Object Level Grouping for Video Shots

Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman

Robotics Research Group, Department of Engineering Science,
University of Oxford
http://www.robots.ox.ac.uk/~vgg

Abstract. We describe a method for automatically associating image patches from frames of a movie shot into object-level groups. The method employs both the appearance and motion of the patches.
There are two areas of innovation: first, affine invariant regions are used to repair short gaps in individual tracks and also to join sets of tracks across occlusions (where many tracks are lost simultaneously); second, a robust affine factorization method is developed which is able to cope with motion degeneracy. This factorization is used to associate tracks into object-level groups.
The outcome is that separate parts of an object that are never visible simultaneously in a single frame are associated together. For example, the front and back of a car, or the front and side of a face. In turn this enables object-level matching and recognition throughout a video.
We illustrate the method for a number of shots from the feature film 'Groundhog Day'.

1 Introduction

The objective of this work is to automatically extract and group independently moving 3D semi-rigid (that is, rigid or slowly deforming) objects from video shots. The principal reason we are interested in this is that we wish to be able to match such objects throughout a video or feature length film. An object, such as a vehicle, may be seen from one aspect in a particular shot (e.g. the side of the vehicle) and from a different aspect (e.g. the front) in another shot. Our aim is to learn multi-aspect object models [19] from shots which cover several visual aspects, and thereby enable object level matching.

In a video or film shot the object of interest is usually tracked by the camera — think of a car being driven down a road, and the camera panning to follow it, or tracking with it. The fact that the camera motion follows the object motion has several beneficial effects for us: the background changes systematically, and may often be motion blurred (and so features are not detected there); and, the regions of the object are present in the frames of the shot for longer than other regions. Consequently, object level grouping can be achieved by determining the regions that are most common throughout the shot.

In more detail we define object level grouping as determining the set of appearance patches which (a) last for a significant number of frames, and (b) move (semi-rigidly) together throughout the shot. In particular (a) requires that every appearance of a patch is identified and linked, which in turn requires extended tracks for a patch – even associating patches across partial and complete occlusions. Such thoroughness has two benefits: first, the number of frames in which a patch appears really does correspond to the time that

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 85–98, 2004.

it is visible in the shot, and so is a measure of its importance. Second, developing very long tracks significantly reduces the degeneracy problems which plague structure and motion estimation [5].

The innovation here is to use both motion and appearance consistency throughout the shot in order to group objects. The technology we employ to obtain appearance patches is that of affine co-variant regions [9,10,11,18]. 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 invariant tracked regions are used to repair short gaps in tracks (section 3) and also associate tracks when the object is partially or totally occluded for a period (section 5). The result is that regions are matched throughout the shot whenever they appear. Second, we develop a method of robust affine factorization (section 4) which is able to handle degenerate motions [17] in addition to the usual problems of missing and mis-matched points [1,3,7,13].

The task we carry out differs from that of layer extraction [16], 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.

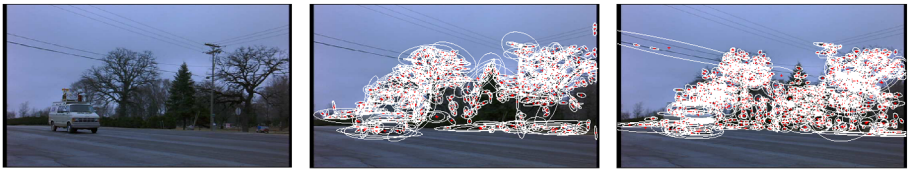
In section 6 we demonstrate that the automatically recovered object groupings are sufficient to support object level matching throughout the feature film 'Groundhog Day' [Ramis, 1993]. This naturally extends the frame based matching of 'Video Google' [14].

2 Basic Segmentation and Tracking

Affine invariant regions. Two types of affine invariant region detector are used: one based on interest point neighborhoods [10,11], the other based on the "Maximally Stable Extremal Regions" (MSER) approach of Matas *et al.* [9]. In both the detected region is represented by an ellipse. 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 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. The union of both provides a good coverage of the image provided it is at least lightly textured, as can be seen in figure 1. The number of regions and coverage depends of course on the visual richness of the image.

To obtain tracks throughout a shot, regions are first 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. It is here that we benefit significantly from the affine invariant regions: first, incorrect matches can be removed by requiring consistency with multiple view geometric relations: the robust estimation of these relations for point matches is very mature [6] and can be applied to the region centroids; second, the regions can be matched on their appearance. The latter is far more discriminating and invariant than the usual cross-correlation over a square window used in interest point trackers.



