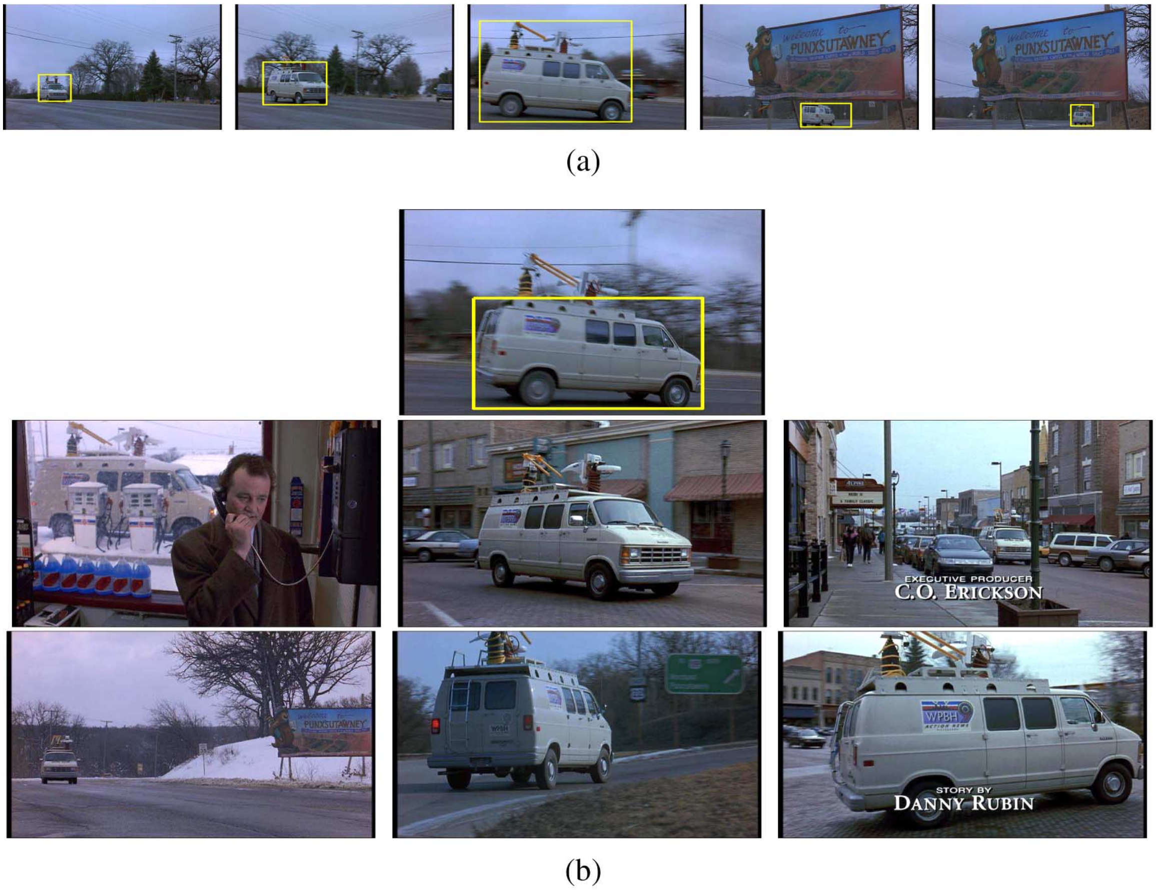


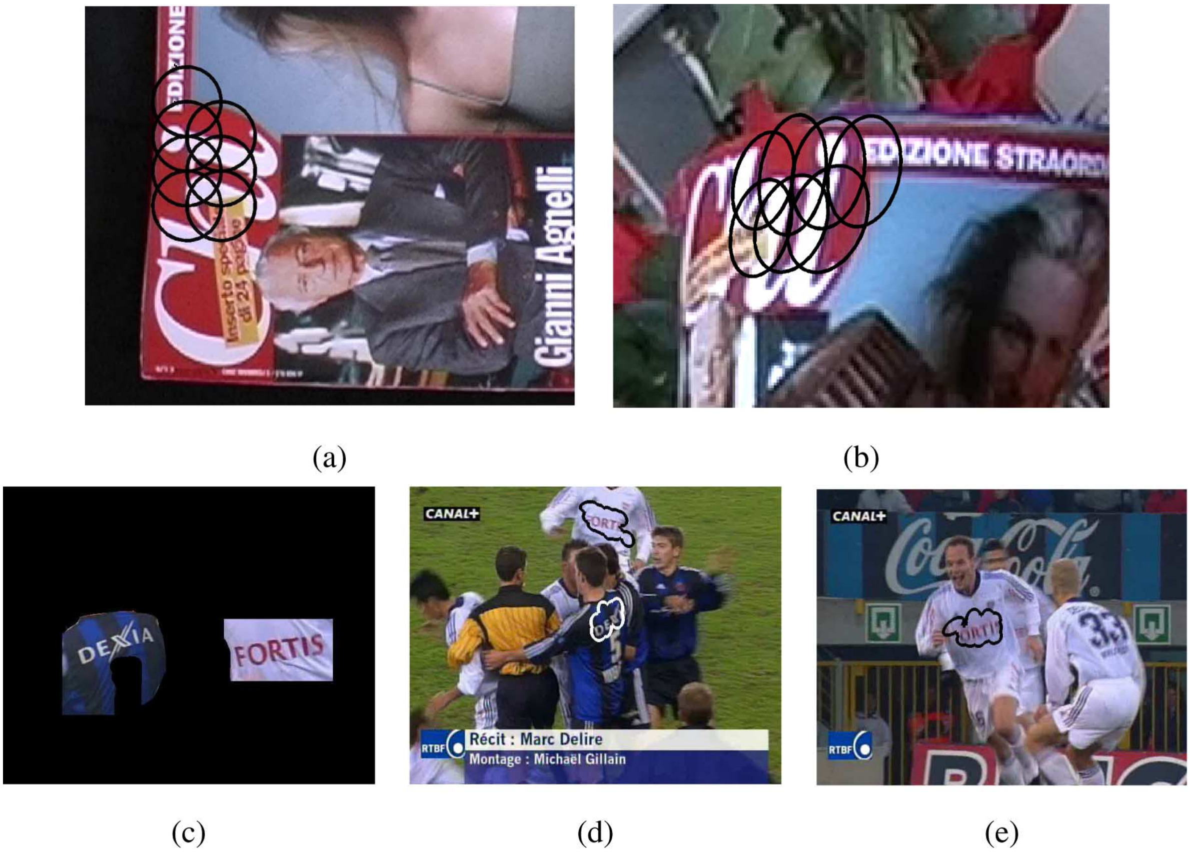
Fig. 17. ***Automatic association and querying multiple aspects of 3-D object. (a) Five frames from one shot (188 frames long) where camera is panning right, and van moves independently. Regions of interest, shown in yellow, are associated automatically by tracking affine covariant regions and motion grouping tracks belonging to rigidly moving 3-D objects. Note that regions of interest cover three aspects (front, side, and back) of van. (b) Multiple aspect video matching. Top row: Query frame with query region (side of the van) selected by the user. Query frame acts as a portal to automatically associated frames and query regions, shown in (a). Next two rows: example frames retrieved from entire movie by multiple aspect query. Note that views of van from back and front are retrieved.***

covariant regions are used to build an explicit 3-D model of the object/scene automatically from a video shot. Scenes containing a small number of independently moving objects are also handled. Although, in principle, 3-D models can be used for retrieval in videos, and we return to this point in the next section, the focus of [37] is more on model building than matching, and only rigid objects are considered.

C. Challenge III: Verification Using Spatial Structure of Objects

The spatial consistency reranking (Section II-C3) was shown to be very effective in improving the precision and removing false positive matches. However, the precision could be further improved by a more thorough (and more expensive) verification, based on a stricter measure of spatial similarity. Examples include angular ordering of regions [41], region overlap [16], deformable mesh matching [34], common affine geometric transformation [24], [33], or multiview geometric constraints [36]. Unless the system is being designed solely to retrieve rigid objects, care must be taken not to remove true positive matches on deformable objects, such as people's clothing, by using measures that apply only to rigid geometry. To reduce the computational cost, verification can be implemented as a sequence of progressively more thorough (and more expensive) filtering stages. The cost of spatial verification can be further reduced by including some of the spatial layout of regions in the precomputed index. In the following, we review in more detail two spatial verification methods based on matching local image regions. The two methods are complementary as they are designed, respectively, for matching deformable and 3-D objects.

