Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2011, Article ID 537372, 19 pages
doi:10.1155/2011/537372

Research Article

A Similarity-Based Approach for Audiovisual Document Classification Using Temporal Relation Analysis

Zein Al Abidin Ibrahim,[1] Isabelle Ferrane,[2] and Philippe Joly[2]

[1] *LERIA Laboratory, Angers University, 49045 Angers, France*
[2] *IRIT Laboratory, Toulouse University, 31062 Toulouse, France*

Correspondence should be addressed to Zein Al Abidin Ibrahim, zibrahim@info.univ-angers.fr

Received 1 June 2010; Revised 28 January 2011; Accepted 1 March 2011

Academic Editor: Sid-Ahmed Berrani

We propose a novel approach for video classification that bases on the analysis of the temporal relationships between the basic events in audiovisual documents. Starting from basic segmentation results, we define a new representation method that is called Temporal Relation Matrix (TRM). Each document is then described by a set of TRMs, the analysis of which makes events of a higher level stand out. This representation has been first designed to analyze any audiovisual document in order to find events that may well characterize its content and its structure. The aim of this work is to use this representation to compute a similarity measure between two documents. Approaches for audiovisual documents classification are presented and discussed. Experimentations are done on a set of 242 video documents and the results show the efficiency of our proposals.

1. Introduction

Motivated by the fact that large scale document indexing cannot be handled by human operators, researches tend to use high-level automatic indexing with the recent existing huge masses of digital data. Several automatic tools are based on low-level feature extraction. For audiovisual documents, low-level features can be the result of audio, image or video processing. However, finding the discriminating characteristics is still a challenging issue, especially if one wants to keep the detection of the basic events reliable and robust enough. Another challenge is to detect events of a sufficiently high semantic level and to produce indexes that are highly relevant according to the document content and structure. Such indexes will then allow requests such as, "*I am looking for an interview of Mister X by Miss Y about the movie Z*" to be in a high-level information retrieval task.

From an automatic indexing point of view, answering these requests requires searching among the available audiovisual documents in order to find the ones that contain such events. This requirement results in two major objectives. The first consists in proposing a method for automatically standing out a document structure due to the different events that are occurring in it. This leads to the second objective which is to make automatic classification according to these document structures.

To reach those goals, we first analyze the audiovisual document content from a temporal and generic point of view. For audiovisual documents, time is a central component, so each document has a beginning, an end, and a length and contains different events. In its turn, each event has also a beginning, an end, and a length. However, the detection of these events depends on what the underlying definition of an event is and also on the granularity of the events themselves. Generally, results of automatic analysis of the audio/video components indicate when a specific feature has been detected. Thus, whatever the media is and whatever its basic characteristics are, we already have temporal basic events that can be described by elementary descriptors. Combining these temporal events can be a way to detect more relevant events and to improve the semantic level of the document content analysis. The generic side of our approach lies in the fact that we are trying to be independent from any prior knowledge about the document type (sports, news, movie...), its production rules (how it is structured), or the specific events it may contain. Even

if some tools are extracting basic events from a single medium (image [1], video [2], or audio [3]), most of the approaches are recently focusing on the combination of basic events (color, shape, activity rate, texture...) extracted from several mediums. Even though some of them are based on multimodal extraction [4], they remain limited by the fact that they are looking for well-defined semantic classes of events (goal, play, and break phases in sports games, reports in news programs) in a specific type of document (sports, news programs...) or a specific content (soccer, baseball, football, ...). Some efforts have been made to generalize event detection techniques but they are still bound to a specific domain like sports [5]. In its turn, our approach can be also considered a generic characteristic as it is based on the temporal analysis of document content without being constrained by the video type, its structure, or the containing events.

The rest of this paper is organized as follows. In Section 2, we give a short summary concerning the basic principles of our approach, and we show what kind of events can stand out from document content. Then, in Section 3 we explain how we define a similarity measure which will be the basis of our two document classification methods. We also describe and discuss the results of our experiments on document classification before concluding and presenting some work perspectives in Section 4.

2. Temporal Relationship Analysis

Temporal representation has been already addressed by some of the existing works [6–9]. These approaches aim at defining basic units and to express temporal relationship between them. The existing models depend on the type of the temporal unit used (point or interval) and the temporal constraints taken into account (qualitative, quantitative). In the qualitative models, the interest is to observe the nature of the relations. For example, the relation "I before J" is a qualitative temporal relation. On the other hand, the quantitative ones focus on numerical features such as the distance between the start of J and the end of I.

The temporal models of the literature are point-based [10–12], interval-based [13–16], or the mixture of the two [17–20].

In [21], Allen has proposed the well-known model that describes the relationships between intervals by means of thirteen relationships. This approach is more generic than others but we still want to be more generic and to take into account any relation between events whatever the events may be (points, intervals, or the two) without losing the quantitative information. The first step of our method consists in analyzing a document content by studying the temporal relations between the events that it may contain, basing on the parametric representation of these relations as it is explained in the next paragraph.

2.1. Parametric Representation of Temporal Relations. In a previous paper [22], we have presented all the basic principles of the parametric representation of temporal relations which is the core of our work. Here we present the main points of this representation.

As an input, we use a set of N elementary segmentations made on a same document. Such segmentations are defined as a set of temporally disjointed segments, where each segment is a temporal interval. Each temporal interval represents the occurrence of a specific type of event. Each event indicates the presence of a specific low-level or mid-level feature in the document, such as speech, music, applauses, speaker (from audio), and color, texture, activity rate, face detection, costume (from video). These segmentations can be done automatically or manually, and are represented as follows. Any elementary segmentation Seg that contains M segments is defined by: $\text{Seg} = \{S_i\}$; $i \in [1, M]$. In its turn, each segment S_i is characterized by its two endpoints: its beginning (S_{ib}) and its end (S_{ie}) and will be written: $S_i = [S_{ib}, S_{ie}]$.

As it is proposed in [23], any temporal relation R between two segments is defined by three parameters: the distance between segments-ends (DE), the distance between segments-beginnings (DB) and the gap between the two segments (Lap). Let us consider $(\text{Seg}_1, \text{Seg}_2)$ a pair of elementary segmentations that contain (M_1, M_2) segments, respectively. We compute the three parameters between all the possible couples of segments $(S_{1i}, S_{2j}) \in \text{Seg}_1 \times \text{Seg}_2$; $i < j$.

$$\text{DE} = S_{2je} - S_{1ie}; \quad \text{DB} = S_{1ib} - S_{2jb}; \quad \text{Lap} = S_{2jb} - S_{1ie}. \tag{1}$$

More formally, R can be written:

$$S_{1i} R(\text{DE}, \text{DB}, \text{Lap}) S_{2j}. \tag{2}$$

Considering all the possibilities, we will have $M_1 \times M_2$ temporal relations. However, if the two segments are too far from each other, the temporal relation between them will be less relevant. In other words, two events e_1 and e_2 may probably be semantically related if they are not far away from each other. To avoid considering all the possible temporal relations between the events that are very far in distance from each other, we limit the scope of our considered temporal relations by introducing a threshold α for the distance between any compared pair of segments. α is chosen empirically basing on some observations that we have made on some audiovisual documents, and then it is used to select only the relations that verify the condition $\text{Lap} < \alpha$. Next we explain how we based on these first steps in order to compute a Temporal Relation Matrix.

2.2. Temporal Relation Matrix (TRM). As we have seen, each temporal relation is represented by a set of three parameters. This set can be considered as the coordinates of a point in a three-dimensional space, or as cell-indexes in a three-dimensional matrix. The former representation allows us to visualize all the observations made between two elementary segmentations Seg_1 and Seg_2 in a graphical way, while the latter can be used as a vote (co-occurrence) matrix, in which the occurrence of each temporal relation will be counted.

Each time the same values for the three parameters (DE, DB, Lap) is observed, the value of the corresponding cell in the matrix is incremented. This helps in calculating the temporal relation frequency and to further study the TRM content.

Building such a matrix is not quite simple. Each elementary segmentation is based on the detection of a specific feature in a given medium. Thus, a quantization step must be done before computing the TRM because the audio and video components do not possess the same temporal units. Audio segments must be aligned on each point corresponding to an image (the video lower unit). This step is also interesting because it takes us from the real space to the integer one in the parameter computation. However, for a document with a set of N elementary segmentations, the number of corresponding TRMs will be $(N^*(N-1))/2$. The first step of a TRM analysis is to study the number of the observed temporal relations and their distribution in the 3D space. This distribution is related to the nature of the temporal relations to observe. If they are predefined like the Allen's ones (see Section 2), so the semantic of these relations will introduce some constraints on the parameters scope and significant subparts of the TRM can be identified. For example, the (DE, DB, Lap) parameters of the "MEETS" relation take values in $(]0\ +\infty], [-\infty 0[, \{0\})$. On the other hand, if these relations are completely unknown, we will need an automatic method that can put forward the main zones in which temporal relations are distributed. After this step, we will be able to find if any relevant interpretation can be deduced. The importance of the latter method is that we base on the distribution of the point in the space in order to induce the temporal relations. In other words, we base on the quantitative information in order to obtain the qualitative one. In contrast, we do not have any prior semantic interpretation of the observed temporal relations.

2.3. Distribution of Temporal Relations. We are considering, as an example, four of the thirteen Allen's relations between couple of temporal intervals (or segments): before, overlaps, meets, and equals. Each relation defines a zone in which observations will be located. Considering each temporal relation as a point p having the coordinates (x, y, z) with $x = \text{DE}, y = \text{DB}$, and $z = \text{Lap}$. We can see that the first and the second zones are subparts of a 3D space, while the third one is a plane space and the fourth zone is a half straight line.

$$\begin{aligned}
\text{BEFORE} &= \{p = (x,y,z),\ 0 < z \le \alpha,\ y < -z,\ x > z\},\\
\text{OVERLAPS} &= \{p = (x,y,z),\ z < 0,\ y < 0,\ x > 0\},\\
\text{MEETS} &= \{p = (x,y,z),\ x > 0,\ y < 0,\ z = 0\},\\
\text{EQUALS} &= \{p = (x,y,z),\ x = 0,\ y = 0,\ z < 0\}.
\end{aligned} \tag{3}$$

The scope limit α has been taken into account when it has been relevant. The graphical representation of the "MEETS" and "OVERLAPS" relations are shown in Figure 1.

A temporal relation alone may not be significant. The occurrence number of the temporal relations, inside of the subpart of the TRM to which they belong, will be significant and will determine if the associated class of temporal relations is or is not relevant enough. Thus, once the different subparts have been identified, a global occurrence number is associated to each one. This number is the sum of all the votes corresponding to the temporal relations belonging to the subpart.

Our objective is to be as generic as possible and not to limit ourselves to a set of predefined relations like Allen's ones. For this aim, we need to use an automatic classification method to identify classes of relations. Then, each class is represented by the occurrence number of the temporal relations it contains.

