EURASIP Journal on Applied Signal Processing 2004:6, 786–797

Interaction between High-Level and Low-Level Image Analysis for Semantic Video Object Extraction

Andrea Cavallaro
Multimedia and Vision Laboratory, Queen Mary University of London (QMUL), London E1 4NS, UK
Email: andrea.cavallaro@elec.qmul.ac.uk

Touradj Ebrahimi
Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland
Email: touradj.ebrahimi@epfl.ch

Received 21 December 2002; Revised 6 September 2003

The task of extracting a semantic video object is split into two subproblems, namely, object segmentation and region segmentation. Object segmentation relies on *a priori* assumptions, whereas region segmentation is data-driven and can be solved in an automatic manner. These two subproblems are not mutually independent, and they can benefit from interactions with each other. In this paper, a framework for such interaction is formulated. This representation scheme based on region segmentation and semantic segmentation is compatible with the view that image analysis and scene understanding problems can be decomposed into low-level and high-level tasks. Low-level tasks pertain to region-oriented processing, whereas the high-level tasks are closely related to object-level processing. This approach emulates the human visual system: what one "sees" in a scene depends on the scene itself (region segmentation) as well as on the cognitive task (semantic segmentation) at hand. The higher-level segmentation results in a partition corresponding to semantic video objects. Semantic video objects do not usually have invariant physical properties and the definition depends on the application. Hence, the definition incorporates complex domain-specific knowledge and is not easy to generalize. For the specific implementation used in this paper, motion is used as a clue to semantic information. In this framework, an automatic algorithm is presented for computing the semantic partition based on color change detection. The change detection strategy is designed to be immune to the sensor noise and local illumination variations. The lower-level segmentation identifies the partition corresponding to perceptually uniform regions. These regions are derived by clustering in an N-dimensional feature space, composed of static as well as dynamic image attributes. We propose an interaction mechanism between the semantic and the region partitions which allows to cope with multiple simultaneous objects. Experimental results show that the proposed method extracts semantic video objects with high spatial accuracy and temporal coherence.

Keywords and phrases: image analysis, video object, segmentation, change detection.

1. INTRODUCTION

One of the goals of image analysis is to extract meaningful entities from visual data. A meaningful entity in an image or an image sequence that corresponds to an object in the real world, such as a tree, a building, or a person. The ability to manipulate such entities in a video as if they were physical objects is a shift in the paradigm from pixel-based to content-based management of visual information [1, 2, 3]. In the old paradigm, a video sequence is characterized by a set of frames. In the new paradigm, the video sequence is composed of a set of meaningful entities. A wide variety of applications, ranging from video coding to video surveillance, and from virtual reality to video editing, benefit from this shift.

The new paradigm allows us to increase the interaction capability between the user and the visual data. In the pixel-based paradigm, only simple forms of interaction, such as fast forward and reverse, slow motion, are possible. The entity-oriented paradigm allows the interaction at object level, by manipulating entities in a video as if they were physical objects. For example, it becomes possible to copy an object from one video into another.

The extraction of the meaningful entities is the core of the new paradigm. In the following, we will refer to such meaningful entities as *semantic video objects*. A semantic video object is a collection of image pixels that corresponds to the projection of a real object in successive image planes of a video sequence. The meaning, that is, the *semantics*, may change according to the application. For example, in a building surveillance application, semantic video objects are people, whereas in a clothes shopping application, semantic video objects are the clothes of the person. Even this simple

example shows that defining semantic video objects is a complex and sometimes delicate task.

The process of identifying and tracking the collections of image pixels corresponding to meaningful entities is referred to as *semantic video object extraction*. The main requirement of this extraction process is *spatial accuracy*, that is, precise definition of the object boundary [4, 5]. The goal of the extraction process is to provide pixelwise accuracy. Another basic requirement for semantic video object extraction is *temporal coherence*. Temporal coherence can be seen as the property of maintaining the spatial accuracy in time [6, 7]. This property allows us to adapt the extraction to the temporal evolution of the projection of the object in successive images.

The paper is organized as follows. In Section 2, the need of an effective visual data representation is discussed. Section 3 describes how the semantic and region partitions are computed and introduces the interaction mechanism between low-level and high-level image analysis results. Experimental results are presented in Section 4, and in Section 5, we draw the conclusions.

2. VISUAL DATA REPRESENTATION

Digital images are traditionally represented by a set of unrelated pixels. Valuable information is often buried in such unstructured data. To make better use of images and image sequences, the visual information should be represented in a more structured form. This would facilitate operations such as browsing, manipulation, interaction, and analysis on visual data. Although the conversion into structured form is possible by manual processing, the high cost associated with this operation allows only a very small portion of the large collections of image data to be processed in this fashion. One intuitive solution to the problem of visual information management is content-based representation. Content-based representations encapsulate the visually meaningful portions of the image data. Such a representation is easier to understand and to manipulate both by computers and by humans than the traditional unstructured representation.

The visual data representation we use in this work mimics the human visual system and finds its origins in active vision [8, 9, 10, 11]. The principle of active vision states that humans do not just *see* a scene but *look* at it. Humans and primates do not scan a scene in raster fashion. Our visual attention tends to jump from one point to another. These jumps are called *saccades*. Yarbus [12] demonstrated that the saccadic pattern depends on the visual scene as well as on the cognitive task to be performed. We focus our visual attention according to the task at hand and the scene content. In order to attempt to emulate the human visual system to structure the visual data, we decompose the problem of extracting video objects into two stages: content-dependent and application-dependent. The *content-dependent* (or data-driven) stage exploits the redundancy of the video signal by identifying spatio-temporally homogeneous regions. The *application-dependent* stage implements the semantic model of a specific cognitive task. This semantic model corresponds to a specific human abstraction, which need not necessarily be characterized by perceptual uniformity.

