Image Coding I

Automatic Video Segmentation through Editing Analysis

J. M. Corridoni[1] and A. Del Bimbo[1,2]

[1] Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, via S. Marta 3, 50139 Firenze, Italy.
[2] Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia, via Branze, 38, 25133 Brescia, Italy

Abstract. In the present paper the problem of video segmentation for indexing and retrieval in multimedia databases is addressed. This work proposes a statistically based new algorithm to detect cuts between two shots in a video sequence. Such algorithm is data driven, since no a priori knowledge about the shots is available. A mathematical model is formulated for detecting fades, dissolves and mattes. All the algorithms are inserted into a global framework, in order to generate a low level parsing and segmentation of video streams.

1 Introduction

With the emergence of multimedia applications, a growing interest is being focused on databases of digital images, image sequences and video, concerning the way in which contents of these pictorial data are described, indexed and retrieved. The central problem for the effective access and usage of a video database lies in representing video content meaning, which is determined not purely by the visual content of the video stream, but also by the way in which frame sequences are ordered and organized into the entire stream. Visual content of frame sequences and video may be expressed either by detecting the spatial occurrences of objects in the individual frames,[3], and encoding through appropriate representation languages the temporal change of the spatial relationships between imaged objects, [1],[4], or at a higher level, by describing spatio-temporal interactions among imaged objects, through statements taken from descriptive languages [2]. Detection of cuts between clips, based on data analysis, is developed in [3]. There, the *Template Matching* technique and the χ^2 *Test* are applied to the color histograms of two subsequent frames. Segmentation obtained not purely considering data analysis, is presented in [5], where a different cut technique is presented (*Intensity Average Difference*) and Different editing procedures like page translate, fades and dissolves are modeled through mathematical functions. The film editing is to be considered not only the "glue" between two shots, but also an essential contribute to the meaning conveyed by the video. Where film becomes art, editing techniques create a *meta–language*, through which a part of the visual message is communicated to the viewer. The detection of the editing features in a film can be therefore a useful help

for technical analysis on films as forms of art. In this paper we present a new prototypal system for automatic segmentation of video into its syntactic units through the detection of the editing operations that are present in the film. Experimental evidence of the system operation is provided by presenting results selected from the set of experiments carried out over critical video sequences. A critical comparison with the performance of systems discussed in the literature is also presented.

2 Editing Analysis

The parsing procedure herein introduced is aimed to detect all the editing operations that are present in a film. The stream of frames recorded between the time in which a video camera is turned on and turned off is called *shot.* A *scene* is a sequence of shots related by semantic features. The content of a single scene must have the four Aristotelian properties of unity of *space, place, time* and *action.* All the shots sharing these properties are part of the same scene. The shot is the syntactic unit of a video, while the scene constitutes the semantic atom upon which the visual speech is based. In the following we will address both the detection of cuts, and of the other editing techniques.

Cut Detection: A cut between two shots hasn't undergone any editing process and has therefore no mathematical model, since no information about the shots connected is available. The basic idea that lies under the design of a *cut detector* is that consecutive frames belonging to the same shot are in some way *more similar* than frames belonging to different shots. Measures for the cut detection in a gray-level or a RGB video proposed in [5], [3] show poor performance whenever there is large motion in the scene, for example in presence of rapid moving objects or camera zoom. Even in such cases the global image background does not change significantly, while, on the contrary, a cut implies a global change in the scene. As a result, we expect a cut to be present when the difference between two frames is much larger than the *standard* difference between frames belonging to the same shot. Therefore a relative difference between frames should be introduced, which is expressed as an incremental ratio: $\mathbf{CD} = \mathbf{D}(f_{t+1}, f_t)/\mathbf{D}(f_t, f_{t-1})$, where $\mathbf{D}(f_{t+1}, f_t)$ is a suitable difference measure of two frames. Previous equation states that a cut is detected when the difference between two frames is larger with respect to the former. This statement eliminates the difference as an absolute value, but states its relativeness. In order to increase robustness to local changes in the image, such as the appearance of an object in the visual field, each frame has been divided into $n \times n$ rectangular regions. Instead of comparing two whole frames, as in [5], we compare every pair of subframes between two frames, to obtain n^2 difference values. Discarding the largest values, we make $\mathbf{D}$ more robust against local changes between consecutive frames of the same shot. The problem of defining a meaningful measure of difference has been addressed, using some statistical parameters of the color histograms. Instead of encoding the color levels, as in [3] and [5], the three color histograms H_r, H_g, H_b, are collected for each subframe $r_t(i)$ of frame f_t. Such histograms can be parameterized well