2.4. Temporal Relation Matrix Classification. For data classification in the TRMs, we use the well-known K-means method. On the other hand, the errors produced by the segmentation tools affect the TRM calculation. We have presented some experiments that mainly concern the way of handling segmentation errors and studying their effects on TRM calculation. In these experiments, the fuzzy C-means method is used in the classification phase [24]. The main problem of the former method is that the number of classes (K) needs to be initially set. To make our approach more generic, we have tried to automatically determine the optimal number of classes for each TRM. So, we use a splitting algorithm that is coupled with an information criterion as the one of Rissanen type [25]. The main idea of this algorithm is to apply the k-means several times with different values for k and then to retain the value that gives the most pertinent distribution.

Next, we explain some experimentation results that we have already obtained by applying this classification method [26]. In these experiments, we use a video document with thirty-one minutes duration. This video is a TV-game in which two teams, each with two players (speaker #2, speaker #3) and (speaker #4, speaker #5), are playing. This program contains three animators. Speaker #1 is the principal one that animates the game while speaker #6 and speaker #7 are secondary ones that present the audiences that participate to the game and the lot of gifts to be won. One audience is also appearing in the program (speaker #8).

Figures 2 and 3, respectively, represent two experiments of two steps: (1) determining the optimal K number, and (2) classifying TRMs. From an audio point of view, our TV-game video contains eight elementary speaker segmentations that have been manually extracted. While from a video point of view, face segmentations have been automatically extracted using the tool proposed in [27]. A TRM is computed for each couple of speakers (speaker segmentations) which means a total of twenty-eight TRMs for the speaker segmentations. α threshold in its turn has been fixed to 10 seconds. The value of 10 gave more significant results, due to the nature of the document content as we will explain later. The maximum value of the Rissanen criterion has determined the optimal number of classes to 2, and the obtained results are shown in Figure 2.

In Figures 3(a) and 3(b), respectively, we can see the graphical representation of the temporal relations computed between the speaker #4 and the speaker #5 (speaker #2

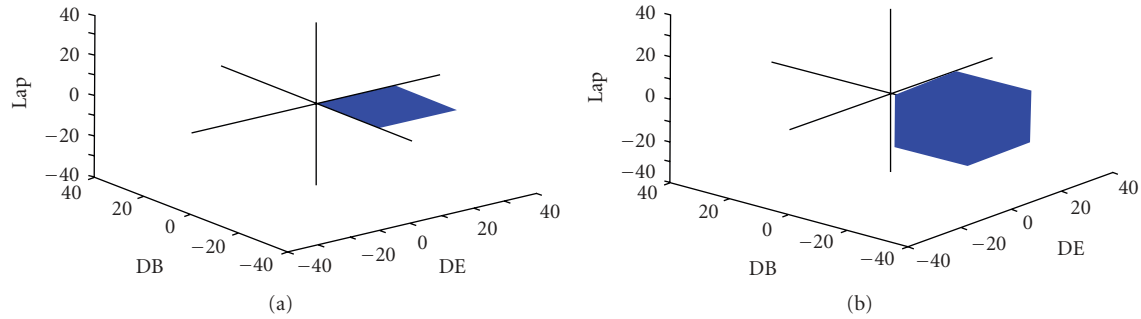

Figure 1: Graphical representation of the MEETS (a) and OVERLAPS (b) relations.

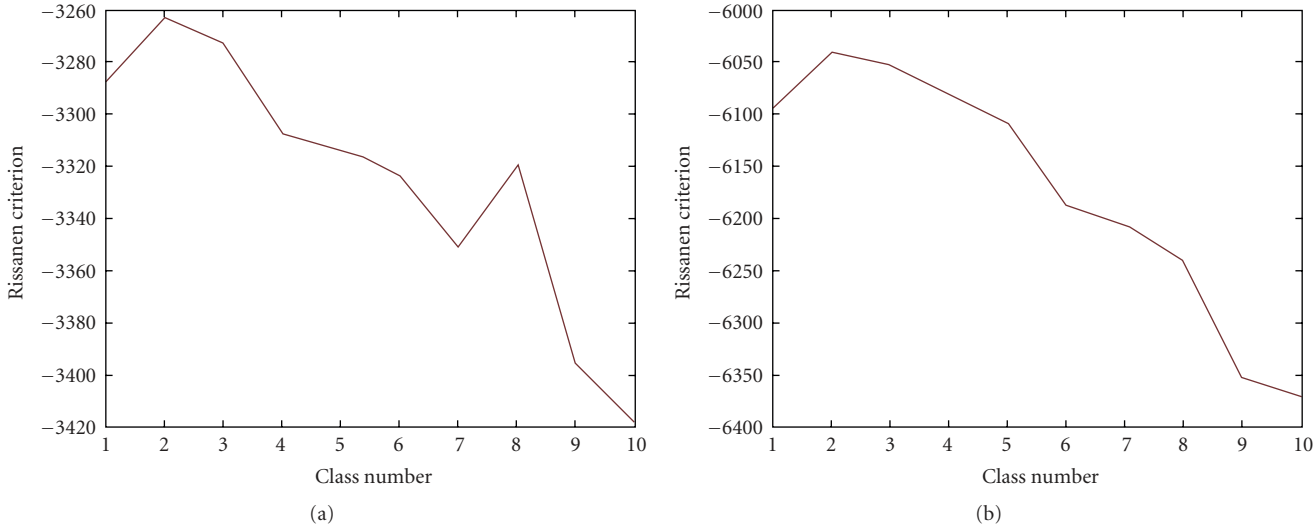

Figure 2: (a) Rissanen criterion value function of the class number for the $TRM_{S(2,3)}$, (b) Rissanen criterion value function of the class number for the $TRM_{S(4,5)}$.

and speaker #3, resp.), on which the result of a 2-class classification has been applied. The TRM in the Figure 3(b) ($TRM_{S(4,5)}$, S means speaker segmentation) contains 450 votes (245 for class C1 and 205 for class C2) while the TRM ($TRM_{S(2,3)}$) in the Figure 3(a) contains 247 votes (123 for class C1 and 124 for class C2).

Table 1 contains the distribution of votes between classes in each TRM where we can notice the previously commented results.

We then apply the same process on faces. Figures 4 and 5 represent two other examples for the face #4 and face #5 (face #2 and face #3, resp.). Figure 4 shows the optimal class number decision (here it is 3 for the two TRMs), and Figure 5 shows the TRMs and the three-class classification results.

Table 2 shows the distribution of votes between classes in each TRM. In this table, C3 is equal to zero when the optimal number of clusters in the corresponding TRM is equal to two.

After applying these first analysis steps, the question that may be asked is: "Are these classes of temporal relations related to more semantic events than those initially used?"

2.5. TRM Content Analysis and Event Detection. Regarding the set of TRMs and the occurrence (vote) numbers of their classes, the question is now to determine if these numbers are carrying any semantic information about the document content. In this section, we give a glance about what this semantic information may be, particularly, in the case of our TV-game video.

The first note that we can make is that, in practice, an empty TRM between two segmentations means no interaction or relevance between them. For example, considering the TRM computed between face and applauses segmentations. An empty TRM (or even low number of votes) means that the applauses segments are not related to the appearance of a face on the screen. In other words, the applauses segments are completely independent from the appearance of a face on the screen. On the other hand, having an important number of votes between two segmentations certainly indicates a specific event that relates them. Returning to the previous example, the high number of votes indicates that the applauses segments are related directly to the appearance of the face on the screen. An additional remark is that, a two-class classification with a quite balanced number of occurrences between them may result in having a kind of exchange between the two segmentations in use. More specifically, considering our TV-game example, further

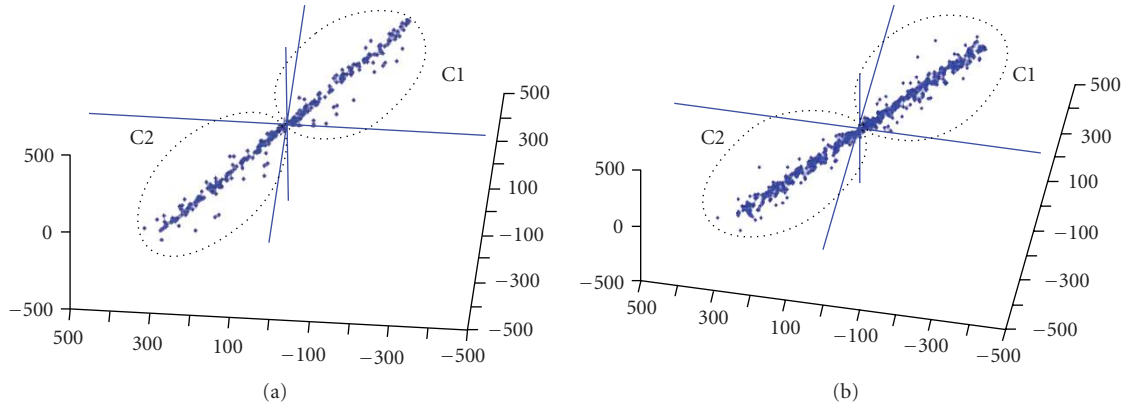

Figure 3: (a) Graphical representation of the $TRM_{S(2,3)}$ and two-class classification results, (b) graphical representation of the $TRM_{S(4,5)}$ and two-class classification results.

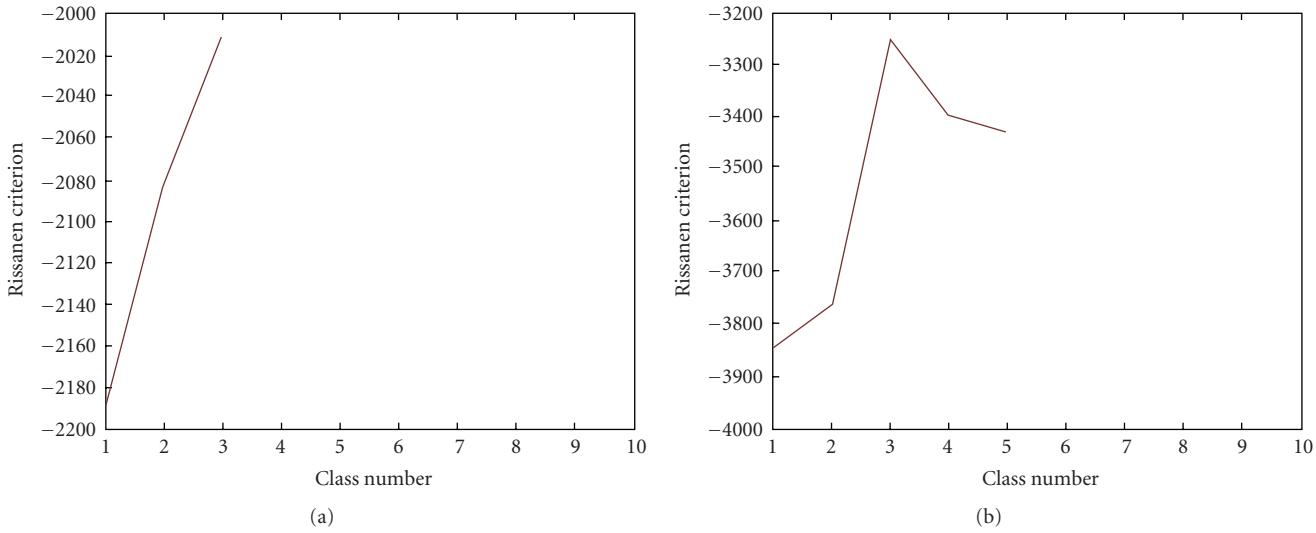

Figure 4: (a) Rissanen criterion value function of the class number for the $TRM_{F(2,3)}$, (b) Rissanen criterion value function of the class number for the $TRM_{F(4,5)}$.

investigations on number, duration, and exchanges alternation would give other clues on the nature of these exchanges (interview, conversation, debate…). Moreover, in case of having a segmentation with no empty TRM with any of the other segmentations, this implies that this segmentation is interacting in a significant way, with each of the other segmentations. This clue is not only interesting from a semantic content point of view, but it also indicates the specific role of this segmentation.

