A Probabilistic Framework of Selecting Effective Key-Frames for Video Browsing and Indexing [1] [2]

Riad Hammoud and Roger Mohr
MOVI-INRIA Rhône-Alpes and GRAVIR-CNRS
655 avenue de l'Europe, 38330 Montbonnot Saint Martin, FRANCE
riad.hammoud@inrialpes.fr

International workshop on **R***eal-Time* **I***mage* **S***equence* **A***nalysis, pages 79-88*
August 2000

Abstract: To represent effectively the video content, for browsing, indexing and video skimming, the most characteristic frames (called key-frames) should be extracted from given shots. This paper, briefly reviews and evaluates the existing approaches of key-frames extraction; and then introduces a framework of selecting effective key-frames using an unsupervised clustering method. The mixture of Gaussians is used to model the temporal variation of the feature vectors of all frames in the shot. As a result, the feature-based representation of the shot is partitioned into several clusters. From each obtained cluster, firstly the closest frame to the median of its frames is selected as a reference key-frame. Then depending on the variation in time and appearance of the cluster content against the reference key-frame multiple frames can be extracted to represent effectively the cluster. The number of clusters is determined automatically by the Bayes Information Criterion. Experimental results on tracked objects in a real-world video stream are presented which illustrate the performance of the proposed technique.

1 Introduction and motivation

As the amount of video data grows rapidly, the ability to manipulate it efficiently becomes of greater importance, for the purpose of selection of appropriate elements of information [6]. The selection or extraction of limited and meaningful informations is a way to resolve a set of challenging problems for recently emerging multimedia applications: video browsing and navigation, content-based indexing, video summarization and trailers, storage and transmission bandwidth of digitized video information [14] [9].

The access to video is still a hard task due to video's length and unstructured format. Video abstraction and summarization techniques are needed to solve this difficulty. Shot boundary detection and key-frame extraction are two bases for abstraction and summarization techniques [15] [1] [11].

A *shot* is defined as an unbroken sequence of frames recorded from a single camera, which forms the building block of a video. The purpose of shot boundary detection is to segment the video stream into multiple shots. There exist many already effective shot boundary techniques [2].

Beyond the shot level an abstraction level could be constructed by mapping the entire shot to a small number of representative frames, called *key-frames* [14]. Indeed, an index

[1] `This work is supported by Alcatel CRC Grant Alcatel-Inria No. 198G098.`
[2] Demos of this work are available at `http://www.inrialpes.fr/movi/people/Hammoud/`

may be constructed from key-frames, and retrieval may be directed at key-frames, which can subsequently be displayed for browsing purposes.

This paper focuses on the key-frame extraction techniques. There exist many different approaches to extract key-frames [14] [15] [1]. However, they can not effectively capture the major visual content, and/or are not friendly-user where a set of parameters must be adjusted by the user, and/or also are computationally expensive.

In this paper, a new strategy to extract the most characteristic key-frames is proposed. The main idea is to cluster similar or redundant views within the shot together. The clusters are approximated by a mixture of Gaussians using the standard Expectation-Maximization (EM) algorithm [4]. Here, the estimation is performed in the color histogram feature space. The Bayes Information Criterion [12] is used to chose the appropriate number of clusters (i.e. the number of key-frames) for each shot differently, depending on its complexity. From each obtained cluster, firstly the closest frame to the median of its frames is selected as a reference key-frame. Then depending on the variation in time and appearance of the cluster content against the reference key-frame multiple frames can be extracted to represent effectively the cluster. A temporal filter is applied on the set of all selected key-frames in order to eliminate the overlapping case between constructed clusters of frames. Using the proposed framework only sufficient separated frames in time and appearance are kept. The selection of key-frames is fully automatic, no parameters to be adjusted by the user.