We implement this decomposition by modeling an image or a video in terms of partitions. This partitional representation results in spatio-temporal structures in the iconic domain, as discussed in the next sections.

The application-dependent and the content-dependent stages are represented by two different partitions of the visual data, referred to as *semantic* and *region* partitions, respectively. This representation in the iconic domain allows us not only to organize the data in a more structured fashion, but also to describe the visual content efficiently.

3. PROPOSED METHOD

To maximize the benefits of the object-oriented paradigm described in Section 1, the semantic video objects need to be extracted in an automatic manner. To this end, a clear characterization of semantic video objects is required. Unfortunately, since semantic video objects are *human abstractions*, a unique definition does not exist. In addition, since semantic video objects cannot generally be characterized by simple homogeneity criteria[1] (e.g., uniform color or uniform motion), their extraction is a difficult and sometimes loose task.

For the specific implementation used in this paper, motion is used as a clue to semantic information. In this framework, an automatic algorithm is presented for computing the semantic partition based on color change detection. Two major noise components may be identified: the sensor noise and illumination variations. The change detection strategy is designed to be immune to these two components. The effect of sensor noise is mitigated by employing a probability-based test that adapts the change detection threshold locally. To handle local illumination variations, a knowledge-based postprocessing stage is added to regularize the results of the classification. The idea proposed is to exploit invariant color models to detect shadows. Then homogeneous regions are detected using a multifeature clustering approach. The feature space used here is composed of spatial and temporal features. Spatial features are color features from the perceptually uniform color space CIELab, and a measure of local texturedness based on variance. The temporal features used here are the displacement vectors from the dense optical flow computed via a differential technique. The selected clustering approach is based on fuzzy C-means, where a specific functional is minimized based on local and global feature reliability. Local reliability of both spatial and temporal features is estimated using the local spatial gradient. The estimation is based on the observation that the considered spatial features are more uncertain near edges, whereas the considered temporal features are more uncertain on uniform areas. Global reliability is estimated by considering the variance of the features in the entire image compared to the variance of the features in a region.

[1] This approach differs from many previous works that define *objects* as areas with homogeneous features such as color or motion.

The grouping of regions into objects is driven by a semantic interpretation of the scene, which depends on the specific application at hand. Region segmentation is automatic, generic, and application independent. In addition, the results can be improved by exploiting domain dependent information. Such use of domain dependent information is implemented through interactions with the semantic partition (Figure 1).

The details of the computation of the two partitions and their interactions are given in the following.

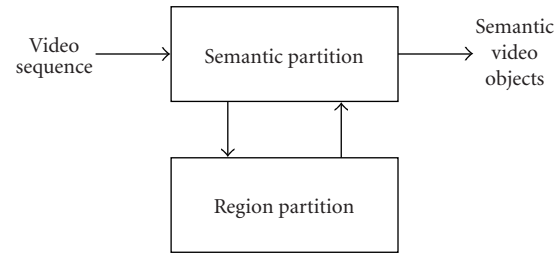


Figure 1: The interaction between low-level (region partition) and high-level (semantic partition) image analysis results is at the basis of the proposed method for semantic video object extraction.

3.1. Semantic partition

The semantic partition takes the cognitive task into account when modeling the video signal. The semantic (i.e., the meaning) is defined through a human abstraction. Consequently, the definition of the semantic partition depends on the task to be performed. The partition is then derived through *semantic segmentation*. In general, human intervention is needed to identify this partition because the definition of semantic objects depends on the application. However, for the classes of applications where meaningful objects are the moving objects, the semantic partition can be automatically computed. This is possible through color change detection. A change detection algorithm is ideally expected to extract the precise contours of objects moving in a video sequence (*spatial accuracy*). An accurate extraction is especially desired for applications such as video editing, where objects from one scene can be used to construct other artificial scenes, or computational visual surveillance, where the objects are analyzed to derive statistics about the scene.

The temporal changes identified by the color change detection process are here used to compute the semantic partition. However, temporal changes may be generated not only by moving objects, but also by noise components. The main sources of noise are illumination variations, camera noise, uncovered background, and texture similarity between objects and background. Since uncovered background is originated by applying change detector on consecutive frames, a frame representing the background is used instead (Figure 2). Such a frame is either a frame of the sequence without foreground objects or a reconstructed frame if the former is not available [13]. Camera noise and local illumination variations are then tackled by a change detector organized in two stages. First, sensor noise is eliminated in a classification stage. Then, local illumination variations (i.e., shadows) are eliminated in a postprocessing stage.

(a) (b)

Figure 2: (a) Sample frame from the test sequence Hall Monitor and (b) frame representing the background of the scene.

3.1.1. Classification

The classification stage takes into account the noise statistics in order to adapt the detection threshold to local information. A method that models the noise statistics based on a statistical decision rule is adopted. According to a model proposed by Aach [14], it is possible to assess the probability that the value at a given position in the image difference is due to noise instead of other causes. This procedure is based on the hypothesis that the additive noise affecting each image of the sequence respects a Gaussian distribution. It is also assumed that there is no correlation between the noise affecting successive frames of the sequence. These hypotheses are sufficiently realistic and extensively used in literature [15, 16, 17, 18]. The classification is performed according to a significance test after windowing the difference image. The dimension of the window can be chosen according to the application. In Figure 3, the influence of window size on the results of the classification by comparing the sizes of the window 3×3, 5×5, and 7×7 is presented. For the visualization of the results, a sample frame from the test sequence Hall Monitor is considered. The choice corresponding to Figure 3b, a window of 25 pixels, is a good compromise between the presence of halo artifacts, the correct detection of the object, and the extent of the window. This is the window size maximising the spatial accuracy and is therefore used in our experiments. The results of the probability-based classification with the selected window size are compared in Figure 4 with state-of-the-art classification methods so as to evaluate the difference in accuracy. The comparison is performed between the probability-based classification, the technique based on image ratioing presented in [19], and the edge-based classification presented in [20]. Among the three methods, the probability-based classification (Figure 4a) provides the most accurate results. A further discussion on the results is presented in Section 4.