By mapping the previous semantic clues to our TV-game video, several results can be obtained. First, we can notice the high number of votes in $TRM_{S(2,3)}$ and $TRM_{S(4,5)}$. This is due to the fact that in this TV game, these two couples of speakers (2,3) and (4,5) correspond to two teams of players who are playing together and this is why many exchanges between players of each couple can be observed. On the other hand, regarding the nature of the document, a player can take few seconds to think about the answer he or she is going to give. Within a scope threshold of only one second, many temporal relations were missed. For this reason, we tend to raise the α value to 10, which helps us to get more significant results.

Furthermore, by considering $TRM_{S(2,4)}$, $TRM_{S(2,5)}$, $TRM_{S(3,4)}$, and $TRM_{S(3,5)}$, we can observe that they are practically empty. That is because each player of a team has no occasion to exchange words with any player in the other team.

Moreover, in our example, speaker #1 has no empty TRMs and this is logical as it is the animator which interacts with all the other speakers.

More advanced analysis step is applied to retrieve more semantic information about the content. The idea consists in taking two temporal relations belonging to different classes C1, C2, and looking for a third temporal relation that may be the composition (composition operator) of the two previous ones. For example, let C1 represent the event "the speaker A is talking to the speaker B", and the class C2 associated to the event "speaker B is talking to speaker A". A new class of relations may be the result of the composition of C1

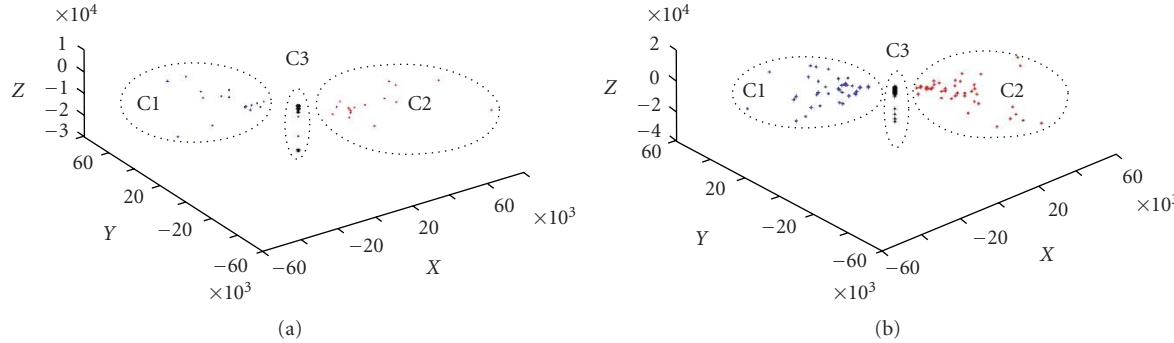

Figure 5: (a) Graphical representation of the $TRM_{F(2,3)}$ and three-class classification results (C1, C2, C3), (b) graphical representation of the $TRM_{F(2,3)}$ and three-class classification results (C1, C2, C3).

Table 1: Distribution of votes between classes in each TRM for speaker segmentations.

TRM	C1	C2	TRM	C1	C2	TRM	C1	C2
$TRM_{S(1,2)}$	65	60	$TRM_{S(1,3)}$	49	49	$TRM_{S(1,4)}$	84	71
$TRM_{S(1,5)}$	106	97	$TRM_{S(1,6)}$	6	5	$TRM_{S(1,7)}$	89	79
$TRM_{S(1,8)}$	3	5	$TRM_{S(2,3)}$	123	124	$TRM_{S(2,4)}$	4	7
$TRM_{S(2,5)}$	6	6	$TRM_{S(2,6)}$	0	0	$TRM_{S(2,7)}$	6	7
$TRM_{S(2,8)}$	0	0	$TRM_{S(3,4)}$	6	5	$TRM_{S(3,5)}$	10	5
$TRM_{S(3,6)}$	0	0	$TRM_{S(3,7)}$	7	4	$TRM_{S(3,8)}$	0	0
$TRM_{S(4,5)}$	245	205	$TRM_{S(4,6)}$	4	8	$TRM_{S(4,7)}$	15	19
$TRM_{S(4,8)}$	0	0	$TRM_{S(5,6)}$	0	0	$TRM_{S(5,7)}$	39	26
$TRM_{S(5,8)}$	0	0	$TRM_{S(6,7)}$	1	0	$TRMs_{(6,8)}$	0	3
$TRM_{S(7,8)}$	4	3						

and C2 (C1∧C2) which represents the event "A is talking to B who is then talking to A". Using such type of operations may put forward more complex pattern. Several compositions can be hierarchically performed to detect really consecutive exchanges. By marking the beginning and the end of such sequences with appropriate indexes or descriptors, parts of the document content can be indexed as zones of oral interactions involving two persons. This is not only highly semantic information about the document content, but also information about the document structure. Applying this analysis step on the TV-game video, we have retrieved all the consecutives exchanges (conversations) between speakers. The longest exchanges were observed between the couple (speaker#2, speaker#3) and (speaker#4, speaker#5) [26, 28].

As a last example, to illustrate the structuring aspect, we have introduced a new elementary segmentation of the same document containing applause segments. All the TRM between this new segmentation and each one of the speaker segmentation has been computed and classified in two classes. The composition operation is applied several times and then has put forward exchange zones ended by applause. These patterns correspond to the different game phases. Considering the duration of this last segment (patterns), short game phases can be differentiated from main phases of the game. The detection of such events is really important in terms of document structuring. We have applied our approach on several video documents in order to detect events and to identify their structures. The obtained results are presented and explained in details in [26].

The video structuring approach is used in the scope of a three-year ANR project started in 2008 (National Research Agency project: EPAC) on Masses of Data-Ambient Knowledge in which the involved task is the document content analysis and structuring.

All the processes described above through these few examples can help us to climb a new step in the semantic analysis of audiovisual document content and to detect more complex motives. Resulting from the aggregation of more basic events, as illustrated in previous examples, these motives are themselves clues about the temporal structure of a document. Then, if an audiovisual document can be described by its temporal structure, can it be compared to another one on such structuring features? This leads us to the second objective of our work, which is to make document clustering on the basis of the documents temporal structure.

3. Similarity Measure Based on the TRM Representation

The first step of the document clustering process is to define a similarity measure thanks to which differences or resemblances between audiovisual documents might be evaluated. This will then help to process large sets of documents by sorting and clustering them according to their similarity. Generally, similarity is expressed by means of a distance

Table 2: Distribution of votes between classes in each TRM for face segmentations.

TRM	C1	C2	C3	TRM	C1	C2	C3	TRM	C1	C2	C3
$TRM_{F(1,2)}$	25	15	0	$TRM_{F(1,3)}$	10	10	0	$TRM_{F(1,4)}$	19	19	5
$TRM_{F(1,5)}$	19	13	2	$TRM_{F(1,6)}$	9	19	0	$TRM_{F(1,7)}$	2	1	8
$TRM_{F(1,8)}$	2	1	0	$TRM_{F(2,3)}$	*38*	*36*	53	$TRM_{F(2,4)}$	11	7	0
$TRM_{F(2,5)}$	3	2	0	$TRM_{F(2,6)}$	6	6	0	$TRM_{F(2,7)}$	2	1	0
$TRM_{F(2,8)}$	0	0	0	$TRM_{F(3,4)}$	1	2	0	$TRM_{F(3,5)}$	1	0	0
$TRM_{F(3,6)}$	4	4	0	$TRM_{F(3,7)}$	1	0	0	$TRM_{F(3,8)}$	0	0	0
$TRM_{F(4,5)}$	*52*	*50*	76	$TRM_{F(4,6)}$	14	5	0	$TRM_{F(4,7)}$	4	1	0
$TRM_{F(4,8)}$	4	5	0	$TRM_{F(5,6)}$	22	14	0	$TRM_{F(5,7)}$	1	1	0
$TRM_{F(5,8)}$	4	5	0	$TRM_{F(6,7)}$	2	2	0	$TRM_{F(6,8)}$	1	0	0
$TRM_{F(7,8)}$	2	1	0								

computed in a specific feature space (a metric space) [29]. The shorter this distance is, the higher the similarity, or the lower the dissimilarity, between the compared objects will be.

Measuring document similarity is a question that has already been addressed in different works. From the one side, similarity depends on the kind of features that are used to compute its value. On the other side, it depends on the underlying task to which it is done.

From the kind of features point of view, they are drawn from three modalities: text, audio, and video. Regardless of the features used, some methods have defined the video similarity as the ratio of similar images or subsequences between two documents [30]. In such methods, similarity is computed on multidimensional feature vectors extracted from key-frames or video shots. Here, similarity measures may be based on textual features (closed caption, transcript of the dialog, ...), visual features (color, texture, shape, activity rate...), audio features (silence detection, speech, music, ...), or multimodal ones. In some works, the video similarity bases on the extraction and characterization of points of interest by local path descriptions using temporal and spatial features for video copy detection [31]. While in [32, 33], the similarity estimation rates are obtained from chronological series and dynamic programming in order to segment and structure video documents basing on multimodal features. The proposed similarity methods in [34–36] base on *a priori* models, while the similarity values are computed without any previous knowledge to detect repeated objects in the multimedia streams [37].

From another point of view, similarity measure can also differ according to the kind of the underlying task to carry out: document retrieval and classification (i.e., [30, 38–43]), document segmentation and structuring (i.e., [37, 44–46]), TV stream structuring (i.e., [47]), or document copy identification (i.e., [31, 48]). For example, Foote and cooper. [46] define a similarity measure in order to compute a similarity matrix using a set of well-chosen features. This similarity matrix allows a visual representation of the structural information of a video or audio signal. The structure of the similarity matrix can be analyzed to find structure boundaries. In contrast, the aim of the work proposed in [39] is the video categorization. Each video is represented by a multidimensional series of multimodal features. The videos are then classified using the SVM classifier. The SVM classifier is used also by Manson and Berrani. in [47] but the aim of their work is different. During the TV stream macrosegmentation process, a TV program may be split in several parts. The aim of the work of Manson et al. is to provide a method to fuse consecutive program segments that belong to the same program. A set of visual features is extracted from each pair of consecutive programs and then used within an SVM classifier in order to decide if they should be fused or not.

