Segmentation and Tracking of Multiple Video Objects ⋆

A. Colombari, A. Fusiello, V. Murino

Dipartimento di Informatica, University of Verona
Strada Le Grazie 15, 37134 Verona, Italy.

Abstract

This paper describes a technique that produces a content-based representation of a video shot composed by a background (still) mosaic and one or more foreground moving objects. Segmentation of moving objects is based on ego-motion compensation and on background modelling using tools from robust statistics. Region matching is carried out by an algorithm that operates on the Mahalanobis distance between region descriptors in two subsequent frames and uses singular value decomposition to compute a set of correspondences satisfying both the principle of proximity and the principle of exclusion. The sequence is represented as a layered graph, and specific techniques are introduced to cope with crossing and occlusion. Examples of MPEG-4 (Main Profile) encoding are reported.

Key words: Content-based representation, MPEG, Video coding, Video sequence analysis, Mosaicing, Motion segmentation

1 Introduction

Digital video is nowadays widespread on the World Wide Web and in multimedia databases. Unfortunately, the usefulness of such large amount of information is limited by the adequacy of the retrieval method. Whereas text documents are self-describing, digital videos do not give any explicit description of their content (see [1] for a review on video indexing). Moreover, transmission of video requires high compression rates to make it viable.

⋆ Preliminary versions of this work appeared in ICIP'03 and WIAMIS'04.

Email address: colombari@sci.univr.it, {andrea.fusiello, vittorio.murino}@univr.it (A. Colombari, A. Fusiello, V. Murino).

By exploiting the content-based representation offered by MPEG-4 [2], video shots can be encoded as a stationary background mosaic – obtained after compensating for camera motion – plus moving objects (MOs) represented individually. This allows to achieve a high compression rate in the transmission of the sequence, since all the information about the background (which does not change) is sent only once. Besides, this representation of the video is a true content-based description, that allows manipulation and adaptation (as in the MPEG-7 standard) [3].

The challenge is to create a system that is able to do this segmentation automatically and accurately, and to cope with complex situations, such as crossing between MOs and occlusion with elements of the static background.

Several techniques have been proposed for motion segmentation (see [4] for a review), as image differencing (see [5] for example), temporal analysis of gray-levels based on probabilistic models [6], robust motion estimation [7], or misalignment analysis based on the normal flow [8]. In [9], body parts are segmented and tracked, using a body model to help resolving ambiguities and tracking failures. Our tracking approach was inspired by [10], where a graph is used to represent objects and both shape and colour features are used to match them. A graph-based approach is also used in [11] to track features.

In our work, MOs are obtained from the original video shot by differencing with the background. For each frame, the mosaic of the background is backwarped onto the frame and each pixel is labelled as belonging to a MO or not by comparing it with a statistical model of the background. Then, the resulting binary image is cleaned, and connected regions (blobs) are identified as candidate MOs. The next step is to exploit temporal coherence: blobs are tracked (non-causally) through the sequence. Finally, noisy tracks are discarded and tracks belonging to the same object are merged. Our work builds on a previous research [12], and improves radically the blob tracking algorithm, allowing for occlusions between MOs, occlusions between a MO and a background object, MOs entering and leaving the scene at any point.

Specific contributions of this papers include the model of the background, based on robust statistics, and the blob matching technique based on a generalisation of the method for feature-matching proposed in [13,14].

2 Overview

The input is a video shot [1] with a static background and negligible parallax. The background must be the dominant part of the scene (see Sec. 3.1). As the processing is non-causal, all the frames composing the video shot are needed simultaneously.

The output is a representation of the sequence suitable to be encoded in MPEG-4 (Main Profile). The central concept in MPEG-4 is that of the Video Object (VO). Being content-based oriented, MPEG-4 considers a scene to be composed of several VOs, which are separately encoded. Each VO is characterised by intrinsic properties such as shape, texture, and motion. Shape is represented by a binary mask or by an 8-bit transparency mask (this feature is available in the MPEG-4 Main Profile).

In our case, the video shot is represented as being composed by a *sprite panorama* (i.e., a still image describing the content of the background over all the frames in the shot) and one arbitrary-shape VO for each foreground moving object, with a binary mask as shape descriptor. For each frame, the global motion parameters are given by the coordinates of the four corners of the image transformed in the mosaic reference frame. Only moving VOs are extracted: If a VO is static in the whole shot, it is considered part of the background (as the referee and the ball boys in the "Stefan" sequence, Figure 9).

The method we are proposing is based on (i) segmenting moving objects from the background using mosaicing and (ii) tracking them in the video sequence using frame-to-frame matching and a graph representation. The procedure can be decomposed into several steps, according to the scheme depicted in Figure 1. The details of each step will be given in the corresponding section.

(1) Ego-motion compensation (Sec. 3.1). The projective transformations (homographies) linking each pair of consecutive frames are recovered by tracking features.
(2) Background modelling (Sec. 3.2). A mosaic is built using the median to assign pixel colours and computing a robust estimate of the colour variability. The resulting mosaic depicts only the static background.
(3) Foreground segmentation (Sec. 3.3). The X-84 outlier rejection rule is used to threshold the difference between mosaic and each frame, thereby obtaining binary masks for moving objects.
(4) Blob matching (Sec. 4.1). Tracks are initialised by matching blobs using a technique that enforces simultaneously the proximity and the exclusion

[1] A video shot is defined as an image sequence captured with a single operation of the camera and presenting a continuous action in time and space [1].

principles. The output is a set of tracks in a layered graph G representing the blobs in the sequence.

