EFFICIENT OBJECT RETRIEVAL FROM VIDEOS

Josef Sivic, Frederik Schaffalitzky and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science, The University of Oxford
http://www.robots.ox.ac.uk/∼vgg

ABSTRACT

We describe an approach to video object retrieval which enables all shots containing the object to be returned in a manner, and with a speed, similar to a Google search for text. The object is specified by a user outlining it in an image, and the object is then delineated in the retrieved shots.

The method is based on three components: (i) an image representation of the object by a set of viewpoint invariant region descriptors so that recognition can proceed successfully despite changes in viewpoint, illumination and partial occlusion; (ii) the use of contiguous frames within a shot in order to improve the estimation of the descriptors and motion group object visual aspects; (iii) vector quantization of the descriptors so that the technology of text retrieval, such as inverted file systems, can be employed at run time.

The method is illustrated on a full length feature film.

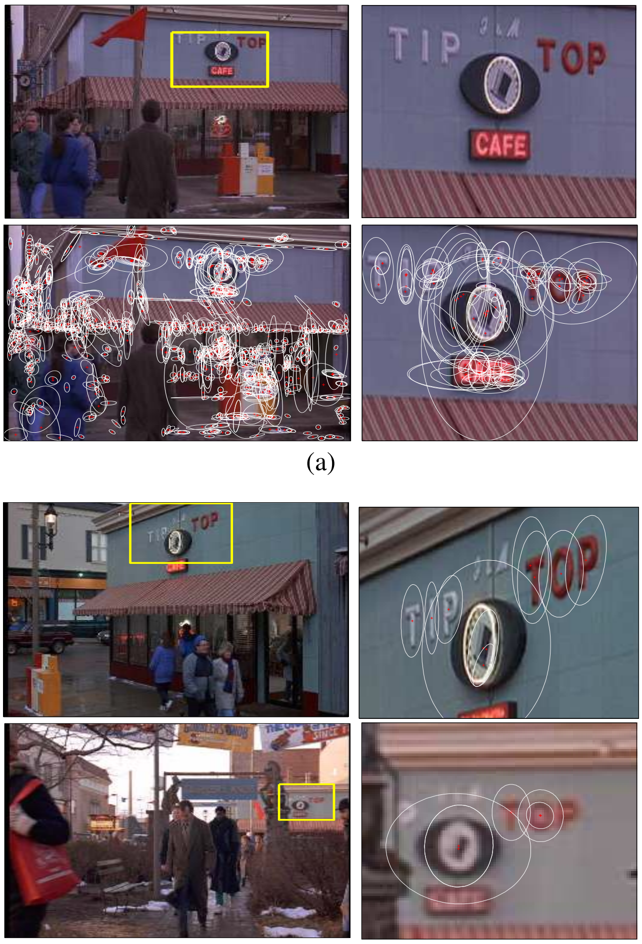
1. INTRODUCTION AND OBJECTIVES

The aim of this work is to retrieve those key frames and shots of a video containing a particular object with the ease, speed and accuracy with which Google retrieves text documents (web pages) containing particular words.

Identifying an (identical) object in a database of images is a challenging problem because an object's visual appearance may be very different due to viewpoint and lighting, and the object may be partially occluded. However, recently a number of successful approaches [2, 3, 6, 7, 8, 9, 10, 14, 15] have been developed based on a *weak segmentation* of the image. Rather than attempt to 'semantically' segment the image, e.g. into foreground object and background, an image is represented by a set of overlapping (local) regions. The region segmentation, and their descriptors, are built with a controlled degree of invariance to viewpoint and illumination conditions. Recognition of a particular object then proceeds by matching the descriptor vectors, followed by disambiguation using local spatial coherence. The result is that objects can be recognized despite significant changes in viewpoint and, due to the multiple local regions, despite partial occlusion since some of the regions will still be visible in such cases.

In this work we cast this approach as one of text retrieval. In essence this requires a visual analogy of a word, and we provide this by quantizing the descriptor vectors. The benefit of this casting is that matches are effectively pre-computed so that at run-time frames and shots containing any particular object can be retrieved immediately. 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.

The method will be illustrated for the feature length film 'Groundhog Day' [Ramis, 1993]. We show examples of shots retrieved based on: (i) objects specified within the film; (ii) other common objects specified by images; and, (iii) different visual aspects of the same object. Examples of similar retrievals using other films are given in [11, 12].



