A Fuzzy Approach to Video Scenes Detection and its Application for Soccer Matches

Angelo Chianese, Riccardo Miscioscia, Vincenzo Moscato, Sergio Parlato and Antonio Picariello
Dipartimento di Informatica e Sistemistica, University of Naples Federico II
via Claudio 21, 80125 Naples, Italy
{angchian,rmiscios,vmoscato,sparlato,picus}@unina.it

Abstract— **An efficient process for content-based video retrieval and indexing requires an effective scene segmentation technique to divide a long video into meaningful high-level aggregates of shots called scenes. Each scene has an autonomous semantic content and can be used as starting point in the video classification work. The main focus of this paper is the design of a fuzzy-based system with the aims of detecting video blocks belonging to a general semantic class. In particular the system has been tested for automatic GOAL event detection in a sport video.**

I. Introduction

The rapid advances in video technology, the widespread diffusion of video products - such as digital cameras, camcorders and storage devices - and the explosive growth of the internet have quickly made of digital video an essential component of today multimedia applications, including video-on-demand, video conferences, multimedia authoring systems and so on. The major bottleneck limiting a wider use of digital video is the ability of finding desired information from a huge database using content. Traditional video database use keywords as index to quickly access to a great deal of data. However, this kind of data representation requires a burdening manual processing. It is widely agreed that video analysis refers to the computerized understanding of the semantic meanings of a video sequence. Tools that enable such automated analysis are becoming indispensable to be able to efficiently access and retrieve video information.

The video shot segmentation is the first step in an automatic video indexing process. The objective of such process is to partition a video into basic parts called shots. After this preliminary stage, an efficient process for content-based video retrieval requires an effective scene segmentation technique to divide a video into meaningful high-level aggregates of shots called scenes. Each scene has an autonomous semantic content and can be used as starting point in the video classification and annotation work. Such task involves the segmentation of the video into semantically meaningful units, classifying each unit into a predefined scene type, and indexing and summarizing the video for efficient retrieval and browsing. In the literature, several automatic techniques for video scene detection have been proposed. The majority of such methods uses audio and visual information jointly for accomplishing the above task. In [7], the main audio and visual features that can effectively characterize scene content and some algorithms for video segmentation and classification are reported. In [3] some visual useful metrics for scene change detection based on scene lighting and intensity distribution are presented; in opposite [4] focuses the attention on the associated audio information for video scene analysis. In [8] a framework to group shots based on the analysis of video content (in terms of visual, position, camera, motion and audio features) continuity is performed. In [6] video scenes are detected on the base of cromacity, lighting conditions and ambient sound video properties. In [1] a Markov model approach for scene detection based on audio and visual video analysis is proposed. Eventually, in [5] a scheme for identifying scenes on the base of video genre has been developed.

In this paper we describe a flexible approach to video scenes detection and classification based on the definition of a fuzzy system capable of discovering, on the base of visual and audio features, the video blocks belonging to the same semantic class [12]. The paper is organized as follows. In Section 2, we outline the proposed system architecture for scene detection. In section 3, we describe the application of our model to GOAL detection for sport videos. In section 4 the experimental protocol and related results, are provided. Eventually some concluding remarks are given in Section 5.

II. System Architecture

A. Architecture Layout

We define a fuzzy-based flexible system in order to:

- decompose video streams in shots;
- discover the semantic content of each shot choosing the finest recognition method according to genre and type of the video;
- classify shots in homogeneous semantic sets;
- regroup semantically-alike shots in a single video matching the timeline dependencies and dynamic aggregation rules (Block Making);
- track shots that don't fit in any particular semantic category and create online new *fact* classes, relying on the minimal necessary user interaction.

We propose a *smart block maker* that doesn't simply manage shots and their timeliness: it groups the video segments that match a semantic proximity criteria. In the general schema of figure 1, the main elements of the proposed block-maker architecture can be observed.

