Video Google: A Text Retrieval Approach to Object Matching in Videos

Josef Sivic and Andrew Zisserman
Robotics Research Group, Department of Engineering Science
University of Oxford, United Kingdom

Abstract

We describe an approach to object and scene retrieval which searches for and localizes all the occurrences of a user outlined object in a video. The object is represented by a set of viewpoint invariant region descriptors so that recognition can proceed successfully despite changes in viewpoint, illumination and partial occlusion. The temporal continuity of the video within a shot is used to track the regions in order to reject unstable regions and reduce the effects of noise in the descriptors.

The analogy with text retrieval is in the implementation where matches on descriptors are pre-computed (using vector quantization), and inverted file systems and document rankings are used. The result is that retrieval is immediate, returning a ranked list of key frames/shots in the manner of Google.

The method is illustrated for matching on two full length feature films.

1. Introduction

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. This paper investigates whether a text retrieval approach can be successfully employed for object recognition.

Identifying an (identical) object in a database of images is now reaching some maturity. It is still a challenging problem because an object's visual appearance may be very different due to viewpoint and lighting, and it may be partially occluded, but successful methods now exist. Typically an object is represented by a set of overlapping regions each represented by a vector computed from the region's appearance. The region segmentation 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 neighbour matching of the descriptor vectors, followed by disambiguating using local spatial coherence (such as neighbourhoods, ordering, or spatial layout), or global relationships (such as epipolar geometry). Examples include [5, 6, 8, 11, 13, 12, 14, 16, 17].

We explore whether this type of approach to recognition can be recast as text retrieval. In essence this requires a visual analogy of a word, and here we provide this by vector quantizing the descriptor vectors. However, it will be seen that pursuing the analogy with text retrieval is more than a simple optimization over different vector quantizations. There are many lessons and rules of thumb that have been learnt and developed in the text retrieval literature and it is worth ascertaining if these also can be employed in visual retrieval.

The benefits of this approach is that matches are effectively pre-computed 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. However, we must also determine whether this vector quantized retrieval misses any matches that would have been obtained if the former method of nearest neighbour matching had been used.

Review of text retrieval: Text retrieval systems generally employ a number of standard steps [1]. The documents are first parsed into words. Second the words are represented by their stems, for example 'walk', 'walking' and 'walks' would be represented by the stem 'walk'. Third a stop list is 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 (described in more detail in section 4), and in the case of Google the weighting of a web page depends on the number of web pages linking to that particular page [3]. 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* [18] 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

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

document) in which the word occurs.

A text is retrieved by computing its vector of word frequencies and returning the documents with the closest (measured by angles) vectors. In addition the match on the ordering and separation of the words may be used to rank the returned documents.

Paper outline: Here we explore visual analogies of each of these steps. Section 2 describes the visual descriptors used. Section 3 then describes their vector quantization into visual 'words', and section 4 weighting and indexing for the vector model. These ideas are then evaluated on a ground truth set of frames in section 5. Finally, a stop list and ranking (by a match on spatial layout) are introduced in section 6, and used to evaluate object retrieval throughout two feature films: 'Run Lola Run' ('Lola Rennt') [Tykwer, 1999], and 'Groundhog Day' [Ramis, 1993].

Although previous work has borrowed ideas from the text retrieval literature for image retrieval from databases (e.g. [15] used the weighting and inverted file schemes) to the best of our knowledge this is the first systematic application of these ideas to object retrieval in videos.

2. Viewpoint invariant description

Two types of viewpoint covariant regions are computed for each frame. The first is constructed by elliptical shape adaptation about an interest point. The method involves iteratively determining the ellipse centre, scale and shape. The scale is determined by the local extremum (across scale) of a Laplacian, and the shape by maximizing intensity gradient isotropy over the elliptical region [2, 4]. The implementation details are given in [8, 13]. 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 [7]. 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 gray 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 1600. An example is shown in Figure 1.

Each elliptical affine invariant region is represented by a 128-dimensional vector using the SIFT descriptor developed by Lowe [5]. In [9] this descriptor was shown to be superior to others used in the literature, such as the response of a set of steerable filters [8] or orthogonal filters [13], and we have also found SIFT to be superior (by comparing scene retrieval results against ground truth as in section 5.1). The 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 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, colour information is not currently used in this work.