(5) Connection (Sec. 4.2). At this point, a single object can correspond to several tracks, due to occlusions or over-segmentation. Tracks are then pruned to remove spurious ones and joined (when possible) using template matching. Connected components in G are computed.

(6) Object recovery (Sec. 4.3). In the easiest case, each connected component represents an object, but if objects can merge and disappear behind occlusors, this is not the case: One connected component might represent more than one object, and one object can be associated to more that one component. Specific heuristics are used to associate tracks and objects.

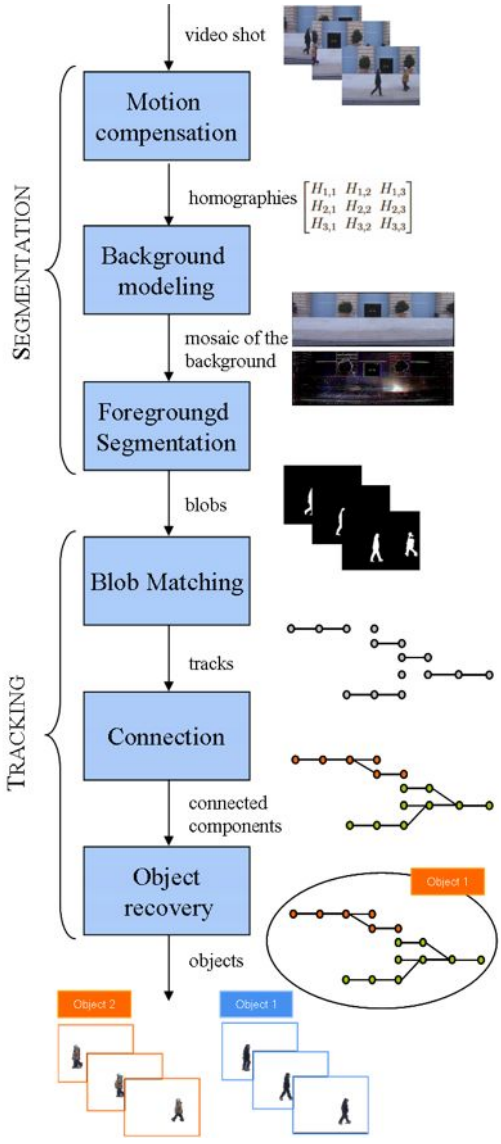

Fig. 1. Overview of the method. At a high level it consists of two parts: segmentation and tracking. The input is a video shot, the output are shape descriptors for each moving object.

3 Segmentation

3.1 Ego-motion compensation

Two pictures of the same scene are related by a non-singular linear transformation of the projective plane (or *homography*) in two cases: i) the scene is planar or ii) the point of view does not change (pure rotation). In these cases, which can be summarised by saying that there must be no *parallax*, images can be composed together to form a *mosaic*.

Inter-frame homography computation is based on correspondences produced by the Kanade-Lucas-Tomasi (KLT) tracker [15], initialised with phase-correlation to reduce search range. As in [12], Least Median of Squares (LMedS) is used to be robust against tracking errors and features attached to moving objects. Finally, given the set of *inlier* point matches, the homography is computed according to a technique proposed in [16], which obtains an optimal estimate and reduces the instability of images alignment even with a small overlap between frames. These homographies are then combined to obtain frame-to-mosaic homographies and frames are warped accordingly and blended to produce a mosaic of the background. The use of LMedS implicitly assumes that the background is dominant, i.e., that the majority of the tracked features belong to the background, because LMedS has a breakdown point of 50%. Using a robust estimator with a higher breakdown point (such as RANSAC [17]) would allow, in principle, to cope with highly cluttered sequences, where the dominant background assumption fails.

More details on the mosaicing technique in [12,18].

3.2 Background modelling

Starting from a single mosaic pixel P, a temporal line piercing all the aligned frames will intersect pixels that correspond to the background and pixels belonging to MOs. The colour histogram of these pixels is modelled as a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ corrupted by outliers, corresponding to the MOs. Therefore, the median of the distribution $\bar{c} = \underset{i}{\text{med}}\{c_i\}$, being a robust estimate of the mean μ, is taken as the background colour and assigned to P (see Figure 2). As a result, only the pixels corresponding to the background contribute to the colour of P. Thus, the moving objects are removed. Actually, anything that keeps the same position in the mosaic for most of the time is included in the background.