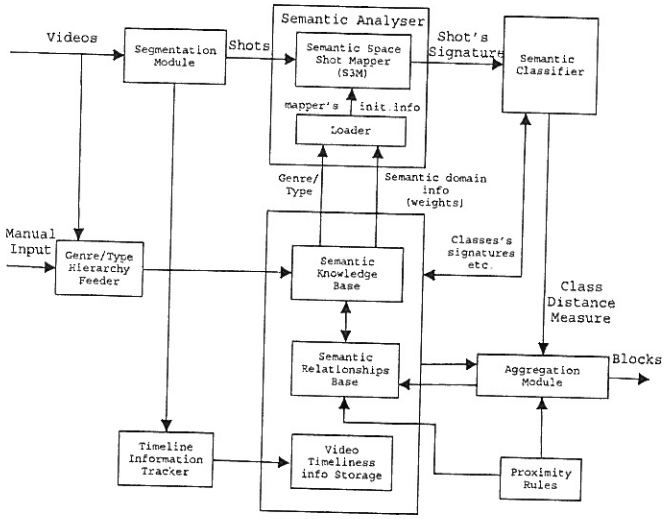

Fig. 1. 'Smart' Block Maker Architecture

1) Video Segmentation: Video streams are split into shots by the means of a **Segmentation Module**. This module identifies shot changes searching for abrupt (cuts) and gradual (fades, cross fades etc.) transitions.
In order to track the chronological order of the shots, a **Timeline Information Tracker** stores absolute time position information about the segments of the originating video in the **Video Timeliness Info Storage** block.

2) Upper level video classification: Upon the video stream submission, its genre and type are stored by the **Genre/Type Hierarchy Feeder** in a **Semantic Knowledge Base** in order to enable the correct semantic detection modules to work accordingly.

3) Semantic analysis: The **Semantic Analyser** (SA) module queries the **Semantic Knowledge Base** in order to determine the shot's general semantic properties and to choose the correct knowledge data for the activation of the semantic discovering modules (eg., in the case of a neural network analyser, the genre and type of the video determine the net weights to be loaded). Shots are serially submitted to the SA which produce a semantic *checksum* of the shot called *Shot Signatures* (S2). The **Semantic Knowledge Base** (SKB) uses S2's to fill a specific semantic field in the core data of the system. Shots are mapped to vector space elements by the **Semantic Space Shot Mapper** (S3M) module included in the SA. The *Loader* dynamically feeds S3M with the correct data to make the detection possible.

This mechanism keeps knowledge data and semantic analysis tools separated, ensuring flexibility to the underlying detection architecture. The benefits are a wide reusability of the semantic analysis systems and scalability of the working architecture.

4) Semantic classification: Once the S2s are obtained, the **Semantic Classifier** determines which semantic contents are to be taken in account for shot processing in the aggregation phase.

Obviously a single shot may be included in one, none or many semantic classes and the classification criteria must be flexible. It's also necessary to consider that the semantic classes have no strict membership rules. Classes are much more like sets with no sharp border lines with a smooth *membership function*: this project uses basic *fuzzy sets theory* (see [9] and [10]) to manage the complexity of the classification task.

5) Semantic Relationship Base: When the **Semantic Classifier** determines the *affinity vector* (the similarity distance between two S2s) for the current shot, it fills the **Semantic Knowledge Base** with shot's information which can be optionally used to redefine inter/intra-classes relationships (eg. discovery of a new semantic class, changing an existing class properties, naming de-facto classes, deleting duplicate classes, etc.).
We call the collection of optional architecture compounds **Semantic Relationship Base** (SRB).
A simplified model of this subsystem will be discussed later.

6) Aggregation Module: This module does the hard job. We can briefly expose its operation model as follows.
The **Aggregation Module**:

- Gets general semantic properties from the **Semantic Knowledge Base**.
- Evaluates the minimum required confidence level necessary to correctly compose a semantically homogeneous output video stream.
- Associates the timeline information to respective shots querying the **Video Timeliness Information Base**.
- Checks all the significative shots in a given semantic class against the determined confidence level (this ensures a more or less precise semantic aggregation).
- Checks candidate adjacent shots against proximity rules (the semantic content change less than a threshold and the time distance is not greater than an adaptive value). Proximity rules give inter-shot aggregation directives.
- Builds a new video stream with the sorted selected shot set.