(a) (b) (c)

Figure 3: Influence of the window size on the classification results. The dimensions of the window used in the analysis are (a) 3×3, (b) 5×5, and (c) 7×7.

(a) (b) (c)

Figure 4: Comparative results of change detection for frame 67 of the test sequence Hall Monitor: (a) probability-based classification, (b) image ratioing, and (c) edge-based classification.

3.1.2. Postprocessing

The postprocessing stage is based on the evaluation of heuristic rules which derive from the domain-specific knowledge of the problem. The physical knowledge about the spectral and geometrical properties of shadows can be used to define explicit criteria which are encoded in the form of rules. A bottom-up analysis organized in three levels is performed as described below.

Hypothesis generation

The presence of a shadow is first hypothesized based on some initial evidence. A candidate shadow region is assumed to correspond to a darker region than the corresponding illuminated region (the same area without the shadow). The color intensity of each pixel is compared to the color intensity of the corresponding pixel in the reference image. A pixel becomes a candidate shadow pixel if all color components are smaller than the corresponding pixel in the reference frame.

Accumulation of evidence

The hypothesized shadow region is then verified by checking its consistency with other additional hypotheses. The presence of a shadow does not alter the value of invariant color features. However, a material change is highly likely to modify their value. For this reason, the changes in the invariant color features $c_1c_2c_3$ [21] are analyzed to detect the presence of shadows. A second additional evidence about the existence of a shadow is derived from geometrical properties. This analysis is based on the position of the hypothesized shadows with respect to objects. The existence of the line separating the shadow pixels from the background pixels (the *shadow line*) is checked when the shadow is not detached, that is, an object is not floating, or the shadow is not projected on a wall. If a shadow is completely detached, the second hypothesis is not tested. In case a hypothesized shadow is fully included in an object, the shadow line is not present, and the hypothesis is then discarded.

Information integration

Finally, all the pieces of information are integrated to determine whether to reject the initial hypothesis.

The postprocessing step results in a spatio-temporal regularization of the classification results. The sample result presented in Figure 5 shows a comparison between the result after the classification and the result after the postprocessing. To improve the visualization, the binary change detection mask is superimposed on the original image.

3.2. Region partition

The semantic partition identifies the objects from the background and provides a mask defining the areas of the image containing the moving objects. Only the areas belonging to the semantic partition are considered by the following step, which takes into account the spatio-temporal properties of the pixels in the changed areas and extracts spatio-temporal

(a) (b)

Figure 5: Comparison of results from the test sequence Hall Monitor. The binary change detection mask is superimposed on the original image. The results of the classification (a) is refined by the postprocessing (b) to eliminate the effects of shadows.

homogeneous regions. Each object is processed separately and is decomposed in a set of nonoverlapping regions. The region partition Π_r is composed of homogeneous regions corresponding to perceptually uniform areas. The computation of this partition, referred to as *region segmentation*, is a low-level process that leads to a signal dependent (data-driven) partition.

The region partition identifies portions of the visual data characterized by significant homogeneity. These homogeneous *regions* are identified through segmentation. It is well known that segmentation is an ill-posed problem [9]: effective clustering of elements of the selected feature space is a challenging task that years of research have not succeeded in completely solving. To overcome the difficulties in achieving a robust segmentation, heuristics such as size of a region and maximum number of regions may be used. Such heuristics limit the generality of the approach.

To obtain an adaptive strategy based on perceptual similarity, we avoid imposing the above mentioned constraints but rather seeking an over-segmented result. This is followed by a region merging step.

Region segmentation operates on a decision space composed of multiple features, which are derived from transformations of the raw image data. We represent the feature space as

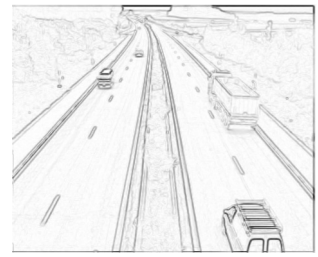
$$\mathbf{g}(x, y, n) = \big(g_1(x, y, n), g_2(x, y, n), \ldots, g_K(x, y, n)\big), \quad (1)$$

where K is the dimensionality of the feature space. The importance of a feature depends on its value with respect to other feature values at the same location, as well as to the values of the same feature at other locations in the image. Here we refer to these two phenomena as *interfeatures reliability* and *intrafeature reliability*, respectively. In addition to the feature space, we define a *reliability map* associated to each feature:

$$\mathbf{r}(x, y, n) = \big(r_1(x, y, n), r_2(x, y, n), \ldots, r_K(x, y, n)\big). \quad (2)$$

The reliability map allows the clustering algorithm to dynamically weight the features according to the visual content. The details of the proposed region segmentation algorithm are given in the following sections.

(a)



(b)

Figure 6: The reliability of the motion features is evaluated through the spatial gradient in the image: (a) test sequence Hall Monitor; (b) test sequence Highway. Dark pixels correspond to high values of reliability.

3.2.1. Spatial features

To characterize intraframe homogeneity, we consider color information and a texture measure. A perceptually linear color space Lab is appropriate, since it allows us to use a simple distance function. The reliability of color information is not uniform over the entire image. In fact, color values are unreliable at edges. On the other hand, color information is very useful in identifying uniform surfaces. Therefore, we use gradient information to determine the reliability of features. We first normalize the spatial gradient value to the range $[0, 1]$. If $n_g(x, y, n)$ is the normalized gradient, the reliability of color information $r_c(x, y, n)$ is given by the *sigmoid function*:

$$r_c(x, y, n) = \frac{1}{1 + e^{-\beta n_g(x,y,n)}}, \quad (3)$$

where β is the slope parameter. Low values correspond to shallow slopes, while higher values produce steeper slopes. Weighting color information with its reliability in the clustering algorithm improves the performance of the classification process.