Figure 1: Top row: Two frames showing the same scene from very different camera viewpoints (from the film 'Run Lola Run'). Middle row: frames with detected affine invariant regions superimposed. 'Maximally Stable' (MS) regions are in yellow. 'Shape Adapted' (SA) regions are in cyan. Bottom row: Final matched regions after indexing and spatial consensus. Note that the correspondences define the scene overlap between the two frames.

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

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

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. This gives a measurable improvement in the signal to noise of the descriptors (which again has been demonstrated using the ground truth tests of section 5.1).

3. Building a visual vocabulary

The objective here is to vector quantize the descriptors into clusters which will be the visual 'words' for text retrieval. Then when a new frame of the movie is observed each descriptor of the frame is assigned to the nearest cluster, and this immediately generates matches for all frames throughout the movie. The vocabulary is constructed from a subpart of the movie, and its matching accuracy and expressive power are evaluated on the remainder of the movie, as described in the following sections.

The vector quantization is carried out here by K-means clustering, though other methods (K-medoids, histogram binning, etc) are certainly possible.

3.1. Implementation

Regions are tracked through contiguous frames, and a mean vector descriptor $\bar{\mathbf{x}}_i$ computed for each of the i regions. To reject unstable regions the 10% of tracks with the largest diagonal covariance matrix are rejected. This generates an average of about 1000 regions per frame.

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 48 shots is selected (these shots are discussed in more detail in section 5.1) covering about 10k frames which represent about 10% of all the frames in the movie. Even with this reduction there are still 200K averaged track descriptors that must be clustered.

To determine the distance function for clustering the Mahalanobis distance is computed as follows: it is assumed that the covariance Σ is the same for all tracks, and this is computed by estimating from *all* the available data, i.e. all descriptors for all tracks in the 48 shots. The Mahalanobis distance enables the more noisy components of the 128–vector to be weighted down, and also decorrelates the components. Empirically there is a small degree of correlation. The distance function between two descriptors (represented by their *mean* track descriptors) $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2$, is then given by $d(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2) = \sqrt{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \Sigma^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)}$. As is standard, the descriptor space is affine transformed by the square root of Σ so that Euclidean distance may be used.

About 6k clusters are used for Shape Adapted regions, and about 10k clusters for Maximally Stable 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 number of clusters is chosen empirically to maximize retrieval results on the ground truth set of section 5.1. The K-means algorithm is run several times with random initial assignments of points as cluster centres, and the best result used.

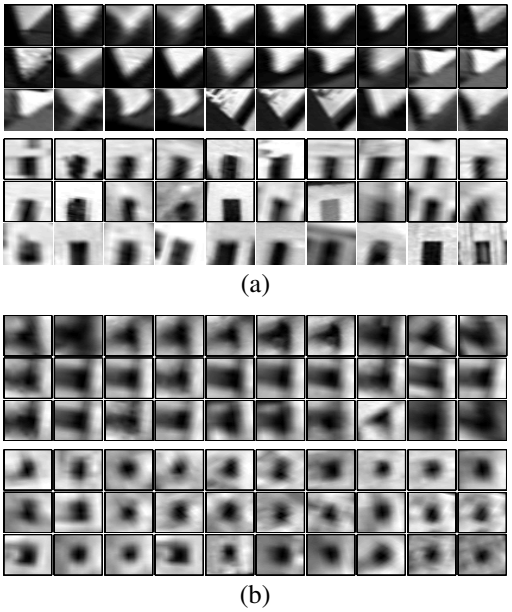


Figure 2: Samples from the clusters corresponding to a single visual word. (a) Two examples of clusters of Shape Adapted regions. (b) Two examples of clusters of Maximally Stable regions.

Figure 2 shows examples of regions belonging to particular clusters, i.e. which will be treated as the same visual word. The clustered regions reflect the properties of the SIFT descriptors which penalize variations amongst regions less than cross-correlation. This is because SIFT emphasizes orientation of gradients, rather than the position of a particular intensity within 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 state of repair of a building.