B. Semantic Base Operation model

Semantic Knowledge Base (SKB) is used by the **Semantic Analyser module** to correctly process the shots and store its semantic properties. SKB subsystem memorizes, updates and maintains a hierarchical classification structure in order to accomplish its task. A three level semantic hierarchy is defined as shown in Fig. 2.

When a shot is processed by the analyser, general semantic properties (*Genre* and *Type*) are inherited from the original video; SKB is queried and semantic tree searched by a depth-first algorithm. The results of this query determine the analyser behavior, the **Semantic Classifier**'s output updates *last hierarchy level* deciding *facts* accordingly. Last hierarchy level (facts hierarchy) tracks shot's semantic content dinamically memorizing the **Semantic Classifier** module's decisions necessary to operate the **Aggregation Module**.
Semantic Class Tree is managed matching a set of **Tree**

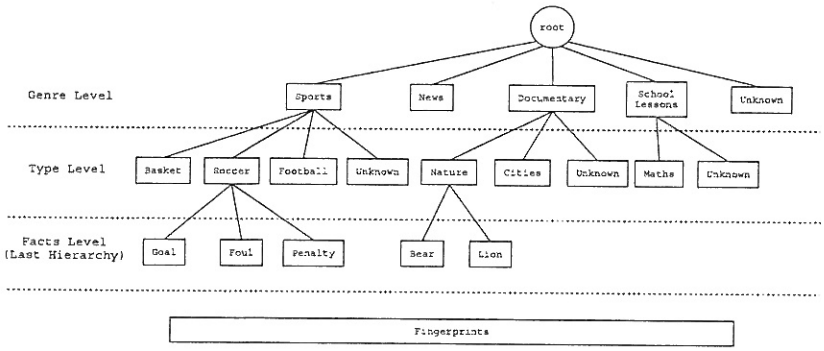

Fig. 2. Semantic tree

Update Rules by the **Hierarchy Manager** which acts as follows:

- When a video is submitted to the system its *Genre* info is required; if the incoming genre already exists, a type information node should be searched to match *Type* info. If the submitted genre doesn't exist, a new genre is created updating root's tree level links list.
- If no *genre* info is supplied nor available genre matches, system defaults *unknown* genre.
- Applies the same *Genre* insertion rules to Type-level tree management
- By using available information about these two upper semantic levels, feeds **semantic analyzer** module permitting SNs computation and store in a signature base.
- Allows **Semantic Classifier** to access the stored shot's SNs and estimate the membership degree of a shot related to each semantic class (also known as *fact*) enabling *proximity rules*.

C. Evaluating semantic distance

Semantic Analiser maps every shot $s_{vs(i)}(k)$ in to a semantic space vector: $\underline{v}_i(k)$, where $vs(i)$ is the i-th video stream, $s_{vs(i)}(k)$ is the k-th shot in time order extracted by the segmentation module from $vs(i)$. Let us call $s_{vs(i)}(k)$ shot's *signature*: the semantic classes are identified in a center-surround approach by specifying the central originating class sample as shown in figure 3. Class sample is the signature of the best determined semantically fitting example.

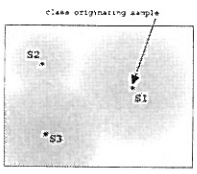

Fig. 3. Semantic classes graphical representation

Let be $s_{vs(i)}(k)$ a shot, its *semantic distance* from the j-th class of a given *semantic class set* is a function $d(\cdot,\cdot)$ defined by means of a distance $\delta(\cdot,\cdot)$ defined over the feature vector metric-space (for example the *Euclidean distance*) as follows:

$$d(s_{vs(i)}(k), c(j)) = \delta(\underline{v}_i(k), \underline{s}_{c(j)}) = d^i(j,k) = \delta^i_{j,k} \quad (1)$$

where:

- $c(j)$ is the j-th semantic class
- $\underline{s}_{c(j)}$ is class's sample signature vector

Obviously, when j-k maximal semantic matching occurs $\delta^i_{j,k} = 0$ so: $d^i(j,k) = 0$.