The first method, proposed by Ferrari *et al.* [16], is based on measuring spatial overlap of local regions as illustrated in Fig. 18(a) and (b). A set of local regions in the query (model) image, shown in Fig. 18(a), is deemed matched to a set of local regions in the retrieved image,



(a) (b)

(c) (d) (e)

Fig. 18. ***(a), (b) Spatial verification based on overlap of local regions. (a) Set of local regions (shown in black) on a model view of object (magazine cover). (b) Same set of regions matched to a different view of same object. Note that pattern of intersection between neighboring regions is preserved despite the fact that object is imaged from a different viewpoint and deformed. (c), (e) Retrieval of deformable objects (logos on t-shirts) in video. (c) Two query objects delineated by user. Rest of the frame is blanked out. (d), (e) Examples of retrieved frames. Automatically found regions of interest are delineated in white and black. Figure courtesy of Ferrari, Tuytelaars, and Van Gool [15], [16].***

shown in Fig. 18(b), if regions are matched individually on appearance and the pattern of intersection between neighboring regions is preserved. This verification procedure has been applied to retrieval of objects from video [15] and example results are shown in Fig. 18(c)–(e). This verification was shown to work well for deformations, which can be approximated locally by affine 2-D geometric transformation. On the downside, the matching is computationally expensive as it involves search over parameters of the affine map for each local region.

The second verification method, proposed by Rothganger *et al.* [36], is based on matching a 3-D object model composed of a set of local affine covariant regions placed in a common 3-D coordinate frame. The model is built automatically from a collection of still images. An example 3-D model is shown in Fig. 19. Note that the model explicitly captures the structure of the object. During verification, the consistency of local appearance descriptors as well as the geometric consistency of the projected 3-D object model onto the target image is required. The benefit of this approach is that the object can be matched in a wide range of poses, including poses unseen during the model building stage, as shown in Fig. 19(c) and (d). On the downside, the 3-D model is currently built offline, and the model building requires several (up to 20) images of the object taken from different viewpoints and with fairly clean background.

V. DISCUSSION AND CONCLUSION

We have demonstrated a scalable object retrieval architecture, which utilizes a visual vocabulary based on vector-quantized viewpoint invariant descriptors and efficient indexing techniques from text retrieval.

It is worth noting two differences between document retrieval using a bag-of-words, and frame retrieval using a bag-of-visual-words: 1) because visual features overlap in the image, some spatial information is implicitly preserved (i.e., randomly shuffling bits of the image around will almost certainly change the bag-of-visual-words description). This is in contrast to the bag-of-words representation of text, where all spatial information

