M. Detyniecki, "Browsing a Video with Simple Constrained Queries over Fuzzy Annotations," *Proceedings of the International Conference on Flexible Query Answering Systems - FQAS'2000,* Warsaw, Poland, pp. 282-287, October, 2000.
(*edited by Physica Verlag)*

# Browsing a Video with Simple Constrained Queries over Fuzzy Annotations

**Marcin Detyniecki**

LIP6 - University of Paris VI
4, place Jussieu
75230 Paris Cedex 05, France
Marcin.Detyniecki@lip6.fr

## 1. Introduction

Video as format of computer related material is becoming more and more common [1,2]. Every day new multimedia information systems appear on the market containing more and more video. Also the format of information on Internet is clearly evolving to a video form. Initially we saw the embedding of images on text pages, now we see simple animation on almost every web page. We also know that the amount of information stored in computers is growing. So, the question that naturally arises is: "How to get the information you want?" We propose here a new tool for leading the user to make the right question to retrieve the information he wants inside a video.

Useful information may be automatically extracted from multimedia streams. For instance, cuts and camera motion can be detected from video, while cues such as applause, silence and speaker identity can be found from the audio. This may help the still used human annotation. But it is not always easy to aggregate the confidence scores coming from automatic programs and the manual ones. Here we point out this problem and suggest some solutions.

We finish this paper by describing the technology used to create our query prototype and we present the structure of our XML annotations.

## 2. Fuzzy Annotations

The present works on query-systems for video are based on the use of annotations [3-6]. These annotations can be considered as information contained in a database associated to the video and indexed by the time. These annotations can be extracted from the video manually or automatically. The automatically derived information can be generally described as a time-dependent value or values that are synchronous with the source media. For example, annotations might come from output of a face-recognition or speaker identification algorithm. In this case it is clear that a lot of uncertainty arises. But even the manual indexing, which we can consider as the most reliable way of obtaining the annotations, contains some uncertainty, not because of human errors, but because of the complexity of the world. For example when annotating night and day scenes, we can have a smooth passage from the day to the night.

So we propose here to use fuzzy annotation to enrich the descriptions. We call fuzzy annotation a classical annotation accompanied by a degree of certainty of the information (and not of the time indexing this annotation). So, for example an annotation can be: "At minute 6 the actor on the scene is Brigitte Bardot with a degree of certainty 0.75", which means that we are not totally, but almost, sure that it is Bardot. We assume that the indexing time (6 minutes) is certain.

The range of this degree is not the classically used [0,1] interval, but a more natural scale [-1,1], where –1 is the complete falsity, 0 the total ignorance and 1 the complete truth (for more details about this see [7]). This scale allows us to points out a classical mistake in aggregation of annotation. Automatic detection programs are positively oriented, which means that they are confident in their good results, but not in their bad results. A degree 0 in a detection program means "we do not know" and not "certainly the object is not there". So for automatic indexing we use the [0,1] range. But, since human annotation is more reliable, when the user does not annotate we assume that the object is not visible (for sure).

From the point of view of the time scale, we segmented it into shots. In video production a shot corresponds to the segment of video captured by a continuous camera recording and is classically used in all the annotation systems. The use of shots for annotation reduce considerably the research algorithm and provides some granularity (structure) to the video, similarly to paragraphs in text document.

In the following we are going to explain how we extract the information from the video stream.

### *2.1. Extraction*

Useful information may be automatically derived from multimedia streams. For example, we detect cuts and camera motion from video. The cuts are typically found by computing an image based distance between consecutive frames of the video. Over a certain threshold we consider that there is a cut. The distance between frames can be based on statistical properties of pixels [8], histogram difference [9], compression algorithms [10], edge differences [11] or motion

detection [12]. We use an automatic shot boundary detection developed in our laboratory [13-14]. And we automatically annotate the camera motion.

Using the resulting video segmentation, we annotate each shot with keywords and a degree of certainty. We use an annotation program developed in our laboratory [16] that uses a XML DTD (described in the next paragraph) to lead the annotation process.

### *2.2. Storage and transfer of the fuzzy annotations*

In order to store and to transfer the annotation we attach to the video file an XML file. XML is an eXtensible text based Markup Language that is fast becoming the standard for data interchange on the Web. As with HTML, identification of data is done with tags (identifiers enclosed in angle brackets). It is extensible because there is no fixed set of tags. New tags can be created as for instance in our application.