1) μ-function: For each semantic class $c(j)$, using shots and the considered classes signatures (SNs) we can estimate the degree of membership of $s_{vs(i)}(k)$ to $c(j)$ as a generic fuzzy function $\mu(\cdot,\cdot) : (s_{vs(i)}(k), c(j)) \rightarrow m \in [0,1]$. In our case the μ function is specialized as:

$$\mu^i_{(j,k)} = \mu(d^i(j,k)) : d \in R^+ \rightarrow m \in [0,1] \quad (2)$$

In our center-surround approach we decided to define such μ-function as a "*x is about Y*" membership relation as in figure 4, SNs that match sample classe's SNs determine full membership of the shot; this value decreases as $d^i(j,k)$ increases so membership grade smoothly goes to zero but there's always a not-null semantic matching level for the sample to each class. The following equation remaps $d^i(j,k)$ to $\mu^i_{j,k}$:

$$\mu^i_{j,k} = e^{-\left(\frac{d^i(j,k)}{\sigma}\right)^2} \quad (3)$$

where the σ parameter takes in account how much j-th class semantic propagates it's effects into the entire space. The higher is σ, the more general is the semantic class and the *fact*. Indeed, specific *facts* must be focused onto their generating sample so they have very little σ.

While processing shot information, in the selected semantic space, **Semantic Analyzer** computes distances $d^i(j,k)$ from shot to each class sample and originates a *semantic signatures vector*.

D. Fact managing rules

Semantic Classifier determines *semantic distance vector* to enable operations of a *Fact Determining Module* (FDM) which acts accordingly to *Fact Determining Rules* (FDR). Some

simple rules, useful for determining facts, are then applied in the classification phase to find major semantic membership candidate classes for a shot: the index j of the highest element of $\underline{\mu}$ array is the corresponding semantic class for the selected shot that's why it's distance element is the minimum possible.

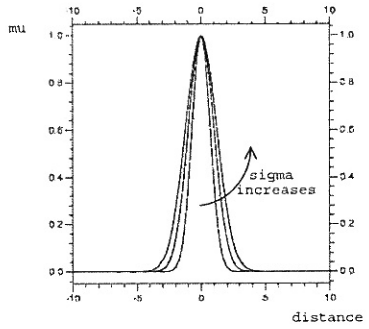

Fig. 4. μ-function representation and Semantic domain influence regions related to σ values

In this example the SNs are checked against two classes to determine which of the two facts is more probable in the selected shot. Distance functions fix the membership degrees m1 and m2 so we decide that *fact1* is taking place because m1 is greater than m2. This is a minimum-distance classification criteria widely adopted in many correlation problems and guarantees good computational performances. Anyway it is also possible that two non-exclusive facts are simultaneously happening and we are interested in determining the presence of a *related fact* which is more or less implicitly bound to the *fact1* and the *fact2*. It becomes necessary to define fact composition rules to enable FDM's simple logic operations.

E. Fact Determining Module basic logic operations

While *primary facts* can be determined directly applying the minimum distance approach, a class of *related/derived facts* can be deducted by identifying them through set operations on primary facts membership rules:

$$\mu^i_{(j_1 \wedge j_2),k} = min(\mu^i_{j_1,k}, \mu^i_{j_2,k}) \quad (4)$$

$$\mu^i_{(j_1 \vee j_2),k} = max(\mu^i_{j_1,k}, \mu^i_{j_2,k}) \quad (5)$$

$$\mu^i_{\neg j,k} = 1 - \mu^i_{j,k} \quad (6)$$

Thus, a *derived fact* membership function can be defined without fixing a new fact *sample*.

1) Facts naming: FDM tracks shots occourrencies distribution in the semantic space. When the *density of shots* increases in an area far from classes samples we are in presence of new unreported facts. So, accordingly to last level managing rules, a new tree's leaf is created or optionally a new *type* subtree is grafted. New fact is an *unnamed fact* because it has been discovered by the system and not fixed by a human operator which however must periodically approve/disapprove fact birth and assign a significative name. Unknown facts often could be replaced by logical derived facts accepting some kind of approximation.

F. Block aggregation rules

Once S2s are computed and classes memberships assigned, a *block aggregation module* reconstructs a video stream extracting and merging shots having common semantic properties and checks *proximity rules*.