The literature is very rich in methods proposed for video classification. The reader can refer to the survey of Brezeal and cook. in [48] for more information about the features and the video similarity approaches proposed or to the phds of Haidar and Ibrahim [26, 49].

As it is explained in the previous section, the results of our temporal analysis of audiovisual document content are independent from the kind of features and the type of documents (α value set to one for all documents). Consequently, we aim at using the temporal relations to compute similarity between audiovisual documents. Our video similarity method differs from the existing ones in two important points.

(1) Our method can be applied on any video type, and we do not need to select specific pertinent features for document clustering or using a weighting method. We use the set of available features. In other words, we do not base on any *a priori* knowledge about the video content in order to select the set of the most relevant features or to compute a weight for each of them.

(2) The second point is related to the features used in the classification process. Any video classification (clustering, resp.) method should provide classes (clusters, resp.) where each class (cluster, resp.) should contain videos that share the same structure. Thus, the features used for video classification should represent information about the structure of the document. To our knowledge, this axis is not well addressed in the literature. Usually, the similarity

between two documents A and B was defined as a weighted sum of distances D_i ($i = 1, 2, \cdots N$ where N = number of features used). Each distance D_i is computed between the feature F_i belonging to the document A and the same feature F_i belonging to B. In such a way, we do not consider, when computing the distance, the temporal relations or interactions that may present between the heterogeneous features (F_i and F_j), which may hold useful information about the documents structure. For example, given the two documents A and B and the two features F_1 and F_2 where F_1 represents the appearance of the newscaster on the screen, and F_2 represents the reports segments, the temporal relations presented between F_1 and F_2 give structural information about the document (news or not). Even though the existing approaches tend to compute the distance as the fusion of two distances D_1 and D_2, each one is computed between the same features where each feature belongs to one of the two documents. By this way, the structure of the document that is represented as the presence of some specific temporal relations between F_1 and F_2 was not considered at all because each feature is processed independently from the others. In our case, the basic features that feed into the similarity measure are the temporal relations observed between heterogeneous segmentations of the document. Consequently, our proposed measure for video classification (clustering, resp.) is based on features that may represent efficiently the structure of video documents. The results presented later show the efficiency of this video representation in the classification process.

3.1. Distance between Two Documents. As it is explained in Section 2, temporal relations are observed between segmentations and represented by a set of TRMs. For each TRM, a set of classes (classes of temporal relations) is identified and then each class is represented by the number of its relations-occurrences.

Let us consider a set Seg, of N segmentations that are automatically or manually extracted from an audiovisual document. This set can be defined as $\text{Seg} = \{\text{Seg}_1, \text{Seg}_2, \ldots, \text{Seg}_N\}$, where Seg_i is an audio segmentation (speech, speakerX, applause…) or a video one (color, texture…). In this case, M TRMs are computed per document ($M = N * (N-1)/2$), and a document D^i can be described by the set of its TRMs, here called TRMSD^i:

$$\text{TRMSD}^i = \left\{\text{TRMD}_1^i, \text{TRMD}_2^i, \ldots, \text{TRMD}_M^i\right\}. \quad (4)$$

Furthermore, a classification method is applied on each TRMD_j^i (TRM #j of the document D^i with $1 \le j \le M$). Let NCD_j^i be the number of classes (NC) in the TRMD_j^i. The number of classes may be different from TRM to another. Whatever this number is, each TRMD_j^i can be described by a set of NCD_j^i classes of temporal relations (CR),

$$\text{TRMD}_j^i = \left\{\text{CRD}_{j1}^i, \text{CRD}_{j2}^i, \ldots, \text{CRD}_{jLij}^i\right\} \quad \text{with } Lij = \text{NCD}_j^i. \quad (5)$$

Each class of relations CRD_{jk}^i, (class of relations #k in the TRMD_j^i which contains NCD_j^i classes with $1 \le k \le \text{NCD}_j^i$) has its own number of occurrences of temporal relations NRD_{jk}^i. Therefore, a TRM is represented by a set of real values, where each value is equivalent to the number of occurrences of a class. Mathematically, each TRM is represented by a vector of real values. The size of this vector equals to the number of classes in the TRM,

$$\text{TRMD}_j^i \approx \text{NRSD}_j^i = \left\{\text{NRD}_{j1}^i, \text{NRD}_{j2}^i, \ldots, \text{NRD}_{jLij}^i\right\} \quad \text{with Lij} = \text{NCD}_j^i. \quad (6)$$

An analogous method for classifying TRM is to discretize the 3D space in a set of predefined subspaces. The subspaces may correspond to predefined models such as the Allen's relations (refer to Section 2.3). They may be chosen randomly (i.e., the number of subspaces and their limits), where their number may also differ from TRM to another. As for the classes, each sub-space in each TRM is represented by the number of temporal relations contained in it.

Our aim is to find a way to compare two audiovisual documents (A and B) based on the extracted information from the last steps. Therefore, two levels of comparison have to be applied as we explain later.

In a first level, The TRMs of the document A (TRMSD^A) are compared to those of B (TRMSD^B). The results of this level are then combined in the second level to obtain an overall comparison. Two TRMs are compared if both are built on the same couple of segmentations (TRMD_j^A is compared with TRMD_j^B for each j). Comparing two TRMs that have not been built on the same couple of segmentations is nonsensical. In this paper, we compare the TRM_j^A to the TRM_j^B only if both have the same number of classes ($\text{NCD}_j^A = \text{NCD}_j^B = \text{NC}_j$). In other words, two vectors are compared if they have the same size. However, in order to compare two TRMs that do not have the same number of classes, different solutions have been proposed but not yet tested: the first is to fix a priori a number of classes for each TRM or one for all the TRMs. The second is to determine the optimal number of classes on each TRM of one video and then use these numbers for the same TRM (the TRMs built on the same couple of segmentations) of the other videos. The more advanced solution is to determine the optimal number of classes for all the TRMs of all the videos and then choose the majority number presenting in the same TRMs.

The TRMs classification leads us to consider each NRSD (NRSD_j^A or TRMD_j^B) as a vector. Two TRMs are compared by computing a distance between the corresponding vectors

(NRSD_j^A and NRSD_j^B) or more precisely between the vectors normalised by the document length (t_A for document A and t_B for document B). Then the global distance $d(A,B)$ between the two documents A and B is computed as the sum of the M normalized distances:

$$d(A,B) = \sum_{j=1}^{M} \alpha_j \left[\sum_{k=1}^{NC_j} \beta_k \left| \frac{\mathrm{NRD}_{jk}^A}{t_A} - \frac{\mathrm{NRD}_{jk}^B}{t_B} \right| \right]. \quad (7)$$

Using the two weight vectors α and β, we can give more or less importance to certain features (TRMs and classes of relation in the TRMs).

We make some new experiments on the set of audiovisual video documents composed of several collections. A collection here is a subset of documents belonging to the same category. A category contains documents of the same type (i.e., news, sports, documentary…). Some video categories may also be composed of several subcategories. For example, a sports collection may be subcategorized in soccer, basketball, volleyball, and so forth. In our case, the news collection contains several subcategories where each is produced by a TV channel (France2, ABC, CNN…). The subcategory will help us to show that our method is able to put the subcategories that belong to the same category in the same collection (i.e., the different subcategories of news collection). However, the purpose of these experiments is to figure out whether or not, the distance between audiovisual documents gives relevant clues that help in classifying documents and automatically organizing them into categories.

3.2. Experiments on Distance between Documents. Two hundred-forty-two audiovisual documents of different categories have been chosen to constitute our experimental video corpus (VC). Details about the categories of documents, the dataset they belong to, the subcategories in each dataset (if any) and the number of documents in each subcategory are given in Table 3 with the minimum, maximum, and average documents duration (hh:mm:ss).

In Table 3, four of the twenty soccer sequences are belonging to the same match, and the movie extracts are coming from the same movie.

C = {*7 different values for the dominant colour, 3 different values for the mean of luminance, 4 different values for the contrast, 3 different values for the activity rate, shot transition, speech, music, noise, applause, laugh*} is the set of 23 features used as elementary segmentations (18 from the video component and 5 from the audio).

This set of features represents the available segmentation tools in our research team. Studies about the pertinence and relevance of the used features for video classification are beyond the scope of this paper. Our aim is to show that basing on some available features (whatever they are or their number), the observation of temporal relations that may present between heterogeneous features may be very useful to classify video documents basing on their structures. Nevertheless, we should not ignore the impact of the features used on the clustering process. This impact will be studied in future works.

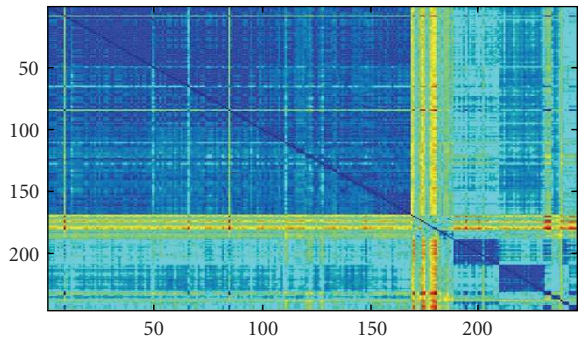

Figure 6: Similarity matrix between documents.

For each document, 253 TRMs have been computed. After TRMs computation, we calculate the distances between each pair of documents where we set the weights α and β to 1. To simplify the video classification task, we have chosen, in this paper, to use Allen's discretization of the 3D space to represent the temporal relation in spite of determining the optimal number of clusters in each TRM, classifying the votes, and then representing each cluster with the number of votes it contains. More advanced works that base completely on TRM-classification will be presented in our future works.

By this way, each TRM is represented by the number of occurrences of the thirteen Allen's relations. In other words, each TRM is represented by a vector of thirteen entries, where each entry corresponds to the number of occurrences of one of the Allen's relations. In order to get a more normalized distance between TRMs, we have chosen to transform the vector to a binary vector and discard the video duration. As a consequence, each TRM will provide the information whether or not Allen's relations are present in the video, and the proposed distance becomes

$$d(A,B) = \frac{1}{M} \times \frac{1}{13} \sum_{j=1}^{M} \left[\sum_{k=1}^{13} \left| \mathrm{NRD}_{jk}^A - \mathrm{NRD}_{jk}^B \right| \right]. \quad (8)$$

We use this distance in different experiments in which the purpose is to study how document set classification (supervised and nonsupervised) could be handled according to document temporal structure.