Moreover, an estimate of the background colour variability at that point is

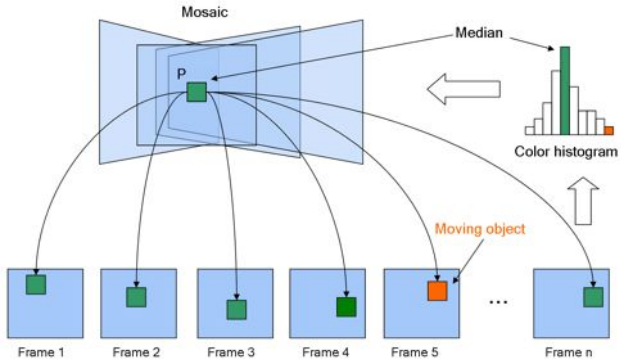

Fig. 2. Background modelling. The colour assigned to a mosaic pixel P is the median of the colours of the corresponding pixels in the video frames.

attached to each mosaic pixel P. A robust estimator of the spread of the distribution is given by the median absolute difference (MAD):

$$\mathrm{MAD} = \mathrm{med}_i\{|c_i - \bar{c}|\}.$$

It can be seen [19] that, for symmetric distributions, the MAD coincides with the *inter-quartile range*: $\mathrm{MAD} = (\xi_{3/4} - \xi_{1/4})/2$, where ξ_q is the q^{th} quartile of the distribution (for example, the median is $\xi_{1/2}$). For a normal distribution Φ, we infer the standard deviation σ from

$$\mathrm{MAD} = \Phi^{-1}(3/4)\sigma \approx 0.6745\sigma.$$

In the above discussion we considered the colour as if it was a scalar quantity, but in fact it is a three-dimensional vector. Each channel (R, G, B) can be considered as an independent stochastic variable with normal distribution. The mean and variance for each channel are estimated independently as described above and then packed into a vector $\bar{\mathbf{c}}$ and a diagonal 3×3 covariance matrix $\mathbf{\Lambda}$ respectively. The Mahalanobis distance between the colour $\bar{\mathbf{c}}$ of a background pixel and the colour $\mathbf{c}_i$ of a corresponding pixel in a given frame i, writes:

$$(\mathbf{c}_i - \bar{\mathbf{c}})^{\mathsf{T}}(\mathbf{\Lambda} + 0.5\mathbf{I})^{-1}(\mathbf{c}_i - \bar{\mathbf{c}}) \qquad (1)$$

where it is assumed that in the frames each colour channel is affected by a Gaussian noise with variance equal to 0.5. Hence, a pixel with colour c_i is deemed to belong to the background with confidence α provided that the Mahalanobis distance is below the $\alpha{-}th$ quartile of the Chi-square distribution with three degrees of freedom. We used the value 19.339, corresponding to 3.5 standard deviation (99.977% confidence) .

In the one dimensional case, this rule is known in robust statistics as the X-84 outlier rejection rule [19].

3.3 Foreground segmentation

MOs are obtained from the original video shot by differencing with the background. Each frame is warped onto the mosaic of the background, and each pixel is labelled as belonging to a MO or not according to the X-84 rule. Then, the resulting binary image is cleaned with morphological filtering. In detail, we performed the following sequence of operations: majority [2] iterated three times, opening followed by closing with a 3×3 structuring element. Finally, connected regions (blobs) are identified as candidate MOs. Examples of binary masks are shown in Figure 3.

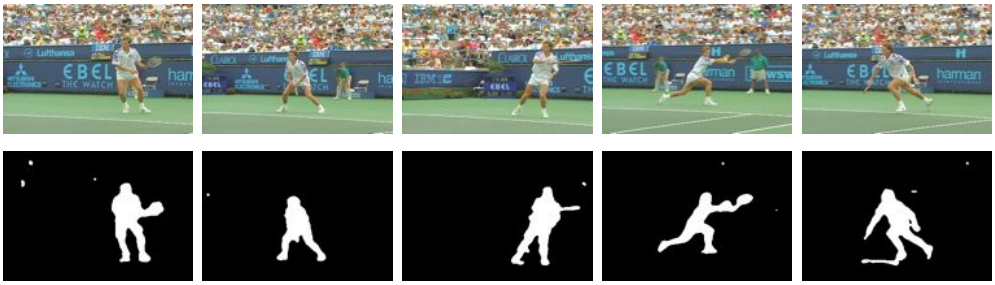

Fig. 3. Selected frames from the "Stefan" sequence and corresponding binary masks.

4 Tracking

Tracking is carried out on a graph G representing the whole sequence. G is a layered graph, where each layer corresponds to a frame and each node to a blob. The tracking starts with an empty edge set and proceeds by adding edges. Ideally, an edge links two nodes from consecutive layers if they represent the same MO (or part of it) at different time instants.

Some definitions are needed to set up the terminology that will be used in the sequel. A *track* is a chain of at least two nodes belonging to consecutive frames. Every node in a track, other than the first and the last, has degree [3] two. Any node with a degree of two, one or zero is respectively referred as *internal*, *external*, or *isolated* node. An isolated node does not belong to any track. The *length* of a track is the number of nodes belonging to that track, the *area* of a track is the median area of the blobs corresponding to the nodes of that track.

The goal is to find one track for each MO, thereby identifying blobs as objects (Figure 4).

[2] The majority operation sets a pixel to 1 if five or more pixels in its 3 ×3 neighborhood are 1's. See the `bwmorph` function in the MATLAB Image Processing Toolbox.
[3] The degree of a node is the number of edges incident on that node.

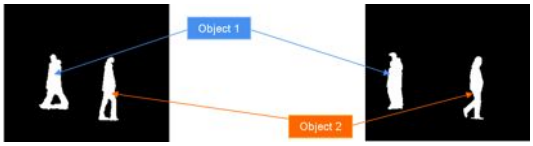

Fig. 4. The goal of tracking is assigning an object identity to blobs.