A video algebra system can retrieve shots from an annotated video-base and process them accordingly to the well defined video-algebraic relational operations. In our simplified approach we will refer to the following proximity rules:

1) Semantic proximity rules: A semantic proximity rule tells the system how selective it has to be in shot selection and how confident we want to be on the video contents. This kind of rule can be checked verifying whether a shot belongs to a *feature confidence set*:

$$F_{c(j)}(\tau) = \{k \in S_n \subset \mathcal{N} : \mu^i_{j,k} > \tau\} \quad (7)$$

where τ is the confidence level we want to associate to our produced video stream and $\mu^i_{j,k}$ is the membership function related to the j-th class. The larger $F_{c(j)}(\tau)$ is the greater is the uncertainty we bind to $F_{c(j)}(\tau)$ elements and so to the presence j-th *fact*.

A right level of confidence must be fixed in order to operate correctly the aggregation module.

2) Time proximity rules: As two close classes can be replaced by just one class with proper center/deviation settings, it's necessary to determine if two shots are semantically *the same shot*. A simple time-distance rule is formulated to merge chronologically near shots in more complete scenes recostructing the single-fact continuity.

Timely adjacent shots with near semantic signatures may often belong to the same video content. Time proximity rules are essential to rejoin bad-segmented shots and so to guarantee robustness against detection errors and segmentation errors

III. An Audio-Visual Features Based Approach for Goal Detection in a Sport Video

The proposed system is a particularization of the shot aggregation module for videos of sport events, in particular soccer matches. The system aims to aggregate the shots to form scenes characterized by the presence of goal actions to help the realization and the presentation of the match's highlights.

A segmentation architecture feeds the scene maker infrastructure splitting the video stream in its compounding shots by evaluating a human-alike attention function as in **AVS** system architecture [2].

The SA module has been trained for classifying the *GOAL fact* by sensing the presence of player (foreground, portrait or whole figure), of the ball, of the field and of the goal-mouth [11].

A. Implementing the Semantic Classifier

Since the features individualization can't be easily implemented according to deterministic methodologies, we preferred to apply a neural-network enabled approach. We avoid, in such way, a formal description of the algorithm of detection

and we introduce a flexible and *intelligent* element in the critical process of 'detection'. We have adopted a *feed-forward* neural network with *Back-Propagation* algorithm. The number of neurons in the *input layer* is equal to the number of pixels that compose the image. We experimentally determined that best performances are achieved with a number of neurons in the *hidden layer* equal to halves of those present in the input one. Net's weights data structure in its entirety is memorized in a three-dimensional vector **W** whose dimension is:

$$dim(W) = i \cdot h + h \cdot u \ (= \frac{i^2}{2} + \frac{i}{2} \ in \ our \ case)$$

where *i,h,u* are, respectively, the number of neurons in the input, in the hidden and in the output layer.

Therefore it is deduced that the memory occupation due to the allocation of the weights vector makes particularly critical to make effective the recognition algorithm. In fact, if we work in the resolution of 320x240 pixels, it would be necessary a quantity of RAM memory equal to about 22GB, without considering the increase of the elaboration time. It is necessary, thus, a radical simplification of the *visual scheme* of the *Semantic Classifier* by reducing, through an affine transformation, the dimension of the frames. Experimental data have underlined that a good compromise among performance, memory occupation, speed of elaboration and level of detail of the vision is got by scaling down the resolution to 80x60 pixels.

B. Definition of the visual scheme

Visual scheme is a synthetic representation of the reality in which it is prese1nt the essential information is present. We have decided to adopt a conversion of the native image in grey scale extracting its luminance samples. Such approach has been followed both in the training and the in the querying phase. The output values of the net are submitted to the Semantic Classifier that fixes the confidence level of the decision. It remains to be established, for every video, what is the value of this threshold that maximizes the performances. To take into account this dependence from the video, we have decided to make adaptive such threshold mediating on the values got in the output from the net, as these are only a measure of the average luminance of the frames.

