

Efficient Visual Content Retrieval and Mining in Videos

Josef Sivic and Andrew Zisserman

Robotics Research Group, Department of Engineering Science
University of Oxford

<http://www.robots.ox.ac.uk/~vgg>

Abstract. We describe an image representation for objects and scenes consisting of a configuration of viewpoint covariant regions and their descriptors. This representation enables recognition to proceed successfully despite changes in scale, viewpoint, illumination and partial occlusion. Vector quantization of these descriptors then enables efficient matching on the scale of an entire feature film. We show two applications. The first is to efficient object retrieval where the technology of text retrieval, such as inverted file systems, can be employed at run time to return all shots containing the object 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 second application is to data mining. We obtain the principal objects, characters and scenes in a video by measuring the reoccurrence of these spatial configurations of viewpoint covariant regions. The applications are illustrated on two full length feature films.

1 Introduction and objectives

Identifying an (identical) object in frames of a video is a challenging problem because an object's visual appearance may be very different due to viewpoint, scale and lighting changes, and partial occlusion. However, recently a number of successful approaches [5, 8, 9, 11, 12, 17] have been developed in the Computer Vision literature 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 which act as 'barcodes' for that object. 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. Matches of descriptor vectors can be pre-computed by vector quantizing, and this in turn enables very efficient applications to be built.

In this work we describe two applications: object retrieval in videos, and data mining in videos. In object retrieval the aim 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.

The second application is to data mining. The objective is to extract significant objects, characters and scenes in a video by determining their frequency of occurrence.



Fig. 1. Object query example 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. Querying all the 5,640 keyframes of the entire movie took 0.36 seconds on a 2GHz Pentium.

For example, the principal actors will be mined because their face or clothes will appear often throughout a film. Similarly, a particular set or scene that re-occurs (e.g. Rick’s bar in ‘Casablanca’) will be ranked higher than those that only occur infrequently (e.g. a particular tree by the highway in a road movie).

There are a number of reasons why it is useful to have commonly occurring objects/characters/scenes. First, they provide entry points for visual search in videos and image databases, or for generating a visual thesaurus [3]. Second, they can be used in forming video summaries [1, 4, 16]. A third application area is in detecting product placements in a film – where frequently occurring logos or labels will be prominent.

The retrieval and data mining methods will be illustrated for the feature length films ‘Groundhog Day’ [Ramis, 1993] and ‘Casablanca’ [Curtiz, 1942].

2 Quantized viewpoint invariant descriptors

We build on work on viewpoint invariant descriptors which has been developed for wide baseline matching [6, 8, 11, 17], object recognition [5, 9, 10], and image/video retrieval [12, 13].

The approach taken in all these cases is to represent an image by a set of overlapping regions, each represented by a vector computed from the region’s appearance. The region segmentation is designed so that the pre-image of the region corresponds to the same surface region, i.e. their shape is not fixed, but automatically adapts based on the underlying image intensities so as to always cover the same physical surface. Note that the regions are computed independently in each image. In short, the segmentation commutes with the viewpoint transformation between images, and such regions are known as *affine covariant* (since the transformation is locally an affinity). Similar descriptors

are computed for all images, and region matches between images are then obtained by, for example, nearest neighbour matching of the descriptor vectors, followed by disambiguating using local spatial coherence or global relationships (such as a homography transformation). This approach has proven very successful for lightly textured scenes, with robustness up to a five fold change in scale reported in [7].

Affine co-variant regions: In this work, two types of affine co-variant regions are computed for each frame. The first is constructed by elliptical shape adaptation about an interest point. The implementation details are given in [8, 11]. The second type of region is constructed using the maximally stable procedure of Matas *et al.* [6] where areas are selected from an intensity watershed image segmentation. 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. An example is shown in figure 1a.

Each elliptical affine covariant region is represented by a 128-dimensional vector using the SIFT descriptor developed by Lowe [5]. Combining the SIFT descriptor with affine covariant regions gives region description vectors which are invariant to affine transformations of the image.