4.1 Blob matching

In a first phase, tracks are initialised by matching nodes from one layer to the next. A dissimilarity (distance) measure between blobs is defined taking into account the appearance (shape and colour) of the blob and its position. In particular, each blob is described by a feature vector $\mathbf{b}$ composed of solidity, eccentricity, orientation [4], area, dimensions of the bounding box, average colour, contrast (standard deviation of the colour) and position of the centroid in the mosaic reference frame.

The dissimilarity of blobs I_i and J_j is computed as the Mahalanobis distance between the respective feature vectors:

$$d_{ij} = (\mathbf{b}_i - \mathbf{b}_j)^{\mathsf{T}} (\mathbf{\Lambda}_I + \mathbf{\Lambda}_J)^{-1} (\mathbf{b}_i - \mathbf{b}_j) \tag{2}$$

where $\mathbf{\Lambda}_I$ and $\mathbf{\Lambda}_J$ are the covariance matrices of the feature vectors in images I and J respectively. In practice, $\mathbf{\Lambda}$ is a diagonal matrix containing normalising weights for each feature-vector element.

Matching is carried out with a technique introduced by Scott and Longuet-Higgins [13] and elaborated upon by [14] (henceforth referred to as "SL matching"), where the singular value decomposition (SVD) of a suitable matrix is used for associating features of two images.

Let $\{I_i\}_{1\ldots n}$ and $\{J_j\}_{1\ldots m}$ the two sets of blobs which are to be put in one-to-one correspondence. The first stage is to build a *proximity matrix* $\mathbf{P}$ of the two sets of features: $P_{ij} = e^{-d_{ij}/2}$. The next stage is to perform the SVD of $\mathbf{P}$

$$\mathbf{P} = \mathbf{USV}^{\mathsf{T}}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and $\mathbf{S}$ is a non-negative $m \times n$ diagonal matrix. Finally, $\mathbf{S}$ is converted into a new $m \times n$ matrix $\mathbf{D}$ by replacing every diagonal element S_{ii} with 1, thus obtaining another matrix $\mathbf{Q} = \mathbf{UDV}^{\mathsf{T}}$ of the same

[4] Solidity is defined as the ratio between the area of the region and the area of its convex hull. Eccentricity is the eccentricity of the ellipse that has the same second-moments as the region. Orientation is the angle between the major axis of this ellipse and the x-axis. For further details, see the `regionprops` function in the MATLAB Image Processing Toolbox.

shape as the original proximity matrix and whose rows are mutually orthogonal. The element Q_{ij} indicates the extent of pairing between the blobs I_i and J_j. This matrix incorporates the principle of proximity (that favours a match with the closest feature) by construction of $\mathbf{P}$ and the principle of exclusion (that prohibits many-to-one correspondences) by virtue of its orthogonality.

If Q_{ij} is both the largest element in its row and in its column, then I_i and J_j are regarded as corresponding with each other, provided that their Mahalanobis distance is below a certain threshold. This threshold is computed as $\alpha - th$ quartile of the Chi-square distribution with 14 degrees of freedom (since $\mathbf{b} \in \mathbb{R}^{14}$), where α is the desired confidence level (the choice of α turns out not to be particularly critical).

When Q_{ij} is the greatest element in row i but not the greatest in column j, "then we may regard I_i as competing unsuccessfully for partnership with J_j" [13], and a matching is not established.

The use of Mahalanobis distance is customary in data association [20], but it is often used in a nearest-neighbour scheme (proximity principle). The approach described above extends it by introducing also the exclusion principle. On the other hand, our proximity matrix $\mathbf{P}$ generalises the solution proposed by [14], because using Mahalanobis distance in a feature space allows to take into account both appearance and spatial position (and possibly other features) in a consistent way.

This matching across the sequence produces tracks (Figure 5). Many of them are due to noise, and only a few correspond to moving objects. A track may represent only a *part* of an object, in the case of occlusion with a static element or because of over-segmentation.

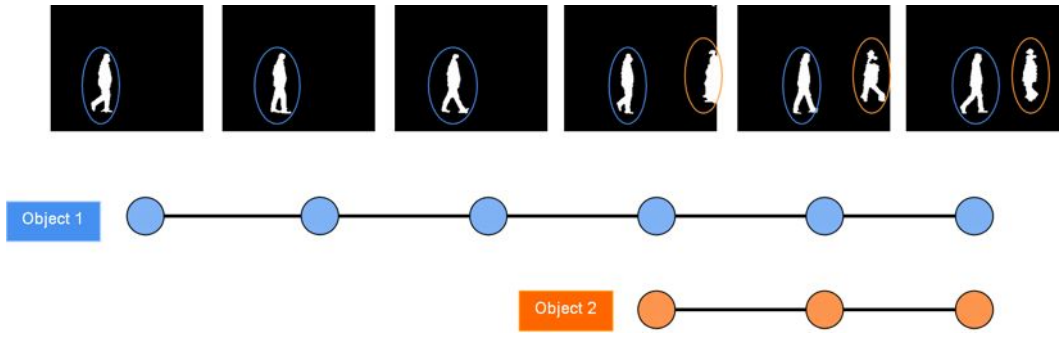

Fig. 5. Blob matching. Two tracks (chains of nodes) are produced by SL matching along the sequence.