(a)

(b)

Figure 1: **Example object query I.** (a) Top row: (left) a frame from the movie 'Groundhog Day' with a query region in yellow and (right) a close-up of the query region delineating the object of interest. Bottom row: (left) all 1039 detected affine co-variant regions superimposed and (right) close-up of the query region. (b) (left) two retrieved frames with detected region of interest in yellow and (right) a close-up of the images with affine co-variant regions superimposed. These regions match to a subset of the regions shown in (a). Note the significant change in foreshortening and scale between the query image of the object, and the object in the retrieved frames.

This work was funded by EC project VIBES.

2. VIEWPOINT INVARIANT OBJECT RETRIEVAL

In this section we overview a method for retrieving images (for example key frames) containing a particular object, given a single image of the object as a query. The method is based on the viewpoint invariant descriptors which have been developed for wide baseline matching [6, 7, 8, 14, 15], object recognition [2, 7], and image/video retrieval [10, 12].

Each image of the database is represented by a set of overlapping regions (see figure 1a), with each region represented by a descriptor vector computed from its appearance. The regions are segmented in a viewpoint co-variant manner – so that for images of the same scene, the pre-image of the region covers the same scene portion. Details of this segmentation are given below. An object is retrieved by segmenting its image, and computing matches between these regions and those of the database by, for example, nearest neighbour matching of the descriptor vectors. Thus object retrieval is cast as a set of nearest neighbour matches. Incorrect region matches are then excised using local spatial coherence, or global relationships (such as epipolar geometry). Images are then ranked by, for example, the number of remaining region matches. An example is shown in figure 1b. This approach has proven very successful for lightly textured scenes, with matching up to a five fold change in scale reported in [5].

Affine co-variant regions: Two types of affine co-variant regions are used here. The first is constructed by an elliptical shape adaptation about an interest point. The implementation details are given in [6, 8]. The second type of region is constructed using the maximally stable procedure of Matas *et al* [4], where areas are selected from an intensity watershed image segmentation. The regions are those for which the area is approximately stationary as the intensity threshold is varied. 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 between 1000-2000. Each elliptical affine co-variant region is represented by a 128-dimensional vector using the SIFT descriptor developed by Lowe [2]. Combining the SIFT descriptor with affine co-variant regions gives region description vectors which are invariant to affine transformations of the image. Both region detection and the description are computed on monochrome versions of the frames, colour information is not currently used in this work.

Temporal smoothing: 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. Any region which does not survive for more than three frames is rejected. Each region of the track can be regarded as an independent measurement of a common scene region (the pre-image of the detected region), and the estimate of the descriptor for this scene region is computed by averaging the descriptors throughout the track.

3. VISUAL INDEXING USING TEXT RETRIEVAL METHODS

In this section we describe how the retrieval method of section 2 can be recast as a text retrieval system, and thereby benefit from the run-time advantages of such systems. In text retrieval each document is represented by a vector of word frequencies – the 'bag of words model'. Documents are then retrieved, in the first instance, by specifying a query as a set of words, and obtaining the documents by the vectors containing these words as components. However, it is usual to apply a weighting to the components of this vector [1], rather than use the frequency vector directly for indexing.

Here we build a visual analogy of textual words by quantizing the descriptor vectors (see below). The video is then represented as a set of key frames (analogy with documents), and each key frame is represented by a weighted vector of the visual word frequencies it contains. During retrieval, the query vector is given by the visual words contained in a user specified sub-part of a frame. The retrieved frames are ranked (in the first instance) according to the similarity (measured by angles) of their weighted vectors to this query vector. More details of the method and other lessons learnt from text retrieval [13] are given below.

Building a visual vocabulary: The objective here is to vector quantize the descriptors into clusters which will be the visual 'words' for text retrieval. Each descriptor is a 128-vector, and to simultaneously cluster all the descriptors of the movie would be a gargantuan task. Instead a subset of 474 frames is selected. Even with this reduction there are still 200K averaged track descriptors that must be clustered. About 6k clusters are used for Shape Adapted regions, and about 10k clusters for Maximally Stable regions. The number of clusters is chosen empirically to maximize retrieval results on a ground truth data, see [12] for more details. Once the visual words are defined, all the descriptors for a new frame of the movie are assigned to visual words according to the nearest cluster centre to their SIFT descriptor.

