A Rule-Based Video Annotation System

Andres Dorado, Janko Calic, and Ebroul Izquierdo, Senior Member, IEEE

Department of Electronic Engineering
Queen Mary, University of London
{andres.dorado, janko.calic, ebroul.izquierdo} @ elec.qmul.ac.uk

Abstract—This paper introduces a system for automatic annotation of videos based on knowledge extracted from a pre-annotated datasets. Constructing knowledge representations from a customised set of low-level features, the system generates inference rules linking low-level descriptors with user defined semantic concepts, whether they are keywords or any other symbol that conveys the video semantics to the user. The system utilises fuzzy logic and data mining to achieve human-like approximate reasoning. Using a rule- knowledge-base created in the learning process, keywords from a user defined lexicon are assigned to new video clips added to the database. Exploiting efficient low-level video representations, the system performance was assessed on a dataset containing over 100 broadcasting videos. The experimental evaluation showed robust and high annotation accuracy. The system architecture offers straightforward expansion to relevance feedback and autonomous learning capabilities.

Index Terms—Inference mechanisms, Multimedia databases, Semantic video annotation
I. INTRODUCTION

Driven by development of high-capacity storage devices and the omnipresence of digital media in global networked environments, image and video retrieval has emerged in the last decade as one of the most important functionality of pervasive multimedia systems. Although the initial excitement and hype-boosted research effort dedicated to achieve this functionality, the outcome was limited to few specialised systems based on the query by example retrieval model. Relying only on low-level image or video features the query by example paradigm manage to reveal perceptual similarity between images or videos but it lacks potential to convey their meaning to the user. This model exclusively based on low-level primitives appears to be the source of most problems restraining the development of user centred multimedia systems. It is not until more recently that the research community has focused on the challenging problem originated by the gap between the information that can be extracted automatically from visual data and the interpretation the same data has for a user in a given situation: The Semantic Gap [1].

Meaning is not a datum that is present in the image or video and that can be computed and decoded prior to the query process. It is rather a complex instantiation of static and dynamic elements emerging from relations within the system: database record itself, temporal context, user's circumstances, etc [2]. During the last few years there has been increasing research effort put towards automatic generation of links between low-level features and high-level concepts. In [3], a redefinition of the concept and role of meaning in a visual information system is proposed to overcome the semantic gap. The work presented in [4] introduces an approach to learning-based semantic indexing of multimedia content using cues derived from audio, visual and textual features. It addresses the problem by developing a set of statistical models that utilise Gaussian mixtures, Hidden Markov Models, and support vector machines for a predefined lexicon. Two similar approaches to identify the underlying links between the low-level features and the high-level concepts associated with images are introduced in [5]. This method utilizes classification trees and k-means clustering algorithms. In [6], [7] several different low-level visual primitives are combined together by domain-
specific rules in order to capture semantics of video content at a higher level of significance. Although these systems achieved reasonable accuracy the complexity of the statistical approach remained an implementation obstacle. Jardon et al. introduced a rule-based approach for the generation of inference rules using fuzzy logic [8]. However, the used knowledge representations are predefined and static, limiting the adaptability to different contexts.

The complex analysis required in any solid multimodal approach needs sensible low-level descriptions of the underlying media. To facilitate creation of useful inference rules, a system needs appropriate representation of automatically extracted features to distinguish between videos conveying different concepts. Based on a broad understanding of visual media elements and their role in synthesis and manipulation of their content an approach called Computational Media Aesthetics studies the dynamic nature of the narrative via analysis of the integration and sequencing of audio/visual elements [9], [10]. Following this approach, Ba et al. exploits a set of features commonly used by film editors and directors to classify films [11]. In [12] a mechanism to discover video intervals as consecutive sequences of video shots is presented. Each sequence represents an action or a situation and the mechanism generates annotations of the discovered video intervals from annotations of shots. In order to facilitate high-level semantics-based content annotation and interpretation, Ba et al. proposed two different techniques using intershot analysis to tackle the problem of automatic decomposition of motion pictures into meaningful story units [13].

Another important aspect that adds complexity to the problem is the high dimensionality of signification spaces. Since the dimensionality and size of the signification space can bring out high computational cost and even divergence of the algorithms, it is crucial to find low-dimensional but meaningful subspaces. Context information can be used to constrain the content information, reducing the number of possible interpretations. Few approaches have been proposed to extract and use context in the process of semantic inference. A two layered Bayesian network integrating multimodal video features was proposed in [14]. Naphade and Huang introduced a probabilistic framework for semantic understanding using specific multimedia objects to extract the context involved [15]. Work on video
classification as a way of context creation was also reported in [16]. Here a set of syntactic and style attributes is used to classify the video sequences into five predefined genre classes.

The research leading to this paper is based on a premise that one cannot predefine the rules needed to infer a set of high-level concepts from low-level descriptors, but rather learn them using knowledge embedded in the database and exploiting interaction with the users. Using this idea a system for automatic semantic annotation of videos is introduced. Underpinning this system at the implementation level is pre-annotated data that dynamically creates signification links between a set of low-level features extracted directly from the video dataset and high-level semantic concepts defined in the lexicon. The lexicon can consists of words, icons or any set of symbols that conveys the meaning to the user. Thus, the lexicon is contingent on the user, application, time and the entire context of the annotation process.