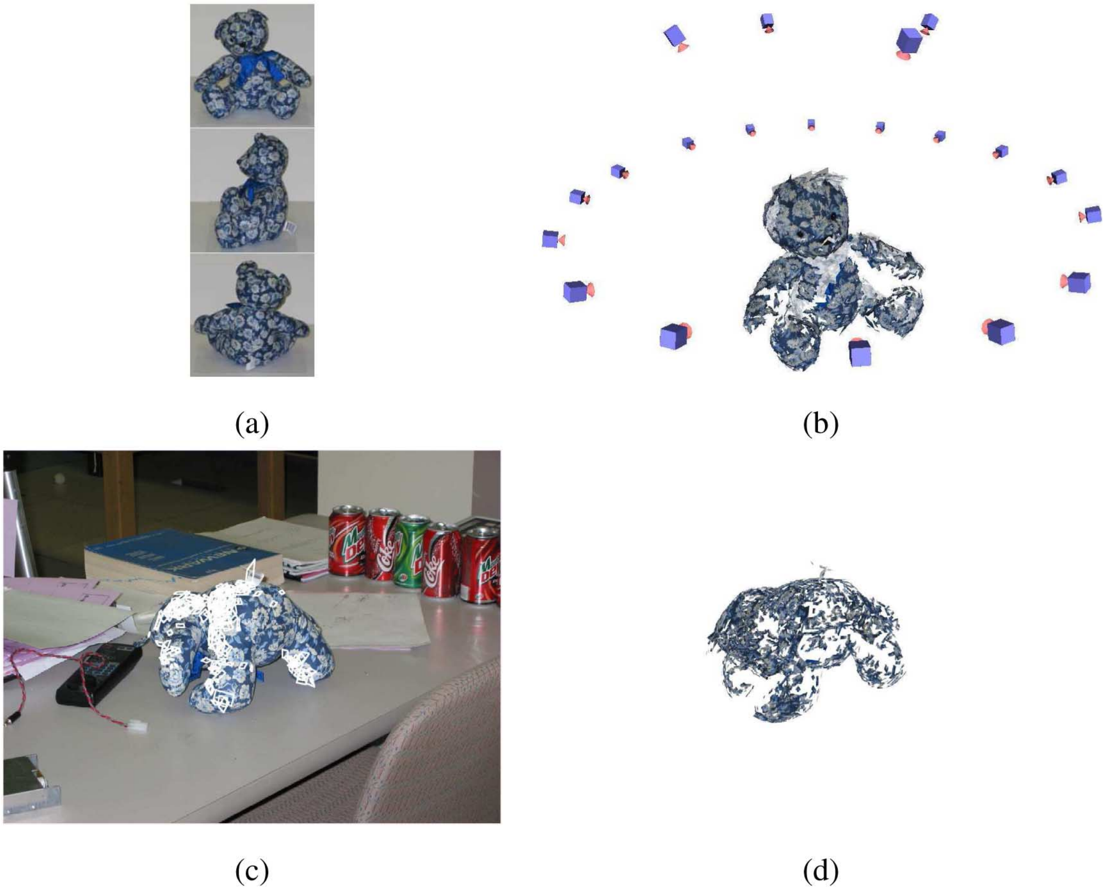


Fig. 19. ***Representing 3-D object by local patches with explicit 3-D structure. (a) Three (out of 20) images used to build 3-D model shown in (b). During verification, object can be matched in a previously unseen pose, as shown in (c) and (d). Figure courtesy of Rothganger, Lazebnik, Schmid, and Ponce [36].***

between words (e.g., the word order or proximity) is discarded. 2) An image query typically contains many more visual words than a text query—as can be seen in Fig. 2 a query region of a reasonable size may contain 50–100 visual words. However, since the visual words are a result of (imperfect) detection and also might be occluded in other views, only a proportion of the visual words may be expected to match between the query region and target image. This differs from the web-search case where a query is treated as a conjunction, and all words should match in order to retrieve a document/web page.

This paper demonstrates an application of text retrieval techniques for efficient visual search for objects in videos. Probabilistic models from statistical text analysis and machine translation have been also adapted to the visual domain in the context of object category recognition [6], [44], [48] and scene classification [8], [12], [20], [35].

A live demonstration of the object retrieval system on two publicly available movies ("Charade" [Donen, 1963] and "Dressed to Kill" [Neill, 1946]) is available online at [3]. ■

Acknowledgment

The authors are very grateful for suggestions from and discussions with M. Brady, A. Efros, M. Everingham, M. Isard, and D. Lowe.

REFERENCES

[1] [Online]. Available: http://www.vis.uky.edu/~stewe/ukbench/data/

[2] [Online]. Available: http://www.flickr.com/

[3] [Online]. Available: http://www.robots.ox.ac.uk/~vgg/research/vgoogle/

[4] [Online]. Available: http://www-nlpir.nist.gov/projects/trecvid/

[5] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM, ISBN: 020139829, 1999.

[6] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan, "Matching words and pictures," *J. Machine Learning Res.*, vol. 3, pp. 1107–1135, Feb. 2003.

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 2, pp. 509–522, Feb. 2002.

[8] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proc. Eur. Conf. Computer Vision*, 2006.

[9] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proc. 7th Int. WWW Conf.*, 1998.

[10] O. Carmichael and M. Hebert, "Shape-based recognition of wiry objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 12, pp. 1537–1552, Dec. 2004.

[11] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," in *Proc. Workshop Statistical Learning Computer Vision, ECCV*, 2004, pp. 1–22.

[12] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2005.

[13] R. Fergus, "Visual object category recognition," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., Dec. 2005.

[14] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2003, vol. 2, pp. 264–271.

[15] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Retrieving objects from videos based on affine regions," in *Proc. 12th Eur. Signal Processing Conf.*, 2004.

[16] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation by image exploration," in *Proc. Eur. Conf. Computer Vision*, 2004, vol. 1, pp. 40–54.