Final representation: The video is represented as a set of key frames, and each key frame is represented by the visual words it contains and their position. The original raw images are not used other than for displaying the results. Thus the film is represented by a n_w by n_k matrix $\mathtt{M}$ where n_w is the number of visual words (the vocabulary) and n_k the number of key frames. Each entry of $\mathtt{M}$ specifies the number of times the word appears in that frame. The corresponding positions of visual words within the frame are stored in the inverted file structure (see below).

Stop list: The frequency of occurrence of single words across the whole video (database) is measured, and the top 5% are stopped. This step is inspired by a stop-list in text retrieval applications where poorly discriminating very common words (such as 'the') are discarded. In the visual word case the large clusters often contain specularities (local highlights) that are distributed throughout the frames.

Spatial consistency ranking: Up to this point we have simply used the 'bag of (visual) words' frequency representation, but we have not employed the spatial organization of the words. In a text search engine, such as Google, the ranking is increased 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 a subpart of the image, where matched co-variant regions in the retrieved frames should have a similar spatial arrangement [8, 10] (e.g. compactness) 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 re-ranking them based on a measure of spatial consistency. A search area is defined by the 15 nearest spatial neighbours (in the image) 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 total number of votes determines the rank of the frame.

Implementation – use of inverted files: In a classical file structure all words are stored in the document they appear in. An inverted file structure [1] is organized as an ideal book index. It has an entry (hit list) for each word where all occurrences of the word in all documents are stored. In our case the inverted file has an entry for each visual word, which stores all the matches, i.e. occurrences of the same word in all frames. Each visual word occurrence contains the position of the affine co-variant region within the image and its (precomputed) 15 nearest neighbours which are used for the spatial consistency ranking.

4. EXAMPLES AND CAPABILITIES

Example queries: Figures 1, 2 and 4 show results of object queries for the movie 'Groundhog Day'. The movie contains 5,640 keyframes (1 keyframe a second). Both the actual frames returned and their ranking are excellent – as far as it is possible to tell, no frames containing the object are missed (no false negatives), and the highly ranked frames all do contain the object (good precision).

Searching for objects from outside the movie: Figure 3 shows an example of searching for an object outside the 'closed world' of the film. The object (a Sony logo) is specified by a query image downloaded from the internet. The image was preprocessed as outlined in section 3. Searching for images from other sources opens up the possibility for product placement queries, or searching movies for company logos, particular types of vehicles or buildings.

Automatic association of multiple aspects of a 3D object: An object, such as a vehicle, may be seen from one aspect in a particular shot (e.g. the side of the vehicle) and from a different aspect (e.g. the front) in another shot. Our aim is to automatically associate several visual aspects from shots where these are visible, and thereby enable 3D object matching and retrieval. This is achieved by tracking the affine co-variant regions and grouping the tracks based on independent 3D rigid motion constraints. More details can be found in [11]. When the user selects one aspect of the object, the system queries for all associated aspects (which are precomputed). An example of multiple aspect query is shown in figure 5.