Since color provides information at pixel level, we supplement color information with texture information based on a neighborhood $\mathcal{N}$ to better characterize spatial information. Many texture descriptors have been proposed in the literature, and a discussion on this topic is outside the scope of this paper. In this work, we use a simple measure of the *local texturedness*, namely, the variance of the color information over $\mathcal{N}$. To avoid using spurious values of local texture, we

do not evaluate this feature at edges. Thus, the reliability of the texture feature is zero at edges, and uniform elsewhere.

3.2.2. Temporal features

To characterize interframe homogeneity, we consider the horizontal and vertical components of the displacement vector at each pixel and their reliability. According to [22], the best performance for optical flow computation in terms of reliability can be obtained by the differential technique proposed in [23], and by the phase-based technique of [24]. We select the differential technique (see [23]) since it is gradient-based and therefore allows us to reuse the spatial gradient already computed for color reliability.

The results of motion estimation are noisy due to apparent motion. We mitigate the influence of this noise in two successive steps. First, we introduce a postprocessing (median filter) which reduces the noise in the dense optical flow field. Second, we associate a reliability measure to the motion feature, based on its spatial context. The reliability value derives from the fact that motion estimation performs poorly (i.e., it is not reliable) in uniform areas, whereas it shows better results in textured areas. Methods based on optical flow do not produce accurate contours (regions with homogeneous motion). For this reason, the reliability is given by the complement of the sigmoid function defined in (3). The motion reliability $r_m(x, y, n)$ is defined as follows:

$$r_m(x, y, n) = 1 - r_c(x, y, n). \tag{4}$$

Equation (4) allows the clustering algorithm to assign a lower weight to the motion feature in uniform areas than in those characterized by high contrast (edgeness). An example of motion reliability is reported in Figure 6.

3.2.3. Decision algorithm

The decision algorithm operates in two steps. First, a partitional algorithm provides over-segmented results, then a region merging step identifies the perceptually uniform regions. The partitional algorithm is a modified version of the fuzzy C-means algorithm described in [25]. Such modified version is spatially unconstrained so that to allow an improved flexibility when dealing with deformable objects.

The spatially unconstrained fuzzy C-means algorithm is an iterative process that operates as follows. After initialisation, the algorithm assigns each pixel to the closest cluster in the feature space (*classification*). For the computation of the distance, each cluster is represented by its centroid. The classification step results in a set of partitions in the image plane. The difference between two partitions is calculated as a point-to-point distance between the centroids of the respective partitions. This difference controls the number of iterations of the algorithm: the iterative process stops when the difference between the two consecutive partitions is smaller than a certain threshold (*cluster validation*).

The feature space includes information from different sources that are encoded with varying number of features. For example, three features are used for color and two for motion. We refer to such groups of similar features as *feature categories*. To avoid masking important information when computing the distance, we use separate distance measures $\mathcal{D}_f$ for each feature category. Since the results of the separate proximity measures will be fused together, it is desirable that $\mathcal{D}_f$ returns a normalized result, especially in the case of poorly scaled or highly correlated features. For this reason, we choose the Mahalanobis metric. To compute the proximity of the feature point $\mathbf{g}_j$ and the centroid $\mathbf{v}_i$, the Mahalanobis distance can be expressed as follows:

$$\mathcal{D}_f(\mathbf{g}_j, \mathbf{v}_i) = \sqrt{\sum_{s=1}^{K} \frac{\left(g_j^s - v_i^s\right)^2}{\sigma_s^2}}, \tag{5}$$

where σ_s^2 is the variance of the sth feature over the entire feature space. The complete point-to-point similarity measure between the $\mathbf{g}_j$ and $\mathbf{v}_i$ is obtained by fusing the distances computed within each category:

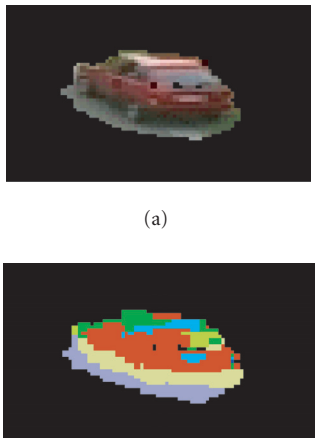
$$\mathcal{D}(\mathbf{g}_j, \mathbf{v}_i) = \frac{1}{F} \sum_{f=1}^{F} w_f \mathcal{D}_f(\mathbf{g}_j^s, \mathbf{v}_i^s), \tag{6}$$

where F is the number of feature categories and w_f the weight which accounts for the reliability of each feature category. The value of F may change from frame to frame and from cluster to cluster.

By projecting the result of the unconstrained partitional clustering back into the data space, we obtain a set of regions which may be composed of unconnected areas. Since this result depends on the predetermined number of clusters C, we adapt the result to the visual content as follows. Disjoint regions are identified by *connected component analysis* so as to form an over-segmented partition. This over-segmented result undergoes a *region merging* step which optimizes the partition by merging together the regions which present perceptually similar characteristics.

Each disjoint region $R_i(n)$ is represented by its own region descriptor $\mathbf{\Phi}_i(n)$. The region descriptor is composed of the same features used in clustering plus the position of the region. The position and the other values stored in the region descriptors are the mean values of the features in the homogeneous regions. We can represent the regions and the region descriptors by a *region adjacency graph*, where each node corresponds to a region, and edges joining nodes represent adjacency of regions. In our case, we explicitly represent the nodes with region descriptors.

Region merging fuses adjacent regions which present similar characteristics. A quality measure is established which allows the method to determine the quality of a merged region and to accept or discard a merging. The quality measure is based on the variance of the spatial and temporal features. Two adjacent regions are merged only if the variance in the resulting region is smaller than or equal to the largest variance of the two regions under test. Adjacent regions satisfying the above condition are iteratively fused together until no further mergings are accepted (Figure 7).