3.3. Documents Representation Using Similarity Matrix. A similarity matrix is a data representation that allows us to compare immediately and visually two documents or video streams contents. In our experiment, each line and each column of the similarity matrix SM are associated to one document of the previously described corpus. The matrix is a set of coefficients C_{ij}, where each represents a distance between two documents of our corpus,

$$\mathrm{SM} = \left\{ \frac{C_{ij}}{C_{ij}} = d\left(D^i, D^j\right) 1 \le i \le Nb\,\mathrm{doc},\ 1 \le j \le Nb\,\mathrm{doc} \right\}. \quad (9)$$

In Figure 6, we can visualize the matrix built on the video corpus VC and showing similarity between contiguous document collections. On the axis are the numbers associated

Table 3: Experimental corpus description.

Category: dataset	Subcategory	Number	Duration		
			Mean	Min	Max
TV news: TREC2003	ABC	49	00:34:20	00:32:50	00:35:50
	CNN	49	00:34:20	00:32:00	00:37:20
TV news: TREC2004	ABC	6	00:34:00	00:33:50	00:34:10
	CNN	6	00:34:40	00:34:10	00:35:55
TV news: TREC2005	CCTV	9	00:53:50	00:34:00	01:10:00
	CNN	7	00:54:30	00:34:00	01:10:00
	LBC	8	00:54:40	00:30:00	01:10:00
	NBC	7	00:33:00	00:28:40	00:34:00
	MSNBC	6	00:34:00	00:34:00	00:34:00
	NTDTV	4	00:34:00	00:34:00	00:34:00
TV news: Argos	France2	17	00:39:40	00:24:40	00:44:40
Sports	Soccer	20	01:26:10	00:25:00	02:45:00
Documentary	Documentary	21	00:29:10	00:14:00	01:05:10
TV series	Stargate	24	00:42:18	00:39:55	0042:20
French TV games	Les amours	5	00:31:55	00:30:30	00:36:00
Movie extracts	Matrix	4	00:31:20	00:24:00	00:38:30
Total duration		242		6 d:6 h:4 m:27 s	

to each document of each collection or subtype: ABC_2003 from 1 to 49; CNN_2003 from 50 to 98; ABC_2004 from 99 to 104; CNN_2004 from 105 to 110; CCTV_2005 from 111 to 119; CNN_2005 from 120 to 126; LBC_2005 from 127 to 134; NBC_2005 from 135 to 141; MSNBC_2005 from 142 to 147; NTDTV_2005 from 148 to 151; JT_INA from 152 to 168; soccer from 169 to 188; documentary film from 189 to 209; TV series from 210 to 233; TV games from 234 to 238; matrix movie extracts from 239 to 242.

The black or dark colors are the higher values of the similarity measure computed between two documents (low distance), while lower values appear in bright color. The darkest color along the diagonal corresponds to similarity values resulting from the comparison of a document with itself (intersimilarity). It can be clearly seen that there are 3 dark blocks in the similarity matrix. These blocks correspond to the news videos, the documentary films, and the Stargate TV series. We can see also a block of videos which is very dissimilar from the others and less dissimilar between them. This set corresponds to soccer videos. Additionally, we observe small blocks in the right bottom of the figure which correspond to the TV game videos and the movie extracts. This is the kind of observations that can be made visually with a similarity matrix. Deeper analysis methods are proposed in the literature to extract automatically clusters from a similarity matrix. The automatic analysis of such matrices is not studied in this work. It will make a part of our future work.

3.4. Document Classification. Using the proposed distance again, two types of experiments on the same set of documents are done. In the first type, we use the k-means and the complete-link hierarchical agglomerative clustering methods to cluster the video documents. In the second one, a set of supervised methods are applied (SVM, random forest, classification tree, KNN, C4.5, CN2, and naïve Bayes). Moreover, we propose our own supervised classification method. The aim in this latter type is not to compare the different classification methods but to show that our proposed method (supervised-classification one) provides results that are very close to the well-known classifiers. On the other hand, we also show how by using our structural-features (the TRMs representation), we obtain good results whatever the used classifier is. In the next section, we start with the experiments based on the clustering (nonsupervised) methods.

3.4.1. Unsupervised Classification. The hierarchical agglomerative and the k-means clustering methods are presented in this section.

Document Clustering Using an Iterative and Hierarchical Clustering Method. In this experiment, we have tested the three well-known hierarchical agglomerative clustering algorithms, the complete-link, the average-link, and the minimum-link. We have adopted the complete-link algorithm as it gave us the best results and it was the most adapted one for separating video categories. For more information about the clustering methods, reader can refer to the review of Jain et al. [50] and Bishop [51]. The first step of the hierarchical algorithm is to consider a number of clusters that equals to the number of the documents. So, we start with 242 different clusters. Later on, in each step the algorithm merges the two closest clusters. At the beginning of the process, the two documents that are the most similar according to our similarity measure are put together in the same cluster. This method yields a *dendrogram* that represents the nested grouping of patterns and similarity levels at which

clusters change. In its turn, this *dendrogram* can be broken at different levels to yield different clusters of the data. To facilitate the comprehension, we use the word stage to refer to the number of clusters. Stage k-means that we have cut the *dendrogram* at the point that gives k clusters. We will not comment on all the steps but the most significant ones. We have 6 video categories, so the *dendrogram* is cut to get 6 clusters (stage 6) (Figure 7).

As shown in Figure 7, the algorithm merges some documents of different categories in the same cluster while spreading some documents of the same category over several clusters. At this stage, the news, the TV series, the TV game, and the movie documents are merged in the same cluster while the documentary films (except two) are merged with some soccer documents. We can notice here that the soccer documents are split in several clusters. By this way, they have consumed four clusters instead of one.

In Figure 8, we return backward to stage 9 (9 clusters) to see how videos are spread over clusters. We discover that the documents have started to be clustered correctly. The only exception is the TV game, the soccer, and the movie extracts documents.

The first cluster represented in Figure 8 by the light blue color contains all the news video documents except three of them which are put in another cluster. When searching the reason in the video corpus, we have observed that these three videos have an acquisition problem. Two of them are composed of a single frame while the third starts normally and then record the same frame till the end of the video. This is a very interesting information since we had no idea about this abused videos before clustering. This result shows the possibility of adapting our method to serve as a prefiltering step of video databases.

Returning to soccer documents, they are distributed among five clusters as we can see in Figure 8 (brown color in clusters 3, 4, 5, 6, 7). We also searched the reason in the video corpus. We have observed that some videos are not composed of only soccer content. Some of them contain other content such as interview, movies, comedy series, and soon. Thus, the obtained distribution over the five clusters is due to what is recorded with each video. The cluster number 5 has collected the videos that are only composed of soccer content. The other clusters contain videos that are recorded with pregame studio analysis and interviews, movie, or a comedy series content. The way the nonpurely soccer videos are clustered depends also on the type of nonsoccer content the videos contain. For example, the cluster number three contains the videos recorded with the pregame analysis interviews while the fourth contains the soccer game recorded with 15 minutes of a movie and 10 minutes of a French comedy series. The nonsoccer video contents recorded at the end of the soccer video are not of short duration. We have noticed that they vary from 10% to 20% of the video document duration. That is why they have such effect on the clustering process.

The documentary films are in the same cluster except two documents that are merged with the Stargate TV series cluster. The first document is a documentary about the space. In this documentary, most of the shots talk about the

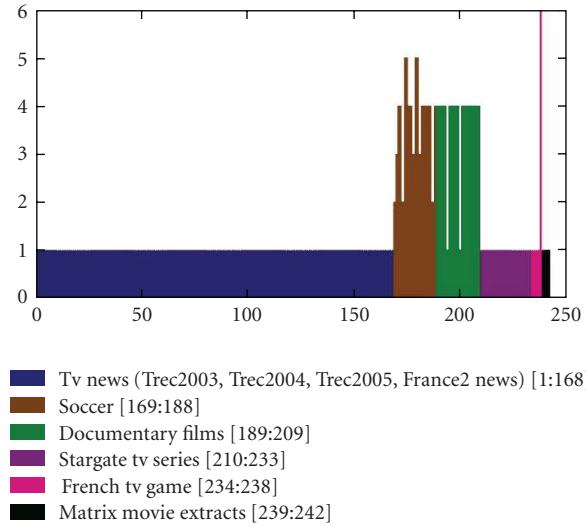

Figure 7: Hierarchical clustering-6 clusters.

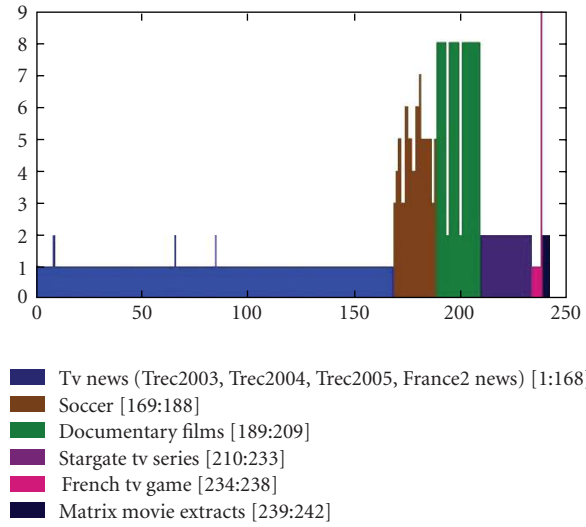

Figure 8: Hierarchical clustering-9 clusters.

space and are recorded in a studio with space instruments. This is what we find also in the most of the Stargate TV series. The second one is composed of scenes filmed in a studio. The number of scenes with different backgrounds is limited. It may be similar to Stargate documents in the way they are produced (studio, number of scenes with different backgrounds, dark scenes...).

The cluster number 2 contains the whole set of TV series documents. It contains also the two documentary films mentioned before and the four movie extracts. The fact that matrix movie extracts are clustered with the Stargate TV series may be due to two reasons. The first one is the presence of lights of fire combats (several combat scenes). The second one is that most of the scenes are filmed in dark places or during nights.

At this stage, we can also observe that the TV game documents are split in two clusters. Four of the videos

are clustered with the news documents and one TV game remains alone in a cluster. The similarity between the news video documents and TV game is due to the way they are produced. In a TV game, you can find the structure of question/reply which is the core of interviews in the news. That is why at this stage, these two types are merged in the same cluster. The TV game document which is clustered alone is recorded with film-content (not purely a TV game).

The first question that we have asked is why different clusters were not separated properly at this stage.

To search for the reason, we have returned several stages backward in the clustering process. We have observed that TV game documents were in the same cluster and that they are merged with the news cluster at stage 13. At this stage, we have observed about seven clusters, each of them containing one soccer video document. It can be clearly seen on the similarity matrix that the soccer videos are not similar between them as the other categories (news, Startgate series, documentary films, movie extracts). That is why the algorithm starts to merge video clusters of different categories before merging the soccer document clusters. The dissimilarity of soccer video documents is due to the fact that some of them contain other content (contain parts of different type).