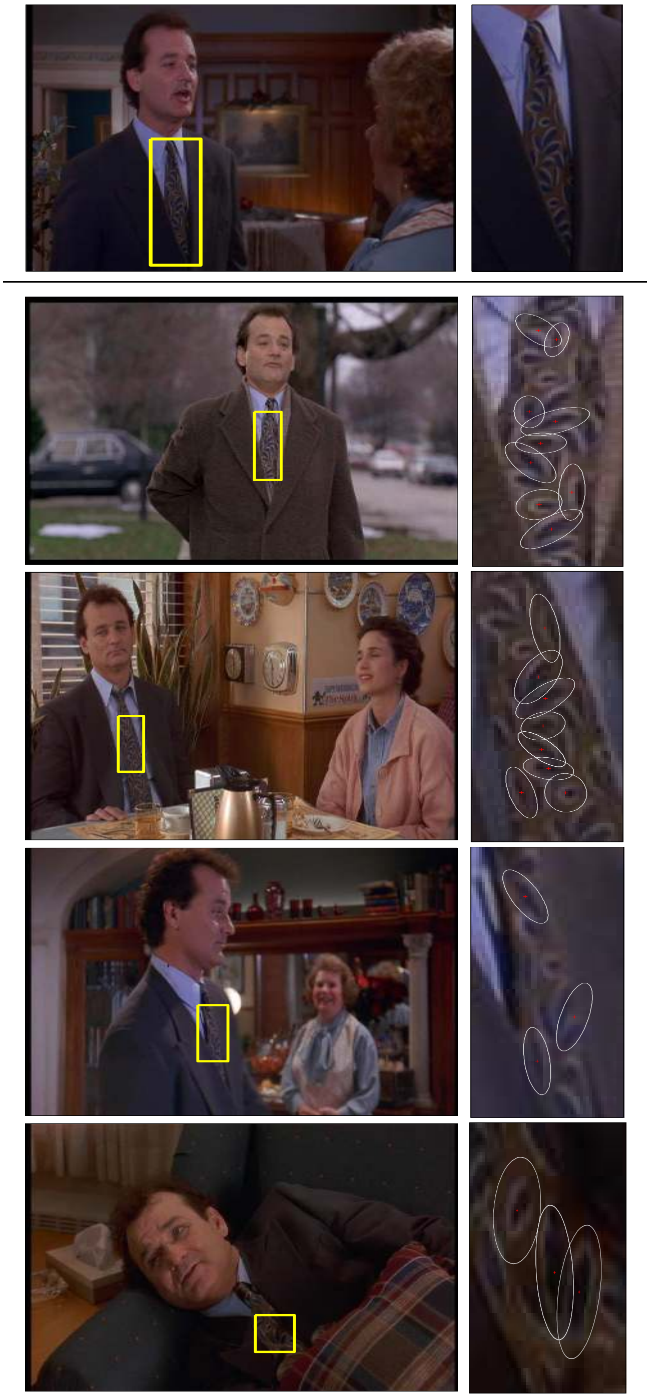


Figure 2: **Example object query II.** First row: (left) frame with user specified query region (a tie) in yellow, and (right) close up of the query region. The four remaining rows show (left) frames from retrieved shots with the identified region of interest shown in yellow, and (right) a close up of the image with matched elliptical regions superimposed. In this case 16 shots were retrieved, 12 correct and the first incorrect shot was ranked 12. Querying all the 5,640 keyframes of the entire movie took 0.38 seconds on a 2GHz pentium.

REFERENCES

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

[2] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.

[3] D. Lowe. Local feature view clustering for 3D object recognition. In *Proc. CVPR*, pages 682–688. Springer, 2001.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC.*, pages 384–393, 2002.

[5] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, 2001.

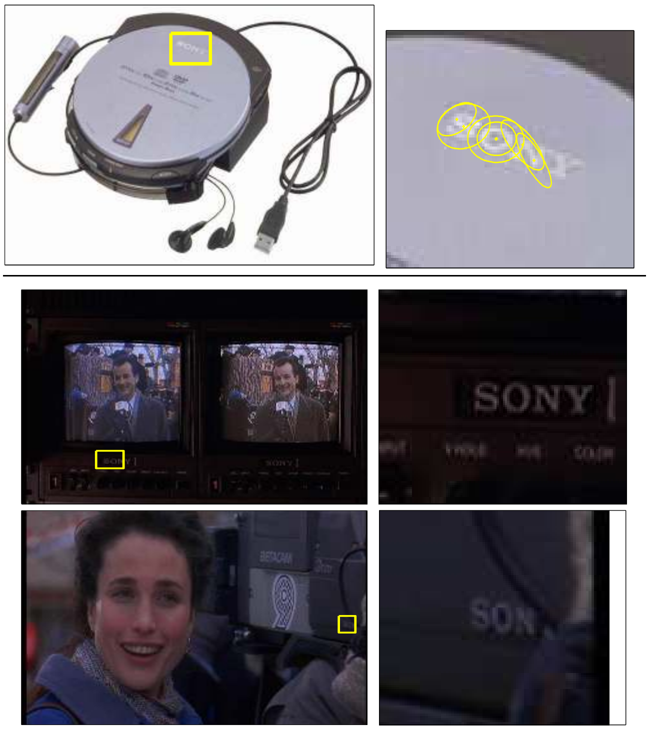


Figure 3: **Searching for a Sony logo.** Top row: (left) Sony Discman image with the query region outlined in yellow and (right) close-up with detected elliptical regions superimposed. Second and third row: (left) frames from two different shots of 'Groundhog Day' with detected Sony logo outlined in yellow and (right) close-up of the image. The retrieved shots were ranked 1 and 4 (from 8 retrieved in total).