So we store the annotations in a XML form :

```
( … )
<annotation plan='16' startTimeCode='00:01:22:75' endTimeCode='00:01:27:73'>
        <personnage personnage='Falbala' acteur='Laëtitia Casta' valeur='0.95'>
        </personnage>
        <personnage personnage='Obélix' acteur='Gérard Depardieu' valeur='1.0'>
        </personnage>
        <object description='Cheval' valeur='1.0'>
        </object>
</annotation>
( … )
```

**Table 1**. Fuzzy annotations in XML

In order to control and lead the right structure of the XML file we use a DTD. A Document Type Definition defines a class of valid XML documents, i.e. it defines which tags, attributes and elements are valid. Our DTD is the following :

```
<Doctype -

    <!ELEMENT annotation (personnage * | object  *) >
    <!ATTLIST annotation
        plan                    CDATA   #REQUIRED
        startTimeCode   CDATA   #REQUIRED
        endTimeCode     CDATA   #REQUIRED
    >

    <!ELEMENT personnage EMPTY >
    <!ATTLIST personnage
        personnage              CDATA   # IMPLIED
        acteur          CDATA   # IMPLIED
        valeur          CDATA   # REQUIRED
```

```
    >

    <!ELEMENT object EMPTY >
    <!ATTLIST object
        description                    CDATA   # REQUIRED
        valeur              CDATA   # REQUIRED
    >
>
```

**Table 2**. DTD for our fuzzy annotations

## 3. Querying

Since the set of possible queries is incredibly large in the case of video, we consider that restriction may be done. For instance for the same video the user may want to see the shot that uses these or that technique, while another user may look for an actor and a third user may want to find where the actor says: "Hasta la vista baby".

Our approach consists in parsing the XML-annotation file in order to guide the user to query on the available annotations. Our program detects automatically the annotations corresponding the our DTD even if they are structured and inserted between other annotations such sought as for instance copyright, video format, etc.

Here we are interested just on simple queries. We understand by simple query a set of keywords that will identify the shot. Future works will deal with more complicated queries as for instance finding two shots related by time constraint.

The construction of the query is incremental. The user has to choose (with combo box) one after the other the keywords he is going to use for the query. Then the query button will start the search and rank process.

### *3.1. Ranking*

After we have constructed the query we launch the research process. We select the matching attributes and we aggregate their certainty values in order to obtain global certainty value. We rank then the shots from the most positive certain to the last one. From this follows that the quality of the result depends directly on the aggregation operator. A theoretical research [7] on this subject suggested that we should use a uninorm [16]. In the next paragraph we supply some explanation about this choice.

### *3.2. The aggregation of attributes*

Besides of the logical arguments announced in [7], we will try here to show on two examples why this operator is more suitable than others classically used: the means [17], and the t-norms [18-19].

Let us first compare the uninorm to a t-norm. Let us imagine that we have the following description for the shot number 11:

| Shot 11 | | | |
|---|---|---|---|
| Actor 1 | Actor 2 | Camera Motion Type 2 | Main background Color = Black |
| 0.7 | 0.9 | 0.2 | 0.8 |

**Figure 1**. Description of Shot Number 11

Using a t-norm will discard this shot from the results, since the low degree 0,2 imposes to the aggregated value to be smaller than 0,2. In this case a uninorm will compensate this low matching with the other good matchings. The suitable property of the uninorms used here is the reinforcement.

Let us now compare it to mean type operator with a uninorm: everytime we do not know we annotate with the value 0 which is neutral element for the operator and will not influence the result.

The neutral element also solves the problem of the missing values, because if there is a value missing then we replace it by the neutral element, which has no influence on total value. People using the mean may imagine that the problem of missing values can be omitted by aggregating just the available values. This is a solution but it is equivalent to replacing the missing value by the mean-value (of the non-missing values). In other words you will give a high score to an attribute you do not know if the other attributes have high scores. We do not think this is a suitable approach.

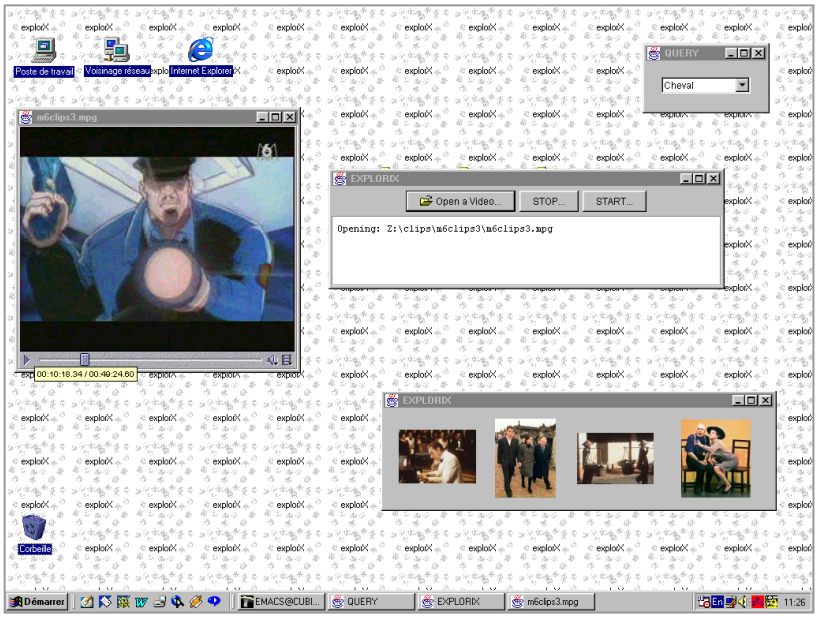

**Figure 2**. Screen shot of ExploriX

## 4. Application: ExploriX

We have prototyped a query program that uses the above ideas, towards the application of browsing and finding musical video clips. All the video clips are low quality (TV quality), MPEG-encoded and made available via network. Our program is being developed in Java language and is able to read any of the current video and sound formats. ExploriX is a first step towards intelligent retrieval and use of fuzzy annotations.