enough by the three statistic moments of first (m_1), second (m_2) and third order (m_3). Denoting with $\mathbf{D}_i^{red}(f_t, f_{t+1})$ the difference between the couple of subframes i, with reference to the red color, we define:

$$\mathbf{D}_i^{red}(f_t, f_{t+1}) = \sum_{i=1}^{3} a_i |m_i^{red}(t+1) - m_i^{red}(t)| \tag{1}$$

being $\mathbf{a} = [a_1, a_2, a_3]^{\mathrm{T}}$ a set of parameters experimentally tuned. These parameters can be modulated, whenever more information is available about the shots. For the generic subframe i we define the difference measure in the following way.

$$\mathbf{D}_i(f_t, f_{t+1}) = \mathbf{D}_i^{red}(f_t, f_{t+1}) + \mathbf{D}_i^{green}(f_t, f_{t+1}) + \mathbf{D}_i^{blue}(f_t, f_{t+1}). \tag{2}$$

Finally, the global measure $\mathbf{D}(f_t, f_{t+1})$ of the difference between frames f_t and f_{t+1} is obtained, discarding the k largest values among the n^2 that have been computed and averaging the remaining values. A shot-cut in a video is detected by thresholding: whenever **CD** is over a predefined threshold, then a cut is inferred. Since there is no mathematical model underlying the production of a shot, nor an *a priori* knowledge about its characteristics, then no conceptual reasoning is applicable to determine the threshold value, which has to be stated experimentally.

Chromatic Editing Detection: While the *cut* has no model, a mathematical approximation can be proposed for the fades, dissolves and mattes, since these effects are obtained in laboratory, according to a specific technique. Let $\mathbf{c} = (r, g, b)$ represent the intensity level of an arbitrary pixel in a video frame, we can identify the frames belonging to a shot $\mathbf{S}$ as $\mathbf{S} = \{(x, y, t) |\ \mathbf{c} = \mathbf{S}(x, y, t)\}$. Let $\mathbf{S}_1(x, y, t)$ and $\mathbf{S}_2(x, y, t)$ be two shots that are being edited, and $\mathbf{S}(x, y, t)$ the edited shot. All the chromatic processes can be described as a linear pixel intensity manipulation, since the machines used (*truka*) follow a linear law. Therefore, the equation that expresses $\mathbf{S}$ has the following form [5]:

$$\mathbf{S}(x, y, t) = \mathbf{S}_1(x, y, t) \left(1 - \frac{t}{l_{out}}\right) \Big|_{(t_1, t_1 + l_{out})} + \mathbf{S}_2(x, y, t) \left(\frac{t}{l_{in}}\right) \Big|_{(t_2, t_2 + l_{in})} \tag{3}$$

The fade–in and fade–out are peculiar cases, in which $\mathbf{S}_1 = \mathbf{0}$ or $\mathbf{S}_2 = \mathbf{0}$, respectively. In the matte case, $\mathbf{S}_2$ is zero outside a specified pixel area expressed by bidimensional function of shape $f(x, y)$. To design a chromatic feature detector, let us consider the case of a fade–in. Let $\mathbf{S}_{\text{fi}}$ be the fade–in sequence; deriving it with respect to time we obtain:

$$\frac{\partial \mathbf{S}_{\text{fi}}(x, y, t)}{\partial t} = \frac{\partial \mathbf{S}_2(x, y, t)}{\partial t} \left(\frac{t}{l_{in}}\right) + \frac{\mathbf{S}_2(x, y, t)}{l_{in}} \tag{4}$$

