Scene Segmentation and Image Feature Extraction in the DiVAN Video Indexing and Retrieval Architecture

Patrick Bouthemy[*] , Christophe Garcia [†], Rémi Ronfard[‡], George Tziritas [†], Emmanuel Veneau[‡], Didier Zugaj[*]

[*]INRIA–IRISA, Campus Universitaire de Beaulieu,35042 Rennes Cedex, France
[†]ICS–FORTH, P.O.Box 1385, GR 711 10 Heraklion, Crete, Greece
[‡]INA, 4 avenue de l'Europe, 94366, Bry-sur-Marne, France
Email:{cgarcia,tziritas}@csi.forth.gr,{rronfard,eveneau}@ina.fr,{bouthemy,dzugaj}@irisa.fr

Abstract We present a video analysis and indexing engine, that can perform fully automatic scene segmentation and feature extraction, in the context of a television archive. A template-based indexing engine is proposed, which interacts with a general-purpose video analysis server (providing spatio-temporal segmentation, feature extraction, feature clustering and feature matching services) and a template server (which provides tool settings, scripts and interpretation rules for specific program collections).

1 Introduction

The ESPRIT project DiVAN aims at building a distributed architecture for the management of - and access to - large television archives. An important part of the system is devoted to automatic segmentation of the video content, in order to assist the indexing and retrieval of programs at several levels of details from individual frames to scenes and stories, using a combination of annotated semantic information and automatically extracted content-based signatures.

In section 2, the template-based architecture of the system is described. The tools developed in this architecture are presented in section 3 In section 4, we briefly sketch how those tools can be integrated into complete analysis graphs, driven by the internal structure of television programs.

2 Template-based indexing and retrieval

In DiVAN, a video indexing session is distributed between several machines, using connections to two remote CORBA servers - a video analysis server and a template library server (1). Each program template contains the taxonomy of event categories (such as shots, scenes and sequences) meaningful to a collection of programs (such as news, drama, etc.), along with their temporal and compositional structure. In addition, the template contains algorithmic rules used

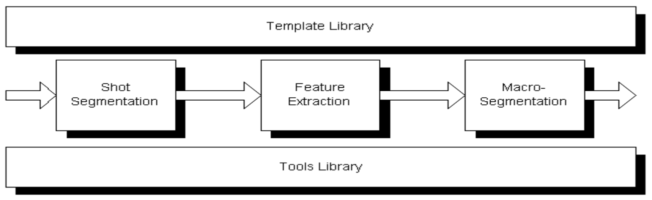

Figure1. *Architecture of the video indexing system in DiVAN: Analysis is controlled by scripts stored in the template library, and launches tools from the tools library. Both libraries are accessed via the CORBA ORB.*

to recover the structure of each individual program, based on observable events from the tools server (such as dissolves, wipes, camera motion types, extracted faces and logos). Section 3 reviews the tools available in the DiVAN system, and Section 4 presents examples of template-based indexing using those tools.

3 Shot segmentation and feature extraction

The first step in the analysis of a video is the partitioning into its elementary subparts, called shots, according to significant scene changes (cuts or progressive transitions). Features such as keyframes, dominant colors, faces and logos can also be extracted to represent each shot.

3.1 Cut detection

Our algorithm involves the frames that are included in the interval between two successive I-frames. It is assumed that this interval is short enough to include only one cut point. Given this assumption, the algorithm proceeds as follows. Step1: Perform cut detection considering the next I-frame pair and using one of the histogram metrics and a predefined cut detection threshold. Step2: Confirm the detected cut in the next subinterval between two consecutive I-, P-frames, included in the interval of I-frames. Step3: Perform cut detection between two successive I-, P-frames using histogram or MPEG macro block motion information. If the cut is not confirmed in this subinterval , go to step 2. Step4: Determine the exact cut point in the subinterval, by applying a histogram or MPEG motion based cut detection method on each frame interval-pair. If the cut is not confirmed, go to step 2, otherwise, go to step 1.