4. Visual indexing using text retrieval methods

In text retrieval each document is represented by a vector of word frequencies. 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 describe the standard weighting that is employed, and then the visual analogy of document retrieval to frame retrieval.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

The standard weighting is known as 'term frequency–inverse document frequency', *tf-idf*, and is computed as follows. Suppose there is a vocabulary of k words, then each document is represented by a k-vector $V_d = (t_1, ..., t_i, ..., t_k)^\top$, of weighted word frequencies with components

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

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 occurrences of term i in the whole database 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 word frequency weights words occurring often in a particular document, and thus describe it well, whilst the inverse document frequency down-weights words that appear often in the database.

At the retrieval stage documents are ranked by their normalized scalar product (cosine of angle) between the query vector V_q and all document vectors V_d in the database.

In our case the query vector is given by the visual words contained in a user specified sub-part of a frame, and the other frames are ranked according to the similarity of their weighted vectors to this query vector. Various weighting models are evaluated in the following section.

5. Experimental evaluation of scene matching using visual words

Here the objective is to match scene locations within a closed world of shots [12]. The method is evaluated on 164 frames from 48 shots taken at 19 different 3D locations in the movie Run Lola Run. We have between 4-9 frames from each location. Examples of three frames from each of four different locations are shown in figure 3a. There are significant viewpoint changes over the triplets of frames shown for the same location. Each frame of the triplet is from a different (and distant in time) shot in the movie.

In the retrieval tests the entire frame is used as a query region. The retrieval performance is measured over all 164 frames using each in turn as a query region. The correct retrieval consists of all the other frames which show the same location, and this ground truth is determined by hand for the complete 164 frame set.

The retrieval performance is measured using the average normalized rank of relevant images [10] given by

$$\widetilde{Rank} = \frac{1}{NN_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \right)$$

where N_{rel} is the number of relevant images for particular query image, N is the size of the image set, and R_i is the rank of the ith relevant image. In essence $\widetilde{Rank}$ is zero if all N_{rel} images are returned first. The $\widetilde{Rank}$ measure lies in the range 0 to 1, with 0.5 corresponding to random retrieval.

5.1. Ground truth image set results

Figure 3b shows the average normalized rank using each image of the data set as a query image with the *tf-idf* weighting described in section 4. The benefit in having two feature types is evident. The combination of both clearly gives better performance than either one alone. The performance of each feature type varies for different frames or locations. For example, in frames 46-49 MS regions perform better, and conversely for frames 126-127 SA regions are superior.

The retrieval ranking is perfect for 17 of the 19 locations, even those with significant viewpoint changes. The ranking results are less impressive for images 61-70 and 119-121, though even in these cases the frame matches are not missed just low ranked. This is due to a lack of regions in the overlapping part of the scene, see figure 4. This is not a problem of vector quantization (the regions that are in common are correctly matched), but due to few features being detected for this type of scene (pavement texture). We return to this point in section 7.

Table 1 shows the mean of the $\widetilde{Rank}$ measure computed from all 164 images for three standard text retrieval term weighting methods [1]. The *tf-idf* weighting outperforms both the binary weights (i.e. the vector components are one if the image contains the descriptor, zero otherwise) and term frequency weights (the components are the frequency of word occurrence). The differences are not very significant for the ranks averaged over the whole ground truth set. However, for particular frames (e.g. 49) the difference can be as high as 0.1.

The average precision recall curve for all frames is shown in figure 3c. For each frame as a query, we have computed precision as the number of relevant images (i.e. of the same location) relative to the total number of frames retrieved, and recall as the number of correctly retrieved frames relative to the number of relevant frames. Again the benefit of combining the two feature types is clear.

These retrieval results demonstrate that there is no loss of performance in using vector quantization (visual words) compared to direct nearest neighbour (or ε-nearest neighbour) matching of invariants [12].

