International Journal of Computer Vision 67(2), 189–210, 2006

DOI: 10.1007/s11263-005-4264-y

Object Level Grouping for Video Shots

JOSEF SIVIC , FREDERIK SCHAFFALITZKY AND ANDREW ZISSERMAN
Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, UK

Received September 21, 2004; Revised April 11, 2005; Accepted May 3, 2005

First online version published in January, 2006

Abstract. We describe a method for automatically obtaining object representations suitable for retrieval from generic video shots. The object representation consists of an association of frame regions. These regions provide exemplars of the object's possible visual appearances.

Two ideas are developed: (i) associating regions within a single shot to represent a deforming object; (ii) associating regions from the multiple visual aspects of a 3D object, thereby implicitly representing 3D structure. For the association we exploit temporal continuity (tracking) and wide baseline matching of affine covariant regions.

In the implementation there are three areas of novelty: First, we describe a method to repair short gaps in tracks. Second, we show how to join tracks across occlusions (where many tracks terminate simultaneously). Third, we develop an affine factorization method that copes with motion degeneracy.

We obtain tracks that last throughout the shot, without requiring a 3D reconstruction. The factorization method is used to associate tracks into object-level groups, with common motion. The outcome is that separate parts of an object that are not simultaneously visible (such as the front and back of a car, or the front and side of a face) are associated together. In turn this enables object-level matching and recognition throughout a video.

We illustrate the method on the feature film "Groundhog Day." Examples are given for the retrieval of deforming objects (heads, walking people) and rigid objects (vehicles, locations).

Keywords: 3D object retrieval in videos, tracking affine covariant regions, independent motion segmentation, robust affine factorization

1. Introduction

In image and video retrieval applications it is usual to specify a query by an image of the object of interest. Such queries enable retrieval of objects with a limited 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. However, shots in a video do contain examples of objects undergoing viewpoint changes and deformations. Our objective in this paper is to use such multiple instances of an object in a shot in order to enable true object-level retrieval, including: (i) deformable objects, e.g. a face changing expression; and (ii) multiple visual aspects of a 3D object, e.g. a vehicle seen from the front, side, and back. Figure 1 shows example shots of a deforming object, and of multiple visual aspects of a 3D object.

The approach we take is to automatically associate regions of frames of the shot into object-level groupings. This is carried out using both motion and appearance consistency throughout the shot. The technology we employ is that of affine covariant regions (Matas et al., 2002; Mikolajczyk and Schmid, 2002; Schaffalitzky and Zisserman, 2002; Tuytelaars and Van Gool, 2000). These regions deform with viewpoint so that their pre-image corresponds to the same surface patch.

To achieve object-level grouping we have developed the state of the art in two areas: first, the affine covariant regions are used to repair short gaps in

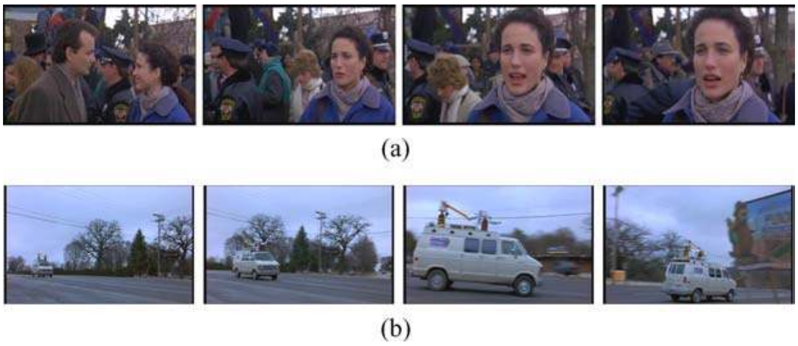


Figure 1. Two example shots from the film 'Groundhog Day' [Ramis, 1993]. (a) Frames from a shot of an actor turning her head and speaking. Tracks of affine covariant regions are used to associate multiple exemplars of the face (from different viewpoints and with different expressions) for retrieval. (b) Frames from a shot where the camera pans to follow a van passing by. Tracks are used to associate the three visual aspects (back, side and front) of the van which are never visible simultaneously in a single frame.

tracks (Section 3), and also to associate a set of tracks when the object is partially or totally occluded for a period (Section 6). The result is that regions are matched throughout the shot whenever they appear. Second, we develop a method of independent motion segmentation using robust affine factorization (Section 5) which is able to handle degenerate motions (Torr et al., 1998) in addition to the usual problems of missing and mis-matched points (Aanaes et al., 2002; De la Torre and Black, 2003; Jacobs, 1997; Shum et al., 1995).

The task we carry out differs from that of layer extraction (Torr et al., 2001), or dominant motion detection where generally 2D planes are extracted, though we build on these approaches. Here the object may be 3D, and we pay attention to this, and also it may not always be the foreground layer as it can be partially or totally occluded for part of the sequence.

Approaches for matching and representing 3D objects using local patches include that of Rothganger et al. (2003) where a 3D object model is built from still images and that of Lowe (2001) and Ferrari et al. (2004a), where a 3D object is modelled as a collection of images with known multiple view region correspondences. In our case we do not enforce global 3D consistency — the 3D object is represented implicitly by a set of exemplar images and this loose coupling allows a degree of deformation (e.g. for facial expressions). Also, we build this object model automatically from video shots despite background clutter. Recently, a similar idea of object model building from video has appeared in Rothganger et al. (2004) but the focus is more on model building rather than matching, recognition and retrieval, and only rigid objects are considered.

Other approaches to building appearance models from video include that of Mahindroo et al. (2002), where optic-flow based motion segmentation is used to extract objects from video, and that of Wallraven and Bulthoff (2001) where an object is modelled by selecting keyframes (using point tracking) from sequences of single objects (some of which are artificial).

The rest of the paper is organized as follows: Sections 2 and 3 review affine region detection and describe the region tracking algorithm. We then give two retrieval applications. First, Section 4, using region tracks alone to associate exemplars for a deforming object — this enables retrieval of the deforming object (a person talking and turning their head). Second, Section 7, using region tracks and independent motion segmentation to associate exemplars for different aspects of a 3D object — this enables object-level retrieval. The independent motion segmentation requires a rigidity grouping, and an algorithm for this is described in Section 5. Section 6 shows how wide base-line matching is used to associate repeated appearances of an object within a shot. The performance of the object-level retrieval is assessed against ground truth in Section 7.1. Finally, in Section 8 the proposed method and its possible extensions are discussed.

We illustrate the method on objects in the feature film 'Groundhog Day' [Ramis, 1993]. The film has 145K frames and 752 shots. This object-level matching naturally extends the frame based matching of 'Video Google' (Sivic and Zisserman, 2003). This paper is an extended version of Sivic et al. (2004).

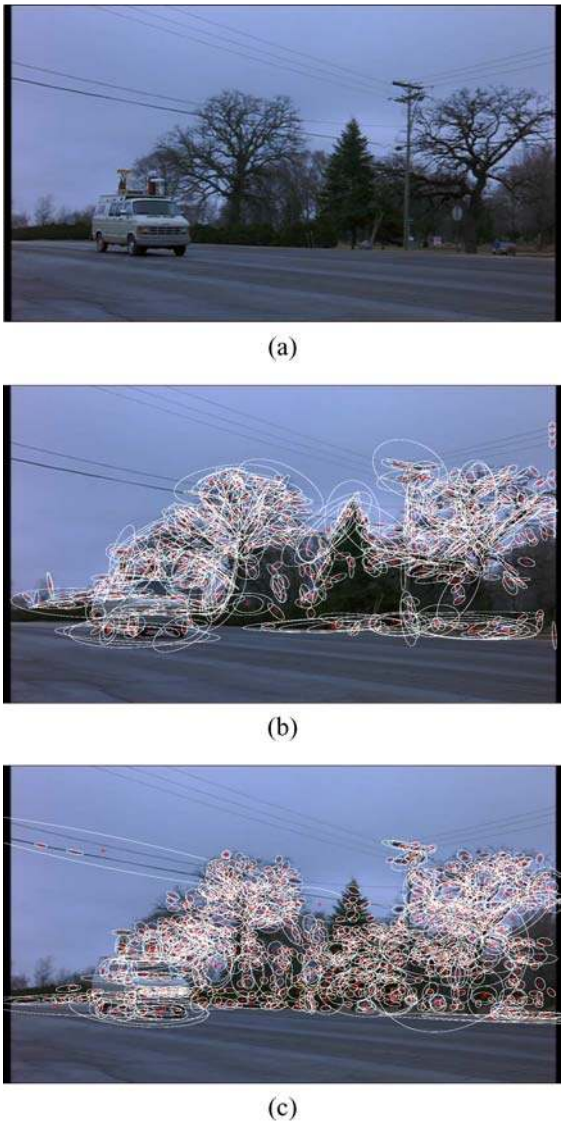
2. Region Detection and Basic Tracking

In this section we describe how regions are detected and tracked (associated) through a shot. Affine covariant regions are detected independently in each frame. The tracking then proceeds sequentially, looking at only two consecutive frames at a time. The objective is to obtain correct matches between the frames which can then be extended to multi-frame tracks. Two matching constraints are used here: first, incorrect matches can be removed by requiring consistency with multiple view geometric relations, second, the regions can be matched on their appearance. The first matching constraint is based on the motion of rigid objects, and the robust estimation of these relations for point matches is mature (Hartley and Zisserman, 2000). The constraint is applied here to the region centroids. The second matching constraint is on the image appearance within the segmented region. It is here that we benefit significantly from using affine covariant regions. This constraint is far more discriminating and tolerant to viewpoint change than the usual cross-correlation over a *square* window used in interest point trackers, since the correct support for the cross-correlation is used here.

2.1. Affine Covariant Regions

Two types of affine covariant region detector are used: one based on interest point neighbourhoods (Mikolajczyk and Schmid, 2002), the other based on the “Maximally Stable Extremal Regions” (MSER) approach of Matas et al. (2002). In both cases the detected region is represented by an ellipse. The region segmentation is designed so that the pre-image of the region corresponds to the same surface region, i.e. their image shape is not fixed, but automatically adapts based on the underlying image intensities so as to always cover the same physical surface. The regions are called *affine covariant* because the segmentation commutes with the viewpoint transformation between images (and the transformation is locally an affinity). Implementation details of these two methods are given in the citations.

It is beneficial to have more than one type of region detector because in some imaged locations a particular type of feature may not occur at all. Here we have the benefit of region detectors firing both at points where there is signal variation in more than one direction (e.g. near “blobs” or “corners”), as well as at high contrast extended regions. These two image areas are quite complementary. Their union provides a good coverage of the image provided it is at least lightly textured, as can be seen in Fig. 2. The number of regions and coverage depends of course on the visual richness of the image. Typically a total of between 1000 and 2000 regions are obtained per frame.


(a)

(b)

(c)

Figure 2. Example of affine covariant region detection. (a) Frame 20 from the van shot. (b) Ellipses formed from 722 affine covariant interest points. (c) Ellipses formed from 1269 MSER regions. Note the large number of regions detected in a single frame, and also that the two types of region detectors fire at different and complementary image locations.