The system consists of two main units: learning and annotation. For the sake of clarity a system overview is given in the next section. The learning unit consist of three sequential modules: low-level feature extraction, knowledge representation and rule mining. The first module extracts a set of low-level features in compliance with the MPEG7 syntax and clusters videos into sub-classes in order to limit the signification space. The second module relays on human knowledge to produce suitable links between low and high level representations on a limited training set. Using initial representations generated from low-level features, the rule mining module identifies whether a relationship between the low-level features and the concepts from the lexicon exists. Afterwards, it unifies these relationships into a set of rules in the form $F \rightarrow C$, where $F$ is a set of features and $C$ is a set of concepts. Given a feature $f \in F$ extracted from a video and a set of rules $R : F \rightarrow C$, the concept $c \in C$ is then associated with $f$ if there is a rule $(f \rightarrow c) \in R$. The variables involved in the description and representation processes are formed as a group of fuzzy sets. Resembling the inexact nature of human reasoning, the system uses fuzzy logic to handle knowledge representations in a relatively simple way. Similarly, inspired by the efficient manner humans link their perception and concepts, association rules are used in the process of data mining to create implications and to assign
support and confidence values to evaluate the generated implications. A detailed description of the different processing steps involved in the learning unit is given in Section III.

The annotation unit automatically infers concepts and assigns them to videos using the rules and supports generated by the learning unit. Each time a new concept emerges because new rules are being established. Thus, the constant population of the database adds up to the overall rule-knowledge base, incrementally improving annotation fidelity. The annotation unit uses a fuzzy inference strategy which involves three basic modules: fuzzification, fuzzy inference and defuzzification. A detailed description of these three modules is given in Section IV.

Relying upon data mining techniques and fuzzy set theory this work introduces a system for automatic annotation of videos using a pre-annotated dataset to initialise the inference process. It utilises information extracted from a customised set of descriptors using inference rules to link low-level video descriptors with context dependent labels from a lexicon. In order to provide a good training for the system using pre-defined lexicons, a small but representative sampling of the database is required to build the pre-annotated dataset. An extensive experimental evaluation was conducted. It showed high accuracy of the annotation process for a limited lexicon. Selected results of this evaluation are given in Section V.

The constant evaluation of the confidence and support measures leads to continuously changing metadata and inference rules. This together with a open and modular system architecture enables easy expansion to include user relevance feedback, contextual input and multimodal information if available. This important features of the system facilitate further developments towards a more robust and accurate autonomous semantic annotation system. These aspects are outlined in the last section of the paper where a summary of this work is presented and prospect research directions are given.

II. SYSTEM OVERVIEW

The proposed system annotates video sequences automatically using knowledge from a pre-annotated dataset. It creates representations from a set of low-level video features and infers the association rules between them and high-level concepts from a pre-defined lexicon. The used
A paradigm for automatic semantic annotation is depicted in Fig. 1.

The system consists of two units: learning and annotation units. The learning unit consists of three sequential modules: low-level feature extraction, knowledge representation, and rule mining. This unit uses pre-annotated videos to generate rules that link a particular low-level representation of the sequence with a corresponding label from the lexicon. Fig. 2 shows a flowchart of the learning unit.

The annotation unit automatically infers concepts and assigns them to videos using the rules and supports generated by the learning unit. Each time new content is added to the database, new concepts can emerge from the constant evaluation of the confidence and support measures leading to continuously changing metadata and inference rules.

The first module of the learning unit parses video into shots and extracts a representative set of key-frames. Exploiting descriptors extracted from the temporal structure and key-frames, a subsequent filtering stage classifies videos into contextual sub-classes in order to limit the signification space and reduce rule mining complexity. Although in this work only few low-level descriptors have been used, the subsequent processing steps are kept general enough to deal with other descriptors tailored to specific application. Each descriptor sub-class has its own lexicon, on which the rule mining algorithm is applied. Since the amount and the complexity of the information encapsulated in visual content is huge, the definition of suitable descriptions of video material is a challenging task. Thus, the initial knowledge representation is essential for the successful automatic annotation of videos. To achieve this task, a dedicated user interface was implemented. This interface is the core of the second module in the learning unit. Using knowledge representation interface an
expert annotator can choose the most appropriate set of descriptors to build the initial representation of selected video content. Exploiting information from the selected descriptors, the annotator defines a fuzzy set by assigning fuzzy boundaries to the numerical descriptors and labelling each fuzzy class with high level representations. This process is illustrated with a simple example in the next section.