(a) (b) (c)

Fig. 1. Example of affine invariant region detection. (a) frame number 8226 from 'Groundhog Day'. (b) ellipses formed from 722 affine invariant interest points. (c) ellipses formed from 1269 MSER regions. Note the sheer number of regions detected just in a single frame, and also that the two types of region detectors fire at different and complementary image locations.

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 generous 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 with a generous 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 figure 3b. This shot is used throughout the paper to illustrate the stages of the object level grouping. Note that the tracks have very few outliers.

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 5.

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.

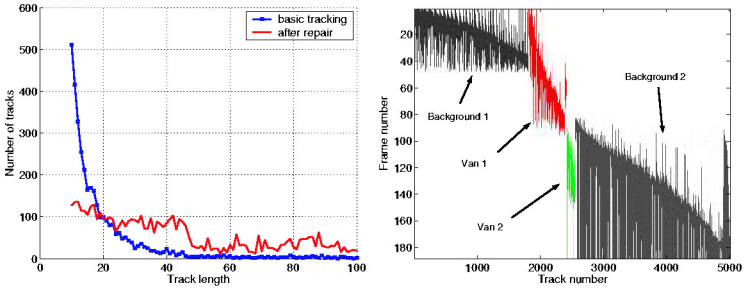


Fig. 2. (a) Histogram of track lengths for the shot shown in figure 3 for basic tracking (section 2) and after short range track repair (section 3). Note the improvement in track length after the repair – the weight of the histogram has shifted to the right from the mode at 10. (b) The sparsity pattern of the tracked features in the same shot. The tracks are coloured according to the independently moving objects, that they belong to, as described in section 4. 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 van object before and after the occlusion.

The repair algorithm works on pairs of neighboring frames and attempts to extend already existing tracks which terminate in the current frame. Each region which has been successfully tracked for more than n 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 spatially neighboring tracks (here $n = 5$ and $k = 5$). In the case of successive frames only translational motion is estimated from the neighboring tracks. In the case of more separated frames the full affine transformation imposed by each tracked region should be employed.

It must now be decided if there is a detectable region near the propagated point, and if it matches an existing region. The refinement algorithm of Ferrari *et al.* [4] is used to fit the 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, then a new region is instantiated. 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, in which case the current track is extended; (iii) if the new instantiated region matches (correlates with) an existing region in its (5 pixel) neighborhood 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 2 gives the 'before and after' histogram of track lengths, and the results of this repair are shown in figure 3. As can be seen, there is a dramatic improvement in the length of the tracks – as was the objective here. Note, the success of this method is due to the availability and use of two complementary constraints – motion and appearance.