The organization of this paper is as follows. Sections 2 and 3 review and evaluate respectively some relevant approaches to the present work. Section 4.1 details the clustering strategy and section 4.2 describes the algorithm to extract key-frames. In this work the key-frames are extracted for only browsing purposes since key-frames summarize the content of a shot [9]. In section 5 experimental results on different tracked objects in a real-world video sequence are presented. The video sequence has been already segmented into shots [3] and moving objects are localized and tracked in shots [5]. These experiments demonstrate the performance of the proposed technique. A short discussion and concluding remarks are given in sections 5.2 and 6 respectively.

2 Related work

Many research effort have been given in the area of key frame extraction [14] [15] [1] [13]. They could be regrouped in three following categories.

1. *Shot boundary based approach.* O'connor et al. use either of the first, the middle or the last frame of the shot as the shot's key frames [11].

2. *Motion analysis based approach.* Wolf proposes a motion based approach to key-frame extraction [13]. He first computes the optical flow for each frame and then computes a simple motion metric based on the optical flow. Finally he analyzes the metric as a function of time to select key-frames at the local minima of motion.

3. *Visual content based approach.*

- Zhang et al. propose to use color and motion features independently to extract key frames [14]. The similarity between the current frame and the last key-frame is identified in each feature space by a thresholding technique.
- Motivated by the same observation as Wolf's and Zhang Avrithis et al. combine the color and motion features in a fuzzy feature vector [1]. The trajectory of feature vectors of all frames of a given shot is analyzed firstly. Then the key-frames are selected on the curve points: the local minima and maxima of the magnitude of the second derivative on the initial trajectory, in the discrete case.
- Zhuang et al. propose an adaptive key frame extraction using a linear clustering technique to regroup similar frames together [15]. The similarity between images of the same shot is computed in the 128-dimensional Hue-Saturation color histogram space. Based on a predefined threshold of similarity for each video sequence, the number of clusters is determined. After that, an arbitrary point of each cluster is selected as a key-frame. Only clusters of proportions greater than a predefined threshold are represented.

3 Evaluation of existing techniques

The approach of O'connor is the easy way to extract key frames. However, it does not capture the visual content of the video shot. The methods of Avrithis and Wolf give interesting results. However they are computationally expensive due to their analysis of motion, and their underlying assumption of local minima does not work very well in the case of constant variation of the feature vectors. The methods of Zhuang and Zhang are relatively fast. However, they are very sensible to the choice of the threshold of similarity. As a result, the number of selected key-frames is very variable. The adjustment of the threshold parameter represents a challenging problem for the user of these methods.

Next section details the theoretic part of the proposed framework to automatically select the effective key-frames for a given shot. Our approach uses an unsupervised clustering algorithm to group similar frames within a shot together. The Gaussian mixture density is used to model the temporal variation of color histograms in the RGB color space. In order to select automatically the number of appropriate components (clusters) the Bayes Information criterion is performed.

4 Probabilistic framework for shot abstraction

Assume that temporal video segmentation into shots was already performed. Then, each frame within a shot a is characterized by a vector of measurements called *feature*. Each feature is represented by a single *point* or *individual* in the d-dimensional feature space, where its coordinates are the values of the feature vector.

Now, for a given shot of n frames (or a tracked object of n occurrences), n points in the d-dimensional space describe the temporal variation (trajectory) of its contents. For example, figure 1 illustrates the temporal variation of the tracked "Ford Car" within a video shot of 66 frames. Some images of this shot are depicted in figure 3. Each point represents a RGB histogram computed on an occurrence of the tracked car in the shot. The 64-dimensional space of this data was already reduced performing the Principal Components Analysis (PCA). In the current framework the method of [5] was used to track non-rigid objects.

In the following both clustering strategy to classify similar frames together, and key-frames extraction algorithm to realize the abstraction level are described.