![Flowchart of the learning unit](image)

**Fig. 2:** Flowchart of the learning unit.

The third module of the learning unit performs rule mining. Initially a set of low-level features are extracted from the pre-annotated dataset. Using the knowledge representation provided by the expert user, for each video clip a set of features is mapped into words and a log of all available transactions is created. From this log and using fuzzy membership values, rules are mined and a list of possible rules is generated. By filtering the initially generated associations rules, the learning unit creates a more dedicated set of rules. These selected rules are then used as a knowledge base in the annotation process.
The annotation unit, as depicted in Fig. 3, automatically assigns high-level concepts from the lexicon to any new video added to the database. As in the learning unit, the automatic annotation process starts with the extraction of the low-level video features and the automatic generation of descriptors. Using the knowledge representation defined beforehand by the expert user, the new videos are mapped into the corresponding feature-related words by a fuzzification process. Finally, a fuzzy inference module generates the pattern of assigned labels and outputs a set of high-level concepts from the lexicon. In this process new rules can be created and added to the rule knowledge base.

III. THE LEARNING UNIT

This section describes the learning unit in more detail. Low-level feature extraction and formation of descriptors is given in the beginning, followed by the knowledge representation module. The section concludes with the description of the rule mining process.

A. Low-Level Feature Extraction

As mentioned before, the selected low-level features and their extraction process are essential to achieve high annotation accuracy. Certainly, a computer cannot extract meaning from low-level
features without any additional inference strategy or learning process. By choosing the most appropriate features to represent a video, the process of giving meaning to the lowest instances of video description may appear to have been finessed. This argument along with the fact that the computational requirements of low-level feature extraction in a video annotation system can heavily impact system's performance, robustness, efficiency and algorithm scalability were the guiding principles for the initial system implementation.

Following the tendencies of the computational media aesthetics approach, temporal features of video and its structure are considered as the foremost expressive elements to be analysed. Certain rules that filmmakers use to arouse various emotional or conceptual impacts on the audience give us insight into importance of temporal video features. Therefore, pace and rhythm of the video sequence appears to convey information vital for higher-concept creation to the user. To facilitate efficient extraction of temporal features assuming video content is stored in compressed form, shot boundary detection and key-frame extraction algorithms were designed to utilise compressed domain features. Initially, the compressed video content is mapped into an one dimensional frame-to-frame difference metric. The metric is proportional to the visual change between adjacent frames and at the same time represents visual activity of the video sequence. It utilises information on prediction type that can be extracted directly from compressed video streams. A more detailed description of the algorithm and its performance can be found in [17]. In order to extract the positions of shot boundaries from the video sequence, the one dimensional curve generated by the difference metric is simplified using the geometric shape evolution algorithm described in [18], [19]. This technique leads to a simplification of the original curve with no peak rounding effects and no dislocation of relevant features. Shot change positions are determined by locating the local maxima in the simplified curve. Since line segments in the simplified curve are linear, thresholding peaks leads to the isolation of local extremes. Since the difference metric reflects the visual dissimilarity between neighbouring frames, local minima should locate the most representative ones in the group of neighbouring frames. Following this approach, the positions of the key-frames are defined as negative peaks in the second derivative
of the simplified metric. Once key-frames have been detected and decoded any visual descriptor can be used. In the current version of the implemented system pace and rhythm of the video sequence along with conventional MPEG7 visual descriptors are used as low-level features.

In order to reduce the complexity of the subsequent rule mining process and improve its accuracy, the database is partitioned into sub-classes according to the properties of the extracted low-level descriptors. The database is clustered by applying a k-means algorithm on the descriptor space. Two main reasons were considered to choose the k-means algorithm in order to achieve this pre-processing step: The huge population of conventional video databases and the excellent performance of the k-means clustering technique when k is known and the set to be clustered is large. Since the number of sub-classes in the underlying annotation problem can be predetermined by the pre-annotated dataset, it can be assumed that k is known. Furthermore, it can be assumed that at least one video per cluster in the dataset has been classified by an expert user. This entry point in the video or in the low-level feature space can be used to define the initial centre of the corresponding clusters.

B. Knowledge Representation

The transformation of low-level feature vectors into concepts which are closer to semantic concepts that are natural to the user is a critical task of the learning module. In order to associate high-level concepts with extracted video features and to enhance the demarcation between videos conveying different concepts in a given context, knowledge representation has to be adequately defined. Thus, a good knowledge about characteristics of the targeted end-users, environment and application is essential for the design of highly effective knowledge representation. For example, numerical values of the editing pace will fall into different classes for different genres like commercials and documentaries. While a given shot length could be characterised as “short shot” in a documentary, the same shot length in a commercial could be labelled as “long shot”. Furthermore, while flashy shots lasting a fraction of a second can be important “special effects” in a commercial, they can be only the result of editing artefacts in a documentary. Therefore, the knowledge representation part requires either adaptive representations tailored to very specific scenarios or it need a set of initial rules
defined by an expert user. In the presented system it is assumed than an expert user defines knowledge representations using a dedicated graphical user interface. The expert user decides which low-level features to use, their range and fuzzy boundaries.