Key-frame extraction is performed during shot detection, and is based upon a simple sequential clustering algorithm, using a single cluster C of the candidate key-frames, which is successively updated by including the I-frame that follows its last element in display order. C is a set of K consecutive I-frames and C_{mf} is an "ideal" frame with the property that its features match the mean vector ("Mean Frame") of the corresponding characteristics of frames C_{fk} ($k = 1..K$). Assuming that the only feature used for classification is the colour/intensity histogram H_{fk} of frames C_{fk}, the "mean frame" histogram H_{mf} is computed. Then, in order to measure the distance $\delta(C_{fk}, C_{mf})$, an appropriate histogram metric is used. Based on this distance, the key-frame C_{KF} (the representative frame of C) is determined, according to a threshold.

3.2 Progressive transitions and wipes

The method we propose for detecting progressive transitions relies on the idea of temporal coherence of the global dominant image motion within a shot. To thus end, we use a method exploiting a M-estimation criterion and a multiresolution scheme to estimate a 2D parametric motion model and to ensure the goal of robustness in presence of secondary motions [8]. The minimization problem is solved by using a *IRLS (Iteratively Re-weighted Least Squares)* technique.

The set of pixels conforming with the estimated dominant motion form its estimation support. The analysis of the temporal evolution of the support size supplies the partitioning of the video into shots [3]. A cummulative sum test is used to detect significant jumps of this variable, accounting for the presence of shot changes. It can accurately determine the beginning and the end time instants of the progressive transitions. This technique, due to the use of the robust motion estimation method, is fairly resilient to the presence of mobile objects or to global illumination changes. More details can be found in [3]. It is particularily efficient to detect dissolve transitions. To handle wipe transitions between two static shots, we have developed a specific extension of this method. In the case of wipes, we conversely pay attention to the outliers. Indeed as illustrated in Figure 2, the geometry of the main outlier region reflects the nature of the wipe with the presence of a rectangular stripe. To exploit in a simple and efficient way this property, we project the outlier binary pixels onto the vertical and horizontal image axes, which supplies characterizable peaks. Then for detecting wipes, we compute the temporal correlation of these successive projections. The location and value of the correlation extremum allow us to characterize wipes in the video. One representative result is reported in Figure 2.

3.3 Dominant color extraction

We characterize the color content of keyframes by automatically selecting a small set of colours, called dominant colours. Because, in most images, a small number of colors ranges capture the majority of pixels, these colours can be used efficiently to characterize the color content. These dominant colors can be easily determined from the keyframe color histograms. Getting a smaller set of colours enhances the performance of image matching, because colour variations due to noise are removed. The description of the image, using these dominant colours, is simply the percentages of those colors found in the image.

We compute the set of dominant colors using the k-means clustering algorithm. This iterative algorithm determines a set of clusters of nearly homogeneous colours, each cluster being represented by its mean value. At each iteration, two steps are performed: 1) using the nearest-neighbour rule pixels are grouped to clusters, and 2) each new cluster is represented by its mean color vector.

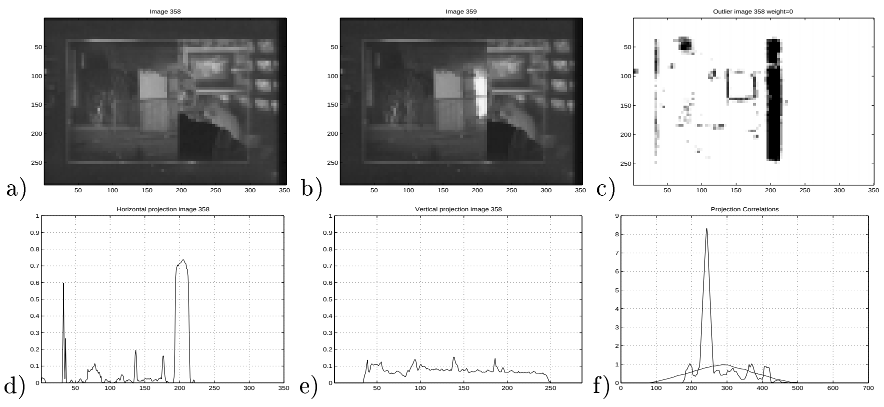

Figure2. *Detection of wipes: (a,b) Images within a wipe effect, (c) Image of outliers in black, (d,e) projections on horizontal and vertical axes, (f) plot of the correlation criterion.*

3.4 Logo detection

We introduce a simple but efficient method to perform the detection of static logos, which is the most frequent case in TV programs. To speed up the detection, we specify the expected screen location of the logo by defining its associated bounding box $\mathcal{B}$ (see Figure 3).