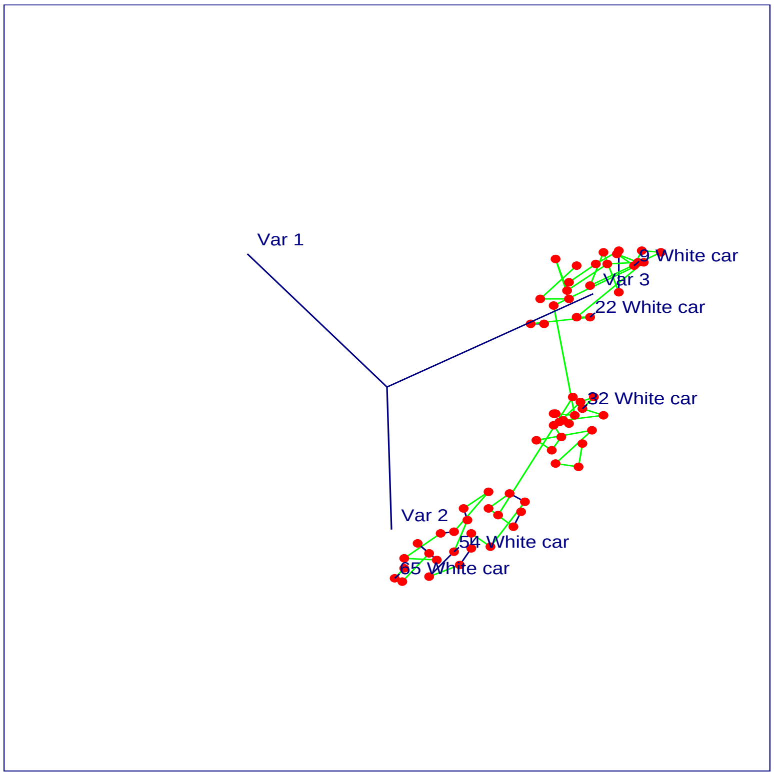

Figure 1: Illustration of the content-based variation of the tracked "Ford Car" within the shot of figure 3, in the 3-principal components of the RGB histogram space. Some labels of points are shown (e.g the corresponding number of frames).

4.1 Clustering by Gaussian mixture densities

Again, assume that a video shot consisting of n images has been selected. Let us denote by y_i the feature vector of dimension d that characterizes the ith frame, and by $Y = \{y_i; i = 1...n\}$ the set of feature vectors collected for all frames of the shot. The distribution of Y is modeled as a joint probability density function, $f(y \mid Y, \theta)$ where θ is the set of parameters for the model f. We assume that f can be approximated as a J-component mixture of Gaussians [10]:

$$f(y|\theta) = \sum_{j=1}^{J} p_j \varphi(y|\alpha) \tag{1}$$

where the p_j's are the mixing proportions and φ is a density function parameterized by the center and the covariance matrix, $\alpha = (\mu, \Sigma)$. In the following, we denote $\theta_j = (p_j, \mu_j, \Sigma_j)$, for $j = 1, \ldots, J$ the parameters to be estimated.

Each cluster approximated by a Gaussian component of the mixture groups a set of similar points (i.e. similar frames) in the feature space. Thus a transition from one Gaussian component to another indicates a significant temporal variation within the shot.

Parameters Estimation. Gaussian mixture density estimation is performed in a semi-parametric way so that the number of components scales with the complexity of the data and not with the size of the data set. The density estimation procedure is a missing data estimation problem to which the EM algorithm [4] can be applied. The type of Gaussian mixture model to be used (see next paragraph) has to be fixed and also the number of components in the mixture. If the number of components is one the estimation procedure is a standard computation (step M), otherwise the expectation (E) and maximization (M) steps are executed alternately until the log-likelihood of θ stabilizes or the maximum number of iterations is reached.