![Diagram](image)

**Fig. 4:** Knowledge representation for changes on dominant colour in key-frames.

For the sake of clarity, a simple system that uses only the dominant colour descriptor of automatically extracted key-frames and a simple lexicon is presented throughout the following descriptions of the implemented system. This illustrative example targets the annotation of broadcasting news and it will be referred as “dominant colour to annotate news” (DCAN) example in the sequel. For a given video sequence, the low-level feature extraction module generates a set of shot boundaries and representative key-frames, as described in the previous subsection. The MPEG7 dominant colour descriptor is extracted from each key-frame. For a lexicon with two concept-related keywords a knowledge representation that exploits dominant colour variations within the sequence is designed. Changes of the dominant colour within the sequence are estimated using the quadratic
colour histogram distances $\gamma$ as defined in the MPEG7 XM. The video descriptor $\delta$ is then defined as the mean value of the $\gamma$-distances between the first key-frame and all the other key-frames in the same sequence. Small dominant colour variations within the sequence should correspond to the label "anchorperson" in the context of news clips, while strong changes should refer to the label "report". These two words form the concept-related part of the lexicon. Knowledge representation for this low-level feature is given by two fuzzy sets related to the mean values. The names of the fuzzy sets, mean of the distances between dominant colour high and low (ddcm_high, ddcm_low), are added to the lexicon as feature-related words. In Fig. 4 the results obtained for 20 samples and the representation of the low-level feature in the fuzzy space are depicted. In this figure the notation related to the knowledge representation module is used to link the obtained result with the description given in the next paragraph.

To describe the process of knowledge representation in a general context, let $W^f$ be a set of words associated with the characteristic of the features, where $W^f \subset L$ and $L$ is the lexicon. The knowledge representation can be expressed as a mapping $\mathcal{K}$ from a set of words in the lexicon $L$ to a corresponding set of interpretations defined on the universe $Y$. Each word $w \in L$ corresponds to a fuzzy set $\mathcal{K} \in Y$, which is the "interpretation" of $w$. This mapping can be denoted by $\mathcal{K}(w, \mathcal{K})$. The fuzzy set $\mathcal{K}$ represents the fuzziness in the mapping between a word and its interpretation, and can be denoted by the membership function $\mu_{wk}(w, y) = \mu_{yk}(y)$. A fuzzy set is a set containing elements that have varying degrees of membership in the set. The degree of membership is also called degree of truth. Elements of a fuzzy set are mapped to a universe of membership values using a function-theoretic form. This function maps elements of a fuzzy set $\mathcal{K}$ to a real value in the interval $[0, 1]$. If a given element $x$ in the universe, is a member of the fuzzy set $\mathcal{K}$, then this mapping is given by $\mu_{yk}(x) \in [0,1]$. As usual membership functions represent a possibility distribution.

In the system, the universe of discourse known as fuzzy set consists of all the possible values that can take a feature $f$. Convex and normal fuzzy sets are used for the knowledge representation. $\mathcal{K}$ is said
to be a convex fuzzy set for any feature values \( f_0, f_1, f_2 \) in a fuzzy set \( \mathcal{K} \), if the relation \( f_0 < f_1 < f_2 \) implies that \( \mu_{\mathcal{K}}(f_1) \geq \min[\mu_{\mathcal{K}}(f_0), \mu_{\mathcal{K}}(f_2)] \). A normal fuzzy set has at least one element \( x \) with membership value 1. Since the universe is continuous and infinite, a mathematical expression for the fuzzy sets is given by \( \mathcal{K} = \left\{ \int \mu_{\mathcal{K}}(x)/x \right\} \).

The process of mapping numerical values into one or more fuzzy sets is called fuzzification. The input of the fuzzification process is a numerical value. The degree of truth for each fuzzy set is calculated using its corresponding membership function. The output of the process is a fuzzy value, which consists of all degrees of truth of the crisp value corresponding to each fuzzy set. Fig. 5 shows an example illustrating the fuzzification process. In this figure, given a value \( x \) representing an instance of a feature \( f \) in a video, by using \( \mathcal{K}(f, x) \) the fuzzification process transforms the numerical value \( x \) into the fuzzy value:

\[
\mathcal{K}(\cdot) = \{\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x), \mu_{\mathcal{C}}(x)\}.
\]

In this particular case the system transforms the value \( x \) into the words: \( A \) and \( B \) only, because for the word \( C \), the relation \( \mu_{\mathcal{C}}(x) = 0 \) is valid.

![Fig. 5: Fuzzification Process.](image)

For the DCAN example the fuzzification process generates feature-related words describing low or high mean values of the difference between dominant colour descriptors as \( ddcm\_low \) and \( ddcm\_high \) for each video clip processed.

C. Rule Mining

Using the set of words generated for the training dataset, the rule mining module prepares the log of
transactions $T$. Basically, $T = [t_0, t_1, \ldots, t_n]$ is a collection of transactions called \textit{log of transactions}. It identifies frequent relationships between features and concepts. A single transaction $t_i$, where $i:1,\ldots,n$ is the union of words associated to each video and is defined as $t_i = W^f \cup W^c$, where $W^f$ are names of fuzzy sets mapped by $\mathcal{A}_f$, and $W^c$ is a set of words denoting concepts. The following properties are satisfied for $W^f$ and $W^c$:

- $W^c = C^r$ such that $W^c \subseteq L$
- $W^f \neq \emptyset$
- $W^c \neq \emptyset$
- $W^f \cap W^c = \emptyset$.