To avoid such problems, we have proposed a new extension of the previously defined distance. The idea is to normalize each distance between two documents D1 and D2 with the sum of distances between D1 and the remaining documents. In other words, we construct a contextual distance between the set of videos which takes into account the distances between the documents and the remaining documents to be clustered. It is a normalized version of the previous defined distance between videos, and it is defined as follows: let $D = \{Di/i = 1 \cdots n\}$ be the set of documents to be clustered. The distance d between the documents D^i and D^j is defined as follows:

$$d_n\left(D^i, D^j\right) = \frac{d(D^i, D^j)}{\sum_{k=1}^{n} d(D^i, D^k)}, \tag{10}$$

where $d(D^i, D^j)$ is the previously defined distance.

Figure 9 displays the similarity matrix built on the normalized distance between the set of videos. As we can see, the blocs corresponding to the same category are clearer.

In its turn, Figure 10 shows the result of documents clustering in 6 clusters using the new distance.

By using this normalized version of distance, clusters will be clearer, and we can observe that the news video documents are clustered together except for the three videos that we have mentioned before which contain problems.

The soccer videos are also clustered together except for three videos that contain video segments of different content. The first soccer video of one hour duration contains 10 minutes of a movie. The second one of two hours duration contains twenty minutes of nonsoccer sequence and ten minutes of a movie. The third video of two hours and twenty minutes duration contains 10 minutes of a French comedy series and thirty five minutes of a movie. As we can notice, these video segments recorded with the soccer documents are not of short duration as mentioned before. That is why

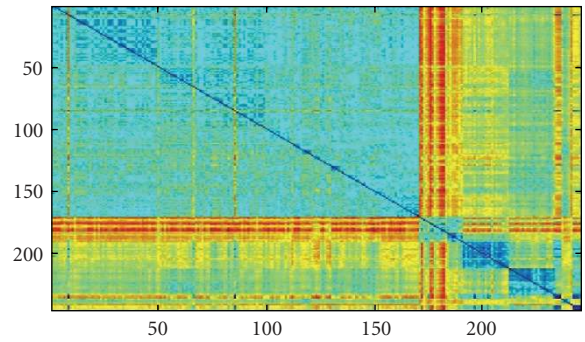

Figure 9: Similarity matrix between documents.

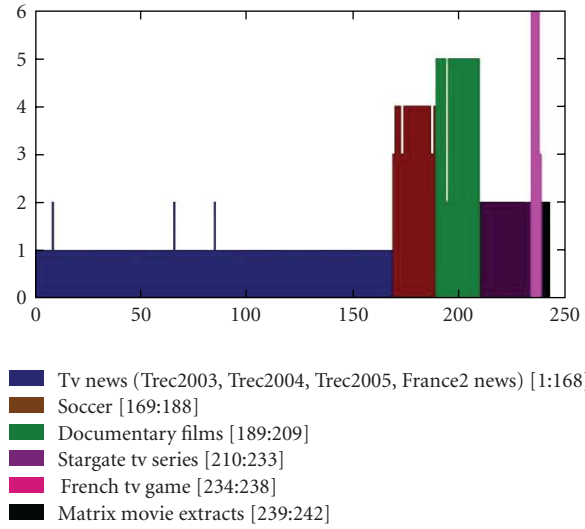

Figure 10: Hierarchical clustering-6 clusters.

the algorithm has put together these documents in the same cluster.

The documentary films are together in the same cluster (cluster five) except the document, previously mentioned, that talks about the space.

The TV series cluster contains also the movie extracts. We have presented before the possible reasons that may merge these two video types together.

The fifth cluster contains four of five TV game documents. The cluster number three has collected the remaining TV game document (the one containing a movie part) and three soccer video documents (the ones containing also movie parts). The TV game document is merged with these soccer video documents for two reasons: the first is that they are recorded with the same type of video parts (movie part). The second reason is that they share the same structure (green dominant color, explosion on the sound track, applauses...).

For more information, at stage 5 (5 clusters), the TV game documents are merged with the news documents while stage 7 shows the same distribution as stage 6 (Figure 9) except for the soccer videos which are split in three clusters (2 clusters at stage 6).

Table 4: F-measure of the clustering—6 clusters.

Clustering results	F-measure (%)	Misclassified in the class	Misclassified out the class
News	99.1%	0	3/168
Soccer	91.9%	0	3/20
TV series	92.33%	4	0/24
Documentary	97.7%	0	1/21
Tv games	88.9%	0	1/5
Movie extracts	0%	0	4/4

Table 4 presents the F-measure of the clustering process at stage 6. As shown in this table, 230 video documents are clustered correctly (**95.1**%). The misclassified videos are, from our point of view, justified.

Document Clustering Using K-mean. We have also tested the k-means clustering method in order to see if it can separate the video types using the defined distance. In this experiment, we have applied the k-means method to cluster the video collection in six clusters (number of video categories). The centers of the clusters are chosen randomly. Figures 11 and 12 show that such a method provides poor results.

As a conclusion, we can say that the complete-link hierarchical algorithm is the best unsupervised classifier used (among the tested ones) with our proposed normalized distance to cluster the video documents. It is the most performant one to separate the clusters of the video categories. Another point that has to be highlighted is that some video documents are not correctly clustered. This was justified by the next two problems that are related to the used video corpus.

The first problem is an acquisition problem. The three news documents are the examples. They should not be considered as miss-clustered. Contrarily, they should be removed from the corpus or at least go through a filtering step before being used.

The second problem is that some video documents contain significant video parts of different content (about 15% of the video duration at the beginning or at the end). This problem occurs when we schedule the recording of a program to begin automatically based on the program guide which is not precise. In this case, we are obliged to schedule the start several minutes before (resp., after) the announced start (resp., end) in the program guide. Nowadays, several works are proposed to correct the electronic program guides in order to obtain the correct TV programs boundaries [52–54].

In the next section, the results provided by several supervised classification methods are presented.

3.4.2. Supervised Classification. Here we present two groups of supervised classifiers. In the first group, we have proposed our own method to train video models and then to compare video documents with the models. On the other hand, the second group is composed of a set of well-known classifiers.

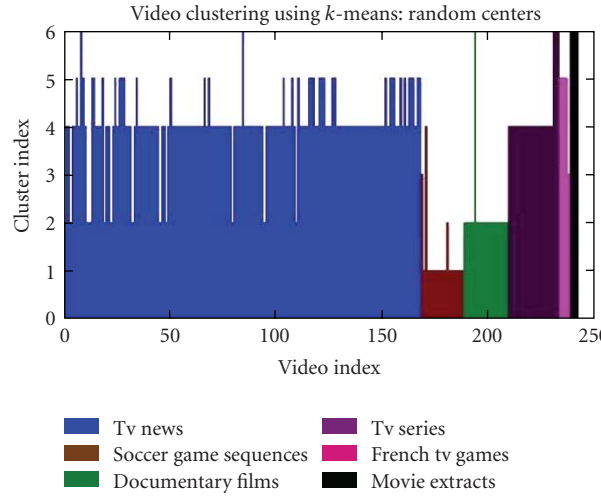

Figure 11: K-means clustering-6 clusters.

A New Proposed Supervised Method. In this paragraph, we propose our own supervised method for audiovisual document classification. This method consists in creating a video model for each video category and then assigning each video document to the nearest model. A model for each video category is created as follows.

As presented before, each document D^i is represented by a set of TRMs. Let TRMSD^i be this set for document D^i:

$$\text{TRMSD}^i = \left\{\text{TRMD}_1^i, \text{TRMD}_2^i, \ldots, \text{TRMD}_M^i\right\}. \tag{11}$$

Each TRMSD_j^i in the TRMs set is represented by the vector of number of occurrences of its clusters: $\text{TRMSD}_j^i = \{\text{NRD}_{j1}^i, \text{NRD}_{j2}^i, \ldots, \text{NRD}_{jLij}^i\}$, where the number of clusters in a specific TRMSD_j^i should be the same in all the TRMSD_j^k.

Let Set $= \{D^1, D^2, \ldots, D^N\}$ be the set of documents used to train a model M,

$$M = \frac{1}{N}\sum_{i=1}^{N} D^i, \tag{12}$$

where the sum of the two documents D^i and D^j is defined as follows:

$$D^i + D^j = \left\{\mathrm{TRMD}_1^i + \mathrm{TRMD}_1^j, \mathrm{TRMD}_2^i + \mathrm{TRMD}_2^j, \ldots, \mathrm{TRMD}_M^i + \mathrm{TRMD}_M^j\right\},$$

$$\mathrm{TRMD}_k^i + \mathrm{TRMD}_k^j = \frac{1}{2} * \left\{\mathrm{NRD}_{k1}^i + \mathrm{NRD}_{\mathrm{k1}}^j, \mathrm{NRD}_{\mathrm{k2}}^i + \mathrm{NRD}_{\mathrm{k2}}^j, \mathrm{NRD}_{kLki}^i + \mathrm{NRD}_{kLkj}^j\right\}. \quad (13)$$

To train the models, we have chosen randomly ten percent (10%) of the video documents in the corpus. Some document categories in the corpus do not contain a large number of videos such as the movie extracts (4 documents) and TV game (5 documents). To avoid this problem in the test step, we decided to train the models on a part of the video categories (10%) that are chosen randomly and then to test the models using the whole video corpus (including the 10% used for training). We have iterated the training and testing process ten times. At the end of the iterations, each video is assigned to the frequently associated model during the ten iterations. Figure 13 shows the obtained results. Three of the news videos are misclassified. These are the previously mentioned videos that contain the acquisition problem. Two soccer videos that contain significant nonsoccer video content are misclassified in the documentary and TV game classes. There is also a TV game video (three documentary films, resp.) misclassified in the TV series class (film extracts class, resp.).

Table 5 shows the F-measure of the proposed method. It also shows the number of video documents of different types that are presented in the cluster (Column 4 and the video documents misclassified in other clusters (column 5) over the total number of documents of this type. As a conclusion, we can see that in total, 231 video documents (**95.5**%) are clustered correctly.

Well-Known Supervised Classifiers. In the literature, several supervised classification methods exist. In this paragraph, we select the most well-known ones in order to test our video representation for clustering (SVM, CN2 rules, random forest, classification tree, C4.5, KNN, naïve Bayes). Our experiments have been done using the Orange data mining software (Orange). As mentioned before, our aim is not to compare the results of the classifiers. In contrast, these experiments show that for most of the well-known classifiers, we obtain good results which put light on the effectiveness of our video representation (representation of the structure and the content). Moreover, the obtained results are close to the ones obtained by applying our proposed supervised-classification method.

In these experiments, we have chosen to represent each TRM in each video by one value. In other words, we have chosen to set the number of clusters in each TRM to one. These experiments show that even if we consider only the number of votes in the TRM without any classification, this representation (one value per TRM) has no significant effect on the video classification.