Let $\mathbf{y} = \left\{y_i;\ 1 \leq i \leq n \ and\ y_i \in I\!R^d\right\}$ be the observed sample from the mixture distribution $f(y|\theta)$. We assume that the component from which each y_i arises is unknown, so that the missing data are the labels c_i $(i = 1, \ldots, n)$. We have $c_i = j$ if and only if j is the mixture component from which y_i arises. Let $\mathbf{c} = (c_1, \ldots, c_n)$ denote the missing data, $\mathbf{c} \in B^n$, where $B = \{1, \ldots, J\}$. The complete sample is $\mathbf{x} = (x_1, \ldots, x_n)$ with $x_i = (y_i, c_i)$. The complete log-likelihood is

$$L(\theta, \mathbf{x}) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{J} p_j \varphi(x_i | \mu_j, \Sigma_j) \right\}. \tag{2}$$

The EM algorithm at iteration "m" is summarized as follow :

Step-E : For $i = 1, \ldots, n$ and $j = 1, \ldots, J$ compute the conditional probability, given $\mathbf{y}$, that y_i arises from the mixture component with density $\varphi(.|\mu_j^m, \Sigma_j^m)$ and mixing proportion p_j^m

$$t_{ij}(\theta^m) = \frac{p_j^m \varphi(x_i, \mu_j^m, \Sigma_j^m)}{\displaystyle\sum_{\ell=1}^{J} p_\ell^m \varphi(x_i | \mu_\ell^m, \Sigma_\ell^m)}. \tag{3}$$

Step-M : Maximize the log-likelihood conditionally on t_{ij}^m. Indeed, in the case of a general Gaussian model we get for θ^{m+1}

$$p_j^{m+1} = \frac{1}{n}\sum_{i=1}^{n} t_{ij}(\theta^m) \; ; \; \mu_j^{m+1} = \frac{\sum_{i=1}^{n} t_{ij}(\theta^m)\, y_i}{\sum_{i=1}^{n} t_{ij}(\theta^m)}$$

$$\Sigma_j^{m+1} = \frac{\sum_{i=1}^{n} t_{ij}(\theta^m)(y_i - \mu_j^{m+1})(y_i - \mu_j^{m+1})^{\mathsf{T}}}{\sum_{i=1}^{n} t_{ij}(\theta^m)}. \tag{4}$$

At each iteration, the following properties hold: For $i = 1, \ldots, n$

$$\sum_{j=1}^{J} t_{ij}(\theta^m) = 1 \text{ and } \sum_{j=1}^{J} p_j^m = 1. \tag{5}$$

More details on the EM algorithm could be found in [4]. Initialization of the clusters is done randomly. In order to limit dependence on the initial position, the algorithm is run several times (10 times in our experiments) and the best solution is kept.

Gaussian models. Gaussian mixtures are sufficiently general to model arbitrarily complex, non-linear distribution accurately given enough data [4]. When the data is limited, i.e. the number of frames of a shot is small, the method should be constrained to provide better conditioning for the estimation. For these reasons and in order to make the method fast some constraints are added on the covariance parameter. In a previous work we have described these Gaussian models and their application [8]. These models are basically introduced in [4].

Choosing models and mixture components' number. To avoid a hand-picked number of Gaussians in the mixture, i.e. the number of clusters and then the number of key-frames to be selected, the Bayes Information Criterion (BIC) [12] is used to determine the best probability density representation (appropriate Gaussian model and number of components). It is an approach based on a measure that determines the best balance between the number of parameters used and the performance achieved in classification. It minimizes the following criterion:

$$BIC(M) = -2L_M + Q_M ln(n) \tag{6}$$

where L_M is the maximized log-likelihood of the Gaussian model M and Q_M is its number of free parameters.

4.2 Key-frames extraction

A few images called "key-frames" can summarize the visual content of a video shot. By definition, a key-frame is an existing frame within the shot which represents a set of redundant similar frames (or views of objects). In addition, two key frames should be visually different.

This section details the extraction of key-frames algorithm for a given shot. As a result of the first part of the approach, a set of clusters are identified. Each cluster is characterized by its center (mass of the distribution), its covariance matrix (dispersion around the center) and the number of individuals belong to it.