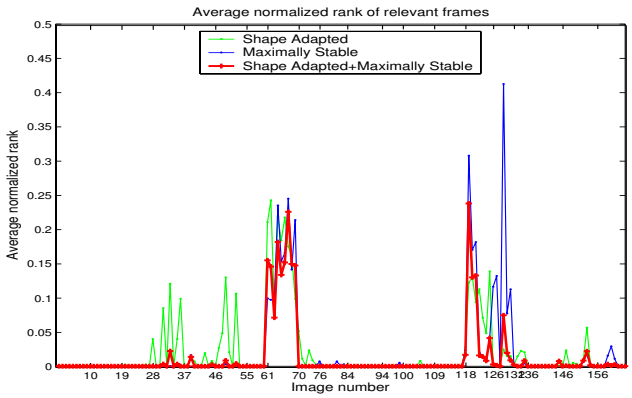
This ground truth set is also used to learn the system parameters including: the number of cluster centres; the minimum tracking length for stable features; and the proportion of unstable descriptors to reject based on their covariance.

6. Object retrieval

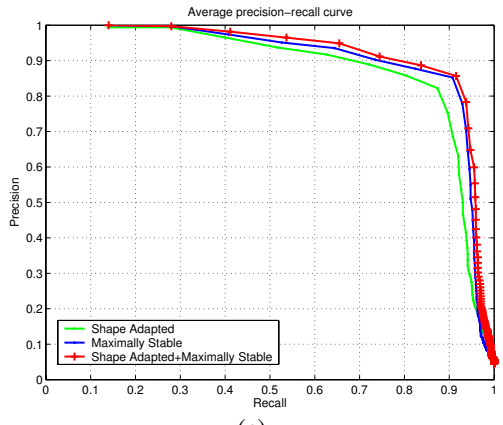
In this section we evaluate searching for objects throughout the entire movie. The object of interest is specified by the

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

(a)



(b)



(c)

Figure 3: **Ground truth data.** (a) Each row shows a frame from three different shots of the same location in the ground truth data set. (b) Average normalized rank for location matching on the ground truth set. (c) Average Precision-Recall curve for location matching on the ground truth set.

	binary	*tf*	*tf-idf*
SA	0.0265	0.0275	0.0209
MS	0.0237	0.0208	0.0196
SA+MS	0.0165	0.0153	0.0132

Table 1: The mean of the $\widetilde{Rank}$ measure computed from all 164 images of the ground truth set for different term weighting methods.

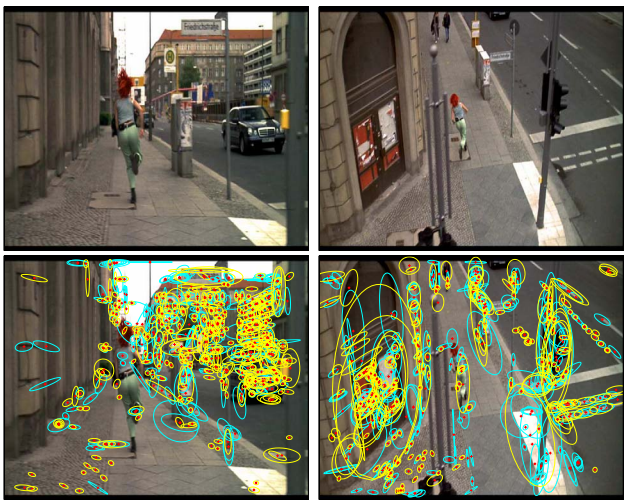


Figure 4: Top: Frames 61 and 64 from the ground truth data set. A poor ranking score is obtained for this pair. Bottom: superimposed detected affine invariant regions. The careful reader will note that, due to the very different viewpoints, only two of the 564 (left) and 533 (right) regions correspond between frames.

user as a sub-part of any frame.

A feature length film typically has 100K-150K frames. To reduce complexity one keyframe is used per second of the video. Descriptors are computed for stable regions in each keyframe and the mean values are computed using two frames either side of the keyframe. The descriptors are vector quantized using the centres clustered from the ground truth set.

Here we are also evaluating the expressiveness of the visual vocabulary since frames outside the ground truth set contain new objects and scenes, and their detected regions have not been included in forming the clusters.