(a)

(b)

Figure 7: Example of region segmentation driven by the results of semantic segmentation: (a) area of interest defined by the semantic segmentation and (b) regions defined by the feature-based segmentation.

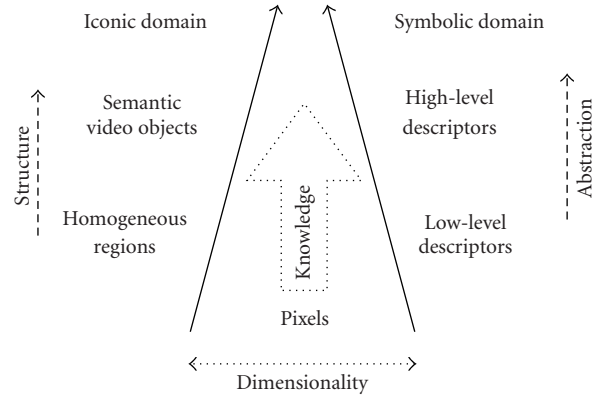


Figure 8: Different levels of visual content description.

3.2.4. Region descriptors

A region defines the topology of pixels that are homogeneous according to a specific criterion. The homogeneity criterion is defined with respect to one or more features in the dense feature space. The values of the features characterizing the region are distinctive of the region itself. We summarize these feature values in a vector, henceforth referred to as *region descriptor*. Region descriptors are the simplest way of representing the characteristics of regions. A region descriptor $\mathbf{\Phi}_i(n)$ can be represented as follows:

$$\mathbf{\Phi}_i(n) = \left(\phi_i^1(n), \phi_i^2(n), \ldots, \phi_i^{K_i^n}(n)\right)^T, \tag{7}$$

where K_i^n is the number of features used to describe region $R_i(n)$. $\mathbf{\Phi}_i(n)$ is an element of the region feature space. The number and the kind of features may change from region to region. Examples of features contributing to the region descriptor are the motion vector, the color, and so on. The selection of the features and their representation is dynamically adapted, based on low-level analysis and on the interaction between the region and semantic partitions.

3.3. Visual content description

The region and semantic partitions are organized in a partition tree. Such tree divides a set of objects into mutually exclusive and jointly exhaustive subsets. The coarsest partition level is the image itself (upper bound); at the finest partition level, every pixel is a distinct partition (lower bound).

The description is the result of a transformation from the iconic domain, constituted by pixels, regions, and objects, to the symbolic domain, consisting of text. This transformation allows us to compact and abstract the meaning buried in the visual information. The description encodes the values of the features extracted at the different stages of the hierarchical representation.

The hierarchy in the iconic domain leads naturally to several levels of abstraction of the description. The different levels of visual content description are depicted in Figure 8. The graphical comparison presented emphasizes the structural organization in the iconic domain as well as the abstraction in the symbolic domain. For the sake of simplicity, here we divide the description into two levels: low-level descriptors and high-level descriptors. The low-level descriptors are derived from the dense and the region feature spaces. The high-level descriptors are derived from the semantic and the image feature spaces.

The two main levels of image data representation defined by segmentation can be used to extract quantitative information from visual data. This corresponds to the transition from information to knowledge and represents a useful filtering operation not only for interpreting the visual information, but also as a form of data compression. The transition from *iconic domain* (pixels) to *symbolic domain* (objects) allows us to represent the information contained in the visual data very compactly.

3.4. Semantic and region partition interaction

The region and the semantic partitions can be improved through interaction with one another. The interaction is realized by allowing information to flow both ways between the two partitional representations so that the semantic information is used to improve the region segmentation result and vice versa.

An example of such interaction is the *combined region-semantic representation* of the visual data. This combined representation can be defined in two ways. One strategy is to define homogeneous regions from semantic objects. Information from the semantic partition is used to filter out the pixels of interest in the region partition. This approach, known as the *focus of attention* approach, corresponds to computing the region partition only on the elements defined by the semantic partition. The other way is to construct semantic objects from homogeneous regions. This

corresponds to projecting the information about the region partition onto the semantic partition.

We use both strategies to obtain a coherent temporal description of moving objects. Semantic video objects evolve in both shape and position as the video sequence progresses. Therefore, the semantic partition is updated over time by linking the visual information from frame to frame through tracking. The proposed approach is designed so as to consider first the object as an entity (semantic segmentation results) and then by tracking its parts (region segmentation results). The tracking mechanism is based on feedbacks between the semantic and the region partitions described in the previous sections. These interactions allow the tracking to cope with multiple simultaneous objects, motion of nonrigid objects, partial occlusions, and appearance and disappearance of objects. The block diagram of the proposed approach is depicted in Figure 9.

The correspondence of semantic objects in successive frames is achieved through the correspondence of objects' regions. Defining the tracking based on the parts of objects, that are identified by region segmentation, leads to a flexible technique that exploits the characteristics of the semantic video object tracking problem. Once the semantic partition is available for an image, it is automatically extended to the following image [26]. Given the semantic partition in the new frame and the region partition in the current frame, the proposed tracking procedure performs two different tasks. First, it defines a correspondence between the semantic objects in the current frame n and the semantic partition in the new frame $n + 1$. Second, it provides an effective initialization for the segmentation procedure of each object in the new frame $n + 1$. This initialization implicitly defines a preliminary correspondence between the regions in frame n and the regions in frame $n + 1$. This mechanism is described in Figure 10 and the results of its applications are shown in Section 4.