Tracks are classified according to their length and area. *Bad* tracks are those shorter than 5% of the longest track and smaller than 5% of the biggest track. Bad tracks are marked but not discarded yet, because some of them could correspond to parts of bigger MOs.

4.2 Connection

If blobs are allowed to split (for the reasons described before) and merge (because the projections of the objects in the image overlap or because they physically touch each other) their descriptors change significantly, hence SL matching is likely to fail. This also happens when objects enter/exit the view frustum or appear/disappear behind large occlusors. Moreover, the output of the blob matching phase are tracks, which are not suited to represent objects in split or merge situations, because nodes should be allowed to have degree greater than two.

A template matching, that is likely to succeed where the SL matching failed, is employed in order to:

- connect tracks representing fragments of the same MO (Figure 6);
- connect tracks representing overlapping MOs (Figure 7);
- prolong tracks representing appearing/disappearing MOs.

The template matching procedure uses colour information and sum of squared differences (SSD) metric. Each external node is template-matched against the nodes (blobs) of the adjacent layer contained in a suitable search window, including internal nodes and those corresponding to a bad track. The role of the template is played by the blob with the smaller area. The search window depends on the target-blob area and it is centred in the predicted position of the centroid, using a moving average. If minimum SSD value is below a threshold, a link between the template blob and the target blob is created.

If one node is isolated we are prolonging the track. If both nodes are external we are chaining the two tracks. If one node is internal, we are increasing its degree above two, thereby catering for splitting and merging situations.

Connected components in G are identified and labelled. At this point the response of classification is taken into account, and the connected components composed only by bad tracks or by a single node are discarded.

4.3 Object recovery

At this stage, one connected component might represent more than one object, and one object can be associated to more that one component. This section describes the heuristics that have been devised in order to associate tracks and objects.

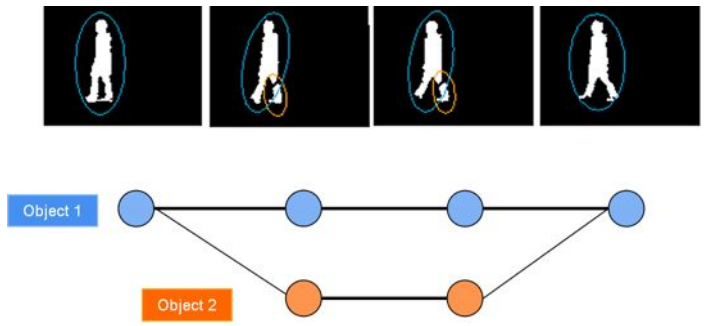

Fig. 6. Splitting object. When an object splits, SL matching (bold line) typically tracks the biggest part. The connected component is created by the edges added by template matching (thin line). The split section is composed by the two central frames, where two blobs represents the same object.

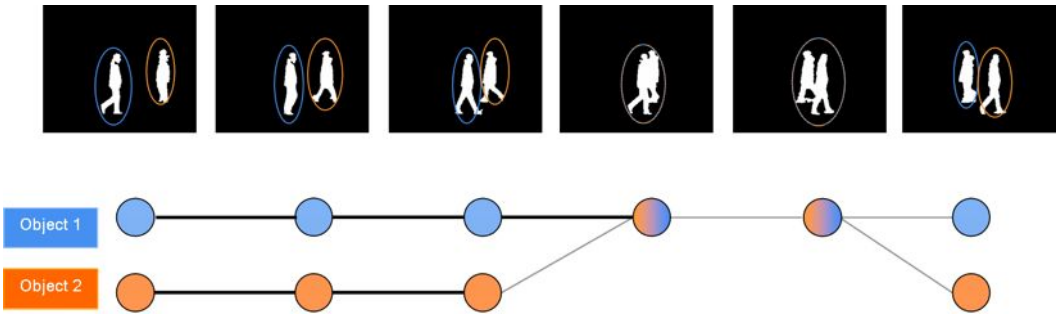

Fig. 7. Overlapping objects. When two objects merge, SL matching (bold line) typically fails in the merge section. The connected component is created by the edges added by template matching (thin line). The merge section is composed by the two frames where one blob represents two objects.

4.3.1 Single connected component representing multiple objects

The first step is to determine how many MOs are associated to each connected component. The rule we conceived says that a connected component represents as many MOs as the number of blobs of that component occurring in the majority of the layers. For example, if in most of the layers a certain connected component consists of two blobs, then that connected component represents two MOs (that have possibly merged at some point). Vice-versa, if in the majority of the layers a connected component consists of one blob, then it represents one MO (that have possibly split at some point). Therefore, frames where a split or a merge has occurred are detected as those where the number of blobs of a certain label is different from the number of objects represented by that connected component. Split sections and merge sections are marked to be subsequently dealt with.

In each frame of a split section, multiple nodes sharing the same label in a layer collapse into one node, thereby obtaining again a simple track (Figure 6).

As for the merge sections, we would like to obtain two separate tracks from

one connected component (Figure 7), but first we need to recover the identity of the objects across the merge section. To this end, we perform SL matching between the layers that bracket the merging section. Then, in each layer of the merge section, the nodes representing the union of two objects are replicated. In this way two separate tracks are generated, and a new label is created. Please note that the blobs belonging to the merge section of both tracks are the same and represents the union of two objects. This prevents from content-based editing such sequences, but does not affect the compression, if only with a negligible overhead. A worth pursuing improvement would be the accurate trimming of the two objects in the merging section, based on colour and motion segmentation, for example.