The method proposed is based on the two following steps : first, determination of the logo binary mask, then, evaluation of a matching criterion combining luminance and chrominance attributes. If the logo to be detected is not rectangular, the first step consists to build the exact logo mask for detection. That is, we have to determine which pixels belong to the logo in the bounding box. From at least two logos from images with different background, a discriminant analysis technique is used to cluster pixels in two classes : those belonging to the logo and the others. The clustering step is performed by considering image difference values for the luminance attribute $\{a_1\}$, and, directly values of chrominance images for chrominance attributes $\{a_2, a_3\}$ in the Y, U, V space. The two classes will be discriminated by considering a threshold T for each attribute $\{a_1, a_2, a_3\}$. We consider the inter-class variance $\sigma^2_{InterC}(T)$ given for each attribute $\{a_k\}$. The optimal threshold T^* then corresponds to an extremum of the inter-class variance $\sigma^2_{InterC}(T)$.

We display in Figure 3 an example of histogram of the image difference pixels within the bounding box, a plot of the inter-class variance $\sigma^2_{InterC}(T)$ supplying the dissimilarity measure between the two classes, and the determined binary mask $\mathcal{M}$. The matching simply consists in comparing each pixel that belongs to the logo mask in the current color image of the sequence with the value in the reference logo image. The criterion is given by the sum of the square differences between these values over the logo mask.

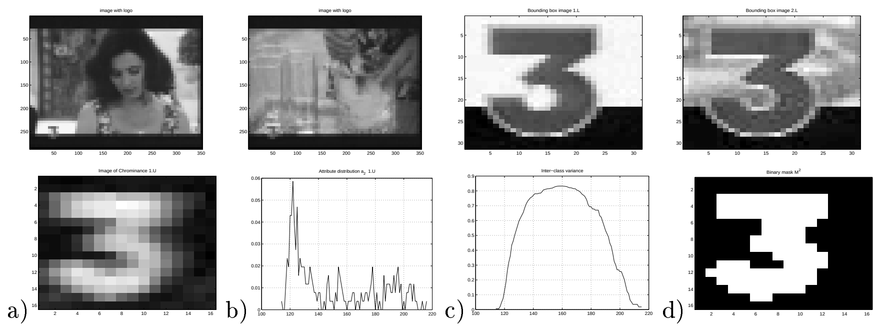

Figure3. Top row: Images of logos and associated bounding boxes. Middle row: Determination of the binary masks (d) from image of chrominance U (a). Plot of the attribute distribution $\{a_2\}$ (b), and the inter-class variance $\sigma^2_{InterC}(T)$ (c).

3.5 Face detection

We developed a fast and complete scheme for face detection in color images where the number of faces, their location, their orientation and their size are unknown, under non-constrained conditions such as complex background [6]. Our scheme starts by performing a chrominance-based segmentation using the YCbCr color model which is closely related to the MPEG coding scheme. According to a precise approximation of the skin color sub-spaces in the YCbCr color model, each macro-block average color value is classified. A binary mask is computed in which a "one" corresponds to a skin color macro-block, and a “zero” corresponds to a non-skin color macro-block. Then, scanning through the binary mask, continuous “one” candidate face regions are extracted. In order to respect the speed requirements, we use a simple method of integral projection like in [10]. Each macro-block mask image is segmented into non-overlapping rectangular regions that contain a contiguous “one” region. Then, we search for candidate face areas in these resulting areas. Given that we don't know a priori the size of the faces regions contained in the frame, we first look for the largest possible ones and we iteratively reduce their size and run over the possible aspects ratios and the possible positions in each binary mask segment area. Constraints related to size range, shape and homogeneity are then applied to detect these candidate face regions and possible overlaps are resolved. The main purpose of the last stage of our algorithm is to verify the face detection results obtained after candidate faces areas segmentation and remove false alarms caused by objects with color similar to skin colors and similar aspects ratios such as exposed parts of the body or part of background. For this purpose, a wavelet packet analysis scheme is applied using a similar method to the one we described in [4, 5] in the case of face recognition. For a candidate face region, we search for a face in every position inside the region and for a possible number of dimensions of a bounding rectangle. This search is based on a feature vector that is extracted from a set of wavelet packet coefficients related to the corresponding region. According to a metric derived from the Bhattacharyya distance, a subregion, defined by its position and size in candidate faces area is classified as a face or non-face area, using some prototype face area vectors, acquired in a previous training step.