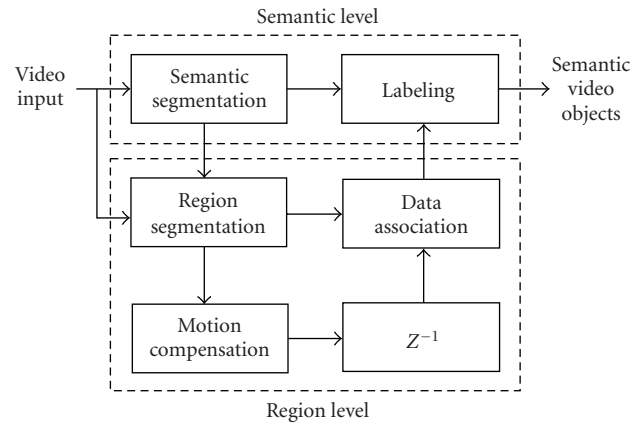


Figure 9: Flow diagram of the proposed semantic video object extraction mechanism based on interactions between the semantic and the region partitions. These interactions help the tracking process to cope with multiple simultaneous objects, partial occlusions, as well as appearance and disappearance of objects.

4. RESULTS

In this section, the results of the proposed algorithm for semantic video object extraction are discussed. The proposed algorithm receives as input a video, then extracts and follows each single video object over time. The results are organized as follows. Semantic video object extraction results are shown first. Then the behaviour of the algorithm for track management issues, such as splitting and merging, is discussed. Finally, the use of the proposed algorithm for content-based multimedia applications is discussed.

In Figures 11 and 12, the sequences Hall Monitor, from the MPEG-4 data set, and Group, from the European project art.live data set, are considered. The sequences are in CIF format (288×352 pixels) and the frame rate is 25 Hz. The results of the semantic segmentation are visualized by superposing the resulting change detection mask over the original sequence.

The method correctly identifies the contours of the extracted objects. In Figure 12b, it is possible to notice that an error occurred: a part of the trousers of the men are detected as background region. This is due to the fact that the color of the trousers and the color of the corresponding background region are similar. To overcome this problem, a model of each object could be introduced and updated over time. At each time, the extracted object can be compared to its model. This would allow to detect instances of a semantic video object which do not present time coherence, as in the case of part of background and moving objects presenting similar color characteristics.

Figure 13 shows examples of track management issues. In the first row, a splitting is reported. Figure 13a shows a zoom on frame 131 of the sequence Hall Monitor. The black line represents the contour of the semantic object detected by the change detector. The man and its case belong to the same semantic object. Figures 13b and 13c show a zoom on frame 135. In this frame, the man and the case belong to two different connected sets of pixels. The goal of tracking is to recognize that the case is coming from the same partition of the man (splitting). In case the splitting is not detected, the identificator for a new object label (coded with the white contour) is generated for the case (Figure 13b). Therefore, the history of the object is lost. Figure 13c show the successful tracking of the case: the case left by the man is detected as coming from the partition of the man in the previous frame. This is possible thanks to the semantic partition validation step. Region descriptors projection allows the tracking algorithm to detect that in two disconnected sets of pixels in the semantic partition, the same label appears.

Figure 13d shows a zoom on frame 110 of the sequence test *Highway*, from the MPEG-7 data set. The truck and the van are identified by two unconnected partitions color coded in white and black, respectively. Figures 13e and 13f show a zoom on frame 115. In this frame, the truck and the van belong to the same semantic partition (merging). In case a

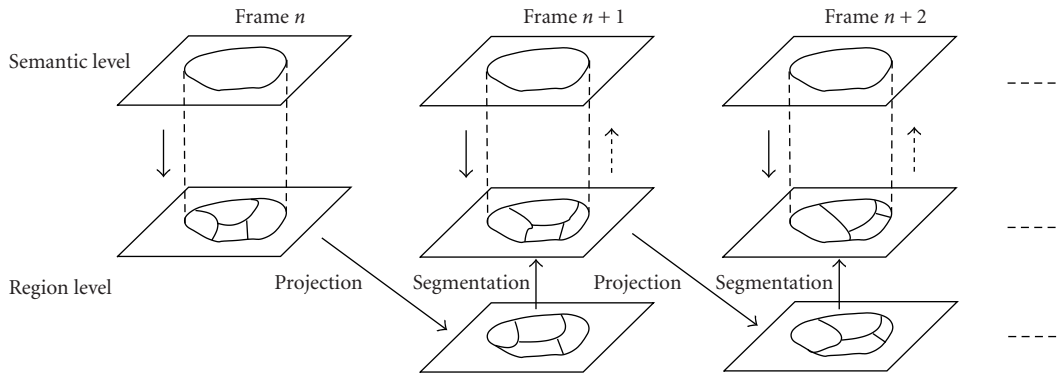


Figure 10: Semantic-region partition interaction in the case of one semantic video object. The semantic level provides the focus of attention and it is improved by the feedback from the region level.

Figure 11: Semantic video object extraction results for sample frames of the test sequence Hall Monitor.

Figure 12: Semantic video object extraction results for sample frames of the test sequence Group.

merging is not detected, the track of one of the two object is lost, thus invalidating the temporal representation and description of the semantic objects. In Figure 13e, the track of the van is lost and the two objects are identified by the same label, that of the truck (color-coded in black). As for the splitting described above, in the case of a merging as well, the semantic partition validation step generates a tentative correspondence that detects such an event. The connected set of pixels of the semantic partition receives from the region descriptors projection mechanism the labels of the two different objects. This condition allows to detect the merging. The semantic partition is therefore divided according to the information of the projection and the segmentation is performed separately in the two partitions. Therefore, the two objects can be isolated, thus allowing to access them separately over time.