[17] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[18] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, Manchester, U.K., 1988, pp. 147–151.

[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd. Cambridge, U.K.: Cambridge Univ. Press, ISBN: 0521540518, 2004.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2006.

[21] R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *Int. J. Image Graphics*, Aug. 2001.

[22] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th Int. Conf. Computer Vision*, Kerkyra, Greece, Sep. 1999, pp. 1150–1157.

[23] D. Lowe, "Local feature view clustering for 3D object recognition," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Kauai, HI, Dec. 2001, pp. 682–688. Springer.

[24] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[25] A. Mahindroo, B. Bose, S. Chaudhury, and G. Harit, "Enhanced video representation using objects," in *Proc. Indian Conf. Computer Vision, Graphics Image Processing*, 2002, pp. 105–112.

[26] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. British Machine Vision Conf.*, 2002, pp. 384–393.

[27] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. 7th Eur. Conf. Computer Vision*, Copenhagen, Denmark, 2002, pp. I:128–I:142. Springer-Verlag.

[28] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2003, pp. II:257–II:263.

[29] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[30] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *Proc. British Machine Vision Conf.*, 2003.

[31] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2006, pp. II:2161–II:2168.

[32] S. Obdrzalek and J. Matas, "Object recognition using local affine frames on distinguished regions," in *Proc. British Machine Vision Conf.*, 2002, pp. 113–122.

[33] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2007.

[34] J. Pilet, V. Lepetit, and P. Fua, "Real-time non-rigid surface detection," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jun. 2005, pp. I:822–I:828.

[35] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica, T. Tuytelaars, and L. Van Gool, "Modeling scenes with local descriptors and latent aspects," in *Proc. Int. Conf. Computer Vision*, 2005, pp. I:883–I:890.

[36] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2003.

[37] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "Segmenting, modeling, and matching video clips containing multiple moving objects," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2004, pp. II:914–II:921.

[38] F. Schaffalitzky and A. Zisserman, "Automated scene matching in movies," in *Proc. Challenge Image Video Retrieval*, London, U.K., 2002, vol. 2383, *LNCS*, pp. 186–197. Springer-Verlag.

[39] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?'" in *Proc. 7th Eur. Conf. Computer Vision*, Copenhagen, Denmark, 2002, vol. 1, pp. 414–431. Springer-Verlag.

[40] F. Schaffalitzky and A. Zisserman, "Automated location matching in movies," *Computer Vision Image Understanding*, vol. 92, pp. 236–264, 2003.

[41] C. Schmid and R. Mohr, "Local grey value invariants for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 530–534, May 1997.

[42] J. Sivic, "Efficient visual search of images and videos," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 2006.

[43] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," in *Proc. Int. Conf. Image Video Retrieval (CIVR 2005)*, Singapore, 2005, pp. 226–236.

[44] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proc. Int. Conf. Computer Vision*, 2005, pp. 370–377.

[45] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Object level grouping for video shots," *Int. J. Computer Vision*, vol. 67, no. 2, pp. 189–210, 2006.

[46] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Computer Vision*, Oct. 2003, pp. II:1470–II:1477.

[47] A. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proc. 8th ACM Int. Workshop Multimedia Information Retrieval*, New York, 2006, pp. 321–330.

[48] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *Proc. Int. Conf. Computer Vision*, 2005, pp. II:1331–II:1338.

[49] D. Tell and S. Carlsson, "Combining appearance and topology for wide baseline matching," in *Proc. 7th Eur. Conf. Computer Vision*, Copenhagen, Denmark, May 2002, vol. 2350, *LNCS*, pp. 68–81, Springer-Verlag.

[50] T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proc. 11th British Machine Vision Conf.*, Bristol, U.K., 2000, pp. 412–425.

[51] C. Wallraven and H. Bulthoff, "Automatic acquisition of exemplar-based representations for recognition from image sequences," in *Proc. IEEE Conf. Computer Vision Pattern Recognition, Workshop Models versus Exemplars*, 2001.

[52] I. H. Witten, A. Moffat, and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. San Mateo, CA: Morgan Kaufmann, ISBN: 1558605703, 1999.

ABOUT THE AUTHORS

Josef Sivic, photograph and biography not available at the time of publication.

Andrew Zisserman, photograph and biography not available at the time of publication.