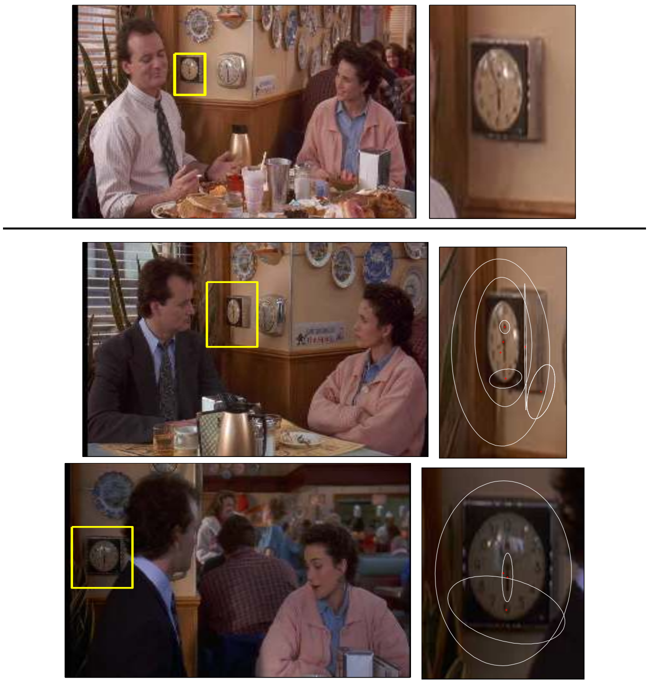
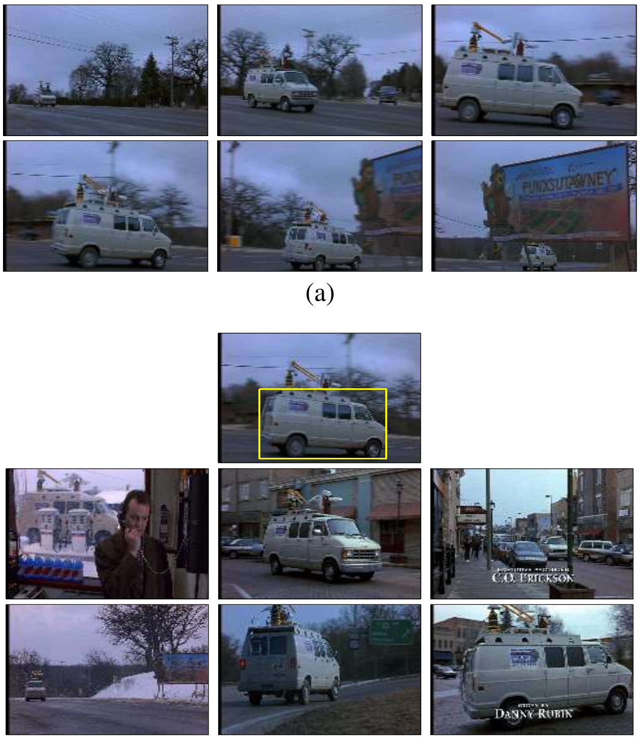


Figure 4: **Query example II.** First row: (left) query region, and (right) its close-up. Next rows: Frames (left) from 6th and 8th retrieved shots and object close-ups (right) with matched regions. The first incorrectly retrieved shot was ranked 7.



(a)

(b)

Figure 5: **Automatic association and querying multiple aspects of a 3D object. (a)** 6 frames from one shot (188 frames long) where the camera is panning right, and the van moves independently. The three aspects (front, side and back) of the van are associated automatically by tracking the elliptical regions and motion grouping tracks belonging to rigidly moving 3D objects. **(b)** Multiple aspect video matching. Top row: The query frame with query region (side of the van) selected by the user. Next two rows: Example frames retrieved from the entire movie by the multiple aspect query. Note that views of the van from the back and front are retrieved.

[6] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.

[7] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. BMVC.*, pages 113–122, 2002.

[8] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, volume 1, pages 414–431. Springer-Verlag, 2002.

[9] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *CVIU*, 92:236–264, 2003.

[10] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE PAMI*, 19(5):530–534, 1997.

[11] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. In *Proc. ECCV*, 2004.

[12] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[13] D.M. Squire, W. Müller, H. Müller, and T. Pun. Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters*, 21:1193–1198, 2000.

[14] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proc. ECCV*, LNCS 2350, pages 68–81. Springer-Verlag, 2002.

[15] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. BMVC.*, pages 412–425, 2000.