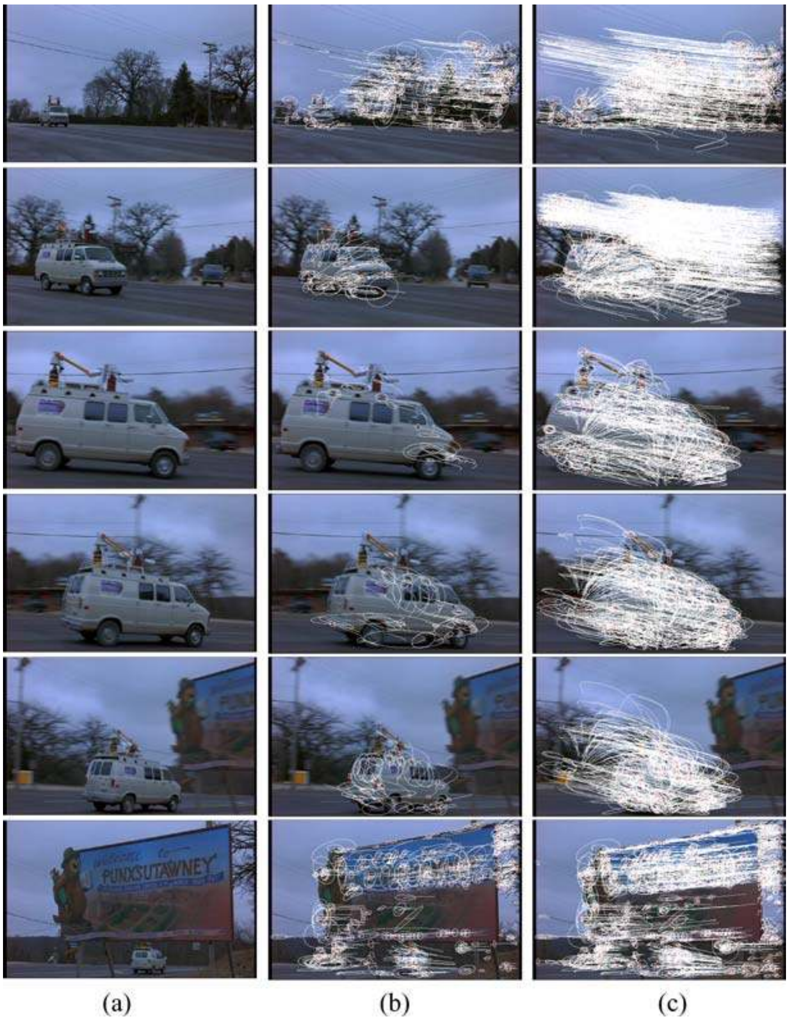


Figure 3. Region tracking: (a) six frames from the van shot. The camera is panning right, and the van moves independently. (b) Frames with the basic region tracks superimposed (before repair). Each frame shows affine covariant regions tracked in that frame. For each tracked region shown, the tracked path of its centroid over the whole life time of the track (i.e. backwards and forwards in time) is shown by its (x, y) position. The path of the region centroid indicates the temporal extent of the track. (c) After short range repair. Note the much longer tracks on the van after applying this repair. For presentation purposes, only tracks lasting for more than 10 frames are shown. Note that the background is not tracked in the middle of the shot due to severe motion blur. A detail of a single region track is shown in Fig. 6.

2.2. *Tracker Implementation*

In a pair of consecutive frames, detected regions in the first frame are putatively matched with all detected regions in the second frame, within a disparity threshold of 50 pixels. Many of these putative matches will be wrong and an intensity correlation computed over the area of the elliptical region removes all putative matches with a normalized cross correlation below 0.90. The 1-parameter (rotation) ambiguity between regions is assumed to be close to zero, because there will be little cyclo-torsion between consecutive frames. All matches that are ambiguous, i.e. those that putatively match several features in the other frame, are eliminated.

Finally epipolar geometry is fitted between the two views using RANSAC (Fischler and Bolles, 1981) with an inlier threshold of 3 pixels. This step is very effective in removing outlying matches whilst not eliminating the independent motions which occur between the two frames.

The results of this tracking on a shot from the movie ‘Groundhog Day’ are shown in Fig. 3b. This shot is used throughout the paper to illustrate the stages of the

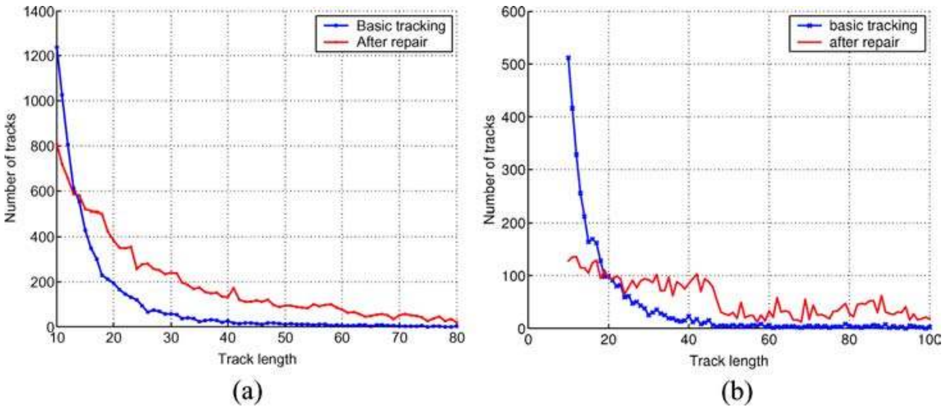


Figure 4. Histograms of track lengths for (a) the face shot, (b) the van shot shown of Fig. 1 for the basic tracking (Section 2) before and after short range track repair (Section 3). Note the improvement in track length after repair. In both cases the weight of the histogram shifts to the right after repair. The step at around frame 45 after repair in (b) is due to the rich background of trees which lasts for about 45 frames at the beginning of the shot.

object-level grouping. Note that the tracks have very few outliers.

It is worth remarking on how this approach compares to the more conventional method of tracking interest points alone. There are two clear advantages in the region case: first, the appearance is a strong disambiguation constraint, and consequently far fewer outliers are generated at every stage; second, far more of the image can be tracked using (two types of) regions than just the area surrounding an interest point. The disadvantage is the computational cost, but this is not such an issue in the retrieval situation where most processing can be done off-line.

3. Short Range Track Repair

The simple region tracker of the previous section can fail for a number of reasons most of which are common to all such feature trackers: (i) no region (feature) is detected in a frame — the region falls below some threshold of detection (e.g. due to motion blur); (ii) a region is detected but not matched due to a slightly different shape; and (iii) partial or total occlusion.

The causes (i) and (ii) can be overcome by short range track repair using motion and appearance, and we discuss this now. Cause (iii) can be overcome by wide baseline matching on motion grouped objects within one shot, and discussion of this is postponed until Section 6.

3.1. Track Repair by Region Propagation

The goal of the track repair is to improve tracking performance in cases where region detection or the first stage tracking fails. The method will be explained for the case of a one frame extension, the other short range cases (2–5 frames) are analogous.

The repair algorithm works on pairs of neighbouring frames and attempts to extend already existing tracks that terminate in the current frame. Each region which has been successfully tracked for more than n (=3) frames and for which the track terminates in the current frame is propagated to the next frame. The propagating transformation is estimated from a set of k (=5) spatially neighbouring tracks. In the case of successive frames only translational motion is estimated from the neighbouring tracks. In more detail, the t_x and t_y components of the translation are estimated as median values of the k translations t_{xi} and t_{yi} suggested by the k spatially nearest tracks i continuing to the next frame. Figure 5 shows an example. In the case of more separated frames the full affine transformation imposed by each tracked region should be employed.

The refinement algorithm of Ferrari et al. (2003) is used to fit the propagated region locally in the new frame (this searches a hypercube in the 6D space of affine transformations by a sequence of line searches along each dimension). If the refined region correlates sufficiently with the original region in the previous frame the region track should continue to the new frame. It is here that the advantage of regions over interest points is manifest: this verification test takes account of local deformations due to viewpoint change, and is very reliable.

The standard ‘book-keeping’ cases then follow: (i) no new region is instantiated (e.g. the region may be occluded in the frame); (ii) a new region is instantiated,

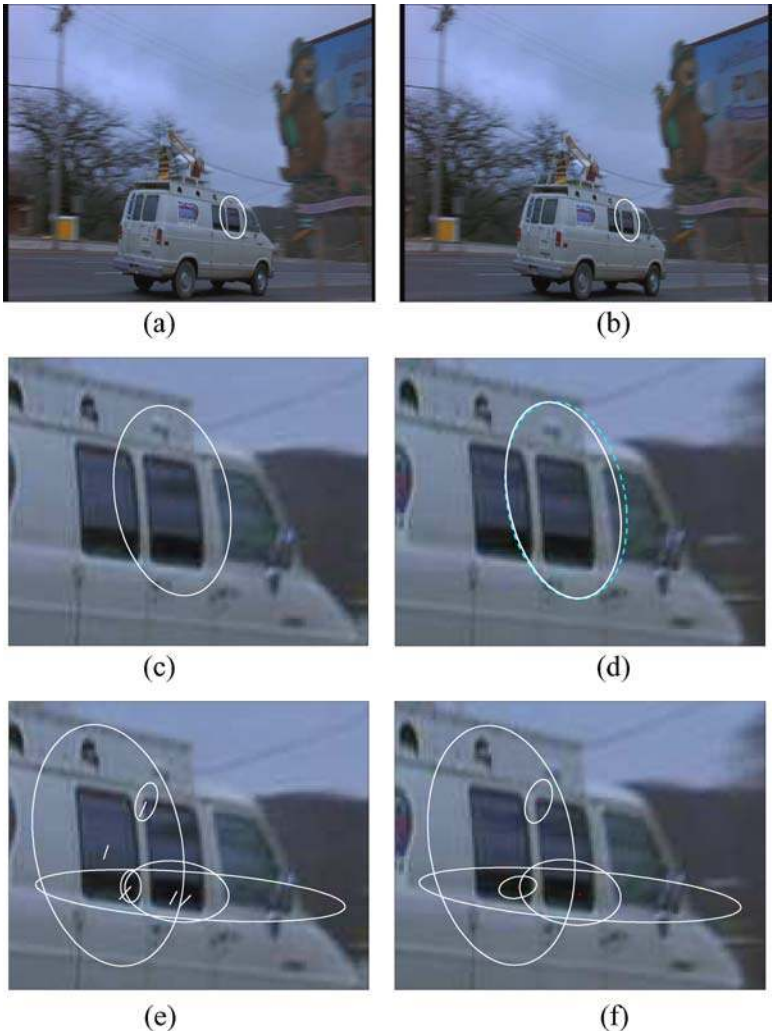


Figure 5. Illustration of the track repair by region propagation. A region track finishing in frame (a) is extended to the following frame (b). The region from the first frame (close-up shown in (c)) is first transformed to the next frame (dashed ellipse in (d)) and then aligned to the image intensities (solid ellipse in (d)). The initial propagation transformation (translation in this case) is estimated from the five (spatially) nearest already existing basic stage tracks. These are shown in (e) and (f). The lines in (e) show the centroid motion of each of the five tracked regions. See text for more details.

in which case the current track is extended; (iii) if the new instantiated region matches (correlates with) an existing region in its (5 pixel) neighbourhood then this existing region is added to the track; (iv) if the matched region already belongs to a track starting in the new frame, then the two tracks are joined.