Vector quantized descriptors: The SIFT descriptors are vector quantized using K-means clustering. The clusters are computed from 474 frames of the video, with about 6K clusters for Shape Adapted regions, and about 10K clusters for Maximally Stable regions. All the descriptors for each frame of the video are assigned to the nearest cluster centre to their SIFT descriptor. Vector quantizing brings a huge computational advantage because descriptors in the same clusters are considered matched, and no further matching on individual descriptors is then required. In an analogy with text retrieval these vector quantized descriptors are termed *visual words*: they provide a vocabulary – visual nouns – for representing an object or scene [13].

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.

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. This is the representation we use from here on for retrieval and data mining. The original raw images are not used other than for displaying the results.

3 Efficient retrieval – Video Google

In this section we describe how the representation of section 2 can be used for object retrieval in videos, making an analogy with text based retrieval systems such as ‘Google’. 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

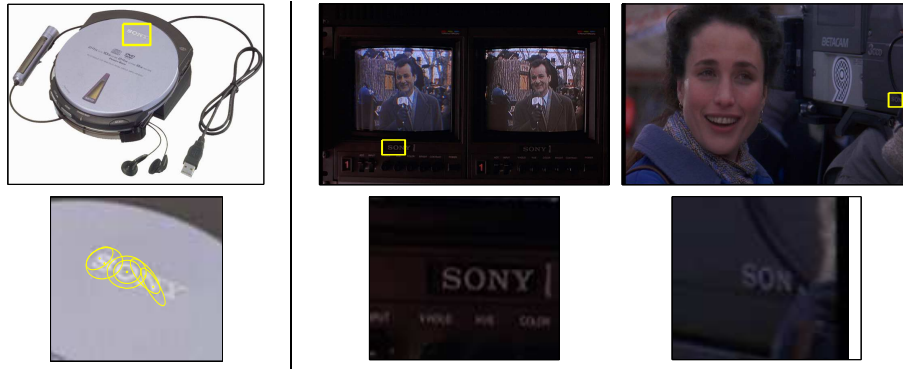


Fig. 2. Object query example II: searching for a Sony logo. First column: (top) Sony Discman image with the query region outlined in yellow and (bottom) close-up with detected elliptical regions superimposed. Second and third column: (top) frames from two different shots of ‘Groundhog Day’ with detected Sony logo outlined in yellow and (bottom) close-up of the image. The retrieved shots were ranked 1 and 4 (from 8 retrieved in total).

a query as a set of words, and obtaining the documents corresponding to the vectors containing those words as components. It is usual to apply a weighting to the components of this vector [2], rather than use the frequency vector directly for indexing.

Here each key frame is represented by a weighted vector of the visual word frequencies it contains. An object query is specified by ‘lassoing’ the image object and thereby specifying its visual words and their configuration. This defines the query vector used for retrieval. The retrieved frames are ranked (in the first instance) according to the similarity (measured by angles) of their weighted vectors to this query vector.

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 [11, 12] to those of the outlined region in the query image. The idea is implemented here by re-ranking the retrieved frames 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. More details of the method and other lessons borrowed from text retrieval [15] are given in [13].

Example queries: Figure 1 shows results of an object query 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). The query takes a fraction of a second on a 2GHz machine.

Searching for objects from outside the movie: Figure 2 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 is preprocessed as outlined in section 2. 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.

A demonstration version of Video Google is available at <http://www.robots.ox.ac.uk/~vgg/research/vgoogle/>.

4 Efficient video mining

In this section we describe how the representation of section 2 can be used for efficient mining of visual content from video.

The objective is to extract significant objects, characters and scenes in a video by determining their frequency of re-occurrence. An object is defined as a spatial configuration of vector quantized viewpoint co-variant regions – visual words. In analogy to the spatial consistency ranking used for retrieving images (described in section 3), a configuration consists of a visual word and its K spatial nearest neighbours. As the segmentation of an object is unknown in advance such a configuration is centred around every visual word in the movie and several scales are used (e.g. $K = 20, 50, 100$). The task then becomes that of measuring the re-occurrence of spatial configurations of visual words over an entire movie. Note the significant difference to the retrieval task described in section 3, where the segmentation of an object is given by a user outlining a query region and essentially only one object is searched for.