4.3.2 Single object represented by multiple tracks

This can occur if, for example, an object gets occluded by a large static element, as in Figure 8. In this case there is no way to connect the two tracks by matching from one layer to another, especially if the object is not visible for some frames. As far as the coding efficiency is concerned this is uninfluential, but for the content-based representation the objects' identity should be preserved.

Our solution is to analyse pairs of tracks that have at least one external node in the internal layers of the sequence. For each such pair, we try to establish a match. With the occlusion example in mind, it is clear that we cannot simply match the external nodes, as they are likely to be partially occluded, hence fairly dissimilar. A *typical* blob in a connected component is a blob whose area does not differ from the median area of its connected component for more than 5.2 MAD (i.e., it is an inlier, according to the X-84 rule). SL matching is carried out between the two typical blobs closest to the external nodes. If the matching is positive the two connected components are given the same label.

5 Results

In this section, we report some results obtained by applying our object segmentation technique to synthetic and real sequences. The real sequences were selected to set different challenges to our algorithm, so as to test each of its stages. The synthetic sequence was used in order to compare our results against a ground-truth segmentation.

In the first real experiment, we used the well known "Stefan" sequence (79 frames) depicting a tennis player during a match (Figure 9). The camera

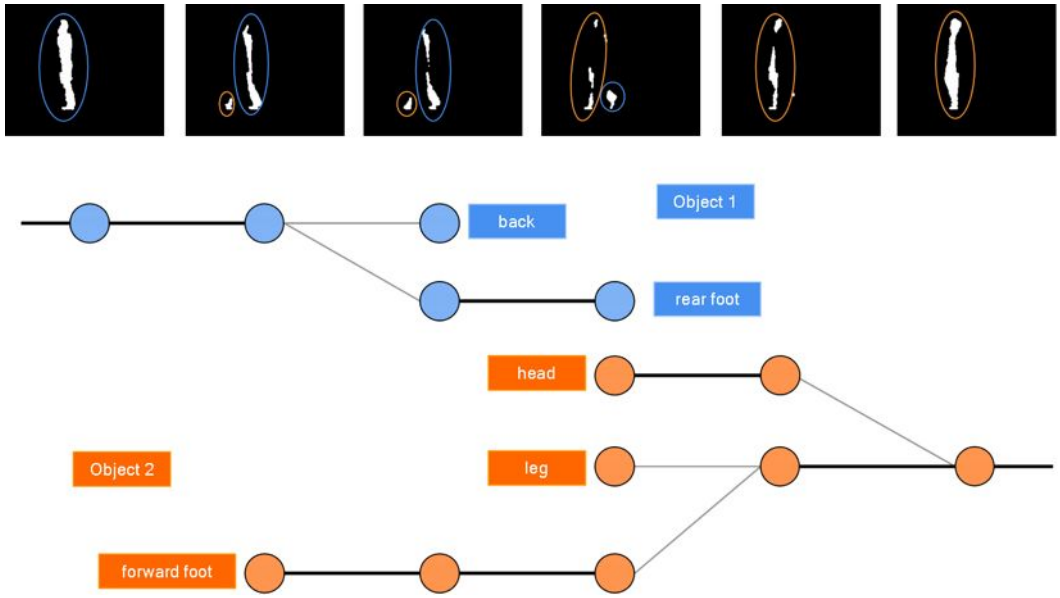

Fig. 8. Splitting and occlusion. When an object is severely occluded, two separate tracks are generated. SL matching (bold line) typically fails in the splitting section, which occurs near the end of both tracks. Connections are recovered by template matching (thin line).

follows the player with a panning motion and varying the focal length. In this case there is neither crossing nor occlusions, but this sequence is a good test-bed for the segmentation algorithm because of the articulated movements of the player, presence of motion blur and zooming.

The background mosaic and the MAD are shown in Figure 10. It is worth noting that pixels affected by the MOs are not significantly brighter than the others in the MAD image, evidence of the fact that MAD consistently estimates the variance of the background colour.

Fig. 9. Some selected frames from the "Stefan" sequence (top) and the result of the segmentation (bottom).

In the second real experiment we tested the case of splitting and occlusion. We acquired the "Pedone" sequence (47 frames) with a digital hand-held camera. While walking from right to left, a man passes behind another man who is standing still (Figure 11); the camera does a panning motion following the walking man. Before and after the occlusion the MO is fragmented in several

(a) (b)

Fig. 10. (a) Mosaic of the "Stefan" background, obtained after motion compensation and background modelling; (b) gray level visualisation of the MAD (values are normalised in [0,255]).

parts, nevertheless our technique can recover from over-fragmentation and recognises it as a single MO. As an example of the segmentation yielded by our technique, Figure 11 shows the MO extracted from the sequence. The mosaic of the background is shown in Figure 12.