In addition, each transaction satisfies a strict partial ordering condition. This condition is formally defined as:

$$\forall w \in t, w \notin w \text{ and } \forall w_i, w_j, w_k \in t, w_i \notin w_j \text{ p w } \Rightarrow w_i \notin w_j \text{ p w }.$$  

The left-part of transaction $t$ consists of words representing feature values or feature-related words ($W^f$). The right-part of the transaction consists of words representing concepts or concept-related words ($W^c$). The technique implemented to perform rule mining has been inspired in the mapping process proposed by Ross in [20]. The basic idea is to generate feature-related words and to represent the features in a user friendly interface. A sketch of the membership function combining a universe $\mathcal{A}_1$ of the feature $f_1$ with a universe $\mathcal{A}_2$ of the feature $f_2$ is depicted in Fig. 6.
Let \( v_i \) be a video, \( f_j \) be values for a given feature \( f \) extracted from \( v_i \), \( c_i \) be concepts annotating the video, and \( W^f \) and \( W^c \) be feature-related and concept-related words respectively. Using this notation, the transaction for the pattern shown in Fig. 6 is given in Tab. 1.

![Table 1: Transaction log.](image)

Frequent relationships are extracted applying a data mining technique on the transactions. This technique is known as *association rule mining*. An *association rule* can be formally described as follows: Let \( I = \{i_1, i_2, \ldots, i_n\} \) be a set of literals called *items* and let \( D \) be a set of transactions where each transaction \( T \) is a set of items such that \( T \subseteq I \). An association rule is an implication of the form \( X \rightarrow Y \) where \( X \subseteq I \), \( Y \subseteq I \) and \( X \cap Y = \emptyset \). The rule \( X \rightarrow Y \) holds in the transaction set \( D \) with confidence \( c \) if \( c \% \) of transactions in \( D \) containing \( X \) also contain \( Y \). The rule \( X \rightarrow Y \) holds in the transaction set \( D \) with support \( s \) if \( s \% \) of transactions in \( D \) contain \( X \cup Y \). Given a set \( D \) of transactions, the problem of mining association rules is to generate all association rules having support and confidence greater than a minimum user-defined support (\( \text{minsup} \)) and a minimum user-specified confidence (\( \text{minconf} \)).

![Table 2: Association rule mining for the transaction log.](image)

In the implemented system the association rules mining process is applied on the log of transactions.
containing feature-related words and concepts. The expected result is the list of rules with the relationships between features and concepts that can be used as patterns for a rule-based inference process. This process is illustrated using the example presented in Tab. 2. In this example the log contains 6 transactions and the lexicon consists of 5 words \( \{w_1, w_2, w_3, w_4, w_5\} \). A pattern is a combination of words. Initially, all possible patterns are generated. For those patterns occurring in the log the support and confidence is calculated. In the upper-right part of Tab. 2 some selected patterns are given with their corresponding support and confidence. In the lower-right part of the same table some results for different \( \text{minsup} \) and \( \text{minconf} \) conditions are given. In this particular example there are not patterns satisfying \( \text{minconf} = 1.0 \) and \( \text{minsup} = 1.0 \). The pattern \( w_2, w_3 \) satisfies the condition \( \text{minsup} = 0.3 \) and \( \text{minconf} = 1.0 \). It means that at least 30\% of the transactions in the log contain these two words and 100\% of the transactions containing the word \( w_2 \) also contain the word \( w_3 \). Therefore, the association rule \( w_2 \rightarrow w_3 \) is generated. In the last row of the same table, two association rules are generated satisfying the conditions \( \text{minsup} = 0.7 \) and \( \text{minconf} = 0.3 \). The corresponding association rules are given in the next cell of the table. The rule mining module has been tested by fixing the \( \text{minconf} \) to the value 1.0 and varying the \( \text{minsup} \) with a step \( \Delta = 0.01 \) until a neck is reached. Fig. 7 shows the behaviour of the process and the location of the neck.

![Fig. 7: Minsup neck defines the optimal number of rules created by the system.](image)

The association rules mining process generates all possible combinations of words in the log of transactions satisfying the \( \text{minsup} \) and \( \text{minconf} \) conditions. However, the inference rules that can be useful for the rule-based inference process are those with feature-related words in the antecedent and
concept-related words in the consequent. Therefore, a filter is required to discriminate meaningful rules from meaningless ones. To achieve this task two types of filters have been used: strong filters and weak filters. A strong filter demands that the antecedent contains only feature-related words and the consequent only concept-related words. In contrast, a weak filter requires that the antecedent contains at least one feature-related word and the consequent contains at least one concept-related word. The result is a set containing zero, one or more association rules which can be used by the system in the rule-based inference process.

In the DCAN example a log of transactions was created using 4 words: ddcm_low, ddcm_high, anchorperson and report. The rule mining process found two rules with 100% of confidence:

\[
\text{ddcm\_low} \rightarrow \text{anchorperson} \\
\text{ddcm\_high} \rightarrow \text{report}
\]

This set of rules form the rule knowledge base for the sample system. Basically, these two rules imply that the mean difference between the dominant colour descriptors is low for "anchorperson" clips and high for "report" clips.