4 Scene segmentation

Shot segmentation and feature extraction alone cannot meet the needs of professional television archives, where the meaningful units are higher-level program segments such as stories and scenes. This paper focuses on programs described by generic templates (special-purpose templates are described in [9].

4.1 Simple program template

The DiVAN simple program template implements a scene segmentation algorithm introduced by Yeo et al. [11]. A scene is defined as a grouping of contiguous shots, based on image similarity. This algorithm uses three parameters : an image similarity function, an average scene duration ΔT and a cluster distance threshold. The similarity function can be defined, using a combination of extracted features : color histograms, dominant colors, camera motion types, number, size and positions of faces, etc. Figure 4 shows the analysis graph for scene analysis using dominant colors. The result of the indexing session consists of a *soundtrack* and a hierarchically organized *scene track.*

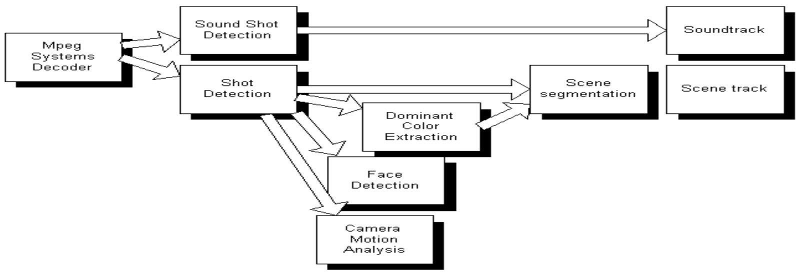

Figure4. *Analysis graph for the simple program template.*

The details of the scene segmentation algorithm follows the original algorithm by Yeo et al. A binary tree is constructed in order to represent the similarity between keyframes. A temporal constraint is introduced in order to reduce computation time. To calculate the $N(N-1)$ distances describing the similarity between clusters, we introduced initially a distance between keyframes with a temporal constraint as done in [2]. Distances are accessed via a priority stack for efficiency reasons. At each step, the smaller distance among the N(N-1) distance is selected and the two corresponding clusters are merged. The whole set of distances is updated through the Lance&Williams formula. A binary tree is built by successive merging of the nearest clusters, until the threshold value is reached.

The scene segmentation is computed according to the methods exposed in [2]. We build an oriented graph where the nodes are the previously selected clusters and the edges are temporal succession links. The oriented graph is segmented into its strongly connected components, which correspond to scenes. Finally, we complete the scene track by adding all progressive transitions at their appropriate level (between shots or scenes). We obtained good scene segmentation results using regional color histograms or dominant colors in programs with long sequences corresponding to one scene (drama, fiction). In many other cases, scene segmentation suffers some limitations, due to the restrictive definition of a scene. For instance, a short story item in a news broadcast will be incorporated into a larger scene, because of the recurrent anchorperson images before and after the story.

4.2 Composite program template

The DiVAN composite program template is used to introduce a broader definition for program segments such as stories, using cues such as logo or face detection. Figure 5 shows the complete analysis graph for a composite program. The program is segmented into stories, based on detected logos, wipes, dissolves and soundtrack changes, and stories are segmented into scenes using the previously described algorithm. in [9], we state the story segmentation problem as a CSP (constraint satisfaction problem), in the framework of an extended description logics.