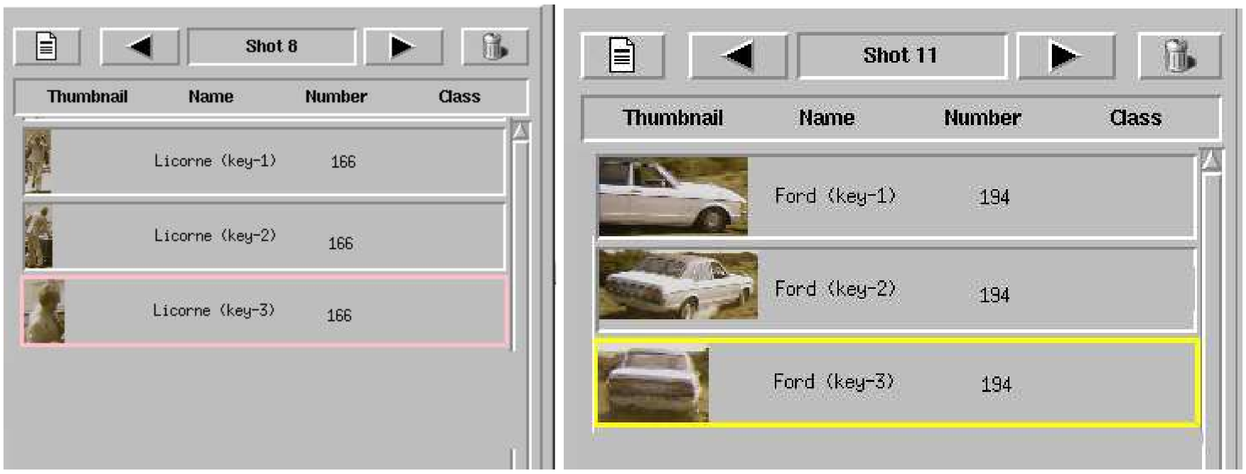

Figure 2: *Key frames browsing interface*

The algorithm to extract key-frames from a given shot is of two stages:

- Perform the following procedure on each cluster C^k with $k = 1..K$ and K denote the number of constructed clusters of frames.

  1- Compute the *median frame*, F_m^k, for the set of frames $\{F_i^k; i = 1..n_k\}$ belong to the cluster C^k, n_k represents the proportion of cluster C^k.

  2- Select as a *reference key-frame*, F_r^k, the closest frame to F_m^k.

  3- For each frame, F_i^k, belongs to C^k, check **(a)** if the temporal distance between it and the reference frames is greater than a predefined temporal threshold. If this condition is verified then check **(b)** if the similarity distance between it and the reference frames is greater than a predefined similarity threshold. If these two conditions are verified add this frame F_i^k to the set of *reference key-frames*. The Mahalanobis distance, $d_M(F_i^k, F_r^k) = (F_i^k - F_r^k)\Sigma_k^{-1}(F_i^k - F_r^k)^t$, is used here to compute the similarity between feature vectors of two frames.

- Merge the set of reference key-frames obtained for all clusters of a shot. From this set of frames keep only the key-frames which verify the two conditions (a) and (b) listed on the above procedure.