The proposed semantic video object extraction algorithm can be used in a large variety of content-based applications ranging from video analysis to video coding and from video manipulation to interactive environments. In particular, the decomposition of the scene into meaningful objects can improve the coding performance over low-bandwidth channels. Object-based video compression schemes, such as MPEG-4, compress each object in the scene separately. For

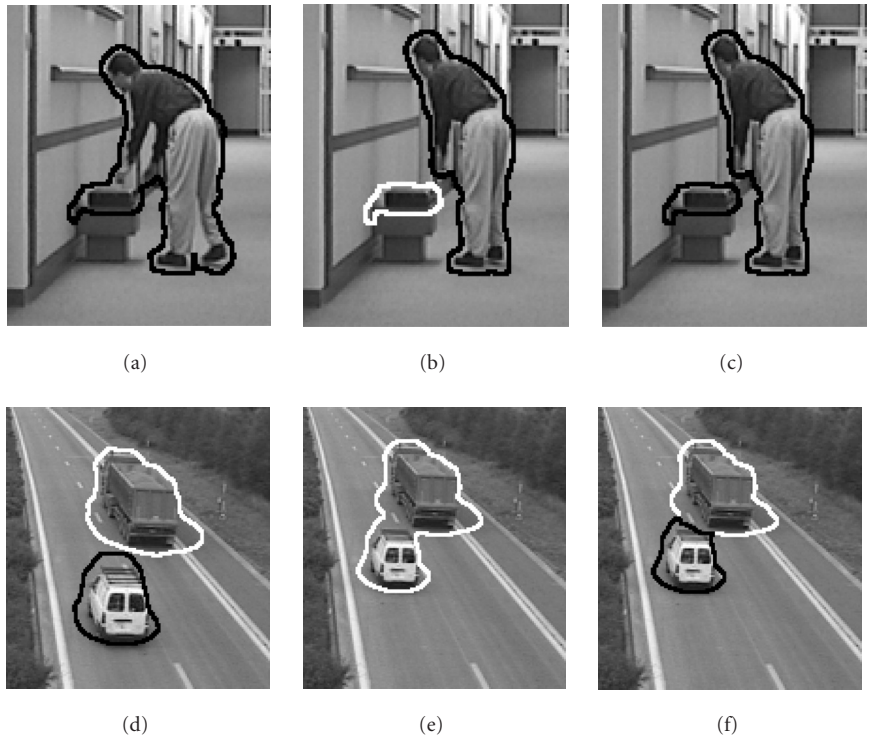


Figure 13: Example of track management issues: splitting of one object into two objects (first row) an merging of two objects into one semantic partition (second row). (a) Zoom on frame 131 of the sequence Hall Monitor, (b) zoom on frame 135, and (c) zoom on frame 135; (d) zoom on frame 110 of the sequence *Highway*, (e) zoom on frame 115, and (f) zoom on frame 115. The contour of the semantic object partition is shown before ((b) and (e)) and after ((c) and (f)) interaction with low-level regions in the proposed semantic video object extraction strategy.

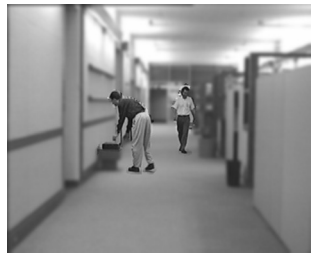
example, the video object corresponding to the background may be transmitted to the decoder only once. Then the video object corresponding to the foreground (moving objects) may be transmitted and added on top of it so as to update the scene. One advantage of this approach is the possibility of controlling the sequencing of objects: the video objects may be encoded with different degree of compression, thus allowing a better granularity for the areas in the video that are of more interest to the viewer. Moreover, objects may be decoded in their order of priority, and the relevant content can be viewed without having to reconstruct the entire image. Another advantage is the possibility of using a simplified background so as to enhance the moving objects (Figure 14a). Finally, the background can be selectively blurred during the encoding process in order to achieve an overall reduction of the required bit rate (Figure 14b). This corresponds to the use of the semantic object as region of interest.

5. CONCLUSIONS

The shift from frame-based to object-based image analysis has led to an important challenge: the extraction of semantic video objects. This paper has discussed the problem of segmenting, tracking, and describing such video objects. A general representation for modeling video based on semantics has been proposed, and its validity has been demonstrated through specific implementations. This representation of visual information can be used in a wide range of applications such as object-based video coding, computer vision, scene understanding, and content-based indexing and retrieval.

The essence of this representation resides in the distinction between the notions of homogeneous regions versus semantic objects. Based on this distinction, the task of semantic video object extraction has been split into two subtasks. One task is fairly objective and aims at identifying areas (i.e., regions) of the image which are homogeneous according to some quantitative criteria such as color, texture, motion, or some combination of these features. Such an area is not required to have any intrinsic semantic meaning. The identification of the appropriate homogeneity criteria and the subsequent extraction of the regions is performed by the system in a completely automatic way. The second task takes the characteristics of the specific implementation into account and aims at identifying areas of the image that correspond to semantic objects. In general, unlike the above-mentioned regions, semantic objects lack global coherence in color, texture, and sometimes even motion. The two subtasks generate two kinds of partitions, namely, the semantic and the region partition that have been generated by two different types of

(a)



(b)

Figure 14: Example of use of the proposed semantic video object extraction algorithm. (a) The extraction of moving objects allows one to reconstruct a scene with a simplified background, thus enhancing the visibility of the moving objects. (b) Example of use of semantic video object extraction for preprocessed frame before coding: the background information is blurred thus requiring less bandwidth while still retaining essential contextual information.

segmentation. Each kind of segmentation exploits the specific nature of the problem to obtain a partition that groups similar data elements together in the selected feature space.

While the advantages of the proposed video object extraction algorithm are evident by the results shown in Section 4, there are several interesting questions that remain to be investigated. Of primary interest is a change detection mechanism which could provide high spatial accuracy in case of global illumination variations. We are currently evaluating the use of edges and photometric invariant color features to this end. Moreover, even if the visual data representation of Section 2 is generic and can deal with static as well as moving cameras, in the implementation of Section 3, we have assumed that the camera is fixed. This scenario is valid for many surveillance type applications. One natural extension is to deal with moving camera sequences by integrating the global motion information. Furthermore, depending on the constraints of the application, such as acceptable levels of delay and complexity, each specific component of the architecture may be replaced with a more adequate one without changing the general approach so as to optimize such modules for each specific application. Finally, the modularity of the system allows us to add other features. This flexibility also allows us to integrate information derived from different sensors, such as an infrared camera, by simply adding the appropriate modules to the same existing structure and other data fusion modules.