Figure 4 gives the 'before and after' histogram of track lengths for the two example shots of Fig. 1. The results of this repair are shown in Figs. 3 and 8. Detail of a single region track after the repair stage is shown in Fig. 6.

As can be seen, there is a dramatic improvement in the length of the tracks — as was the objective here. The success of this method is due to the availability and use of two complementary constraints — motion and appearance.

Note also that the region propagation can develop tracks on deforming objects where the between-frame region deformation can be modelled by an affine geometric transformation. Figure 9 shows an example of such a track on the mouth of a speaking person.

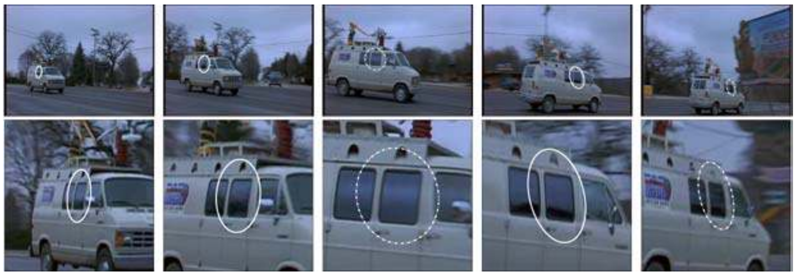


Figure 6. Detail of a single region track after repair. Note the significant change in viewpoint. Top row: Five frames from the shot of Figure 3 with the tracked affine covariant region superimposed. Bottom row: Corresponding close-ups of the tracked region. The solid line denotes regions detected by the affine covariant region detector. The dashdot line denotes regions filled-in by propagation. This particular track extends over 52 frames and consists of 4 basic tracks (a total of 18 detected regions) connected by 34 filled-in regions (including 4 regions which were detected but not associated with any basic track).

4. Application I: Using Multiple Exemplars for Retrieval

The goal here is for a user to be able to specify an object of interest in a single frame, by defining a query region delineating the object, and this to be sufficient input to retrieve all shots containing instances of that object throughout the movie, even though the object may deform or be imaged from a different visual aspect than that of the query frame (see Section 7).

To achieve this, tracks of affine covariant regions throughout the shot are used to automatically associate multiple image exemplars of the object — query regions in other frames — and use the associated exemplars to enhance the original user specified query. The idea is illustrated in Fig. 7.

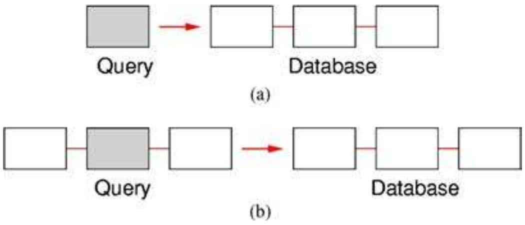


Figure 7. Conceptually, we extend the standard paradigm of image based retrieval (a), where the query is defined by a region within a single image, to retrieval at an object-level (b) where an object is defined over multiple images. A query region in the (shaded) query frame acts as a portal to all the keyframes and search regions within a shot associated by the tracked affine regions.

In detail a query region is 'transported' from the query frame to other frames in the shot as follows: the set of affine covariant regions enclosed by the query region is determined; the tracked regions then determine a corresponding set in each frame of the shot; in turn the rectangular bounding box (or union) of this set determines a query region for that frame. Matching is then carried out for all query regions using the Video Google method (reviewed below).

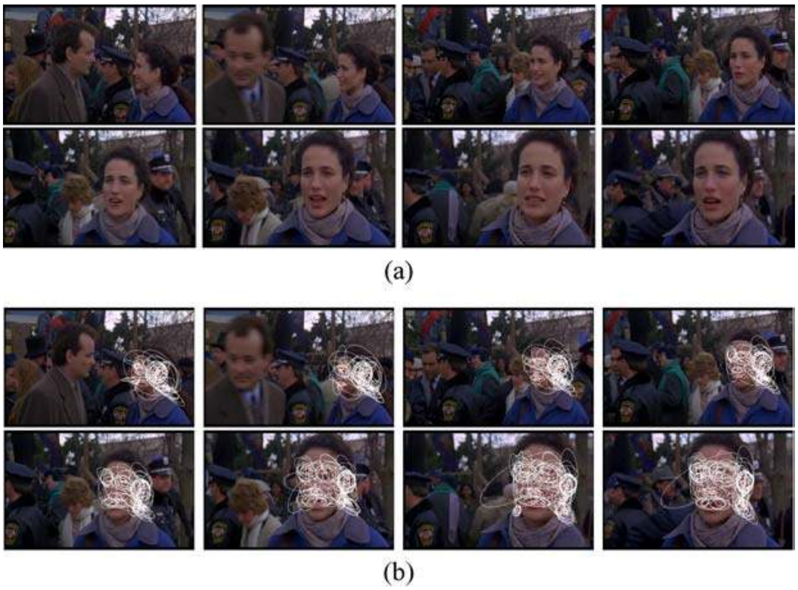
Figure 10 shows an example of an enhanced query. A user outlines a query rectangle in a single frame, as shown in Fig. 10a (top). Tracks on affine covariant regions passing through the user outlined rectangle then define associated query rectangles in other frames. The tracks are shown in Fig. 8. Tracking objects with a limited amount of deformation is possible since the region tracking described in Sections 2 and 3 allows a covariant region to undergo affine geometric transformation between consecutive frames of the video. A detail of a single region track on a deforming mouth is shown in Fig. 9.

The deforming and rotating object (actor's head talking and turning) is represented automatically by multiple exemplars (instances over multiple frames within one shot). The following sub-sections give implementation details.

4.1. *Retrieval on a Single Image Query — Video Google*

This is a brief overview of the Video Google shot retrieval method described in Sivic and Zisserman (2003). The goal is to efficiently and accurately match the object specified by a query region throughout a video. The object is represented by the set of affine covariant regions within the query region (their appearance and position).

In order to match affine covariant regions efficiently each region is first represented as a 128-dimensional vector using the SIFT descriptor developed by Lowe (1999). The SIFT descriptors are then


(a)

(b)

Figure 8. Tracking deforming objects. (a) Eight frames (of 133) for the head turning shot. (b) Tracked viewpoint covariant regions on the actor's head. The tracks are selected in one frame by a user query (see text). Only tracks longer than 10 frames are shown here. A detail of the mouth track is given in Fig. 9.

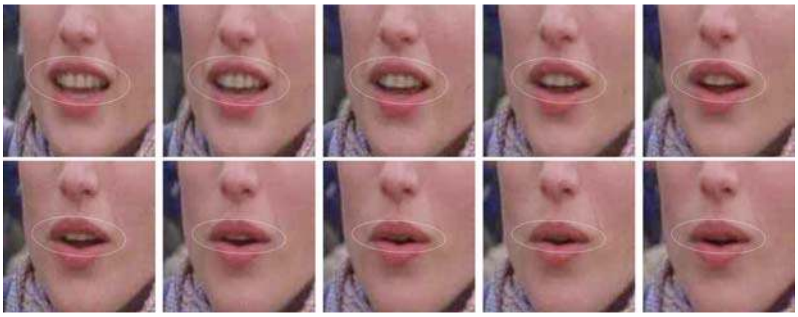


Figure 9. Detail of a region track in 10 consecutive frames covering the deforming mouth whilst the actor speaks. This track extends over 28 frames.

vector quantized using K-means clustering. The clusters are computed from 474 frames of the video, with 6,000 clusters for regions based on interest point neighbourhoods (Schaffalitzky and Zisserman, 2002; Mikolajczyk and Schmid, 2002), and 10,000 clusters for Maximally Stable Extremal Regions (Matas et al., 2002). All the descriptors for each frame of the video are assigned to the cluster centre nearest 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.

The retrieval proceeds in two stages, first keyframes (every 25th frame) are ranked based on the histograms of occurrences of the quantized descriptors, and the top ranked set selected. This set is then re-ranked by a local spatial consistency check which requires that spatially close regions in the query frame map to spatially close regions in the retrieved frame. This spatial consistency requires that a putative affine region match has a supporting match within its nearest spatial neighbours (Schmid, 1997; Sivic and Zisserman, 2003). The number of supporting matches defines the similarity score between two frames. This is quite a loose spatial constraint, and allows object deformation between frames.

4.2. *Collating Search Results from Multiple Queries*

The goal here is to collate search results from multiple associated query frames representing the object level

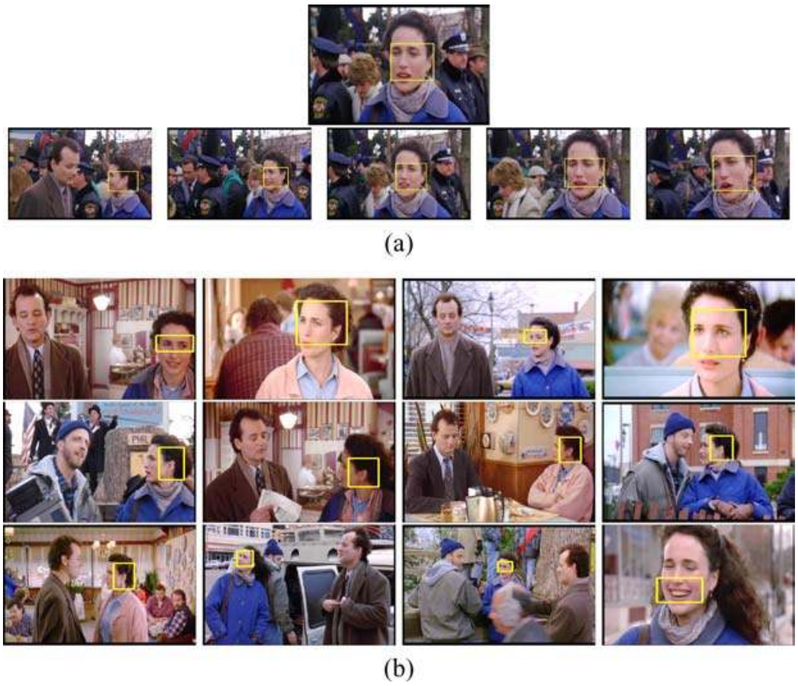


Figure 10. Retrieving a deformable object using multiple exemplars. (a) The user outlined query region (top) in a single frame, and (bottom) 5 (out of 19) automatically associated keyframes and query regions from within the same shot. The associated query regions are obtained as rectangular bounding boxes of the tracks (shown in Fig. 8) passing through the user outlined rectangle in the query frame. Note that full profile views, three quarter views and frontal views with different expressions are associated with the original query frame. (b) The top row shows example of retrieved frames from different shots by searching on only the user outlined query region. The bottom two rows show example retrieved frames by searching on the associated query regions as well. Note that the extended query enables the retrieval of full profile views which would be almost impossible by the original user outlined query. In the first twenty retrieved shots there are five mismatches for other faces and one mismatch for a non-face.

query in order to return a ranked list of shots. In more detail we want to compute a retrieval score Φ_l for shot l, given a set of query frames $S_q = \{q_i\}$ (with query regions), the set of keyframes $S_l = \{k_j\}$ belonging to shot l and keyframe scoring function $\phi(q, k)$ returning similarity score between the query region of the query frame q and keyframe k (as explained in Section 4.1).