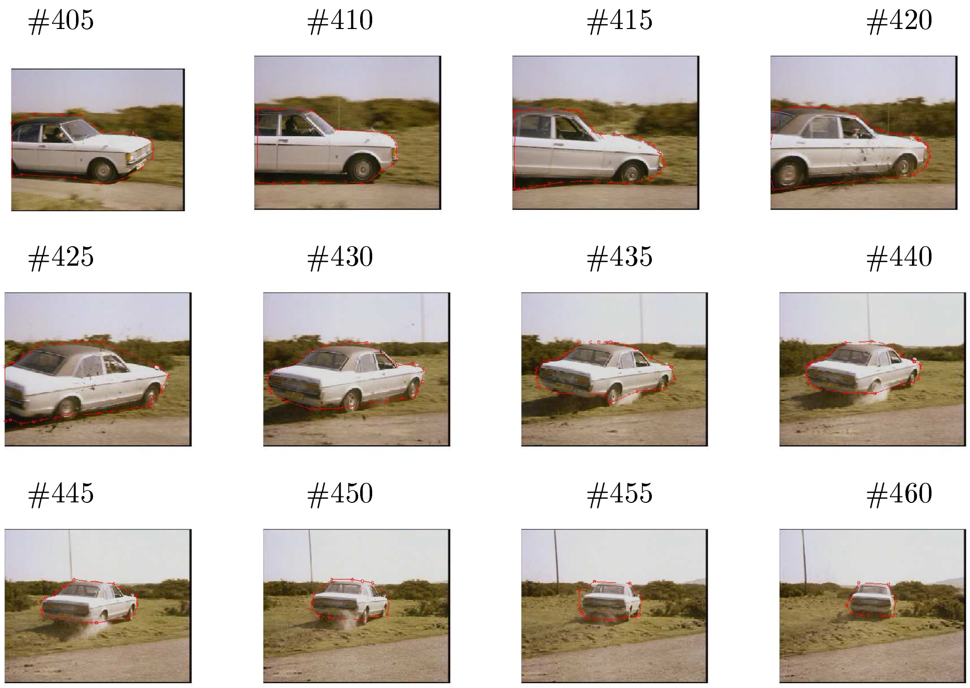

Figure 3: *Subset of views of the "Ford Car" sequence of* 66 *frames.*

5 Implementations and experimental results

In our project for building and browsing interactive video [9] [7], a video sequence is segmented into shots first, using the method of [3], and then moving objects are localized and tracked in each shot separately. The method of [5] was used to track objects. As mentioned previously we extract the key-frames here for a browsing purposes. The browsing of key-frames allows a fast visualization of the content of the shot (see figure 2 for example).

In the current experiments each occurrence of a tracked object is characterized by a histogram computed in the RGB color space. The histogram approach is well known as an attractive method for image retrieval because of its simplicity, speed and robustness. The RGB space is quantized into 64 colors. Then, the Principal Component Analysis was applied on the entire set of vector features in order to reduce their dimensionality. Only 10

Figure 4: Key-frame results for the "Ford Car" sequence

eigenvectors are kept corresponding to the 10 largest eigenvalues. Thus, in this new space the clustering strategy was applied. This makes the method more accurate and speed.

5.1 Results

Experiments are conducted on the MPEG "Avengers" TV movie of "Institut National de l'Audiovisuel en France" (INA). The extraction of key-frames is performed separately on each tracked object within a shot. During the estimation process, the maximum number of permitted Gaussian components, K, depends on the number of frames in the shot. Using the BIC criterion, the appropriate number of cluster ($\in [1..K]$) is chosen automatically i.e the the number of selected key-frames. The size of the temporal window was fixed to 20 frames which is reasonable to separate two key-frames in time.

To evaluate the effectiveness and accuracy of the proposed key-frame extraction technique, we illustrate in this paper the result on two different tracked objects of the database. The "Ford Car" and "la Licorne" sequences, consisting of 66 and 100 frames are illustrated in figures 3 and 5 respectively. One every 5 frames is depicted. The results of the proposed approach are presented in figures 4 and 6.

5.2 Discussion

For each experimented shot, it can be seen that the selected key-frames provide sufficient visualization of the total frames of the shot. They are clearly representative of the different views of tracked objects which are continuously changing with time.

The closest work to our approach is the work of [15]. Zhuang et al. use a linear clustering algorithm with a predefined threshold to determine the number of clusters. Then, they represent each formed cluster of frames by an arbitrary one. The technique presented here uses the EM algorithm which is more adequate to find the partition of a complex distribution where the number of clusters (complexity of the distribution) is determined automatically using the BIC criterion. Also, the key-frames are extracted here for tracked objects.

It is obviously that the method of Zhuang et al. is more speed because the employed clustering strategy is linear and non-parameterized. The proposed approach here is adopted by our project [9] since the estimated Gaussian components are used before the selection of key-frames, by the recognition process of similar tracked objects in the whole video sequence [7].