IV. AUTOMATIC VIDEO ANNOTATION

Once a knowledge base with a set of association rules is created the system uses an inverse rule-based inference process to identify candidate concepts for the annotation of each new video added to the database. The annotation unit uses a fuzzy inference strategy which involves three basic modules: fuzzification, fuzzy inference and defuzzification. The input of the fuzzification module is a real number corresponding to an instance of a variable. Adopting the same principle as in the knowledge representation module of the learning unit, the set of low-level features forms a set of representation variables for the new video clip. The degree of membership for each fuzzy set is calculated using the membership functions defined in the knowledge representation step and mapping features into keywords. The output of this module is a fuzzy value.

The fuzzy inference module uses rules in the form IF <condition> THEN <action>. Inference rules are not a free form of the natural language; they are limited to a set of words and a strict syntax. In
In this case, inference rules are limited to the keywords from the lexicon. Here, $<$condition$>$ expresses the instances of low-level features and $<$action$>$ denotes annotations. Each $<$condition$>$ of a rule corresponds to a specific value of a fuzzy input. This input value is a result of the fuzzification module. Each $<$action$>$ of a rule corresponds to a fuzzy output. In addition, this kind of rules have two representative characteristics: they are qualitative rather than quantitative and each $<$condition$>$ is related to an appropriate $<$action$>$. The importance of these rules lays in the possibility of representing human knowledge by a hierarchical model. Besides, these rules are relatively simple and consistent with the way human reasoning works. The fuzzy inference module calculates the fuzzy output values for the corresponding variable. It uses the relationship between input and output variables using the base of linguistic rules provided by the learning unit. At this point, a number of rules can have different degrees of truth leading to competition between the results. Using an aggregation’s operator the instances of the $<$condition$>$ part of rules are combined in order to determine the value of the rule. This value is used to determine the $<$action$>$ part of the rule. The procedure is repeated for all rules from the rule knowledge base. It is possible that an output fuzzy variable has a fuzzy set as $<$action$>$ in several rules. The composition’s operator is used to determine the final value of this fuzzy set.

The defuzzification module combines the fuzzy values of each output variable to obtain a real number for each variable. In this module a weighted average method is used. It combines fuzzy values using weighted averages to obtain the resulting crisp value.

To illustrate this process the sequence of key-frames KF1-5 extracted form the video news136.mpg is used. These five key-frames are shown in Fig. 8. The aim of this exercise is to show the behaviour of the annotation unit for the rule knowledge database derived for the DCAN example. Following the
procedure for representation, the distances from KF1 to the other key frames are calculated. These values along with the corresponding mean are given in Tab. 3.

<table>
<thead>
<tr>
<th>$\delta_1$</th>
<th>$\delta_2$</th>
<th>$\delta_3$</th>
<th>$\delta_4$</th>
<th>$\bar{X}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>83.4</td>
<td>82.8</td>
<td>78.7</td>
<td>82.4</td>
<td>81.83</td>
</tr>
</tbody>
</table>

**Tab. 3:** Distances between the dominant colour descriptors of KF1 and the other key frames extracted from the video news136.mpg.

The fuzzification process based on the knowledge representation generates $\mu_{B_{d}}(\bar{x}) > 0$ and $\mu_{B_{a}}(\bar{x}) = 0$. So, the mapping function for this instance of the mean is $\mathcal{M}(x) = \{M\}$, where $\mathcal{M}$ is $ddcm_{high}$. Using this word the rule mining process suggests the word “report” for the annotation of the sequence of key-frames KF1-5.

**V. Results**

To evaluate the performance of the whole system, several experiments were conducted using a video database consisting of over 100 randomly chosen MPEG2 digital broadcasts with a duration of 2-6 min. In the first part of this section, selected results obtained from the low-level feature extraction module are presented. This is followed by a report on results obtained for the knowledge representation design. Experimental evidence to evaluate the performance of the rule mining and annotation modules are presented afterwards.
Fig. 9: Different stages of the discrete “activity curve” evolution.

A. Low-level feature extraction

A set of video sequence descriptors is generated using temporal expressive elements, i.e. editing pace and the overall visual activity within a shot. The numerical values that concisely describe these elements are shot length distribution and shot activity. The shot length distribution descriptor is generated as a normalised 3-bin histogram of the shot lengths calculated from the output of the shot detection module. The bin boundaries are defined empirically as: 0-25, 25-150 and 150+ frames for frame rate of 25 fps, so that the descriptor shows the quantity of short, mid and long shots within the sequence. The shot activity is extracted directly from difference metric described in section IIIA. It is defined as the average of the $\Delta(i)$ metric within one shot. Thus, each shot gets assigned a unique activity value. For a shot starting at the $i$-th frame and having a shot length $N_i$, the shot activity is
calculated as: $S_{\Delta t} = \frac{1}{N_i} \sum_{j=1}^{N_i} \Delta(i + j)$.

Fig. 9 depicts different stages of the shot detection algorithm for the news video clip news042.mpg. Since the metric extraction is based on frame triplet units, the raw metric is very noisy. The noise is filtered out by applying a Gaussian filter on the one-dimensional “activity curve” generated by the difference metric. Fig. 9 shows two different stages of the simplification algorithm. The scalability of the activity curve enables analysis at different levels of detail. Shot boundaries and key-frame positions are determined by simple thresholding of the discrete second order derivative of the Gaussian smoothed activity curve.