Two strategies are used for collating results from multi image queries: (i) votes for a particular shot are accumulated across all the associated query frames and retrieved keyframes, i.e.

$$\Phi_l = \sum_{i=1}^{|S_q|} \sum_{j=1}^{|S_l|} \phi(q_i, k_j), \tag{1}$$

or (ii) the best matching keyframe from each shot is used to score the whole shot

$$\Phi_l = \max_{i,j} \phi(q_i, k_j). \tag{2}$$

The advantage of the first method (Eq. (1)) is that a shot can accumulate votes from multiple query frames, whereas false positives tend not to be consistent. For example, if both the query and retrieved shots have profile and frontal view of a face, then the face shot can accumulate votes from both the profile and frontal query frames whereas the false positives would not be the same for the frontal and profile views and would therefore receive lower score. The advantage of the second strategy (Eq. (2)) is that it does not overcount scores for longer shots. In the matching examples of Figs. 10, 21 and 24 the first strategy was used. In the matching example of Fig. 25 the second strategy was used.

5. Object Extraction by Robust Sub-Space Estimation

The previous section used tracked affine covariant regions to associate multiple exemplars of an object.

However, the method is limited in that it can't 'see around corners.' For example, if we select the three-quarter view of the van in Fig. 3(a) (second row), only the side and front of the van will be associated, not the back of the van, because only tracks originating in the original three-quarter view are used. In this section we take the grouping a stage further and partition the tracks into groups with coherent motion. In other words, things that move together are assumed to belong together. For example, in the shot of Fig. 3 the ideal outcome would be the van as one object — grouping the front, side and back even though these are not visible simultaneously in any single frame. We would also expect to obtain several groupings of the background.

The grouping constraint used here is that of common rigid motion, and we assume an affine camera model so the structure from motion problem reduces to linear subspace estimation. For a 3-dimensional object, our objective would be to determine a 3D basis of trajectories $\mathbf{b}_k^i$, $k = 1, 2, 3$, (to span a rank 3 subspace) so that (after subtracting the centroid) all the trajectories x_j^i associated with the object could be written as (Zelnik-Manor and Irani, 1999):

$$\mathbf{x}_j^i = \left(\mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\right)(X_j, Y_j, Z_j)^\top$$

where $\mathbf{x}_j^i$ is the measured (x, y) position of the jth point in frame i, and (X_j, Y_j, Z_j) is the 3D affine structure.

The maximum likelihood estimate of the basis vectors and affine structure could then be obtained by minimizing the reprojection error

$$\sum_{ij} \left\| n_j^i \left(\mathbf{x}_j^i - \left(\mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\right)(X_j, Y_j, Z_j)^\top\right) \right\|^2 \qquad (3)$$

where n_j^i is an indicator variable to label whether the point j is (correctly) detected in frame i, and must also be estimated. This indicator variable is necessary to handle missing data.

It is well known (Torr et al., 1998) that directly fitting a rank 3 subspace to trajectories is often unsuccessful and suffers from over-fitting. For example, in a video shot the inter-frame motion is quite slow so using motion alone it is easy to under-segment and group foreground objects with the background.

We build in immunity to this problem from the start, and fit subspaces in two stages: first, a low dimensional model (a projective homography) is used to hypothesize groups — this over-segments the tracks. These groups are then associated throughout the shot using track co-occurrences. The outcome is that trajectories are grouped into sets belonging to a single object. In the second stage 3D subspaces are sampled from these sets, without over-fitting, and used to merge the sets arising from each object. These steps are described in the following sub-sections. The complete algorithm is summarized in Fig. 19. This approach differs fundamentally from that of Aanaes et al. (2002) and De la Torre and Black (2003) where robustness is achieved by iteratively re-weighting outliers but no account is taken of motion degeneracy.

5.1. *Basic Motion Grouping Using Homographies*

To determine the motion-grouped tracks for a particular frame, both the previous and subsequent frames are considered. The aim is then to partition all tracks extending over the three frames into sets with a common motion. To achieve this, homographies are fitted to each pair of frames of the triplet using RANSAC. In each RANSAC iteration, a four-tuple of tracks extending over the three frames is sampled and three homographies (H_{12}, H_{13}, H_{23}) are computed. The set of inlying tracks is computed based on image reprojection error averaged over the three frames. The inlier threshold is set to a generous number of pixels (around 3 here). The inlying set is removed, and RANSAC is then applied to the remaining tracks to extract the next largest motion grouping, etc. This procedure is applied to all triplets of consecutive frames in the shot, i.e. the neighbouring triplets share two frames. In the next step motion groups are linked throughout the shot into an object.

5.2. *Aggregating Segmentation over Multiple Frames*

The problem with fitting motion models to pairs or triplets of frames are twofold: (i) a phantom motion cluster corresponding to a combination of two independent motions grouped together can arise (Torr, 1995), and (ii) an outlying track will be occasionally, but not consistently, erroneously grouped together with one of the motion groups. In our experience these ambiguities tend not to be stable over many frames, but rather occasionally appear and disappear. To deal with these problems we devise a voting strategy which groups tracks that are consistently segmented together over multiple frames.

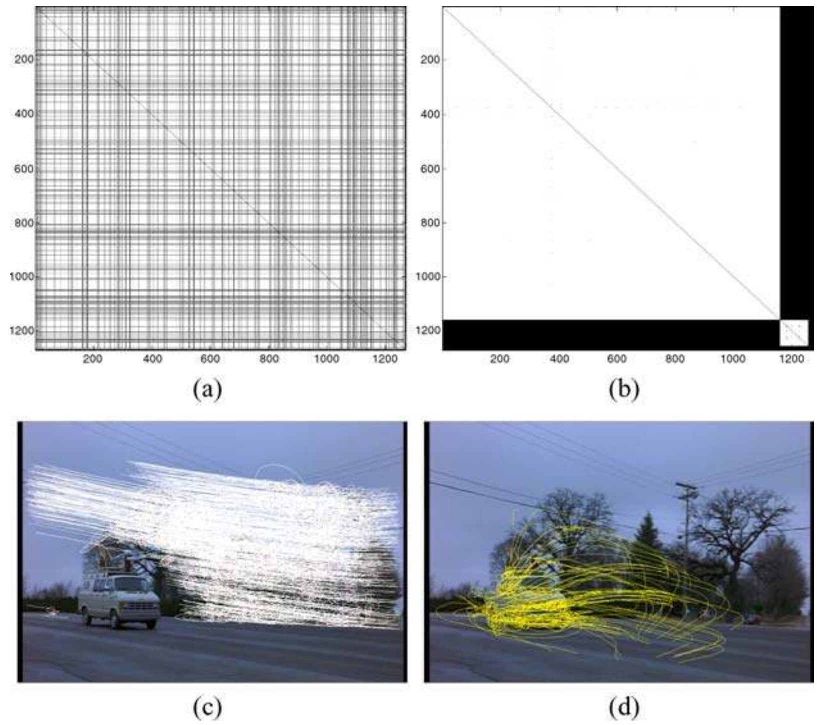


Figure 11. Aggregating segmentation over multiple frames. (a) The track co-occurrence matrix for a ten frame block of the shot from Fig. 3. White indicates high co-occurrence. (b) The thresholded co-occurrence matrix re-ordered according to its connected components (see text). (c) (d) The sets of tracks corresponding to the two largest components (of size 1157 and 97). The other components correspond to 16 outliers.

The basic motion grouping of Section 5.1 provides a track segmentation for each triplet of consecutive frames. The goal is to pull out sets of tracks which are consistently grouped together over a wider baseline. This is achieved by a simple clustering algorithm which operates on a track-to-track similarity matrix, where the track-to-track similarity is based on temporal consistency between the two tracks, i.e. the number of frames over which the two tracks co-occur together in one motion segment (which is given by the basic homography based motion grouping).

In more detail the shot is divided into blocks of frames over a wider baseline of n frames ($n = 10$ for example) and a track-to-track co-occurrence matrix W is computed for each block. The element w_{ij} of the matrix W accumulates a vote for each frame where tracks i and j are grouped together. Votes are added for all frames in the block. In other words, the similarity score between two tracks is the number of frames (within the 10-frame block) in which the two tracks were grouped together. The task is now to segment the track voting matrix W into temporally coherent clusters of tracks. This is achieved by finding connected components of a graph corresponding to the thresholded matrix W. To prevent under-segmentation the threshold is set to a value larger than half of the frame baseline of the block, i.e. 6 for the 10 frame block size. This guarantees that each track cannot be assigned to more than one group. Only components exceeding a certain minimal number of tracks are retained. Figure 11 shows an example of the voting scheme applied on a ten frame block from the shot of Figure 3. This simple scheme segments the matrix W reliably and overcomes the phantoms and outliers.

The motion clusters extracted in the neighbouring 10 frame blocks are then associated based on the common tracks between the blocks. This is achieved by gradually progressing through frame blocks in the shot starting from the first block and associating motion clusters which are connected by a significant number of tracks. Significance is measured relative to the number of tracks in both the motion clusters, i.e. two motion clusters in the neighbouring blocks have to share at least 50% tracks to be associated. The result is a set of connected clusters of tracks which correspond to independently moving objects throughout the shot.