If it is supposed that the scene does not change much during the fade–in, which is often verified in films, the first addend can be skipped. Dividing (3) by $\mathbf{S}_2$, we obtain a constant value. This means that in a fade–in (out) sequence the frame ratio $\frac{f(t+1) - f(t)}{f(t+1)}$ is approximately constant. The verification of the constancy of this ratio can be assumed as an efficient measure for detecting fades. In a typical

dissolve sequence there is always a phase, where the chromatic dynamics of one of the two shots being edited prevails over the other, in such a way that at least two phases of "dominant fade–in" and "dominant fade–out" can be separated. During each of these two phases the model in equation 3 applies and the value of the chromatic feature is meaningful. In [5], this measure is obtained only based on local analysis. This approach contrasts with the way in which these artifices are actually devised in the editing of a video, since dissolves, fades and mattes have no local meaning, being obtained as an optical process, which is performed on a sequence of frames, not on a single frame like the cut. Moreover, such editing processes are performed using the "truka" machine, which can operate only on sequences of standard duration (16, 24, 34, 48 or 96 frames). Therefore, we analyze a sequence over a wide temporal window which is approximately of the size of the minimal chromatic sequence: 16 frames. Whenever the temporal average over such a window overcomes a threshold, then a chromatic editing is detected. Differently from [5], the algorithm allows for a discrimination among fade–in, fade–out, dissolve and matte. By extracting the color histograms of the first and last frames in the edit sequence detected, a distinction between fade-in and fade-out can be easily made, since the average luminance of the first or last frame, respectively, is approximately zero. The dissolve, has first and last frame with non zero luminance and generates typically a local minimum in the feature histogram, as it will be clear in section 3. As to the mattes, they look like fades, apart from the fact that luminance varies over the frames following a geometrical law. Once a fade has been detected, a statistical analysis over one or two frames, typically the central ones, has to be made. If the luminance has a big statistical discontinuity, due to the presence of a black mask the partially covers the frame, then a matte is detected.

3 Experimental Results

The techniques described in previous sections have been applied to movies and to television commercials. The cut and chromatic features detection has been tested on about four hours of video sequences, and a comparison with editing other detection techniques has been made. Results obtained with two sample scenes are reported below. The scenes presented show specific criticality for cut and dissolve detection respectively. **Sequence 1:** Figure 1 shows a sequence taken from a TV commercial, where 6 cuts are present. The longest shot in this sequence is characterized by a large amount of motion, due to the rapid panning of the camera which follows the arrow during its run to the target. This large motion condition could deeply impair the cut feature. The histogram in figure 2a, shows the feature response detected. Its performance is compared with the *Average Intensity* measure [5] (fig. 2b), the *Template Matching* (fig. 2c) and the χ^2 *Test* techniques (fig. 2d), [3]. The comparison among the histograms shows that, while the first measure detects only the cuts, the others give some peak response during the camera panning. **Sequence 2:** The histogram in fig. 3a shows the chromatic

feature response computed on a typical dissolve sequence [3], according to the algorithm described in Sec. 2. In the figure the three phases of *dominant fade–in* (**A**), equal fade–in and out (**B**) and *dominant fade–out* (**C**), detected by the measure are put in evidence. The diagram in fig.3b shows the output response for the technique proposed in [5]. The cut feature response shows no peaks and its diagram is therefore here omitted. While histogram in fig. 3b has peaks outside the dissolve zone and low peaks even in the dissolve region, diagram in 3a has a large region with high values only in correspondence with the dissolved frames. In that region two relative maxima and a local minimum can be detected. They describe the three phases of dominating fade–in, equal fade–in and fade–out and dominating fade–out, that have been highlighted previously. The local minimum in phase (**B**) is to to the fact that the error made approximating a dissolve with a simple fade as in equation 4 becomes larger during that phase. Finally, table 1 summarizes the global results obtained on four hours video. The result summary indicates a 97% correct segmentation.

Edit Type	Correct	False	Total	Error Rate
Cut	798	15	813	2%
Fade–in	23	1	24	4%
Fade–out	34	2	36	5%
Dissolve	16	2	18	11%
Matte	7	0	7	0%
TOTAL	878	20	898	3%

Table 1. Error rates for the video segmentation algorithm, tested on 4 hour video.