6.1. Stop list

Using a stop list analogy the most frequent visual words that occur in almost all images are suppressed. Figure 5 shows the frequency of visual words over all the keyframes of Lola. The top 5% and bottom 10% are stopped. In our case the very common words are due to large clusters of over 3K points. The stop list boundaries were determined empirically to reduce the number of mismatches and size of the inverted file while keeping sufficient visual vocabulary.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

Figures 6 show the benefit of imposing a stop list – the very common visual words occur at many places in the image and are responsible for mis-matches. Most of these are removed once the stop list is applied. The removal of the remaining mis-matches is described next.

6.2. Spatial consistency

Google increases 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 a subpart of the image, where matched covariant regions in the retrieved frames should have a similar spatial arrangement [12, 14] (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.

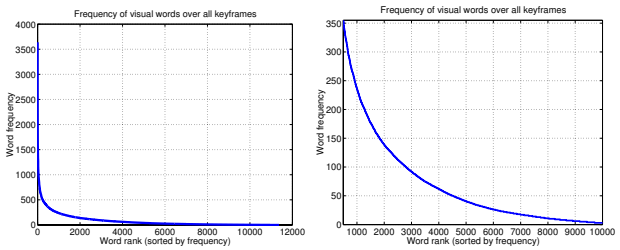
Spatial consistency can be measured quite loosely simply by requiring that neighbouring matches in the query region lie in a surrounding area in the retrieved frame. It can also be measured very strictly by requiring that neighbouring matches have the same spatial layout in the query region and retrieved frame. In our case the matched regions provide the affine transformation between the query and retrieved image so a point to point map is available for this strict measure.

We have found that the best performance is obtained in the middle of this possible range of measures. A search area is defined by the 15 nearest neighbours 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. This works very well as is demonstrated in the last row of figure 6, which shows the spatial consistency rejection of incorrect matches. The object retrieval examples of figures 7 to 9 employ this ranking measure and amply demonstrate its usefulness.

Other measures which take account of the affine mapping between images may be required in some situations, but this involves a greater computational expense.

6.3. Object retrieval

Implementation – use of inverted files: In a classical file structure all words are stored in the document they appear in. An inverted file structure 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. The document vector is very sparse and use of an inverted file makes the retrieval very fast. Querying a database of 4k frames takes about 0.1 second with a Matlab implementation on a 2GHz pentium.



(a) (b)

Figure 5: Frequency of MS visual words among all 3768 keyframes of Run Lola Run (a) before, and (b) after, application of a stoplist.

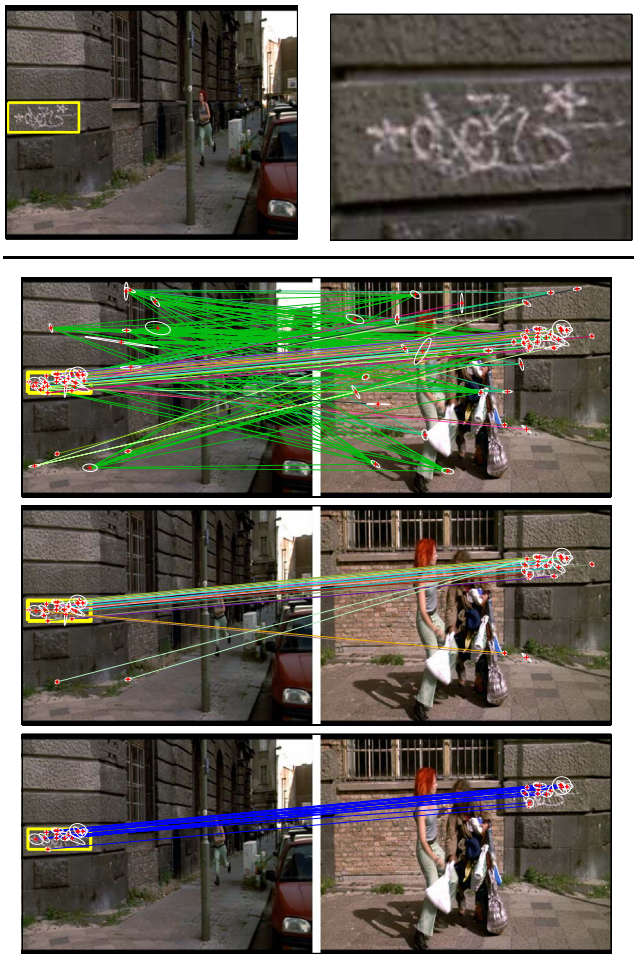


Figure 6: Matching stages. Top row: (left) Query region and (right) its close-up. Second row: Original word matches. Third row: matches after using stop-list, Last row: Final set of matches after filtering on spatial consistency.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