C. Aggregation Module

The output data from the *Semantic Classifier* is, for its nature, *noisy*; particularly the presence of spikes or cluster of spikes determines an excessive fragmentation of the candidate scenes. We have thus chosen to operate a double filtering layer followed by a phase of post-processing for the blocks aggregation. The first filtering layer is a non-casual triangular filter. Purpose of this filtering is the reduction of the isolated spikes probably due to the particular choice of the threshold of the preceding section.

In the second layer an exponential filter is adopted, still non-causal that effects a first agglomeration of clusters. It is extremely probable that, those sequences evaluated as 'goal' and that are temporally close to each other, belong to a same action. By means of the *Scene Maker* algorith we match proximity rules to make a specific aggregation of clusters in order to get visualized long scenes.

Algorithm 1 Scene Maker

```
Input: frameno, DataBaseTable GDBlocks, title
fstart ← 1
fstop ← 1
fstartold ← fstart
fstopold ← fstop
block ← 0
while (fstop <= frameno) do
    if ( (fstop <> frameno) and (verdict_{fstop-1} = verdict_{fstop}) ) then
        fstop ← fstop + 1
    else
        if (verdict_{fstop-1} = "Goal!") then
            if ((fstart − fstopold <= 100)) then
                fstart ← fstartold
                if((fstopold − fstartold) > 75) then
                    deleteBlock(title,block) from GDBLocks
                    block ← block − 1
            fstartold ← fstart
            fstopold ← fstop
            if(fstop − fstart > 75) then
                block ← block + 1
                insertGoal(title,block,fstart,fstop) into GDBlocks
        fstart ← fstop + 1
        fstop ← fstop + 1
return
```

In the algorithm **frameno** is the video frame number, **GDBlocks** is a database table that will contain the scenes, **title** is the working video title and **verdict** is the semantic classifier's verdict on the frame's semantic.

The introduction of the audio features allows a considerable improvement of the *Precision* of the algorithm by adding a very low computation load. Because of the high Recall level we suppose that all the goals of the match have already been identified so the 'audio-detection' algorithm is just applied to the blocks in output from the video-detection module. Obviously the audio information alone is not enough to understand of the video-contents and in general an exhaustive analysis can't put aside the integration of the video/audio analysis. The analysis of the audio signal has the big advantage to require very low *processing times* compared to the video analysis. In our case it is natural to analize the volume of the video in examination: the distribution of the samples of one *'audio clip'* provides information about the variation of the amplitude of the signal that results important for the classification of the scenes. Particularly the RMS (*root mean square*) value can be used to estimate the sound volume associated to a frame or in general to a sequence according to the equation (8):

$$RMS(n) = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} s_n^2 (i)} \qquad (8)$$

where $s_n(i)$ is the *i-th* sample of the *n-th* scene and N is the total number of audio samples contained in the scene under examination. The principle on which is based the audio analysis is the following: it is not difficult to believe that in a scene containing goal actions the volume will be inevitably higher than in normal actions of game. This allows to eliminate those scenes for which the volume is not high enough. The adoption of the RMS makes the choice independent from the length of the scenes.

IV. Experimental Results

The neural network's training phase has been set up with the following parameters: we choose the sigmoid activation function for every node, the learning rate used was 0.2. We have achieved an error equal to $4.73 \cdot 10^{-14}$.

A. Description of the Data Set

The data set used for the testing of the system is composed by a series of matches from which segments of variable duration containing normal scenes of game and goal actions have been drawn out. This data set helped for the evaluation of the performances of the system, while a different one has been used for the detecting threshold values that are necessary to the correct detection operation. The data set is composed of six parts of match for an overall duration of 84 minutes and 26 seconds where 18 candidate scenes are present (sampling 25 fps). The ground truth for the performances evaluation is generated by two human observers.

B. System tuning and results evaluation

1) Indexes of evaluation of the results: The performances of the algorithm are expressed through the parameters *Recall* and *Precision*, and they are evalueted considering the number of *Missed Detections (MD)*, *False Alarms (FA)* and *Correct Detections (CD)*. The parameter *Recall* defines the fraction of correct decisions in comparison to the total number of events, while the *Precision* represents the fraction of correct detections in comparison to the general number of declared events. They are, then, defined as:

$$recall = \frac{CD}{CD + MD}; precision = \frac{CD}{CD + FA} \qquad (9)$$

where CD = number of correct detections, MD = number of missed detections, FA = number of false alarms, CD + MD = number of total events and CD + FA = number of declared events. The performances diagram is shown in figure 6.