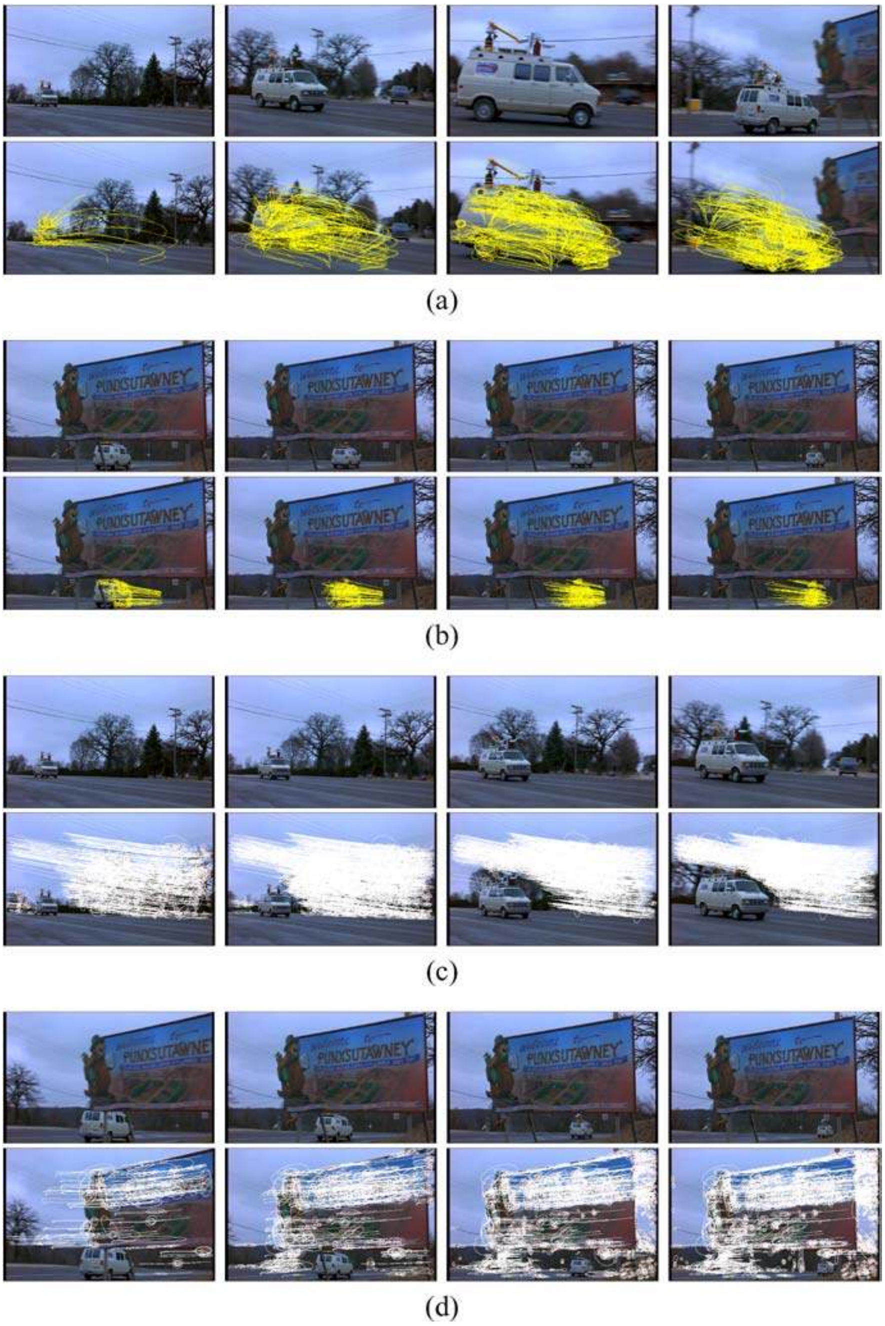


Figure 12. Motion grouping example I: Four dominant objects extracted from the van shot of Fig. 3. (a), (b) The first two objects corresponds to the van before (a) and after (b) the occlusion by the post. Note the billboard post right behind the van in the top left image of (b). This post partially occludes the van in 21 frames. (c), (d) The other two objects correspond to the background at the beginning (a) and the end (b) of the shot. The background in the middle of the shot was not tracked due to severe motion blur.

5.3. *Object Extraction*

The previous track clustering step usually results in no more than 10 dominant (measured by the number of tracks) motion clusters larger than 20 tracks. The goal now is to identify those clusters that belong to the same moving 3D object. This is achieved by grouping pairs of track-clusters over a wider baseline of m frames ($m >$

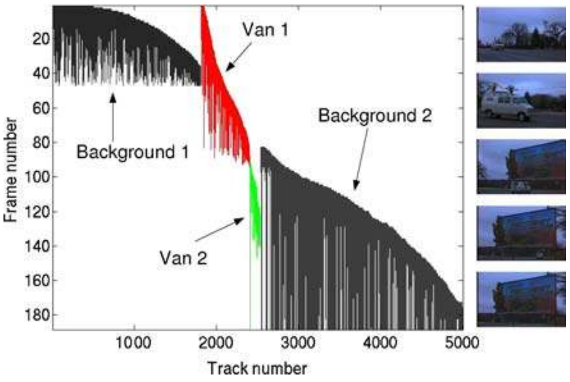


Figure 13. The sparsity pattern of the tracked features (after the short range track repair) in the van shot of Fig. 5. The tracks are sorted according to the frame they start in and coloured according to the independently moving objects, that they belong to, as described in Section 5. The two gray blocks (track numbers 1-1808 and 2546-5011) correspond to the two background objects. The red and green blocks (1809-2415 and 2416-2545 respectively) correspond to the van object before and after the occlusion.

10 here). To test whether to group two clusters, tracks from both sets are pooled together and a RANSAC algorithm is applied to all tracks intersecting the m frames. The algorithm robustly fits a rank 3 subspace as described in Eq. (3).

In each RANSAC iteration, four tracks are selected and full affine factorization is applied to estimate the three basis trajectories which span the three dimensional subspace of the ($2m$ dimensional) trajectory space. All other tracks that are visible in at least five views are projected onto the space. A threshold (1.5 pixels) is set on reprojection error to determine the number of inliers. To prevent the grouping of inconsistent clusters a high number of inliers (90%) from both sets of tracks is required. When no more clusters can be paired, all remaining clusters are considered as separate objects.

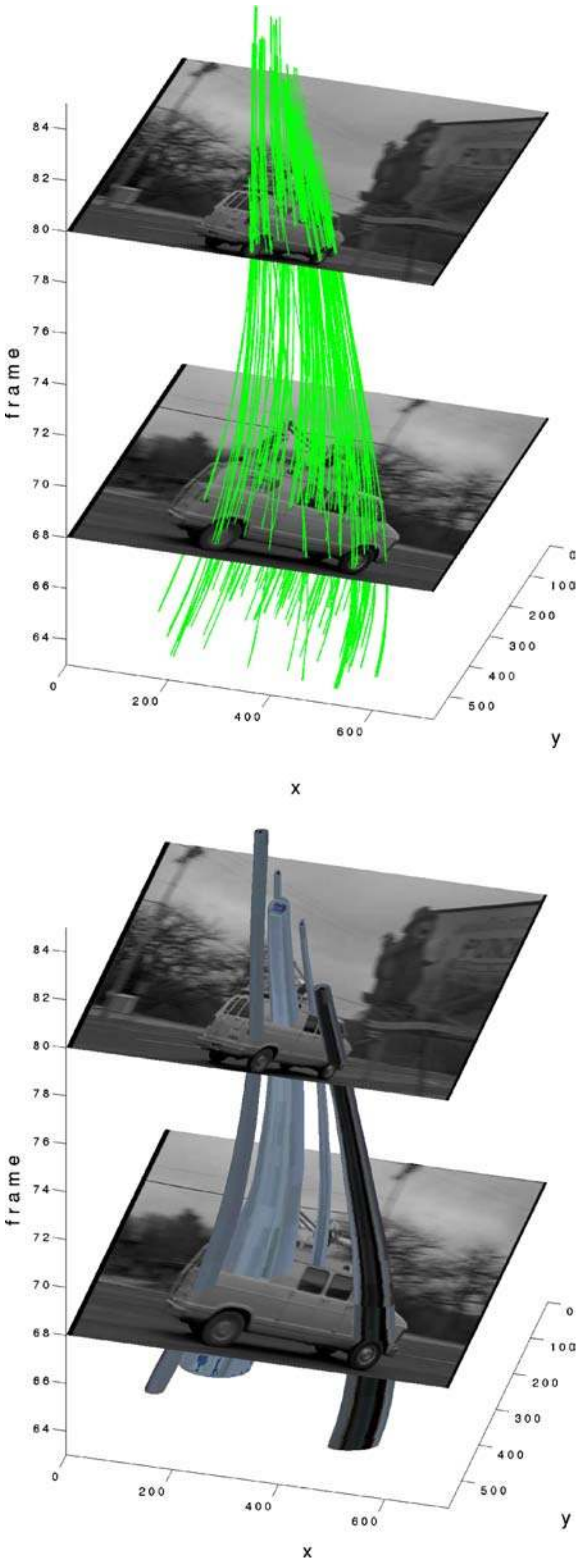


Figure 14. Trajectories following object-level grouping. Top: A selection of 110 region tracks (out of a total of 429 between these frames) shown by their centroid motion. Bottom: Five region tracks shown as spatio-temporal "tubes" in the video volume. The frames shown are 68 and 80. Both figures clearly show the foreshortening as the car recedes into the distance towards the end of the shot. The number and quality of the tracks is evident: the tubes are approaching a dense epipolar image (Bolles et al., 1987), but with explicit correspondence; the centroid motion demonstrates that outlier 'strands' have been entirely 'combed' out, to give a well conditioned track set.

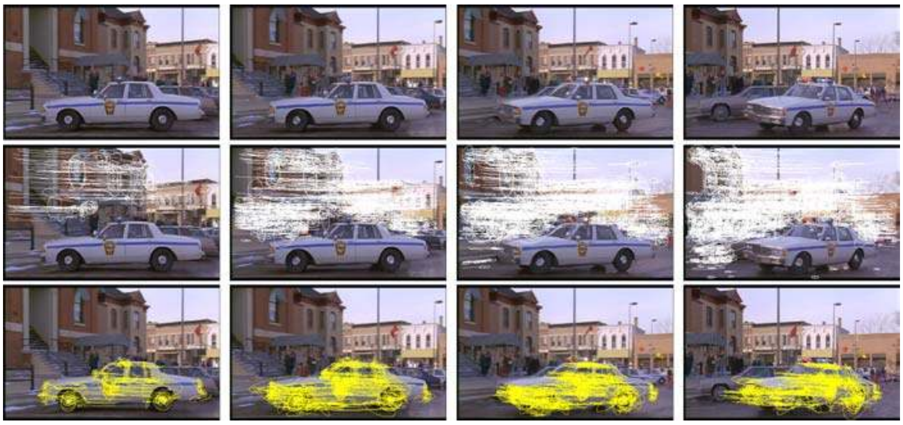


Figure 15. Motion grouping example II: object-level grouping for a 35 frame shot. Top row: The original frames of the shot. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 721 (car) and 2485 (background).

The rigidity based grouping is currently applied only to pairs of track-clusters. Complex objects made of more than two track-clusters could be handled by iterative merging pairs of clusters into larger groups.

5.4. *Object Extraction Results*

Figure 12 shows the four grouped objects for this example shot. Two of the objects correspond to the van (before and after the occlusion by the post, see Fig. 20 in Section 6) and two correspond to the backgrounds at the beginning and end of the shot. The number of tracks associated with each object are 607 (van pre-occlusion), 130 (van post-occlusion), 1808 (background start) and 2466 (background end). The sparsity pattern of the tracks belonging to different objects is shown in Fig. 13. Each of the background objects is composed of only one motion cluster. The van (pre-occlusion) object is composed of two motion clusters of size 580 and 27 which are joined at the object extraction RANSAC stage. The quality and coverage of the resulting tracks is visualized in the spatio-temporal domain in Fig. 14.

Two additional examples of rigid object extraction from different shots are given in Figs. 15 and 16. Figures 17 and 18 show examples of slowly deforming objects. This deformation is allowed because at the first homography based stage rigidity is only applied over a short baseline of three frames.

Computation time: To give some idea of how long the object-level grouping takes we have recorded computation times for the example van shot of Fig. 3. This shot has a total of 187 frames. The region detection and descriptor computation took on average 11 seconds per frame. The basic tracking took 16 minutes (∼5 seconds per frame). The track repair by region propagation took 304 minutes (∼97 seconds per frame). The track repair is currently implemented in Matlab and is the bottleneck of the algorithm. The motion grouping algorithm took 56 minutes of which stage 4a took 12 mins, 4b 23 mins and 4c 21 mins. The different stages refer to the algorithm summary in Fig. 19. The motion grouping algorithm is also entirely implemented in Matlab. All timings are on a 2 GHz machine.