Example queries: Figures 7 and 8 show results of two object queries for the movie 'Run Lola Run', and figure 9 shows the result of an object query on the film 'Groundhog day'. Both movies contain about 4K keyframes. 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 object query results do demonstrate the expressive power of the visual vocabulary. The visual words learnt for Lola are used unchanged for the Groundhog Day retrieval.

7. Summary and Conclusions

The analogy with text retrieval really has demonstrated its worth: we have immediate run-time object retrieval throughout a movie database, despite significant viewpoint changes in many frames. The object is specified as a subpart of an image, and this has proved sufficient for quasi-planar rigid objects.

There are, of course, improvements that can be made mainly to overcome problems in the visual processing. Low rankings are currently due to a lack of visual descriptors for some scene types. However, the framework allows other existing affine co-variant regions to be added (they will define an extended visual vocabulary), for example those of [17]. Another improvement would be to define the object of interest over more than a single frame to allow for search on all its visual aspects.

The text retrieval analogy also raises interesting questions for future work. In text retrieval systems the textual vocabulary is not static, growing as new documents are added to the collection. Similarly, we do not claim that our vector quantization is universal for all images. So far we have learnt vector quantizations sufficient for two movies, but ways of upgrading the visual vocabulary will need to be found. One could think of learning visual vocabularies for different scene types (e.g. city scape vs a forest).

Finally, we now have the intriguing possibility of following other successes of the text retrieval community, such as latent semantic indexing to find content, and automatic clustering to find the principal objects that occur throughout the movie.

Acknowledgements We are grateful to David Lowe, Jiri Matas, Krystian Mikolajczyk and Frederik Schaffalitzky for supplying their region detector/descriptor codes. Thanks to Andrew Blake, Mark Everingham, Andrew Fitzgibbon, Krystian Mikolajczyk and Frederik Schaffalitzky for fruitful discussions. This work was funded by EC project VIBES.