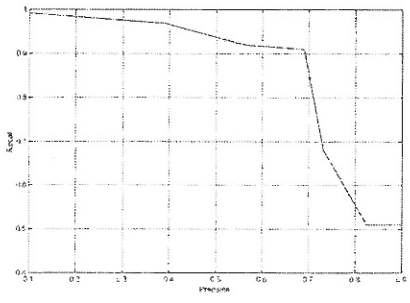

Fig. 5. Precision vs Recall

We note that a Precision value higher than 70 % implies a dramatic Recall reduction. This happens because of the excessive selectivity on the audio that brings to the removal of those blocks that have been correctly recognized by the video detection, but without high enough RMS.

2) Computation times: During the experimentation, the times necessary to the processing according to the video detection have been appraised at 0,133 sec/frames. It is also found that. The video analysis on all the video frames can not provide the possibility of introducing of a decimation factor allowing to consider only the parts of the frames taken with a predefined sampling rate; we have found that the performances of the oracle are nearly unchanged with a decimation factor up to 5, so reducing drastically (of a factor 5 indeed) the elaboration time.

V. Conclusions

In this paper a novel framework to group shots into predefined semantic classes, based on the analysis of variable video content features, has been shown. The proposed system, by means of SNs concept, is fully flexible respect to the video genre and to the video events that have to be discovered. The reported preliminary results show the efficiency and effectiveness of the system performances in the particular case of GOAL detection.

VI. Acknowledgments

This work has been partially supported by the Italian Ministry for Education, University, and Research (MIUR) in the framework of the FIRB Project "Middleware for advanced services over large scale, wired-wireless distributed systems (WEB-MINDS)".

References

[1] A. A. Altan, A. Akansu, and W. Wolf, "Multi-Modal dialog Scene Detection using Hidden Markov Models for Content-Based Multimedia Indexing", Multimedia Tools and Application Journal, Kluwer Pub., 2001.

[2] G. Boccignone, A. Chianese, V. Moscato and A. Picariello, "Foveated Shot Detection for Video Segmentation", IEEE Trans. on Circuits and Systems for Video Techn. (to appear).

[3] R. M. Ford, C. Robson, Daniel Temple, and M. Gerlach, "Metrics for Scene Change Detection in digital Video Sequences", IEEE Int. Conf. on Multimedia Computing and Systems, 1997.

[4] Z. Liu, Y. Wang, znd T. Chen, "Audio Feature Extraction and Analysis for Scene Segmentation and Classification", Journal of VLSI Signal Processing, Kluwer Pub., 1998.

[5] S. Pfeiffer, R. Lienhart, and W. Effelsberg, "Scene Determination based on Video and Audio Features", Multimedia Tools and Application Journal, Kluwer Pub.

[6] H. Sundaram, and S. Chang, "Determing Computable scenes in Films and their Structures using Audio-Visual Memory", ACM Multimedia, 2000.

[7] Y. Wang, Z. Liu, and J. Huang, "Multimedia Content Analysis", IEEE Signal Processing Magazine, pp. 12-36, November 2002.

[8] J. Wang, and T. Chua, "A Framework for Video Scene Boundary Detection", ACM Multimedia, 2002.

[9] L. Zadeh "Fuzzy Logic = Computing with Words", IEEE Trans. on Fuzzy Systems, 2, 103-111, 1996.

[10] L. Zadeh, R. E. Bellman "Decision-making in a fuzzy environment", Management Science 17, B- 141-B-164, 1970.

[11] C.Amoroso, A.Chella, V.Morreale, P.Storniolo "A segmentation System for Soccer Robot Based on Neural Networks", RoboCup, pp. 136-147. 1999.

[12] N. Haering, R. J. Qian, and M. Sezan, "A Semantic Event-Detection Approach and Its Application to Detecting Hunts in Wildlife Video", IEEE Trans. on Circuits and Systems for Video Tech., vol. 10, no. 6, 2000.