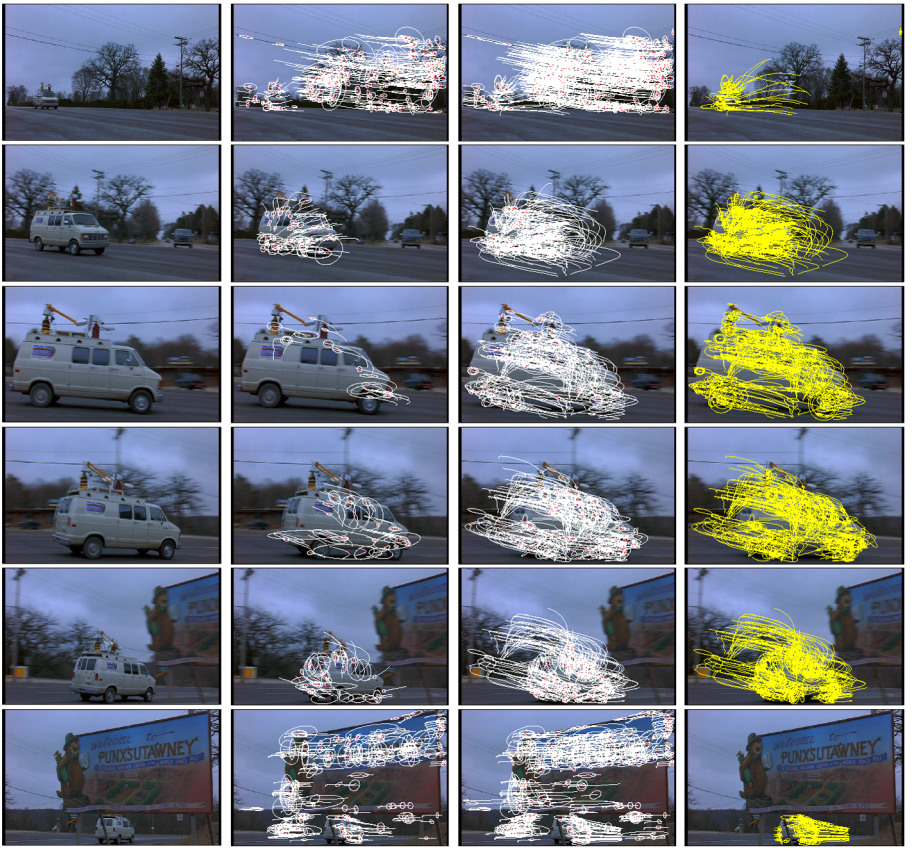


Fig. 3. Example I: (a) 6 frames from one shot (188 frames long) of the movie 'Groundhog Day'. The camera is panning right, and the van moves independently. (b) frames with the basic region tracks superimposed. The tracked path (x, y) position over time is shown together with each tracked region. (c) frames with short range repaired region tracks superimposed. Note the much longer tracks on the van after applying this repair. For presentation purposes, only tracks that last for more than 10 frames are shown. (d) One of the three dominant objects found in the shot. The other two are backgrounds at the beginning and end of the shot. No background is tracked in the middle part of the shot due to motion blur.

4 Object Extraction by Robust Sub-space Estimation

To achieve the final goal of identifying objects in a shot we must 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 figure 3 the ideal outcome would be the van as one object, and then several groupings of the background. The grouping constraint

used here is that of common (semi-)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 $\mathbf{x}_j^i$ associated with the object could be written as [20]:

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

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

$$\sum_{ij} \left|\left| n_j^i \left(\mathbf{x}_j^i - \left(\mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\right) (X_j, Y_j, Z_j)^\top\right)\right|\right|^2 \tag{1}$$

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 [17] 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 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 robustly 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. This approach differs fundamentally from that of [1,3] where robustness is achieved by iteratively re-weighting outliers but no account is taken of motion degeneracy.

4.1 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 [6], and an inlying set is scored by its error averaged over the three homographies. 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 every frame in the shot. This provides temporal coherence (since neighboring triplets share two frames) which is useful in the next step where motion groups are linked throughout the shot into an object.

4.2 Aggregating Segmentation over Multiple Frames

The problem with fitting motion models to pairs or triplets of frames are twofold: phantom motion cluster corresponding to a combination of two independent motions grouped

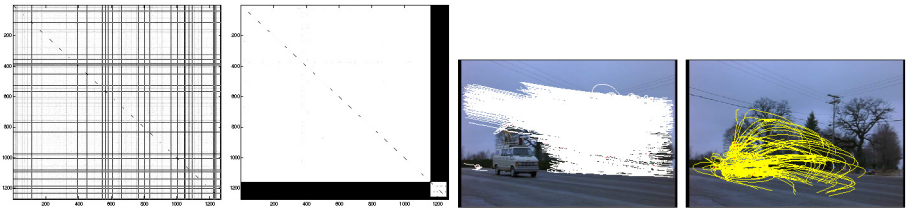


Fig. 4. Aggregating segmentation over multiple frames. (a) The track co-occurrence matrix for a ten frame block of the shot from figure 3. White indicates high co-occurrence. (b) The thresholded co-occurrence matrix re-ordered according to its connected components (see text). (c) (d) The sets of tracks corresponding to the two largest components (of size 1157 and 97). The other components correspond to 16 outliers.