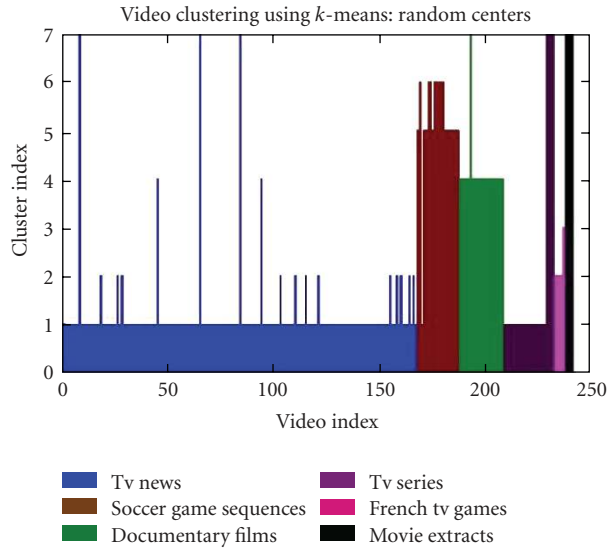

Figure 12: K-means clustering-7 clusters.

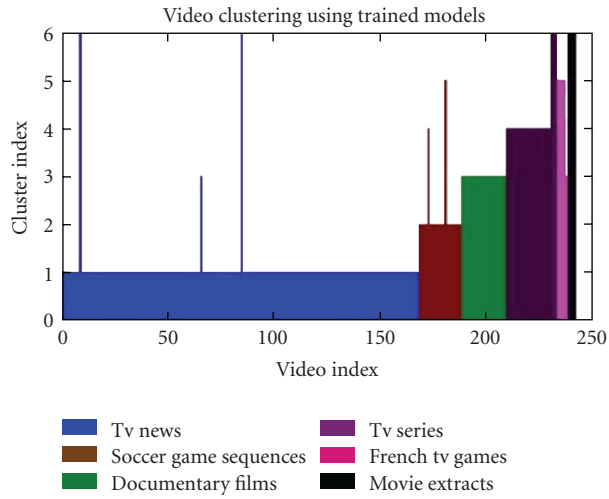

Figure 13: supervised clustering-6 classes.

Using the above representation, each video will be represented by a vector of 253 entries, where each entry counts the number of observed relations (points in the 3D space) in a TRM. Different sampling methods have been also applied (cross-validation, random splitting, leave one out, test on train data).

Three-Folds Cross-Validation Sampling Method. Table 6 shows the F-measure of the supervised classification using a three-folds cross-validation sampling method. The cross-validation method splits the data into the given number of folds (here equals to 3). The algorithm is tested by holding out the examples from one fold at a time; the model is induced from the other folds, and the examples from the held out fold are classified.

Table 5: *F*-measure of the proposed supervised method.

10% models training	*F*-measure (%)	Misclassified in the class	Misclassified out the class
News	99.1%	0	3/168
Soccer	94.8%	0	2/20
TV series	91.32%	1	3/24
Documentary	95.45%	2	0/21
Tv games	80%	1	1/5
Movie extracts	61.6%	5	0/4

Table 6: Three-folds cross-validation sampling method—*F*-measure.

Cross-validation: 3- folds	*F*-measure (%)					
	News	Soccer	TV series	Documentary	TV games	Movie extracts
Random forest	98.23	100	86.96	100	61.54	40
C4.5	96.68	82.05	76.6	93.33	57.14	50
Classification tree	97.9	81.08	82.35	97.67	72.73	22.22
SVM	99.40	94.74	82.35	84.45	88.89	40
CN2 rules	96.83	88.89	84.45	95.45	33.33	66.67
KNN	99.7	94.74	92.30	90.48	88.89	75
Naïve bayes	98.49	97.44	93.33	89.36	88.89	61.53

Table 7 presents the ratio of well- and misclassified number of video documents for the whole video corpus.

Random Sampling Method. The second sampling method tested to select the documents used to train the models is the random sampling method. This method randomly splits the data onto the training and testing set in the given proportion (here 50:50). The whole procedure is repeated several times (5 times in this experiment) to show the stability of the classification even if we change the documents used to train the models.

Table 8 shows the *F*-measure of the classification using the same set of classifiers.

The number of videos classified correctly (misclassified, resp.) is presented in Table 9. The table also contains the ratio of the number of videos classified correctly over the number of video documents in the corpus. As we have mentioned before, the train-test process is repeated five times. For these reasons, the number of videos classified correctly (misclassified, resp.) measure the average number of videos over the five iterations.

Training and Testing on the Same Video Set. The third sampling method used is the "test on train data". In this method, the whole set of video documents are used to train the models and then to test the classification. Table 10 shows the F-measure of the classification.

The classifiers have classified correctly all the video documents except the naïve Bayes which has classified correctly 96.7% of the video corpus.

Leave-One-Out Sampling Method. The last sampling method tested is the "leave-one-out". The leave-one-out is similar to the cross-validation sampling method, but it holds out one document at a time, inducing the model from all others and then classifying the held out one. Table 11 shows the *F*-measure of the classification while Table 12 presents the ratio of number of video documents classified correctly over the number of video documents in the video corpus.

3.4.3. Discussion: Clustering and Classification. We have previously proposed a novel video content representation method, while in this paper we aimed at using this method in studying approaches for video clustering (nonsupervised) and classification (supervised). In the nonsupervised approach, we have defined a novel distance between video documents that is used in the hierarchical clustering algorithm. In the second approach, we have proposed a new supervised method for video classification that is based on the previously defined distance. Additionally, we have tested a set of well-known supervised classifiers, in order to prove the efficiency of our video representation proposal. As shown in the experiments, most of the video documents are clustered and classified correctly. Some video documents have abused the video corpus and their miss-classification was justified by the fact that they contain encoding problems (especially the three news videos) or contain segments of different types.

Two key issues lead to the success of the data classification (and clustering). The first is the data representation. All the experiments done have shown the efficiency of the proposed video representation (applied here for categorization). Most of the video documents are well categorized.

The second key issue is the method used for clustering (nonsupervised) or classification (supervised). The proposed distance for video clustering provides good results

Table 7: Ratio of well-classified video documents over the whole video documents.

Cross-validation	Well classified	Misclassified	Ratio
Random forest	234	8	**96.7**%
C4.5	223	19	92.1%
Classification tree	225	17	93%
SVM	232	10	95.9%
CN2 rules	228	15	94.2%
KNN	235	7	**97.1**%
Naïve Bayes	232	10	95.9%

Table 8: Random sampling method—*F*-measure.

50% sampling	*F*-measure (%)					
	News	Soccer	TV series	Documentary	TV games	Movie extracts
Random forest	96.52	89	88.33	95.15	23.53	77.78
C4.5	95.65	92.5	80	97.08	22.22	82.35
Classification tree	95.97	66	85	89.09	41.38	33.33
SVM	98.93	93.75	95	83.14	60.87	42.10
CN2 rules	96	86.32	77.5	85.22	44.45	25
KNN	98.9	93.75	89.76	88.42	84.6	78.76
Naïve Bayes	99.03	94.95	93.02	97.03	85.72	94.74

Table 9: Ratio of well-classified video documents over the whole video documents.

50% sampling	Well classified	Misclassified	Ratio
Random forest	234.4	7.6	**96.8**%
C4.5	230.4	11.6	95.2%
Classification tree	230.6	11.4	95.3%
SVM	236	6	**97.5**%
CN2 rules	233.4	8,6	96.4%
KNN	237.2	4.8	**98**%
Naïve Bayes	239	3	**98.8**%

Table 10: Test on train data—*F*-measure.

Test on train data	*F*-measure (%)					
	News	Soccer	TV series	Documentary	TV games	Movie extracts
Random forest	100	100	100	100	100	100
C4.5	100	100	100	100	100	100
Classification tree	100	100	100	100	100	100
SVM	100	100	100	100	100	100
CN2 rules	100	100	100	100	100	100
KNN	100	100	100	100	100	100
Naïve Bayes	98.48	100	100	93.33	90.89	66.67

Table 11: Held one out—*F*-measure.

Held one out	Precision (%)					
	News	Soccer	TV series	Documentary	TV games	Movie extracts
Random forest	97.66	91.9	81.63	97.67	75	40
C4.5	96.8	95	72.34	95.45	44.45	50
Classification tree	96.16	76.5	75	93.34	20	50
SVM	98.8	94.74	88.89	81.63	80	75
CN2 rules	96.8	97.43	88.89	100	33.34	85.71
KNN	99.4	94.74	90.56	89.47	88.89	75
Naïve Bayes	98.18	92.31	93.34	91.30	80	57.14

Table 12: Ratio of well-classified video documents over the whole video documents.

Leave one out	Well classified	Misclassified	Ratio
Random forest	229	13	94.6%
C4.5	221	21	91.3%
Classification tree	218	24	90%
SVM	230	12	95%
CN2 rules	231	11	**95.5**%
KNN	234	8	**96.7**%
Naïve Bayes	230	12	95%

Table 13: Summary of the results.

Categorization in 6-clusters	Nonsupervised	Supervised			
Classifiers/Sampling method	—	Random sampling	Cross-validation	Test on train	Held one out
Nonsupervised (proposed)	95.1%	—	—	—	—
Supervised (proposed)	—	95.5%	—	—	—
Random forest	—	**96.8**%	**96.7**%	**100**%	**94.6**%
C4.5	—	95.2%	92.1%	100%	91.3%
Classification tree	—	95.3%	93%	100%	90%
SVM	—	**97.5**%	**95.9**%	**100**%	**95**%
CN2 rules	—	96.4%	94.2%	100%	95.5%
KNN	—	**98**%	**97.1**%	**100**%	**96.7**%
Naïve Bayes	—	98.8%	95.9%	96.7%	95%

(more than 95% are correctly clustered). We have also used this distance in our proposed supervised algorithm and it gave good results as it could correctly separate more than 95% of the classes. These results are very close to the results provided by the well-known classifiers (random forest, SVM...). The obtained results are summarized in Table 13.

We should emphasize here that the proposed video representation is independent from the video categories. Moreover, we should remind that our method takes as input several available segmentations. This set of segmentations may not be the most pertinent one for video classification. We have used a set of available segmentations provided by some tools in our research team. Future works will study the effect of the features on the classification. Finally, we can see that our classification and clustering methods can be considered as an independent method from any preknown information (except models training in the classification methods).

4. Conclusion and Future Works

In this paper, we have proposed a novel approach for video classification that bases on the analysis of the temporal relationships between the basic events in audiovisual documents. We have based on some basic segmentation results in order to propose our new representation method called the Temporal Relation Matrix (TRM). We have shown that interesting clues on document content could be brought to the fore by a detailed analysis of those matrices and their classification in several classes. The proposed technique is applied on video analysis and clustering. For video document analysis, clues about the content may be inferred by clustering the data in the TRMs and composing clusters from different TRMs. Based on the frequencies of the clusters in each TRM, we have proposed a similarity distance that is computed from a set of information resulting from the temporal analysis of audiovisual document content. Then, after defining our similarity measure, we described the set of audiovisual documents we have used in our experiments. Our purpose was to address the problem of video documents classification (supervised and unsupervised). Three different approaches have been studied. The first one computes the similarity between all the possible pairs of documents, the results of which are stored in a similarity matrix and can be easily visualized for further analysis. The second one applies an iterative and hierarchical clustering method to cluster the two most similar clusters at each stage. The third approach is based on a proposed supervised method and a set of well-known classifiers. In each case, results are presented and commented on.