References

1. T.Arndt, S.K.Chang, "Image Sequence Compression by Iconic Indexing", *IEEE VL '89 Workshop on Visual Languages*, Roma, Italy, Sept.1989.
2. M.Davis, "Media Streams, an Iconic Visual language for Video Annotation", *Telektronik*, No.4, 1993, (also appeared in reduced version in *Proc.IEEE VL'93 Workshop on Visual Languages*, Bergen, Norway, Aug.1993).
3. A.Nagasaka, Y.Tanaka, "Automatic Video Indexing and Full Video Search for Object Appearances," in *IFIP Transactions, Visual Database Systems II*, Knuth, Wegner (Eds.), Elsevier Pub. 1992.
4. A.Del Bimbo, E. Vicario, D. Zingoni, "Symbolic Description of Image Sequences with Spatio Temporal Logic, *IEEE Transactions on Knowledge and Data Engineering*, to appear.
5. A. Hampapur, R. Jain, T. Weymouth "Digital Video Indexing in Multimedia Systems" in *Proc. of AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems* Seattle, Wa, Aug. 1994.

[3] Beginning sequence from the episode by Martin Scorsese in "*New York Stories*".

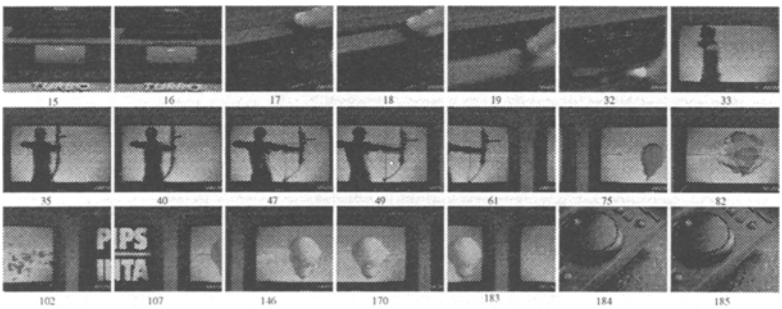

Fig. 1. Frames sampled from a TV spot, characterized by rapid camera panning

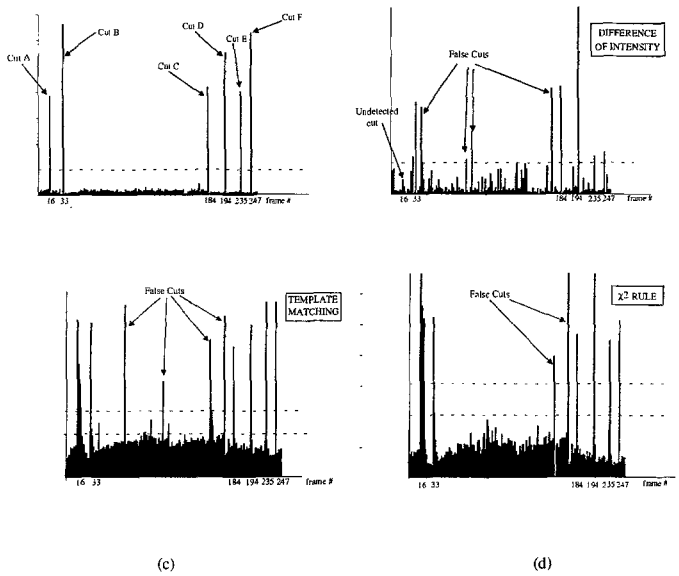

Fig. 2. Comparison among cut detectors. (a) Algorithm described in sect. 2. (b) *Average Intensity Difference.* (c) *Template Matching.* (d) χ^2 *Test.*

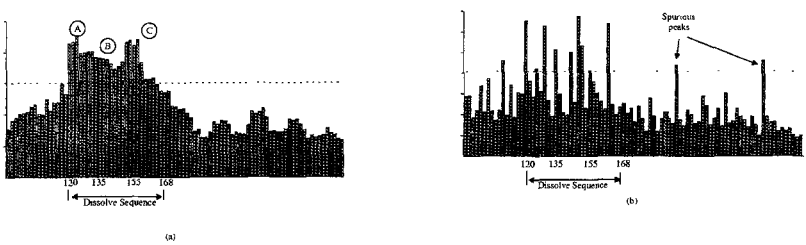

Fig. 3. (a) The response diagram for the dissolve detector presented in section 2. (b) The response diagram for the dissolve detector introduced in [5].