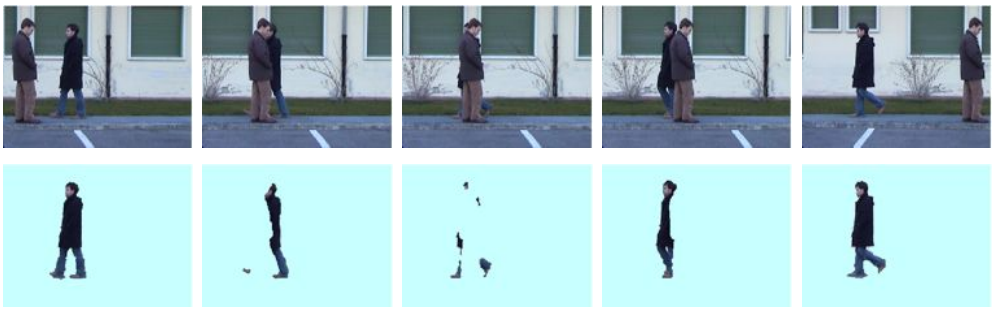

Fig. 11. Some selected frames from the "Pedone" sequence (top) and the result of the segmentation (bottom).

Fig. 12. Mosaic of the "Pedone" background.

In the third real experiment we tested the case of crossing moving objects. The "Granguardia" sequence (51 frames) depicts two persons entering the scene from the opposite sides and crossing in the middle (Figure 13). The camera does a panning motion, following the first man from left to right. Despite the two objects overlap in the image for a significant number of frames, our technique is able to track them trough the video shot. A sample of MOs extracted from the sequence is shown in Figure 13. The mosaic of the background is shown in Figure 14.

In order to perform a quantitative assessment – as in the real case the ground-truth segmentation is not available – we encoded the sequences in MPEG-4 (Main Profile) using our segmentation masks as shape descriptors and then we

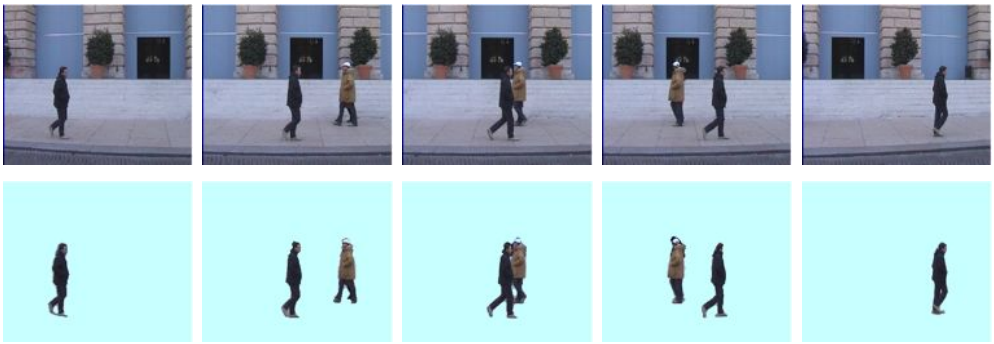

Fig. 13. Some selected frames from the "Granguardia" sequence (top) and the result of the segmentation (bottom).

Fig. 14. Mosaic of the "Granguardia" background.

computed the *peak signal-to-noise ratio* (PSNR) between the original sequence and the coded-decoded one (Figure 15).

As expected, MPEG-4 encoding using a sprite panorama for the background plus arbitrary-shape VOs for moving objects is very efficient: The dimension of the encoded video stream is about 0.5% of the original (uncompressed AVI).

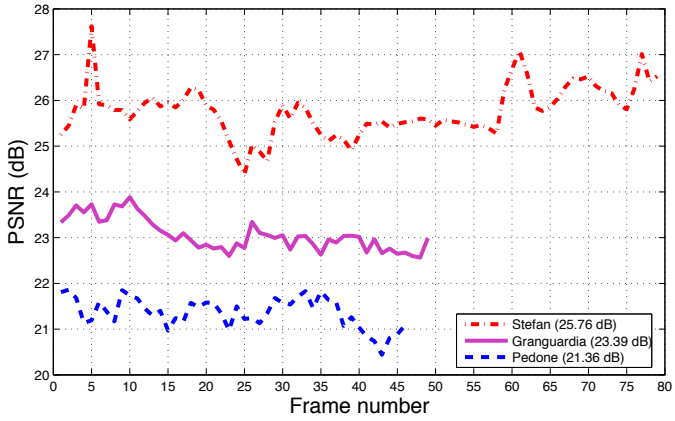

Fig. 15. PSNR for coded-decoded sequences (the mean PSNR is reported in the legend).

In the synthetic sequence (73 frames) we simulated the "Granguardia" sequence with two stylised human body models walking in front of an arcade (Figure 16). In this case, the ground-truth for the segmentation was

available and we could evaluate the percentage of mis-classified pixels (background/foreground) for each frame. The average percentage over the entire sequence before and after morphological filtering are respectively 1.3416% and 0.2893%.

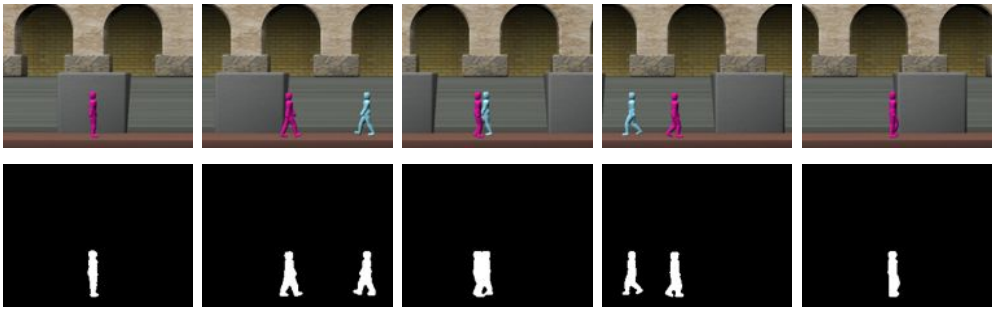

Fig. 16. Some selected frames from the synthetic sequence (top) and the resulting binary masks from segmentation.