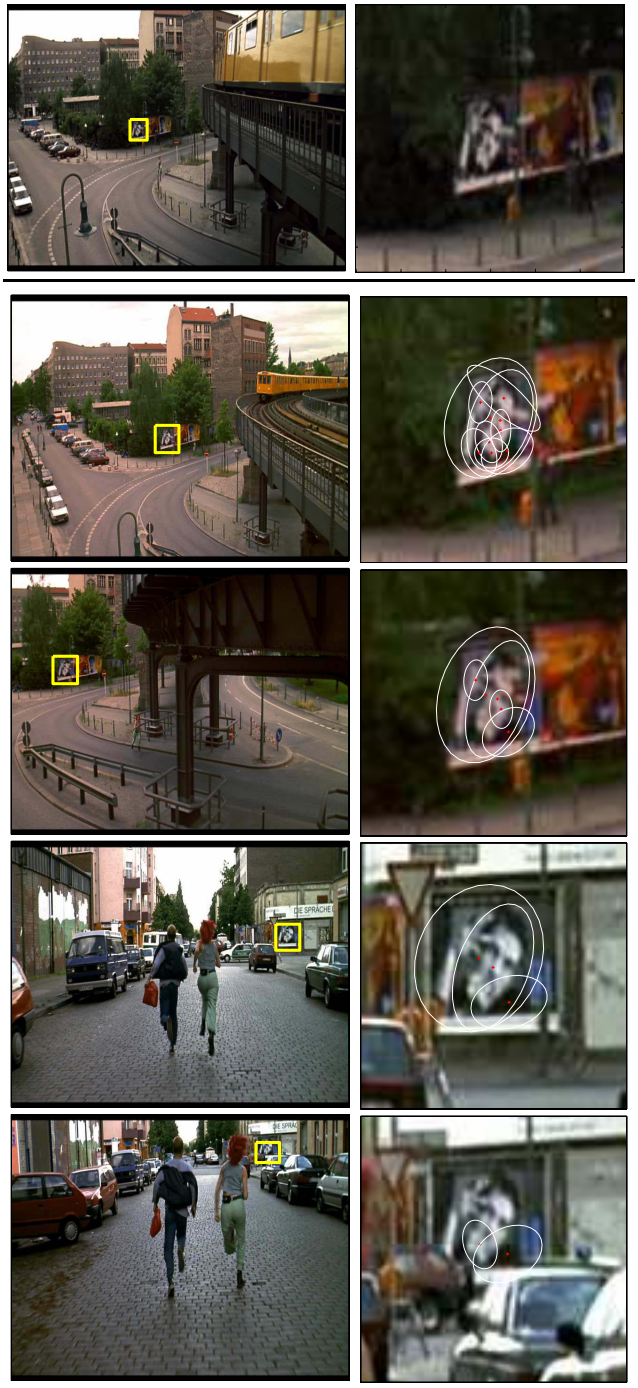


Figure 7: **Object query example I.** First row: (left) frame with user specified query region (a poster) in yellow, and (right) close up of the query region. The four remaining rows show (left) the 1st, 12th, 16th, and 20th retrieved frames with the identified region of interest shown in yellow, and (right) a close up of the image with matched elliptical regions superimposed. In this case 20 keyframes were retrieved: six from the same shot as the query image, the rest from different shots at later points in the movie. All retrieved frames contain the specified object. Note the poster appears on various billboards throughout the movie (and Berlin).

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

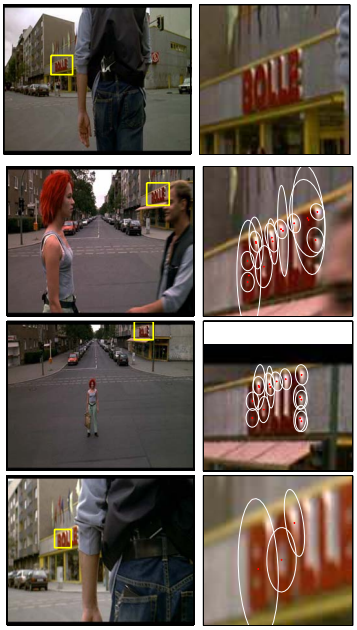


Figure 8: **Object query example II.** Run Lola Run. First row: (left) query region, and (right) its close up. Next rows: The 9th, 16th and 25th retrieved frames (left) and object close-ups (right) with matched regions. 33 keyframes were retrieved. 31 contained the object. The two incorrect frames were ranked 29 and 30.

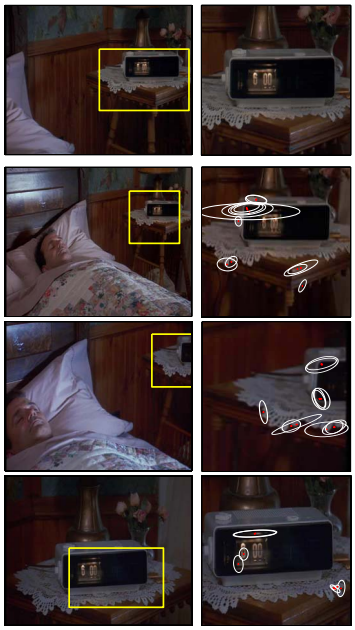


Figure 9: **Object query example III.** Groundhog Day. First row: (left) query region, and (right) its close up. Next rows: The 12th, 35th and 50th retrieved frames (left) and object close-ups with matched regions (right). 73 keyframes were retrieved of which 53 contained the object. The first incorrect frame was ranked 27th.

References

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

[2] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. CVPR*, pages 774–781, 2000.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *7th Int. WWW Conference*, 1998.

[4] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In *Proc. ECCV*, LNCS 800, pages 389–400, 1994.

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

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

[7] 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.

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

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, 2003.

[10] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel - evaluation in image retrieval. In *Proc. CIVR*, 2002.

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

[12] F. Schaffalitzky and A. Zisserman. Automated scene matching in movies. In *Proc. CIVR2002*, LNCS 2383, pages 186–197. Springer-Verlag, 2002.

[13] 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.

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

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

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

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**