ACKNOWLEDGMENTS

This work was supported in part by the European projects ACTS, MODEST, and IST art.live. The authors would like to thank Eduard Solanas Vilar, Elena Salvador, and Olivier Steiger for their contribution to the development of the software.

REFERENCES

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image coding techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–574, 1985.

[2] H. G. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, vol. 1, no. 2, pp. 117–138, 1989.

[3] P. Correia and F. Pereira, "The role of analysis in content-based video coding and indexing," *Signal Processing*, vol. 66, no. 2, pp. 125–142, 1998.

[4] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577–586, 2002.

[5] Y. Tsaig and A. Averbuch, "Automatic segmentation of moving objects in video sequences: a region labeling approach," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 597–612, 2002.

[6] H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75–89, 2002.

[7] C. Kim and J.-N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 122–129, 2002.

[8] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[9] D. Marr, *Vision*, W. H. Freeman, San Francisco, Calif, USA, 1982.

[10] S. Edelman, *Representation and Recognition in Vision*, M.I.T. Press, Cambridge, Mass, USA, 1999.

[11] D. H. Hubel, *Eye, Brain and Vision*, W. H. Freeman, New York, NY, USA, 1995.

[12] A. L. Yarbus, *Eye Movements and Vision*, Plenum Press, New York, NY, USA, 1967.

[13] A. Cavallaro and T. Ebrahimi, "Video object extraction based on adaptive background and statistical change detection," in *Visual Communications and Image Processing 2001*, vol. 4310 of *Proceedings of SPIE*, pp. 465–475, San Jose, Calif, USA, January 2001.

[14] T. Aach, A. Kaup, and R. Mester, "Statistical model-based change detection in moving video," *Signal Processing*, vol. 31, no. 2, pp. 165–180, 1993.

[15] M. Hötter, R. Mester, and F. Müller, "Detection and description of moving objects by stochastic modelling and analysis of complex scenes," *Signal Processing: Image Communication*, vol. 8, no. 4, pp. 281–293, 1996.

[16] R. Mech and M. Wollborn, "A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera," *Signal Processing*, vol. 66, no. 2, pp. 203–217, 1998.

[17] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, no. 2, pp. 219–232, 1998.

[18] M. Kim, J. G. Choi, D. Kim, et al., "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1216–1226, 1999.

[19] E. Durucan and T. Ebrahimi, "Robust and illumination invariant change detection based on linear dependence for surveillance applications," in *Proc. 10th European Signal Processing Conference*, pp. 1041–1044, Tampere, Finland, September 2000.

[20] A. Cavallaro and T. Ebrahimi, "Change detection based on color edges," in *Proc. IEEE Int. Symp. Circuits and Systems*, Sydney, Australia, May 2001.

[21] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[22] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[23] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understanding Workshop*, pp. 121–130, Vancouver, April 1981.

[24] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin, "Phase-based disparity measurement," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 198–210, 1991.

[25] R. Castagno, A. Cavallaro, F. Ziliani, and T. Ebrahimi, "Automatic and interactive segmentation of video sequences," in *Non Linear Model-based Image/Video Processing and Analysis*, I. Pitas and C. Kotropoulos, Eds., John Wiley & Sons, New York, NY, USA, April 2001.

[26] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Multiple video object tracking in complex scenes," in *Proc. 10th ACM International Conference on Multimedia*, pp. 523–532, Juan les Pins, France, December 2002.

Andrea Cavallaro received his M.S. (Honors) in electrical engineering from the University of Trieste, Italy, in 1996, and his Ph.D. in electrical engineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 2002. In 1996 and 1998, he served as a Research Consultant at the Image Processing Laboratory, University of Trieste, Italy, working on compression algorithms for very low bitrate video coding and on digital image sequence de-interlacing. In 1997 he served in the Italian Army as a Lieutenant at the 33rd Electronic Warfare Battalion in Treviso, Italy. From 1998 to 2003 he was a Research Assistant at the Signal Processing Laboratory of the Swiss Federal Institute of Technology (EPFL). Since 2003, he is a Lecturer at Queen Mary University of London (QMUL). His main research interests are image analysis, video compression, and visual information description. Dr. Cavallaro was a Member of the organizing committee of the 2002 IEEE Conference on Multimedia and Expo, Member of the Technical Committee of the 2003 SPIE VCIP conference, 2004 IEEE ICME, and 2004 IEEE ICIP. He organized the special session on object-based video at the 2003 Visual Communication and Image Processing Conference. He is author of more than 25 papers, including 3 book chapters.

Touradj Ebrahimi received his M.S. and Ph.D., both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1989 and 1992 respectively. In 1993, he was a Research Engineer at the Corporate Research Laboratories, Sony Corporation, Tokyo. In 1994, he served as a Research Consultant at AT&T Bell Laboratories. He is currently Professor at EPFL, where besides teaching, he is involved in various aspects of visual information processing and coding for multimedia applications. He is also the Head of the Swiss delegation to MPEG, JPEG, and SC29, and acts as the Chairman of Advisory Group on Management in SC29. In 2002, he founded Emitall SA, an R&D and consulting company in the area of electronic media innovations. He is or has been Associate Editor with various IEEE, SPIE, and EURASIP Journals. His research interests include still, moving, and 3D image processing and coding, visual information security (rights protection, watermarking, authentication, data integrity, steganography), new media, and human computer interfaces (smart vision, brain computer interface). He is the author or the coauthor of more than 100 research publications, and holds 10 patents. Prof. Ebrahimi is a Member of IEEE, SPIE, ACM, and IS&T.