6 Conclusion

In this paper, an efficient video content representation has been presented for realizing an abstraction level beyond the shot one: the key-frame level. The presented framework represents a full automatic method to extract key-frames where no parameters are needed to be adjusted by the user. The use of the PCA and the addition of constraints on the covariance matrix make the method relatively fast.

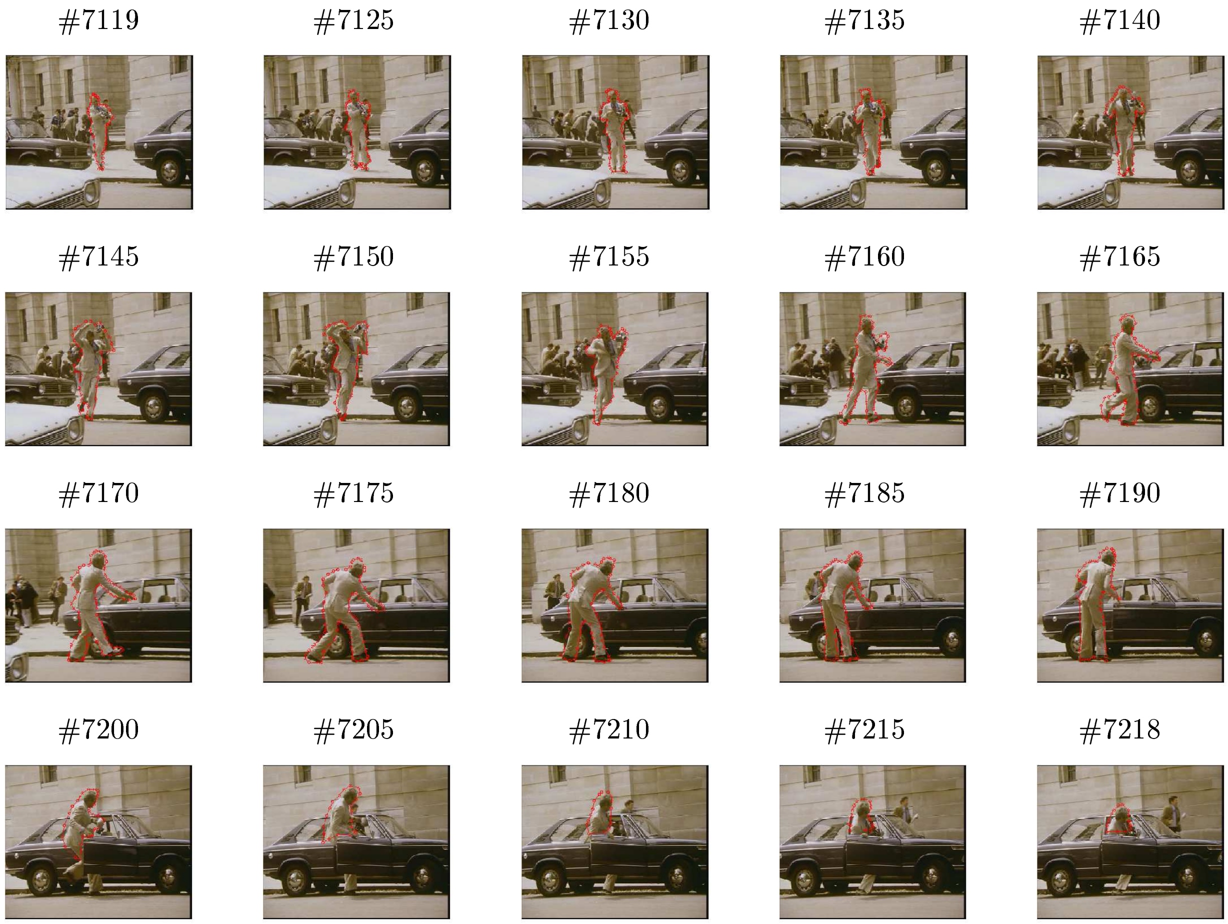

Figure 5: Subset of views of the “la Licorne” sequence of 100 frames.