Our future work will focus firstly on the use of a bigger video corpus that contains more video categories to test our approaches. We will also integrate in the system more features and study their effect on the classification. The study of the relevance of features will allow us to select and/or propose a weighted distance (finding the values of α and β). This work will be completed by the study of the effect of replacing Allen's relations by relation classified automatically in each TRM on the classification. We will also focus in the near future work on the detection of video documents that contain some problems (i.e., acquisition problems).

A challenging future issue will be to propose a method that classifies a new document that does not belong to the initial set of documents in use. A set of documents may be processed using our similarity measure, and a hierarchical clustering method. Then, this clustering result can be used to model each cluster for example through similarity mean and standard deviation if a Gaussian modeling is chosen. We can also use our proposed method to create models for induced clusters or even use one of the previously used learners. Then any new document will be compared with each model and bound to the most likely one. This is one of the clustering approaches we will explore very soon.

One of the interests of our method as well as its generic aspect is also the data mining approach to which it may be related. On the basis of temporal relations that can easily be extracted from any document, and between any basic features, our approach makes events emerge from the available data (audiovisual documents). These events can be associated with high-level information which characterize the document content and structure. Then, automatic clustering of large document sets, regarding the document structure similarity, can be applied on the revealed information, without using any *a priori* knowledge about the document.

References

[1] Y. Avrithis, N. Tsapatsoulis, and S. Kollias, "Broadcast news parsing using visual cues: a robust face detection approach," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '00)*, pp. 1469–1472, August 2000.

[2] V. Tovinkere and R. J. Qian, "Detecting semantic events in soccer games: toward a complete solution," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '01)*, pp. 1040–1043, Tokyo, Japan, August 2001.

[3] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proceedings of the 8th ACM International Conference on Multimedia (Multimedia '00)*, pp. 105–116, Los Angeles, Calif, USA, October-November 2000.

[4] S. Eickeler and S. Mueller, "Content-based video indexing of TV broadcast news using hidden Markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, pp. 2997–3000, March 1999.

[5] L. Y. Duan, M. Xu, X. D. Yu, and Q. Tian, "A unified framework for semantic shot classification in sports videos," in *Proceedings of the 10th International Conference of Multimedia*, pp. 419–420, December 2002.

[6] L. Chittaro and A. Montanari, "Trends in temporal representation and reasoning," *Knowledge Engineering Review*, vol. 11, no. 3, pp. 281–288, 1996.

[7] L. Chittaro and A. Montanari, "Temporal representation and reasoning in artificial intelligence: issues and approaches," *Annals of Mathematics and Artificial Intelligence*, vol. 28, no. 1–4, pp. 47–106, 2000.

[8] A. K. Pani and G. P. Bhattacharjee, "Temporal representation and reasoning in artificial intelligence: a review," *Mathematical and Computer Modelling*, vol. 34, no. 1-2, pp. 55–80, 2001.

[9] L. Vila, "A survey on temporal reasoning in artificial intelligence," *Artificial Intelligence Communications*, vol. 7, no. 1, pp. 4–28, 1994.

[10] P. V. Beek and R. Cohen, "Exact and approximate reasoning about temporal relations," *Computational Intelligence*, vol. 6, pp. 132–144, 1990.

[11] M. B. Vilain and H. A. Kautz, "Constraint propagation algorithms for temporal reasoning," in *Proceedings of the National Conference on Artificial Intelligence (AAAI '86)*, pp. 377–382, Philadelphia, Pa, USA, 1986.

[12] M. B. Vilain, H. A. Kautz, and P. V. Beek, "Constraint propagation algorithms for temporal reasoning: a revised report," in *Readings in Qualitative Reasoning about Physical Systems*, D. S. Weld and J. de Kleer, Eds., pp. 373–381, Morgan Kaufmann, 1990.

[13] J. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[14] T. Drakengren and P. Jonsson, "Twenty-one large tractable subclasses of Allen's algebra," *Artificial Intelligence*, vol. 93, no. 1-2, pp. 297–319, 1997.

[15] M. C. Golumbic and R. Shamir, "Complexity and algorithms for reasoning about time: a graph-theoretic approach," *Journal of the ACM*, vol. 40, no. 5, pp. 1108–1133, 1993.

[16] B. Nebel and H. J. Buerckert, "Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra," *Journal of the ACM*, vol. 42, no. 1, pp. 43–66, 1995.

[17] R. Dechter, I. Meiri, and J. Pearl, "Temporal constraint networks," *Artificial Intelligence*, vol. 38, pp. 353–366, 1991.

[18] H. A. Kautz and P. B. Ladkin, "Integrating metric and qualitative temporal reasoning," in *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI '91)*, vol. 1, pp. 241–246, 1991.

[19] G. Ligozat, "On generalized interval calculi," in *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI '91)*, pp. 234–240, 1991.

[20] I. Meiri, "Combining qualitative and quantitative constraints in temporal reasoning," *Artificial Intelligence*, vol. 87, no. 1-2, pp. 343–385, 1996.

[21] J. F. Allen, "An interval-based representation of temporal knowledge," in *Proceedings of the 7th International Joint Conference on Artificial intelligence*, pp. 221–226, Vancouver, Canada, 1981.

[22] Z. Ibrahim, I. Ferrané, and P. Joly, "Temporal relation analysis in audiovisual documents for complementary descriptive information," in *Proceedings of the 3rd International Workshop on Adaptive Multimedia Retrieval (AMR '05)*, Glasgow, UK, July 2005.

[23] B. Moulin, "Conceptual-graph approach for the representation of temporal information in discourse," *Knowledge-Based Systems*, vol. 5, no. 3, pp. 183–192, 1992.

[24] Z. Ibrahim, I. Ferrané, and P. Joly, "Conversation detection in audiovisual documents: temporal relation analysis and error handling," in *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU '06)*, Paris, France, July 2006.

[25] R. Skibata, "Criteria of statistical model selection," Research Report, University of Keio-Yokohama, Yokohama, Japan, August 1986.

[26] Z. Ibrahim, *Caractérisation des structures audiovisuelles par analyse statistique des relations temporelles*, PHD Thesis, Paul Sabatier University, Toulouse, France, July 2007.

[27] G. Jaffre and P. Joly, "Improvement of a person labelling method using extracted knowledge on costume," in *Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns*, Rocquencourt, France, September 2005.

[28] Z. Ibrahim, I. Ferrané, and P. Joly, "Audio data analysis using parametric representation of temporal relations," in *Proceedings of the IEEE International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '06)*, Damascus, Syria, April 2006.

[29] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871–883, 1999.

[30] S. C. S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 59–74, 2003.

[31] J. Law-To, V. Gouet-Brunet, O. Buisson, and N. Boujemaa, "Local behaviours labelling for content based video copy detection," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 232–235, August 2006.

[32] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, "A framework for measuring video similarity and its application to video query by example," in *Proceedings of the International Conference on Image Processing (ICIP '99)*, pp. 106–110, Kobe, Japan, October 1999.

[33] S. Haidar, P. Joly, and B. Chebaro, "Mining for video production invariants to measure style similarity," *International Journal of Intelligent Systems*, vol. 21, no. 7, pp. 747–763, 2006.

[34] A. Hampapur and R. Bolle, "Feature based indexing for media tracking," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '00)*, pp. 1709–1712, August 2000.

[35] A. Jaimes and J. R. Smith, "Semi-automatic, data-driven construction of multimedia ontologies," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '01)*, 2003.

[36] R. Lienhart, "Reliable transition detection in videos: a survey and practitioner's guide," *International Journal of Image Graph*, vol. 1, pp. 469–486, 2001.

[37] C. Herley, "Extracting repeats from media streams," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 913–916, May 2004.

[38] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, vol. 730 of *Lecture Notes in Computer Science*, pp. 69–84, Springer, Chicago, Ill, USA, October 1993.

[39] E. Bruno and S. M. Maillet, "Prédiction temporelle de descripteurs visuels pour la mesure de similarité entre vidéos," in *Proceedings of the 19ème Colloque sur le Traitement du Signal et des Images (GRETSI '03)*, Paris, France, September 2003.

[40] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very-Large Databases (VLDB '99)*, Edinburgh, Scotland, 1999.

[41] V. Kobla, D. S. Doermann, K.-I. Lin, and C. Faloutsos, "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video," in *Storage and Retrieval for Image and Video Databases V*, vol. 3022 of *Proceedings of SPIE*, pp. 200–211, 1996.

[42] T. Kahveci and A. Singh, "Variable length queries for time series data," in *Proceedings of the 17th International Conference on Data Engineering*, pp. 273–282, Heidelberg, Germany, April 2001.

[43] Y. Wu, Y. Zhuang, and Y. Pan, "Content-based video similarity model," in *Proceedings of the 8th ACM International Conference on Multimedia (Multimedia '00)*, pp. 465–467, Los Angeles, Calif, USA, October-November 2000.

[44] B. Delezoide, "Hierarchical film segmentation using audio and visual similarity," in *Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '05)*, 2005.

[45] L. Zhao, S.-Q. Yang, and B. Feng, "Video scene detection using slide windows method based on temporal constraint shot similarity," in *Proceedings of the IEEE International Conference Multimedia and Expo (ICME '01)*, pp. 649–652, 2001.

[46] J. Foote and M. Cooper, "Media segmentation using self-similarity decomposition," in *Storage and Retrieval for Media Databases*, vol. 5021 of *Proceedings of SPIE*, pp. 167–175, Santa Clara, Calif, USA, 2003.

[47] G. Manson and S. A. Berrani, "Content-based video segment reunification for TV program extraction," in *Proceedings of the 11th IEEE International Symposium on Multimedia (ISM '09)*, pp. 57–64, San Diego, Calif, USA, December 2009.

[48] D. Brezeale and D. J. Cook, "Automatic video classification: a survey of the literature," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 38, no. 3, pp. 416–430, 2008.

[49] S. Haidar, *Comparaison des Document Audiovisuels par Matrice de Similarit*, Ph.D. thesis, University of Toulouse 3, Paul Sabatier, France, September 2005.

[50] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.

[51] C. M. Bishop, *Pattern Recognition and Machine Learning: Information Science and Statistics*, Springer, Berlin, Germany, 2006.

[52] Z. Ibrahim, P. Gros, and S. Campion, "AVSST: an automatic video stream structuring tool," in *Proceedings of the 3rd Networked and Electronic Media Summit*, Barcelona, Spain, October 2010.

[53] G. Manson and S. A. Berrani, "Automatic TV broadcast structuring," *International Journal of Digital Multimedia Broadcasting*, vol. 2010, Article ID 153160, 16 pages, 2010.

[54] X. Naturel, G. Gravier, and P. Gros, "Fast structuring of large television streams using program guides," in *Proceedings of the 4th International Workshop on Adaptive Multimedia Retrieval*, pp. 223–232, Geneva, Switzerland, March 2006.