

ANNOTATION OF VIDEO IMAGES FOR INDEXING SEQUENCES USING ORACLE 10G

Horia Ilie and Alain April, Ph.d.
École de Technologie Supérieure
1100 Notre-Dame West, Montreal, Canada

ABSTRACT

This paper presents the creation of an MPEG-7 compliant annotation using a commercial multimedia database. The contributions of this paper is first identifying the most effective weights, for low level image characteristics, with Oracle Intermedia. Second, describing a video scene identification technique using an Oracle 10g multimedia database.

KEYWORDS

Multimedia database, MPEG-7, video scene identification, image annotation, XML.

1. INTRODUCTION

The growth in the multimedia content of documents, compounded with the emerging Internet multimedia functionalities, undoubtedly accelerates the need for management of multimedia information. One of the current problems faced by research and industry is the rapid and precise retrieval of multimedia information. Commercial multimedia database vendors are now offering more and more tools for efficient image, video and sound storage, but few manipulation tools.

This paper presents scene detection and image annotation using Oracle 10g capabilities. It is organized as follows: Section 2 presents an overview of the project objectives, technology and development approach. Section 3 describes the design choices, as well as the way in which each module operates. It also presents the experimental design chosen to assess the performance of the library developed. Section 4 contains the experimental results, and, finally, section 5 presents conclusions and directions for future work.

2. PROJECT OVERVIEW

In order to achieve the functionality described in the introduction, we placed a multimedia database at the core of our project with a view to addressing the various phases of using and processing metadata. The database stores annotated videos and images, and proposes interfaces for retrieving and presenting documents. Figure 1 presents an architectural overview of the project. First, a video is processed with the goal of automatically detecting a change of scene. Second, a sample image is chosen as a candidate to reference each segment of the video. This step includes image annotation using the MPEG-7 standard. The third and final step is presentation and visualization involving the user.

2.1 Technology used

We used the Oracle 10g Multimedia Database because of its advanced multimedia processing and storage features, and the following Java technologies: JDEV Oracle, Eclipse, Java (JDK 5.0), JMF (Java Media Framework) and JAXP (API Java for XML Parsing).

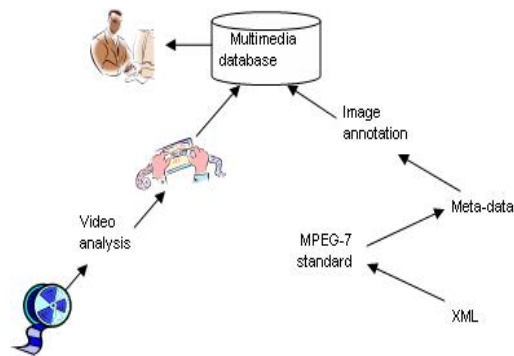


Figure 1. Overview of the three steps of the project

The choice of Java is justified by its portability, availability and object-oriented use, its JDBC features and its XML libraries (XML parser, with SAX (Simple API for XML) and DOM (Document Object Model) [2, 3].) The multimedia database management system, Oracle 10g Intermedia [4], enables management of multimedia documents. The Intermedia multimedia libraries offer pre-developed software (for example: the ORDSYS package) which provides ready-made classes for manipulating and storing multimedia documents.

3. DESIGN AND EXPERIMENTAL PROCEDURE

A modular design approach was chosen using the requirements expressed in use-cases. Three main modules have been developed and are further described as follows: **Operation of the IEU module** – A graphical interface allows the user to choose session parameters; for example, choosing the duration and sampling rate of the video (images/second). Once a video file is loaded, the player starts and the user can watch the video (video flow view frame). The system identifies a representative image frame for each scene (image creation). These images are displayed in a specific window.

Operation of the ITU module – This module, which is responsible for image processing, based on a Java class that collects information about images using the PixelGrabber class from the java.awt.image package. The PixelGrabber class offers a method, named grabPixels(), which allows us to load image pixels into an integer table. The table is used to extract the color associated with each pixel based on a given color model. The class getColorModel() is used in Java to obtain this information. It is then possible to obtain the color of each pixel and approximate a dominant color as defined by MPEG-7. The following information is displayed at each stage of the image processing process:

- List of image names: Images are loaded from the database (image loading) as a java list, sorted according to their name. (IEU had previously created each file name based on its location during the extraction process);
- Database connection information: The status of the connection is shown, respecting the Oracle 10g (*driver.drive@pc:port:db,user, password*) format;
- Creation of a new database: The status is shown when creating a new database;
- Data base image insertion: Specific ORDImage and ORDImageSignature fields are displayed, as well as OrdImage object creation, with entry flow performed by getBinaryOutputStream() and exit flow coming from images (FileInputStream);
- MPEG-7 files describing an image: The Java (JAXP library) classes are used to obtain an MPEG-7-compatible document. The *TransformerFactory* class is used for this purpose. It is based on the *trans.transform(source,result)* model, where *transform* is the method of a “*trans*” instance allowing a DOM-type output stream from an XML document previously created using the *DocumentBuilderFactory* class (of the java JAXP library).
- Image identification: This process analyzes the similarity of successive images in order to identify scene changes. If two successive images are assessed as different, an indicator is placed in the database indicating a possible scene change. The Oracle 10g ‘*IsSimilar*’ method available in the ORDImageSignature class is used for establishing differences (i.e. it returns 1 for similar images and 0 for different ones). The general syntax of this method is as follows: *isSimilar (sign1, sign2, chain attributes, threshold similarity)*,

where *sign1* and *sign2* are the two signatures to be compared, *chain attributes* is an alphanumeric chain specifying the low-level attribute vector of the image (for example: "color=0.3 texture=0.5 shape=0.1 location=0.1") and *threshold similarity* is an integer value (with a maximum of 100) which must be experimentally set and which decides whether or not a picture is to be considered as similar.