Measuring re-occurrence of configurations of visual words: The goal is to group configurations of visual words representing the same objects into clusters and count the number of occurrences within each cluster. A configuration re-occurs (is matched) if at least $M (= 3)$ of the visual words in the configuration are matched. Note that no geometric consistency on visual words (e.g. on their positions) in the configuration is required.

The algorithm consists of three stages. First, only configurations occurring in more than a minimum number of keyframes are considered for clustering. This filtering greatly reduces the data and allows us to focus on only significant (frequently occurring) configurations. Second, significant configurations are matched by a progressive clustering algorithm. At this stage, one object can be represented by multiple clusters which overlap spatially. This is because a configuration is placed at every visual word in a frame and therefore configurations are largely overlapping. Third, clusters representing one object are merged based both on spatial and temporal overlap in multiple keyframes. The algorithm is described in more detail in [14].

Examples: Figures 3 and 4 show samples from different clusters found for the scales of $K = 20, 50$ and 100 neighbourhood in the movie ‘Groundhog Day’. Figure 5 shows samples from clusters found at the 20-neighbourhood scale in the movie ‘Casablanca’. Generally, smaller consistent objects, e.g. faces and logos or objects which change background frequently or get partially occluded, tend to appear at the smaller scale. An example would be the two clocks on the wall in the cafe (objects 7 and 8 of figure 3). On

the larger scales we get (parts of) backgrounds, building fronts or the whole location. An interesting example is the ‘frames’ shop sign (object 9 of figure 3) which is extracted as a separate cluster at the 20-neighbourhood scale, and can be seen again as a subset of the a 100-neighbourhood scale cluster which covers the whole shop entrance (row 1 of figure 4b). Results on other videos and quantitative comparisons with ground truth are given in [14].

Acknowledgements: This research was supported by EC projects Vibes and CogViSys. We are grateful to the ViMining project of IMEDIA INRIA-Rocquencourt for travel funding.

References

1. A. Aner and J. R. Kender. Video summaries through mosaic-based shot and scene clustering. In *Proc. ECCV*. Springer-Verlag, 2002.
2. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, ISBN: 020139829, 1999.
3. N. Boujemaa, J. Fauqueur, and V. Gouet. What’s beyond query by example? In *Trends and Advances in Content-Based Image and Video Retrieval*, 2004.
4. Y. Gong and X. Liu. Generating optimal video summaries. In *IEEE Intl. Conf. on Multimedia and Expo (III)*, pages 1559–1562, 2000.
5. D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, Sep 1999.
6. 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.
7. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, 2001.
8. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.
9. S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. BMVC.*, pages 113–122, 2002.
10. 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. CVPR*, 2003.
11. 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.
12. C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE PAMI*, 19(5):530–534, May 1997.
13. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, Oct 2003.
14. J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. CVPR*, 2004.
15. 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.
16. B. Tseng, C.-Y. Lin, and J. R. Smith. Video personalization and summarization system. In *MMSP*, 2002.
17. T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. BMVC.*, pages 412–425, 2000.



Fig. 3. Mining Groundhog Day I. Examples of mined clusters at the 20 neighbourhood scale. Each row shows ten samples from one cluster. The first two rows correspond to faces of the two main characters. The next two rows show two different ties of the main character. The remaining rows show various objects that occur often in the movie. The images shown cover a rectangular convex hull of the matched configurations of viewpoint covariant regions within the frame plus a margin of 10 pixels. The rectangles are resized to squares for this display.



Fig. 4. Mining Groundhog Day II. Objects and scenes mined on the scale of (a) 50-neighbourhood and (b) 100-neighbourhood. The clusters extend over (a) 7,21,3 shots, (b) 7,3,5 shots (top-down).



Fig. 5. Mining Casablanca. Examples of objects mined on the scale of 20-neighbourhood. Examples include the main characters, parts of clothing of other characters (e.g. uniforms) and objects (lamps) in Rick's bar.