Figure 2 shows the user interface of our browser prototype. To the top left is a classical video playback window (with controls), that listens to the other windows. To the top right there is the "constructing" query window. The user built his query there, using the combo box. Between these two windows we see a control panel, where we can read all the messages transiting between the objects (these need not be visible for the user). To the middle of the screen we see the result window. This window contains a keyframe of the 4 best shots. They are ordered on a the time scale, respecting the order of appearance in the video.

## 5. Conclusions

Note that Browsing a Video with Simple Constrained Queries is only a preliminary attempt and we continue to investigate on more complicated queries and the introduction of other media (audio) annotations.

We pointed out some problems appearing while mixing human and automatic annotations, and we presented a prototype, which demonstrates the feasibility of the proposition.

## 6. Acknowledgments

ExploriX is being developed in the multimedia indexing group at LIP6 (University Paris 6). The group is partially funded by the AGIR project. I would like to thank the whole team for all the help offered.

## 7. References

[1]. Nwosu, K. C., Thuraisingham, B. and Berra, P.B., *Multimedia Database Systems*, Kluwer Academic Publisher: Boston, 1996.

[2]. Hjelsvold, R. and Midstraum, R., Modeling and querying video data, *Proc. the 20th VLDB Conference,* Satiago de Chile, 686-694, 1994.

[3]. Mackay, W.E., EVA: An experimental video annotator for symbolic analysis of video data, *SIGCHI Bulletin 21*, 68-71, 1989.

[4]. Hibino, S. Rundenstein, E. A., A visual multimedia query language for temporal analysis of video data, in *Multimedia Database Systems*, Kluwer Academic Publisher: Norwell, MA. 123-159, 1996.

[5]. Snodgrass, R., The temporal query language TQUEL, *ACM Transactions on Database Systems* 12, 247-298, 1987.

[6]. Yager, R. R., Fuzzy temporal methods for video multimedia information systems, *Journal of Advanced Computational Intelligence 1*, 37-45, 1997.

[7]. Detyniecki M. and Bouchon-Meunier B., Aggregating Truth and Falsity Values, *FUSION'2000*, Paris, France, July 2000.

[8]. Kasturi, R., Jain, R., "Dynamic Vision", in *Computer Vision: Principles*, Kasturi R., Jain, R., Editors IEEE Coputer Society Press, Washington, 1991.

[9]. Zhang, H.J., Kankanhalli, A., Smoiliar, S.W., "Automatic Partitioning of Full-Motion Video", *Multimedia Systems Vol. 1 No1*, 10-28, 1993.

[10]. Arman, F., Hsu, A., Chiu, M-Y., "Image Processing on Encoded Video Sequences", *Multimedia Systems, Vol 1 No 1,* 211-219, 1994

[11]. Zabih, R., Miller, J., Mai, K., "A Feature-based Algorithm for Detecting and Classifying Scenes Breaks", *Proc. ACM Multimedia 95,* 189-200, San Francisco, CA, November 1995.

[12]. Shahraray, B., "Scene Change Detection and Content-Based Sampling of Video Sequences", in *Digital Video Compression: algorithms and Technologies*, Rodriguez, Safranek, Delp, Eds., Proc. SPIE 2419, 2-13, Feb 1995.

[13]. Aigrain P and Joly P., The automatic real-time analysis of film editing and transition effects and its applications, *Computers and Graphics*, *Vol. 18, Nr.1,* 93-103, 1994.

[14]. Ruiloba, R. and Joly P., Framework for Evaluation of Video-to-Shots Segmentation Algorithms: A Description Scheme for Editing Work, Network Information Systems, Special Issue on Video, 2000

[15]. C.Thiénot, C.Seyrat, P.Faudemay and P.Joly, Status of the MPEG-7 parser, *MPEG Document M5767*, Mars 2000

[16]. Fodor, J. C., Yager, R.R. and Rybalov, A., Structure of Uninorms, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 5,* 411-427, 1997.

[17]. Kolmogorov, A., Sur la notion de moyenne*, Atti delle Reale Accademia Nazionale dei Lincei Mem. Cl. Sci. Mat. Natur. Sez. 12*, 323-343, 1930.

[18]. Menger, K., Statistical metrics, *Proc. Nat. Acad. Sci. U.S.A. 8*, 535-537, 1942.

[19]. Schweizer B. and Sklar A., Probabilistic metric spaces, *North Holland*, New York, 1983.