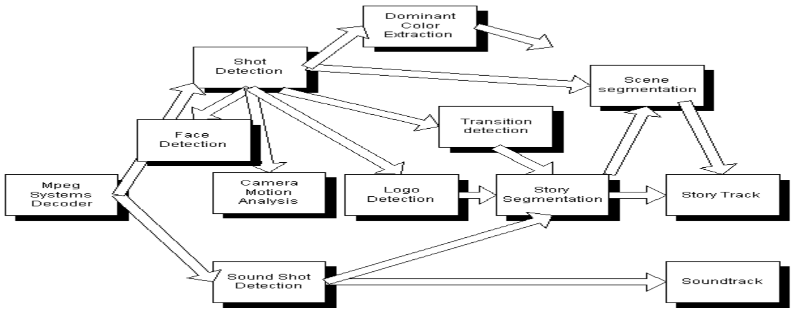

Figure5. *Analysis graph for a composite program template.*

5 Conclusion

This paper describes the first prototype of the DiVAN indexing system, with a focus on the tools library and the general system architecture. Template-based video analysis systems have been proposed in academic research ([2, 7, 1], but no such system is available commercially. The Video Analysis Engine by Excalibur introduces settings for shot segmentation - but the list of settings is closed. The

DiVAN prototype gives the users control of many efficient video analysis tools, though the edition of their own templates, and will be thoroughly tested by documentalists at INA, RAI and ERT in early 1999. The second DIVAN prototype will provide many more functions such as audio analysis, face recognition, query-by-example, moving objects analysis and camera motion analysis.

Acknowledgments

This work was funded in part under the ESPRIT EP24956 project DiVAN. We would like to thank the Innovation Department at Institut National de l'Audiovisuel (INA) for providing the video test material reproduced in this communication.

References

1. P. Aigrain, P. Joly, and V. Longueville. Medium knowledge-based macro-segmentation of video into sequences, *in Intelligent multimedia information retrieval*, M.T. Maybury, Editor. 1997, MIT Press.
2. G.Ahanger and T.D.C. Little. A survey of technologies for parsing and indexing digital video. *Journal of Visual Communication and Image Representation*, 7(1),March 1996.
3. P. Bouthemy and F. Ganansia. Video partioning and camera motion characterization for content-based video indexing. In *Proc. 3rd IEEE Int. Conf. on Image Processing, ICIP'96*, Lausane, September 1996.
4. C. Garcia, G. Zikos, G. Tziritas. A Wavelet-Based Framework for Face Recognition. In *Int. Workshop on Advances in Facial Image Analysis and Recognition Technology, 5th European Conference on Computer Vision*, June 2 – 6, 1998, Freiburg, Germany.
5. C. Garcia, G. Zikos, G. Tziritas. Wavelet Packet Analysis for Face Recognition. To appear in *Image and Vision Computing* (1999).
6. C. Garcia, G. Zikos, G. Tziritas. Face detection in color images using wavelet packet analysis Submitted to *IEEE Multimedia Systems'99* (1999)
7. B. Merialdo and F. Dubois, An agent-based architecture for content-based multimedia browsing, *in Intelligent multimedia information retrieval*, M.T. Maybury, Editor. 1997, MIT Press.
8. J.M. Odobez and P. Bouthemy. Robust multiresolution estimation and object-oriented segmentation of parametric motion models. *Jal of Visual Communication and Image Representation*, **6(4)**, 1995.
9. R. Ronfard, J. Carrive, F. Pachet. A film template library for video analysis and indexing based on description logics. Submitted to *IEEE Multimedia Systems'99.*
10. H. Wang and S.-F. Chang. A Highly Efficient System for Automatic Face Region Detection in MPEG Video. *IEEE Transactions on Circuits and Systems For Video Technology*, **7(4)** (1997) 615-628.
11. M. Yeung, B.-L. Yeo & B. Liu. Extracting story units from long programs for video browsing and navigation. *International Conference on Multimedia Computing and Systems*, June 1996.