Fig. 10: Example of the 6-dimensional temporal descriptor of the video clip news025.mpg.

The shot activity descriptor is generated as a normalised 3-bin distribution of the percentage of the video clips with high, mid and low shot activity. The final temporal descriptor consists of a 6-dimensional vector containing three values for the length distribution (long, mid and short) and three values for the shot activity (high, mid and low). Fig. 10 shows an example of the 6-dimensional feature vector obtained for a news video clip with only anchorperson present. In this particular example the shots are long and the activity is low.
Fig. 11: Video Filtering into three sub-classes: news, commercials and others.

Once the set of low-level descriptors have been extracted, the database is clustered by applying a k-means algorithm on the video descriptors. For the 6-dimensional temporal feature vector the Euclidian distance between points in the 6-dimensional space was used. Fig. 11 shows the partition of a portion of the experimental dataset into the three sub-classes: news, commercials and other clips. In this image each item (square, circle or cross) represents a video clip and the distance between items correspond to the distance between the vectors in the 6-dimensional feature space. In this representation squares correspond to news, circles correspond to commercials and the crosses represent other video clips in the database.

<table>
<thead>
<tr>
<th>Universe</th>
<th>Fuzzy set</th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shot length distributions</td>
<td>( \mathcal{H} \text{= short} )</td>
<td>0.2</td>
<td>0.35</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>( \mathcal{H} \text{= mid} )</td>
<td>0.2</td>
<td>0.35</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>( \mathcal{H} \text{= long} )</td>
<td>0.1</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td>Shot Activity</td>
<td>( \mathcal{H} \text{= low_act} )</td>
<td>0.3</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>( \mathcal{H} \text{= mid_act} )</td>
<td>0.1</td>
<td>0.25</td>
<td>0.4</td>
</tr>
<tr>
<td></td>
<td>( \mathcal{H} \text{= high_act} )</td>
<td>0.1</td>
<td>0.25</td>
<td>0.4</td>
</tr>
</tbody>
</table>
Tab. 4: Knowledge representation boundaries for the temporal descriptors.

B. Knowledge representation

The knowledge representation used in the experiments is given in the Tab. 4. Since news and commercial programmes are produced using a rather unique editing technique, the two temporal features described previously appear to be well suited to represent this sort of video clip in the rule mining process. For each feature a set of three fuzzy variables is generated, as shown in Fig. 12. In this representation the shot length distribution is given by the percentage of shots in the video clip having short, mid and long duration. In addition, representation of the shot activity is given as the percentage of the clip duration having low, mid or high visual activity.

![Diagram of fuzzy variables](image)

Fig. 12: Definition of the fuzzy variables for the knowledge representation.

The normalised fuzzy boundaries were determined empirically, by “manually optimizing” the differences between the three video categories in the feature space.
C. Rule Mining

Applying rule mining based on the knowledge representation described in subsection IIIB, a rule knowledge base was created. Fig. 13 shows temporal descriptors of video clips with the corresponding pre-annotated labels, e.g. anchorperson and report for news clips. At the left side of this figure the shot duration is plotted, while at the right side the shot activity is shown. After fuzzification feature-related words were added to the log of transactions and the rule mining generated the rules given in Tab. 5.

<table>
<thead>
<tr>
<th>SUPPORT</th>
<th>&lt;CONDITION&gt;</th>
<th>&lt;ACTION&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>SLD_shorts is low AND SLD_longs is high</td>
<td>Labeled as anchorperson</td>
</tr>
<tr>
<td></td>
<td>SLD_longs is high AND SA_low is high</td>
<td>Labeled as anchorperson</td>
</tr>
<tr>
<td></td>
<td>SLD_longs is high AND SA_mid is low</td>
<td>Labeled as anchorperson</td>
</tr>
<tr>
<td>0.08</td>
<td>SLD_longs is low AND SA_mid is high</td>
<td>Labeled as report</td>
</tr>
<tr>
<td></td>
<td>SLD_midis is high AND SA_low is low</td>
<td>Labeled as report</td>
</tr>
<tr>
<td></td>
<td>SLD_short is mid AND SA_mid is high</td>
<td>Labeled as report</td>
</tr>
</tbody>
</table>

Tab. 5: Rules created by the rule mining process using the support values 1.0 and 0.08

After applying filtering for support values as low as 0.1 and 0.08 a final set of 6 rules was obtained. These rules appear to be rather intuitive resembling human reasoning and knowledge: clips containing
anchorperson have many long shots with low shot activity, while report clips are characterised by middle length shots having medium visual activity.