6. Long Range Track Repair

The object extraction method described in the previous section groups objects that are temporally coherent. The aim now is to connect objects that appear several times throughout a shot, for example an object that disappears for a while due to occlusion. Typically a set of tracks will terminate simultaneously (at the occlusion), and another set will start (after the occlusion). The situation is like joining up a cable (of multiple tracks) that has been cut.

The set of tracks is joined by applying standard wide baseline matching (Matas et al., 2002; Schaffalitzky and Zisserman, 2002; Tuytelaars and Van Gool, 2000) to a pair of frames that each contain the object. There are two stages: first, epsilon-nearest neighbour search on a SIFT descriptor (Lowe, 1999) for each region, is performed to get a set of putative region matches, and second, this set is disambiguated by a local spatial

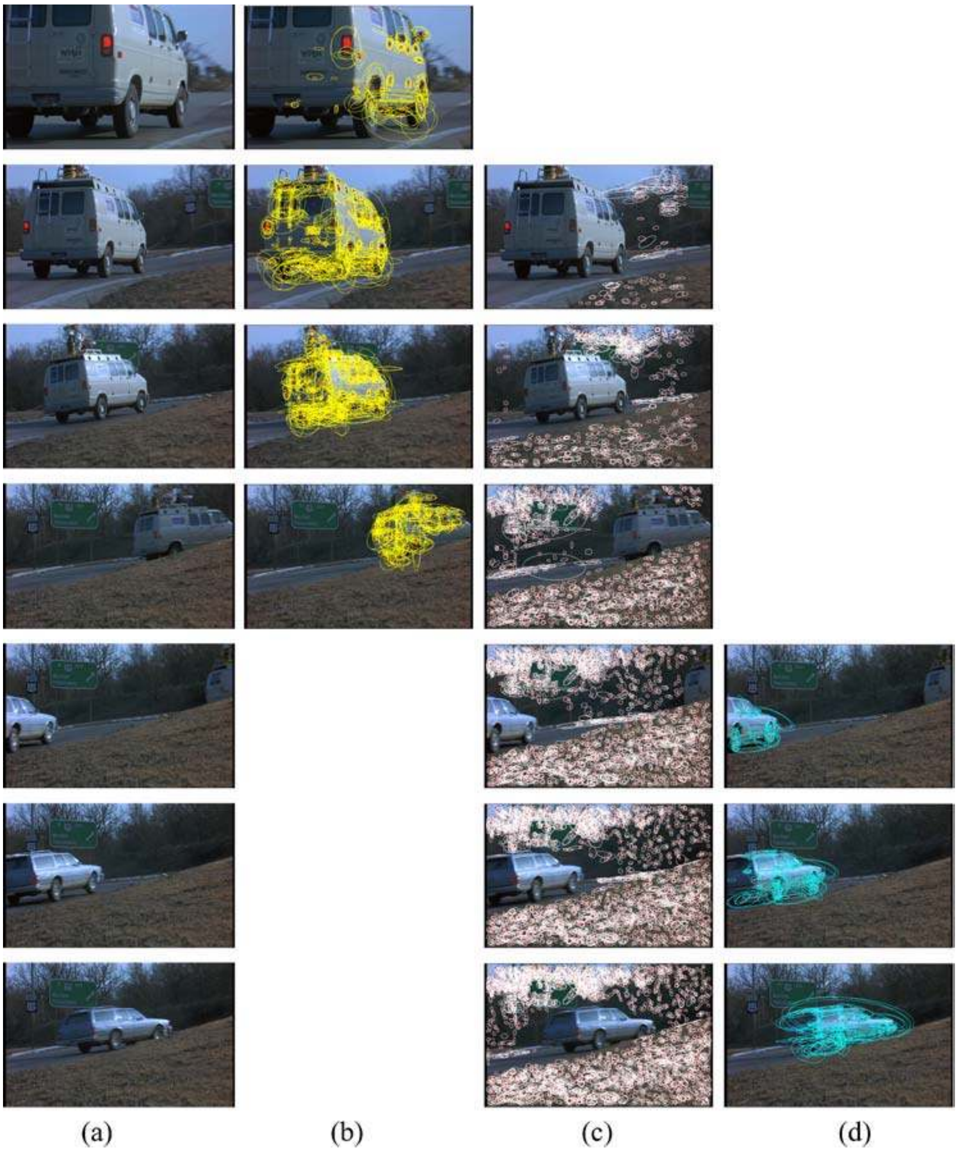


Figure 16. Motion grouping example III: Object-level grouping for a 153 frame shot where the camera is tracking a van followed by another car. (a) Seven frames of the shot. (b)—(d) The three extracted objects correspond to (b) the van (1108 tracks), (c) the background (4481 tracks) and (d) the other car (210 tracks). The trajectory of the regions is not shown here in order to make the clusters visible.

consistency constraint: a putative match is discarded if it does not have a supporting match within its k-nearest spatial neighbours (Schmid, 1997; Sivic and Zisserman, 2003). Since each region considered for matching is part of a track, it is straightforward to extend the matching to join tracks. The two objects are deemed matched if the number of matched tracks exceeds a threshold. Figure 20 shows two examples of long range repair on shots where the object was temporarily occluded.

7. Application II: Object-Level Video Matching

The objective here is to retrieve shots within the film containing the object, even though the object may be imaged from a different visual aspect than in the query image region.

Having computed object-level groupings for shots throughout the film, we are now in a position to retrieve object matches given only one visual aspect of the object as a query region. As in the application en-

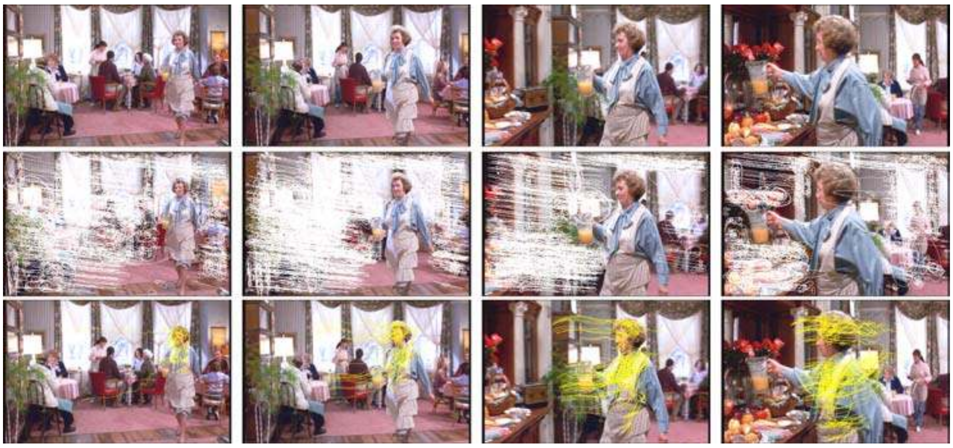


Figure 17. Motion grouping example IV: object-level grouping for a 83 frame shot. Top row: The original frames of the shot. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 225 (landlady) and 2764 (background). The landlady is an example of a slowly deforming object.

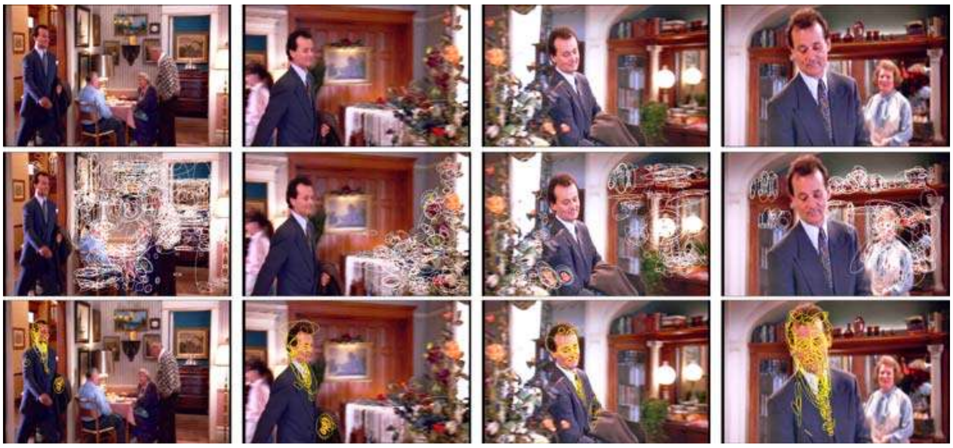
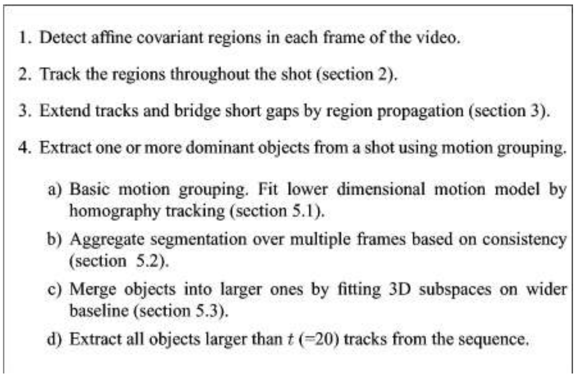


Figure 18. Motion grouping example V: object-level grouping for a 645 frame shot. Top row: The original frames of the shot where a person walks across the room while tracked by the camera. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 401 (the walking person) and 15,053 (background). The object corresponding to the walking person is a join of three objects (of size 114, 146 and 141 tracks) connected by a long range repair using wide baseline matching, see Fig. 20b. The long range repair was necessary because the tracks are broken twice: once due to occlusion by a plant (visible in frames two and three in the first row) and the second time (not shown in the figure) due to the person turning his back on the camera. The trajectory of the regions is not shown here in order to make the clusters visible.

gineered in Section 4, a query region in one frame acts as a portal to a set of associated query regions — but here the association is on common 3D motion as described in Section 5. (In fact since the object has been segmented it is only necessary for the user to 'click' on the object in one frame).

The associated query regions form an implicit representation of the 3D structure, and are sufficient for matching when different visual aspects or parts of the object are seen in different frames of the shot. As shown in Figs. 21 and 24, associated frames naturally span the object's visual aspects contained within the shot.

Examples of object-level matching throughout a database of 5,641 keyframes of the entire movie 'Groundhog Day' is shown in Figs. 21, 24 and 25. In all cases false positives were also retrieved. Retrieval performance for two of the examples is discussed in more detail in the following section.



1. Detect affine covariant regions in each frame of the video.
2. Track the regions throughout the shot (section 2).
3. Extend tracks and bridge short gaps by region propagation (section 3).
4. Extract one or more dominant objects from a shot using motion grouping.
   a) Basic motion grouping. Fit lower dimensional motion model by homography tracking (section 5.1).
   b) Aggregate segmentation over multiple frames based on consistency (section 5.2).
   c) Merge objects into larger ones by fitting 3D subspaces on wider baseline (section 5.3).
   d) Extract all objects larger than t (=20) tracks from the sequence.

Figure 19. Object-level grouping algorithm. Associate independently moving objects within a shot using rigid motion consistency.

7.1. Retrieval Performance

The van query: Ground truth was obtained for the van query in Figure 21 by marking all keyframes and shots where the van appears in the movie. In order to be deemed present in a frame, the van was required to be at least 100 pixels across (in frames that are 720 × 576 pixels).