As an example of content based manipulation, we pasted the MOs from "Granguardia" sequence onto a completely different background, as shown in Figure 17.

Fig. 17. Sample frames from edited "Granguardia", after the substitution of the background.

The content based representation allows many fancy effects to be implemented, limited only by the imagination. For example, Figure 18 shows a stroboscopic-like summary of the "Stefan" sequence obtained by pasting one frame every four onto the background mosaic, with the transparency decreasing with time.

Fig. 18. Stroboscopic-like summary of the "Stefan" sequence.

6 Conclusions

We presented a complete system which produces a content-based representation of a video shot, and, in particular, we addressed the problem of multiple object segmentation and tracking. This paper builds on a previous work [12], and improves both segmentation and tracking. Segmentation is posed as an outlier rejection problem and solved by applying the X-84 outlier rejection rule. Our region matching approach is a generalisation of Scott and Longuet-Higgins algorithm for feature matching [13,14], and it extends the classical nearest-neighbour data association scheme by implementing both the principle of proximity (in Mahalanobis distance) and the principle of exclusion. The proposed tracking technique is rather general, and can take into account occlusions between MOs, occlusions between a MO and a background object, MOs entering and leaving the scene at any point.

Our work can be extended in many ways. For example, transparency in the shape encoding could be introduced for a more realistic blending of the object with the background [21]; accurate trimming of the objects' silhouette in case of merging could also be considered.

Acknowledgements

Thanks to Mosè Bottacini for implementing the MPEG-4 (Main Profile) encoding of the sequences. The use of the code of the KLT tracker by S. Birchfield is here acknowledged.

References

[1] R. Brunelli, O. Mich, C. M. Modena, A survey on the automatic indexing of video data, Journal of Visual Communication and Image Representation 10 (1999) 78–112.

[2] R. Koenen, F. Pereira, L. Chiariglione, MPEG-4: Context and objectives, Signal Processing: Image Communications 9 (4) (1997) 295–304.

[3] J. Martinez, R. Koenen, F. Pereira, MPEG-7: The generic multimedia content description standard, part 1, Multimedia, IEEE 9 (2002) 78–87.

[4] D. S. Zhang, G. Lu, Segmentation of moving objects in image sequence: A review, Circuits, Systems and Signal Processing 20 (2) (2001) 143–183.

[5] P. L. Rosin, Thresholding for change detection, in: Proceedings of the Sixth International Conference on Computer Vision, IEEE Computer Society, 1998, p. 274.

[6] P. Giaccone, G. Jones, Segmentation of global motion using temporal probabilistic classification, in: British Machine Vision Conference, 1998, pp. 619–628.

[7] H. Sawhney, S. Ayer, Compact representations of videos through dominant and multiple motion estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 814–830.

[8] M. Irani, P. Anandan, J. Bergen, R. Kumar, S. Hsu, Efficient representations of video sequences and their applications, Signal processing: Image Communication 8 (4) (1996) 327–351.

[9] S. Park, J. Aggarwal, Segmentation and tracking of interacting human body parts under occlusion and shadowing, in: IEEE Workshop on Motion and Video Computing, 2002, pp. 105–111.

[10] I. Cohen, G. Medioni, Detecting and tracking moving objects in video surveillance, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999, pp. II:319–325.

[11] K. Shafique, M. Shah, A non-iterative greedy algorithm for multi-frame point correspondence, in: Proceedings of the IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 110–115.

[12] F. Odone, A. Fusiello, E. Trucco, Layered representation of a video shot with mosaicing, Pattern Analysis and Applications 5 (3) (2002) 296–305.

[13] G. Scott, H. Longuet-Higgins, An algorithm for associating the features of two images, in: Proceedings of the Royal Society of London B, Vol. 244, 1991, pp. 21–26.

[14] M. Pilu, A direct method for stereo correspondence based on singular value decomposition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 261–266.

[15] C. Tomasi, T. Kanade, Detection and tracking of point features, Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburg, PA (April 1991).

[16] K. Kanatani, N. Ohta, Accuracy bounds and optimal computation of homography for image mosaicing applications, in: International Conference on Computer Vision, Vol. 1, 1999, pp. 73–79.

[17] M. A. Fischler, R. C. Bolles, Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[18] R. Marzotto, A. Fusiello, V. Murino, High resolution video mosaicing with global alignment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. I, IEEE Computer Society, Whashington, D.C., 2004, pp. 692–698.

[19] F. Hampel, P. Rousseeuw, E. Ronchetti, W. Stahel, Robust Statistics: the Approach Based on Influence Functions, Wiley Series in probability and mathematical statistics, John Wiley & Sons, 1986.

[20] I. Cox, A review of statistical data association techniques for motion correspondence., International Journal of Computer Vision 10 (1) (1993) 53–66.

[21] M. Ruzon, C. Tomasi, Alpha estimation in natural images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 18–25.