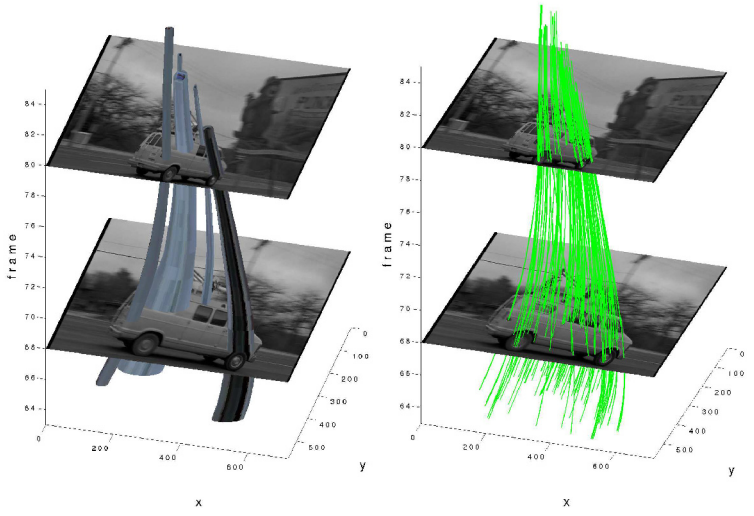


Fig. 5. Trajectories following object level grouping. Left: Five region tracks (out of a total 429 between these frames) shown as spatiotemporal "tubes" in the video volume. Right: A selection of 110 region tracks (of the 429) shown by their centroid motion. The frames shown are 68 and 80. Both figures 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 [2], but with explicit correspondence; the centroid motion demonstrates that outlier 'strands' have been entirely 'combed' out, to give a well conditioned track set.

together can ariser [15], and an outlying track will be occasionally, but not consistently, grouped together with the wrong motion group. 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.

Fig. 6. Example II: object level grouping for another (35 frame) shot from the movie 'Groundhog Day'. 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 basic motion grouping of section 4.1 provides a track segmentation for each frame (computed using the two neighbouring frames too). To take advantage of temporal consistency 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 4 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. The result is a set of connected clusters of tracks which correspond to independently moving objects throughout the shot.

4.3 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 100 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 ($m > 20$ here). To test

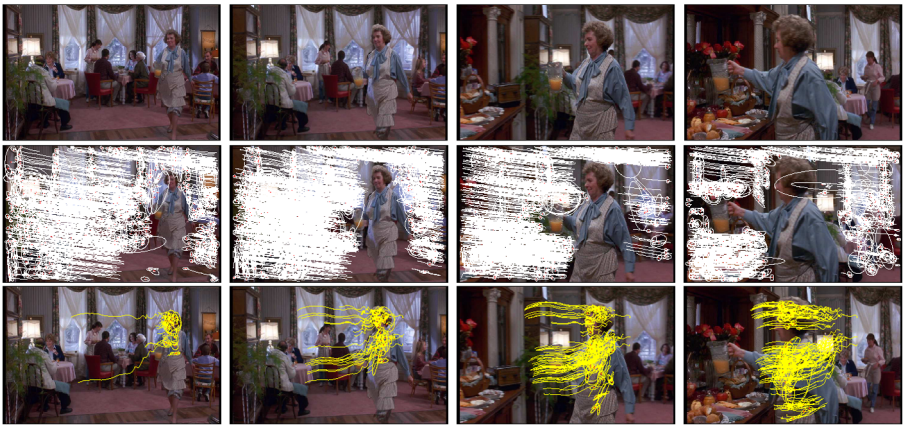


Fig. 7. Example III: object level grouping for another (83 frame) shot from the movie Groundhog Day. Top row: The original frames of the shot. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 225 (landlady) and 2764 (background). The landlady is an example of a slowly deforming object.

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 equation (1).

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 is set on reprojection error (measured in pixels) 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.

4.4 Object Extraction Results

An example of one of the extracted objects (the van) is shown in figure 3d. In total, four objects are grouped for this shot, two corresponding to the van (before and after the occlusion by the post, see figure 9 in section 5) and two background objects 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 figure 2(b). Each of the background objects is composed of only one motion cluster. The van 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 figure 5.

A second example of rigid object extraction from a different shot is given in figure 6. Figures 7 and 8 show examples of slowly deforming objects. This deformation is allowed

Fig. 8. Example IV: object level grouping for another (645 frame) shot from the movie Groundhog Day. 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 figure 9b. 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.

because rigidity is only applied over a sliding baseline of m frames, with m less than the total length of the track. For example we are able to track regions on slowly rotating and deforming face such as a mouth opening.

5 Long Range Track Repair

The object extraction method described in the previous section groups objects which 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 [9,11,18] to a pair of frames that each contain the object. There are two stages: first, epsilon-nearest neighbor search on a SIFT descriptor [8] for each region, is performed to get a set of putative region matches, and second, this set is disambiguated by a local spatial consistency constraint: a putative match is discarded if it does not have a supporting match within its k-nearest spatial neighbors [12,14]. Since each region considered for matching is part of a track, it is straightforward to extend the matching to tracks. The two objects

(a) (b)

Fig. 9. Two examples of long range repair on (a) shot from figure 3 where a van is occluded (by a post) which causes the tracking and motion segmentation to fail, and (b) shot from figure 8 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.

are deemed matched if the number of matched tracks exceeds a threshold. Figure 9 gives two examples of long range repair on shots where the object was temporarily occluded.