Precision-recall curves on the shot level for the object-level matching example from Fig. 21 are shown in Fig. 22. In the case of precision-recall curves (a) and (b) where multiple images were used as query frames, each query frame was used to place a separate query and the results from all queries were then pooled together. Retrieved shots were ranked as described in Section 4.2, Eq. (1).

Note that the user outlined query frame (curve (c) in Fig. 22) recalls only 27% of all the ground truth shots containing the van. This is because the query frame contains only the side of the van (see Fig. 21(a) (top)) and therefore it is possible to retrieve only shots where the side of the van is visible. When the object is represented by a set of keyframes naturally spanning its visual aspects (curve (b) in Fig. 22) the recall jumps to 73%. This is because shots containing the front and back of the van are also retrieved. Representing the object by *all* the frames in the shot (curve (a) in Fig. 22) brings the recall to 97%. The slight improvement in precision of curve (a) is mainly due to score accumulation as described in Section 4.2.

False positives responsible for lower precision at higher recall levels (e.g. 35% precision for 60% recall in Fig. 22(a)) are mainly due to (i) the spatial consistency check failing e.g. on the sparse textured area on the side of the van (where there is a large spatial separation between the individual features) (ii) motion blur, which affects the affine covariant region matching, and (iii) generally low number of good matches on the van.

The precision could be improved further by the removal of false positives based on a more thorough (and more expensive) verification, e.g. by the image exploration algorithm of Ferrari et al. (2004b).

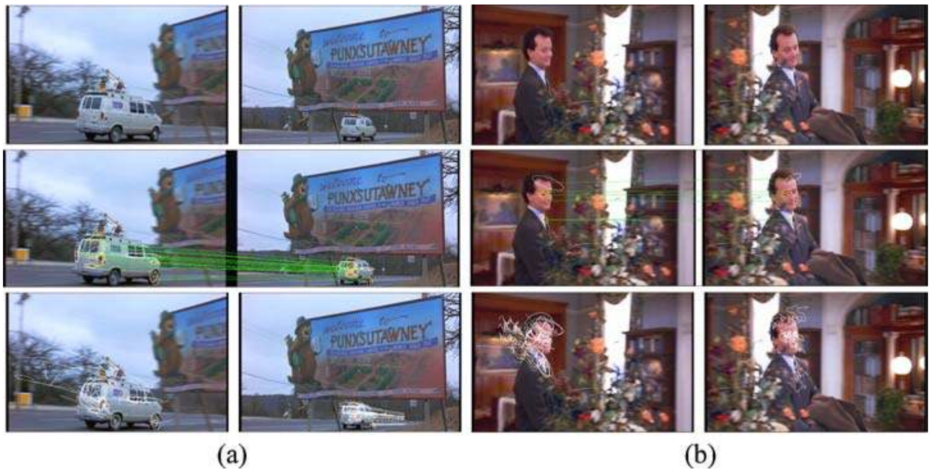


Figure 20. Two examples of long range repair on (a) shot from Fig. 3 where a van is occluded (by a post) which causes the tracking and motion segmentation to fail, and (b) shot from Fig. 18 where a person walks behind a plant. First row: sample frames from the two sequences. Second row: wide-baseline matches on regions of the two frames. The green lines show links between the matched regions. Third row: region tracks on the two objects that have been matched in the shot.

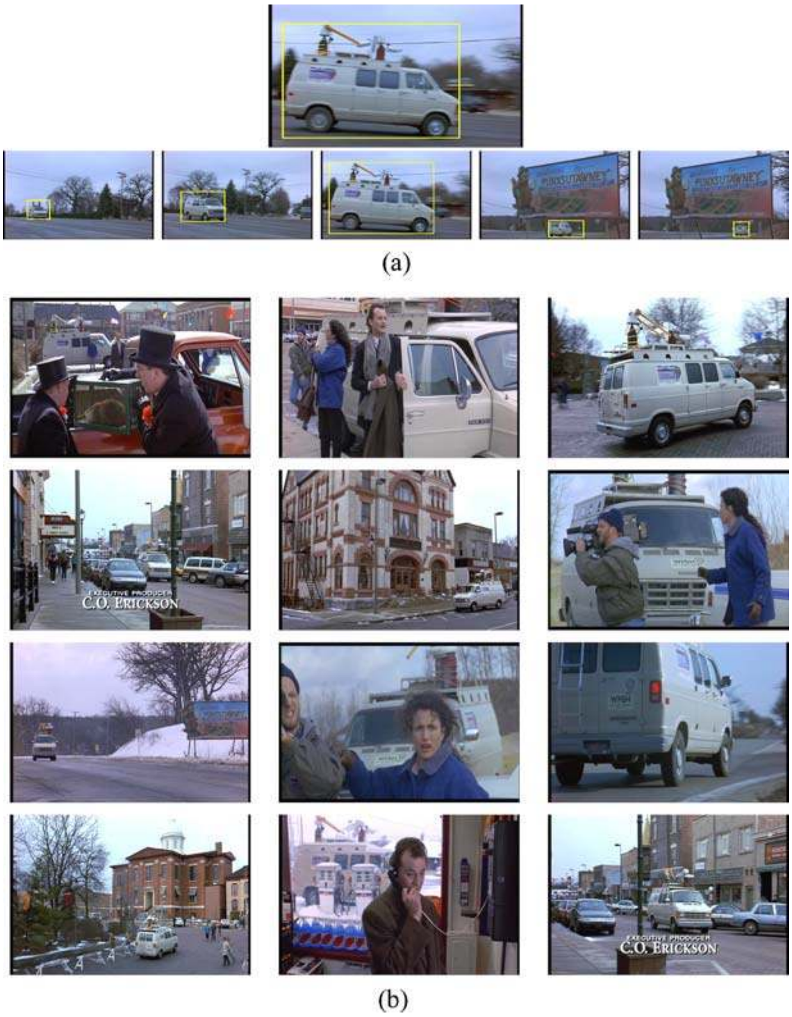


Figure 21. Object-level video matching I. (a) Top row: the query frame with the query region (side of the van) selected by the user. (a) Second row: 5 (out of 6) associated frames and outlined query regions. The query frame acts as a portal to the frames (and query regions) associated with the object by the motion-based grouping. (b) Top row: example frames retrieved from the entire movie when only the original user selected frame with user outlined region is used. (b) Rows 2–4: Example frames retrieved from the entire movie by the object-level query (second row of (a)). Note that views of the van from the back and front are retrieved. This is not possible with wide-baseline matching methods alone using only the side of the van visible in the query image. In this figure, only true positives are shown. Precision recall curves for this query are shown in Fig. 22.

Examples of frames from bottom ranked and missed shots are shown in Fig. 23. They represent very challenging examples for the current object matching method.

The Dining room query : Here the match is on the background location, rather than on the foreground moving object. Ground truth for the query of Fig. 25 was obtained by marking all shots in the movie which are taken in the hotel dining room. The precision-recall curve is shown in Fig. 26. The improved recall of (a) and (b) over (c) is due to the object-level query retrieving shots from the same location but with different background than the original query frame. The improved performance of (a) over (b) is due to better sampling of background in the beginning of the shot with large camera motion where keyframes (every 25th frame) miss some parts of the background. A better keyframe selection technique (Osian and Van Gool, 2004) based on motion within the shot could be used here.

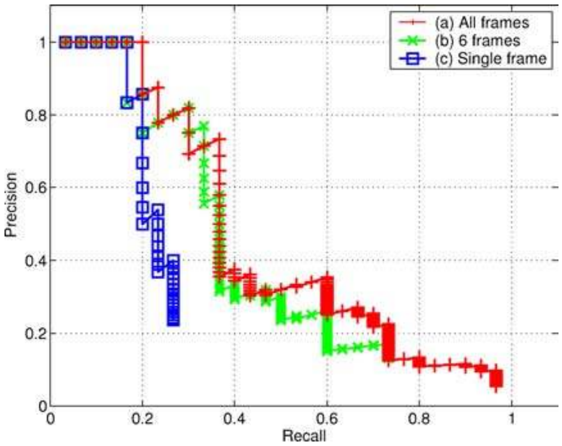


Figure 22. Object-level video matching I. Precision recall curve for the van query at the shot level. Examples of retrieved frames are shown in Fig. 21. (a) All frames in the query shot are used as query frames. (b) 6 frames in the shot are used as query frames. (c) A single frame (the original frame with user outlined region) is used as a query frame. Note the limited recall of (c). This is because only shots where the side of the van is visible are retrieved.

Note that some shots from the dining room are still not retrieved. This is because in the missed shots the camera looks at the other side of the room which is not covered in the query shot. To retrieve these shots a higher level reasoning might be required e.g. the temporal editing structure of shots can be used to group shots into scenes (Goedeme et al., 2005; Kender and Yeo, 1998). An alternative method of matching only background locations using wide baseline matching is given in Schaffalitzky and Zisserman (2003). In our work the user has a choice of whether to search on foreground or background object(s).

8. Discussion and Extensions

We have demonstrated that information available in video shots can be harnessed to enable object-level grouping and retrieval. This is different in spirit to query enhancement techniques in text retrieval (Baeza-Yates and Ribeiro-Neto, 1999), where the high ranked documents are used to enhance the original query. In our case we do not use the retrieved shots or frames to enhance the query but rather we make use of the temporal continuity of the shot. The enhanced query is then performed by making a sequence of associated queries and collating the results.

There are several other research issues: First, in the matching stage of the current method we plan to represent the shot by entire region tracks ('video tubes') rather than the set of separate query frames/keyframes currently used. Using entire 'video tubes' could help to determine the required density of association: Imagine a close-up shot of a speaking person. Deforming region

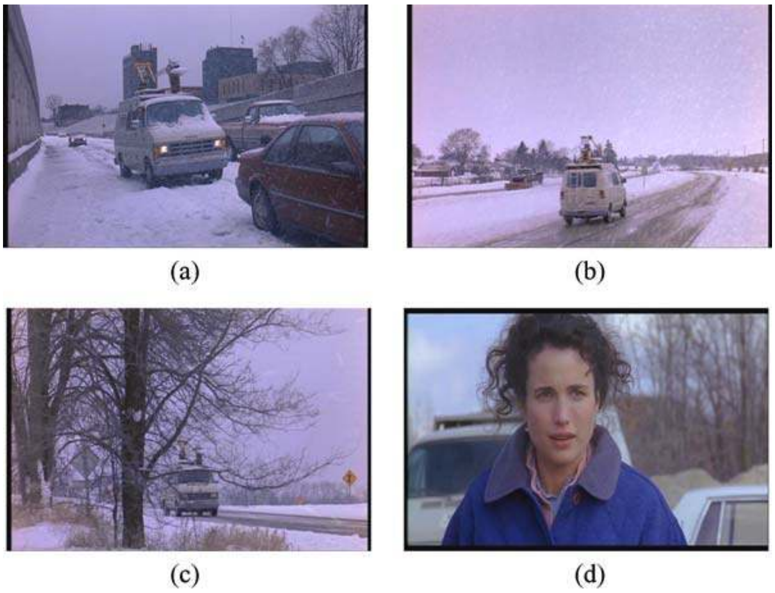


Figure 23. Example frames from low ranked (a–c) and missed (d) shots for the van query (Fig. 21). The altered appearance due to snow in (a, b), and partial occlusion (c, d) affects the affine covariant region extraction and matching methods.