**D. Automatic Annotation**

Using the rule-knowledge base given in Tab. 5, a set of 80 video clips was annotated both automatically and manually. The objective of the manual annotation was to generate ground truths for the evaluation of the automatic annotation process. An example of the particular feature values extracted from three random video sequences belonging to the “news” cluster is given in Tab. 6. Next to each feature value, fuzzy membership values are given as A, B and C. At the bottom of the table the membership values $m^k_f$ for the keywords $k$ from the lexicon are given. These values are obtained as the output of the fuzzy inference module of the annotation unit described in Section IV. In order to assess the accuracy of the annotation procedure, a statistical performance evaluation based on the amount of missed detections (MD's) and false alarms (FA's) for each keyword from the lexicon was conducted. The values for quality of the annotations are defined as recall and precision:

\[
Recall = \frac{D}{D + MD}, \quad Precision = \frac{D}{D + FA},
\]

where the $D$ is the sum of memberships $m^k_f$ for the corresponding keyword $k$, $MD$ is the sum of the distances to the full true membership $m^k_f=1$ and $FA$ is a sum of false memberships:

\[
D = \sum_{i=1}^{M} m^k_f(i), \quad k : true, \quad MD = \sum_{i=1}^{M} 1 - m^k_f(i), \quad k : true, \quad FA = \sum_{i=1}^{M} m^k_f(i), \quad k : false
\]

The obtained recall and precision results are given in the Tab. 7.

<table>
<thead>
<tr>
<th></th>
<th>NEWS032</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>NEWS081</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>NEWS138</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>sld_shorts</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.200</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.333</td>
<td>0.111</td>
<td>0.889</td>
<td>0.000</td>
</tr>
<tr>
<td>sld_mids</td>
<td>0.382</td>
<td>0.000</td>
<td>0.788</td>
<td>0.212</td>
<td>0.143</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.544</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
</tr>
<tr>
<td>sld_longs</td>
<td>0.618</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.657</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.122</td>
<td>0.852</td>
<td>0.148</td>
<td>0.000</td>
</tr>
<tr>
<td>sa_low</td>
<td>0.655</td>
<td>0.000</td>
<td>0.225</td>
<td>0.775</td>
<td>0.418</td>
<td>0.410</td>
<td>0.590</td>
<td>0.000</td>
<td>0.317</td>
<td>0.916</td>
<td>0.085</td>
<td>0.000</td>
</tr>
<tr>
<td>sa_mid</td>
<td>0.153</td>
<td>0.645</td>
<td>0.355</td>
<td>0.000</td>
<td>0.292</td>
<td>0.000</td>
<td>0.718</td>
<td>0.282</td>
<td>0.380</td>
<td>0.000</td>
<td>0.131</td>
<td>0.869</td>
</tr>
<tr>
<td>sa_high</td>
<td>0.191</td>
<td>0.390</td>
<td>0.610</td>
<td>0.000</td>
<td>0.289</td>
<td>0.000</td>
<td>0.738</td>
<td>0.262</td>
<td>0.302</td>
<td>0.000</td>
<td>0.652</td>
<td>0.348</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NEWS032</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>NEWS081</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>NEWS138</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ground truth</td>
<td>Anchorperson</td>
<td>$m_f$</td>
<td>Anchorperson</td>
<td>$m_f$</td>
<td>Report</td>
<td>$m_f$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fuzzy inferences</td>
<td>Anchorperson</td>
<td>1.000</td>
<td>Anchorperson</td>
<td>1.000</td>
<td>Anchorperson</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Report</td>
<td>0.000</td>
<td>Report</td>
<td>0.212</td>
<td>Report</td>
<td>0.916</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Tab. 6:** Temporal representation of three randomly selected news clips.
VI. CONCLUSIONS

A system for rule-based automatic video annotation that utilises fuzzy set theory and data mining techniques is presented. Initially, it extracts knowledge from a pre-annotated dataset. Inference rules associate high-level concepts from a user defined lexicon with the representations generated from a customised set of low-level features. Exploiting the “constantly updating” rule knowledge base, the fuzzy inference system automatically annotates new videos added to the database. By utilising fuzzy set theory and data mining this system simulates the human-like approximate reasoning.

Future work will be focused on the utilisation of the user relevance feedback in the process of the automatic video annotation. Because of the constant evaluation of the rules' confidence and support measures, the system can learn and update rules using relevance feedback information. Furthermore, relying upon the relevance information the knowledge representation model can be redesigned in a self-adaptive manner using soft computing techniques, e.g., a neuro-fuzzy approach. Additional expansion capabilities include learning from oppositions and dissimilarities next to the similarity based approach and the use of multimodal information, specifically considering audio features. The performed experimental evaluation shows high accuracy in the results of automatic annotation, at low computational cost. However, the conducted research also shows that there is plenty of room for improvements and that the semantic gap remains open and will be a subject of study for the years to come.

<table>
<thead>
<tr>
<th>KEYWORDS</th>
<th>DETECTS</th>
<th>MD</th>
<th>FA</th>
<th>RECALL</th>
<th>PRECISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anchorperson</td>
<td>46.65</td>
<td>1.36</td>
<td>2.48</td>
<td>0.97</td>
<td>0.95</td>
</tr>
<tr>
<td>Report</td>
<td>25.74</td>
<td>6.28</td>
<td>1.32</td>
<td>0.80</td>
<td>0.95</td>
</tr>
</tbody>
</table>

**Tab. 7:** Recall and Precision results of the annotation process.
REFERENCES