6 Application: Object Level Video Matching

Having computed object level groupings for shots throughout the film, we are now in a position to retrieve object matches given only part of the object as a query region. Grouped objects are represented by the union of the regions associated with all of the object's tracks. This provides an implicit representation of the 3D structure, and is sufficient for matching when different parts of the object are seen in different frames. In more detail, an object is represented by the set of regions associated with it in each key-frame. As shown in figures 10 and 11, the set of key-frames naturally spans the object's visual aspects contained within the shot.

In the application we have engineered, the user outlines a query region of a key-frame in order to obtain other key-frames or shots containing the scene or object delineated by the region. The objective is to retrieve *all* key-frames/shots within the film containing the object, even though it may be imaged from a different visual aspect.

The object-level matching is carried out by determining the set of affine invariant regions enclosed by the query region. The convex hull of these tracked regions is then computed in each key frame, and this hull determines in turn a query region for that frame. Matching is then carried out for all query regions using the Video Google method described in [14].

An example of object-level matching throughout a database of 5,641 key-frames of the entire movie 'Groundhog Day' is shown in figures 10 and 11.

Fig. 10. Object level video matching I. Top row: The query frame with query region (side of the van) selected by the user. Second row: The automatically associated keyframes and outlined query regions. Next four rows: Example frames retrieved from the entire movie 'Groundhog Day' by the object level query. 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.

7 Discussion and Extensions

We have shown that representing an object as a set of viewpoint invariant patches has a number of beneficial consequences: gaps in tracks can be reliably repaired; tracked objects can be matched across occlusions; and, most importantly here, different viewpoints of the object can be associated provided they are sampled by the motion within a shot.

Fig. 11. Object level video matching II. Top row: The query frame with query region selected by the user. The query frame acts as a portal to the keyframes associated with the object by the motion-based grouping (shown in the second row). Note that in the associated keyframes the person is visible from the front and also changes scale. See figure 8 for the corresponding object segmentation. Next three rows: Example frames retrieved from the entire movie 'Groundhog Day' by the object level query.

We are now at a point where useful object level groupings can be computed automatically for shots that contain a few objects moving independently and semi-rigidly. This has opened up the possibility of pre-computing object-level matches throughout a film – so that content-based retrieval for images can access objects directly, rather than image regions; and queries can be posed at the object, rather than image, level.

Acknowledgements. We are very grateful to Jiri Matas and Jan Paleček for their MSE region detector. Shots were provided by Mihai Osian from KU Leuven. This work was supported by EC project Vibes and Balliol College, Oxford.

References

1. H. Aanaes, R. Fisker, K. Astrom, and J. M. Carstensen. Robust factorization. *IEEE PAMI*, 24:1215–1225, 2002.
2. R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.
3. F. De la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54:117–142, 2003.
4. V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *Proc. CVPR*, pages 718–725, 2003.
5. A. Fitzgibbon and A. Zisserman. Automatic camera tracking. In Shah and Kumar, editors, *Video Registration*. Kluwer, 2003.
6. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
7. D. W. Jacobs. Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images. In *Proc. CVPR*, pages 206–212, 1997.
8. D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
9. 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.
10. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.
11. F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, volume 1, pages 414–431. Springer-Verlag, 2002.
12. C. Schmid. *Appariement d'Images par Invariants Locaux de Niveaux de Gris*. PhD thesis, L'Institut National Polytechnique de Grenoble, Grenoble, 1997.
13. H.-Y. Shum, I. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE PAMI*, 17:854–867, 1995.
14. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.
15. P. H. S. Torr. *Motion segmentation and outlier detection*. PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.
16. P. H. S. Torr, R. Szeliski, and P. Anadan. An integrated bayesian approach to layer extraction from image sequence. *IEEE PAMI*, 23:297–304, 2001.
17. P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degenerate configurations for the fundamental matrix. *CVIU*, 71(3):312–333, 1998.
18. T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. BMVC.*, pages 412–425, 2000.
19. C. Wallraven and H. Bulthoff. Automatic acquisition of exemplar-based representations for recognition from image sequences. In *CVPR Workshop on Models vs. Exemplars*, 2001.
20. L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *Proc. ICCV*, 1999.