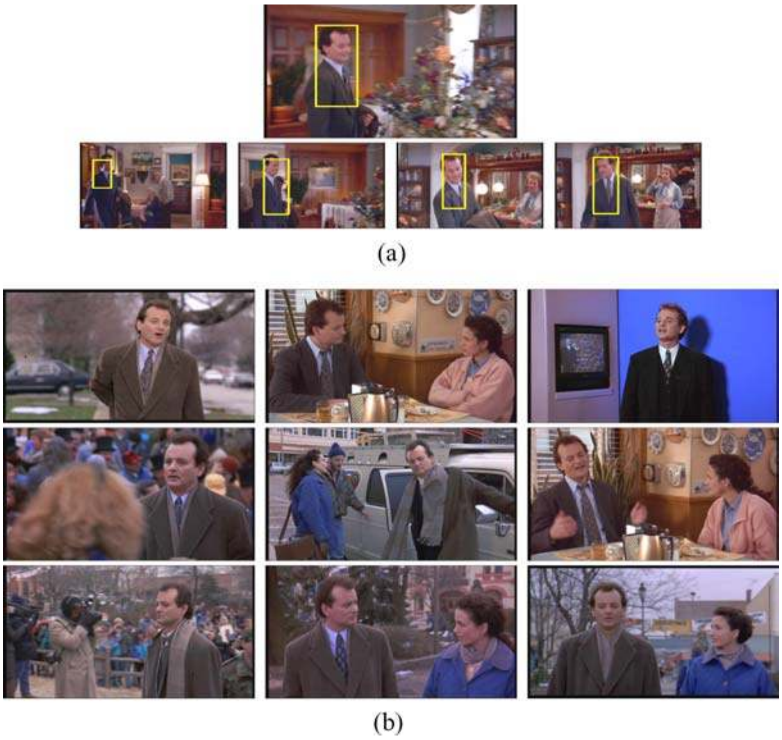


Figure 24. Object-level video matching II. (a) Top row: the query frame with query region selected by the user. (a) Bottom row: The associated keyframes. Note that in the associated keyframes the person is visible from the front and also changes scale. See Fig. 18 for the corresponding object segmentation. (b) Example frames retrieved from the entire movie by the object-level query.

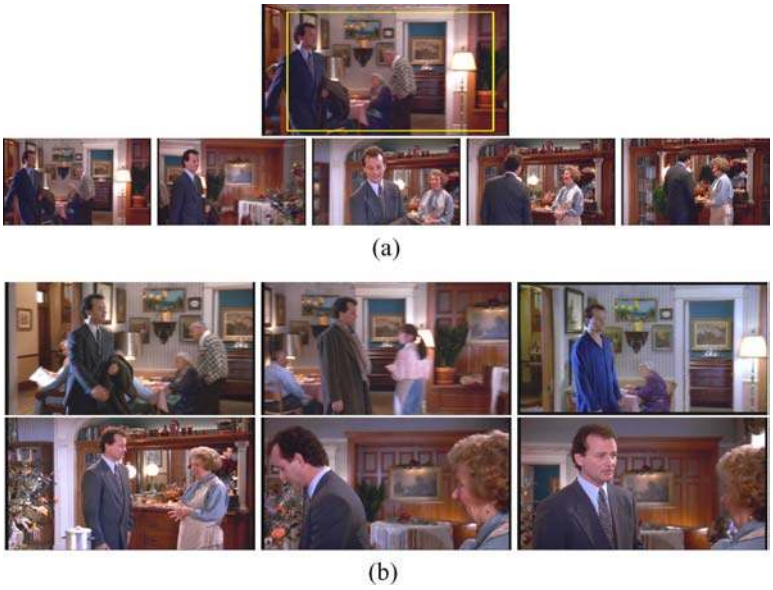


Figure 25. Object-level video matching III. The goal is to retrieve shots in the same location (the hotel dining room). (a) Top row: the query frame with the query region selected by the user. Bottom row: 5 (out of 25) associated keyframes. The object here is the extended background from the object-level grouping example of Fig. 18. The query area in each associated frame is the union of the motion grouped background regions. (b) Top row: Example frames from shots retrieved just by the user selected query frame. Bottom row: Example frames from shots retrieved by the object-level query. Query by the extended background retrieves shots which are from the same location but do not share background with the user selected query frame. The precision-recall curve for this query is shown in Fig. 26.

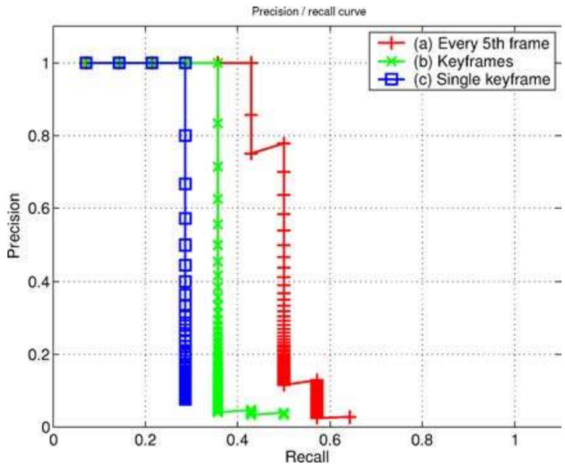


Figure 26. Object-level video matching III. Precision-recall curve for the dining room query. Examples of retrieved frames are shown in Fig. 25. (a) Every fifth frame in the query shot is used as a query frame (127 frames in total). (b) 25 keyframes used as query frames. (c) A single frame (the original frame with the user outlined region) is used as a query frame.

tracks on the person's face would be represented by several different appearance descriptors corresponding to different expressions, e.g. open and closed eyes, whereas region tracks on the (rigid) background would have just one appearance descriptor. 'Video tubes' should provide a complete but at the same time concise representation of video for recognition.

Second, a limitation of the current method is that multiple aspects/deformations have to be present in the query shot. The next step is to use available region tracks within the (correctly) retrieved shots to perform the associations. For example, if the user supplies a query still image of a frontal view of an actor's face. Querying by this image alone will only return close to-frontal views of the face with similar facial expressions. However, region tracks in the retrieved shots can be used to associate other views of the face and different expressions, which can then be used in a second set of queries. This process can be iterated. This would have to be done with some care in order to avoid a 'chain reaction' by matching on retrieved false positives.

Acknowledgments

We would like to thank P. Anandan and Mark Everingham for fruitful discussions. We are very grateful to Jiri Matas, David Lowe and Krystian Mikolajczyk for their region detectors/descriptors. Shot boundaries were provided by Mihai Osian from KU Leuven. This work was supported by EC project Vibes and Balliol College, Oxford.

References

Aanaes, H., Fisker, R. Astrom, K., and Carstensen, J. M. 2002. Robust Factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1215–1225.

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

Bolles, R.C., Baker, H.H., and Marimont, D.H. 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–56.

De la Torre, F. and Black, M. 2003. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1–3):117–142.

Ferrari, V., Tuytelaars, T., and Van Gool, L. 2003. Wide-baseline multiple-view correspondences. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. pp. 718–725.

Ferrari, V., Tuytelaars, T., and Van Gool, L. 2004a. Integrating multiple model views for object recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. pp. 105–112.

Ferrari, V., Tuytelaars, T., and Van Gool L. 2004b. Simultaneous object recognition and segmentation by image exploration. In *Proc. of the European Conference on Computer Vision*, vol. 1. pp. 40–54.

Fischler, M.A. and Bolles, R.C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.*, 24(6):381–395.

Goedeme, T., Tuytelaars, T., Van Gool, L., Sivic, J., and Zisserman, A. 2005. *Cognitive Vision Systems*, EC Project Final Report, IST-2000-29404, Chapt. Location and Object Matching and Discovery in Video. (in press).

Hartley, R.I. and Zisserman, A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049.

Jacobs, D.W. 1997. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 206–212.

Kender, J.R. and Yeo, B.L. 1998. Video scene segmentation via continuous video coherence. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 367–373.

Lowe, D. 1999. Object recognition from local scale-invariant features. In *Proc. of the 7th International Conference on Computer Vision*, Kerkyra, Greece. pp. 1150–1157.

Lowe, D. 2001. Local feature view clustering for 3D object recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, Springer. pp. 682–688.

Mahindroo, A., Bose, B., Chaudhury, S., and Harit, G. 2002. Enhanced video representation using objects. In *Proc. of the Indian Conference on Computer Vision, Graphics and Image Processing*. pp. 105–112.

Matas, J., Chum, O., Urban, M., and Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of the British Machine Vision Conference*. pp. 384–393.

Mikolajczyk, K., and Schmid, C. 2002. An affine invariant interest point detector. In *Proc. of the 7th European Conference on Computer Vision*, Copenhagen, Denmark. Springer-Verlag.

Osian, M. and Van Gool, L. 2004. Video shot characterization. *Machine Vision and Applications Journal*, 15(3):172–177.

Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. 2003. 3D object modeling and recognition using local affine-invariant descriptors and multi-view spatial constraints. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. pp. 272–280.

Rothganger, F., Lazebnik, S. Schmid, C., and Ponce, J. 2004. Segmenting, modeling, and matching video clips containing multiple moving objects. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. pp. 914–921.

Schaffalitzky, F. and Zisserman, A. 2002. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. of the 7th European Conference on Computer Vision,* Copenhagen, Denmark, vol. 1. Springer-Verlag. pp. 414—431.

Schaffalitzky, F. and Zisserman, A. 2003. Automated location matching in movies. *Computer Vision and Image Understanding* 92: 236–264.

Schmid, C. 1997. 'Appariement d'Images par Invariants Locaux de Niveaux de Gris'. Ph.D. thesis, L'Institut National Polytechnique de Grenoble, Grenoble.

Shum, H.-Y., Ikeuchi, I., and Reddy, R. 1995. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):855–867.

Sivic, J., Schaffalitzky, F. and Zisserman, A. 2004. Object level grouping for video shots. In *Proc. of the 8th European Conference on Computer Vision, Prague, Czech Republic*, Springer-Verlag, vol. 2., pp. 85–98.

Sivic, J. and Zisserman, A. 2003. Video google: A text retrieval approach to object matching in videos. In *Proc. of the International Conference on Computer Vision*.

Torr, P.H.S. 1995. Motion segmentation and outlier detection. Ph.D. thesis, Dept. of Engineering Science, University of Oxford.

Torr, P.H.S., Szeliski, R., and Anadan, P. 2001. An integrated bayesian approach to layer extraction from image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(3):297–304.

Torr, P.H.S., Zisserman, A., and Maybank, S. 1998. Robust detection of degenerate configurations for the fundamental matrix. *Computer Vision and Image Understanding* 71(3):312–333.

Tuytelaars, T. and Van Gool, L. 2000. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. of the 11th British Machine Vision Conference,* Bristol. pp. 412–425.

Wallraven, C. and Bulthoff, H. 2001. Automatic acquisition of exemplar-based representations for recognition from image sequences. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Models vs. Exemplars*.

Zelnik-Manor, L. and Irani, M. 1999. Multi-view subspace constraints on homographies. In *Proc. of the 7th International Conference on Computer Vision,* Kerkyra, Greece, vol. 2. pp. 710–715.