The experiments on different real-world tracked objects shown the performance of the proposed technique. Such a technique can be performed on non-segmented frames of shots. It is able to capture the salient visual content of the key clusters and thus that of the underlying shot. The color histogram was computed as a feature where another features could be tested. However, accuracy of the estimation of the Gaussian mixture densities is related to the dimension of the feature space which must be chosen carefully in respect to the size of a shot.

The key-frames are extracted here for a browsing purposes only, but an index may be constructed from key-frames, and retrieval may be directed at key-frames. Finally, the estimated Gaussian models of tracked objects can be used to recognize similar objects in the whole video. A work on this research point is in progress.

#7145 #7187 #7214

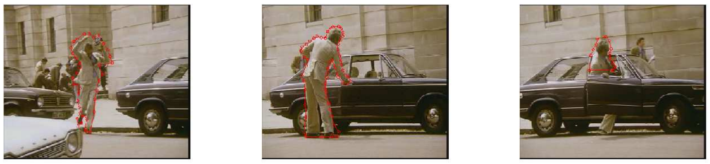

Figure 6: Key-frame results for the “la Licorne” sequence

Acknowledgments

We would like to acknowledge `Alcaltel CRC` for its support of this work, and the “`Institut National` de l’`Audiovisuel` en France”, dept of Innovation, for providing the video used in this paper.

References

[1] Y. S. Avrithis, A. D. Doulamis, N. D. Doulamis, and S. D. Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. *Computer Vision and Image Understanding*, 75(1/2):3–24, July-August 1999.

[2] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *Proc. SPIE Conf. on Vis. Commun and Image Proc.*, 1996.

[3] P. Bouthemy and F. Ganansia. Video partitionning and camera motion characterisation for content-based video indexing. *Proc. 3rd IEEE Int. Conf. Image Processing.*, september 1996.

[4] G. Celeux and G. Govaert. Gaussian Parsimonious Models. *Pattern Recognition*, 28(5):781–783, 1995.

[5] M. Gelgon and P. Bouthemy. A region-level graph labeling approch to motion-based segmentation. In *CVPR*, pages 514–519, Puerto Rico, June 17-19 1997.

[6] N. Guimaraes, N. Correia, I. Oliveira, and J. Martins. Designing computer for content analysis: A situated use of video parsing and analysis techniques. *Multimedia Tools and Applications*, 7:159–180, 1998.

[7] R. Hammoud and R. Mohr. Building and browsing hyper-videos: a content variation modeling solution. *Pattern Analysis and Applications*, 2000. Special Issue on Image Indexation, Submitted.

[8] R. Hammoud and R. Mohr. Gaussian mixture densities for indexing of localized objects in a video sequence. Technical report, INRIA, March 2000. http://www.inria.fr/RRRT/RR-3905.html.

[9] R. Hammoud and R. Mohr. Interactive tools for constructing and browsing structures for movie films. In *ACM Multimedia*, pages 497–498, Los Angeles, California, USA, October 30 - November 3 2000. (demo session).

[10] R. Hammoud and R. Mohr. Mixture densities for video objects recognition. In *International Conference on Pattern Recognition*, volume 2, pages 71–75, Barcelona, Spain, 3-8 September 2000.

[11] B.C. O'Connor. Selecting key frames of moving image documents : A digital environment for analysis and navigation. *Microcomputers for Information Management*, 8(2):119–133, 1991.

[12] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[13] W. Wolf. Hey frame selection by motion analysis. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, 1996.

[14] H. J. Zhang, C. Y. Low, S.W. Smoliar, and J.H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. *ACM Multimedia*, pages 15–24, 1995.

[15] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Proc. of IEEE conf. on Image Processing*, pages 866–870, Chicago, IL, October 1998.