Operation of the VU module – This module allows database image visualization through a graphic interface. Two visualization choices are available to users. One option makes it possible to choose among all the images. The other option gives access only to images associated with the beginning of a new scene. Image visualization can be accompanied by MPEG-7 descriptive data. Figure 2 shows an example of an image associated with the beginning of a new scene. The VU graphical interface shown in Figure 2 contains two main sections:

- Input functions panel, containing editing input fields for personalizing the database connection (user name, password) and used for selecting the parameters for the queries to the server (image title pop-up list, sequences title pop-up list, Visualization action button and Mpeg-7 action button)
- Output panel, with three sub-sections:
 - left sub-section: display the chosen image;
 - center sub-section: display the chosen scene beginning;
 - right sub-section: display MPEG-7 information associated with the image displayed in the center sub-section.

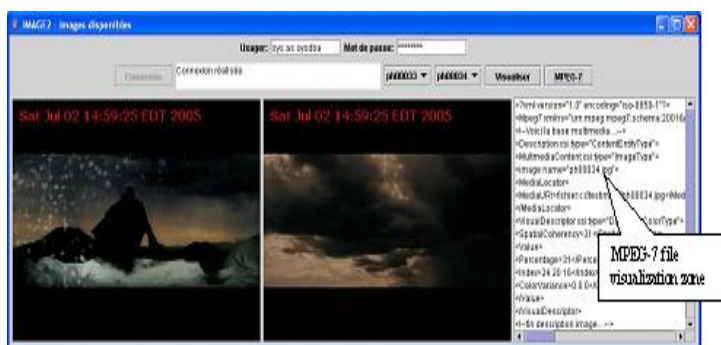


Figure 2. VU interface – MPEG-7 file visualization

The section displayed on the left is principally used as visual confirmation for automatic scene detection. Values chosen for both pop-up lists (images and scenes) were used as query parameters when the visualization button was activated. Results returned from the server database are then displayed.

3.1 Experimental Procedure

Once the unit and system tests had been finalized, there was a need to describe how the project team approached the iterative steps to adjust the scene identification parameters. The following text describes the experimental procedure used to automatically find the images that best represent a new scene beginning:

- video analysis step (to understand the video semantics);
- video image analysis step (to assess image relevance);
- "manual" identification of scene changes step (to identify the target of the automatic identification);
- automatic identification step, which is composed of:
 - quantitative data: precision, recall, noise and silence calculation;
 - qualitative data: direct answer observation (semantic analysis);
 - adjustments: low-level image descriptor weights and threshold value modifications;
 - Experimentation process iterations: application launch → data collection and analysis → parameter modification iterations, in order to obtain the best results.

This approach [8] is similar to that used by students in an undergraduate multimedia database course at the École de Technologie Supérieure who had to experiment with:

- Comparing image "labels" (in our case, labels are image signatures, as viewed by Oracle 10g);
- Analyzing image similarity (students build an alphanumeric chain containing low-level image attributes and their associated weights for purposes of comparison).

4. INTERPRETATION OF RESULTS

4.1 Interpretation context

Managing and handling multimedia documents pose several challenges, at both the technical and interpretation levels, and require a multi-level approach [8] in order to take into account: their spatial dimension (volumes, surfaces, lines, relative positioning, etc.), their time dimension (image order, duration, synchronization, etc), their hierarchical dimension (tree structure: video-clips-scenes-sights) and their content dimension (objects, relations between objects). Moreover, as emphasized in [1, 5, 6], the semantic level of multimedia data is of prime importance as it reveals the high-level content description (representation of the objects, event and action concepts). It was found that a completely automatic indexing process was not possible, because, being purely descriptive (based on temporal discontinuity identification and low-level metadata descriptors), an essential semantic aspect was not considered. Moreover, human intervention was still required to validate the choice of images. This approach of ensuring the necessary "feedback" translates, in our case, to modification of the weight and similarity threshold value associated with the low-level descriptors, i.e. the value beyond which the analyzed model and the target can be regarded as different.

4.2 Results obtained

As a result of executing the experimental steps described in the previous section, we have collected data based on definitions formulated in [8] concerning:

- Precision: number of relevant answers divided by total number of answers: $P = (\text{Relevant Answers} / \text{Total Answers})$;
- Recall: number of relevant answers divided by all relevant data: $R = (\text{Relevant Answers} / \text{Relevant Data})$;
- Noise: no relevant information returned: $N = (\text{Total Answers} - \text{Relevant Answers}) / \text{Total Answers}$;
- Silence: relevant information not returned: $S = ((\text{Relevant Data} - \text{Relevant Answers}) / \text{Relevant Data})$.

By iteratively modifying the values of the image vector attributes (weights and similarity threshold value), it was possible to obtain a gradual improvement in relevance. A visual scene identification is based on the identification of a "whole of the plans" connected in time and space by their semantic content [1, 7, 8]. A 30-second video was analyzed manually and 12 images were chosen as representative of the beginning of a new scene (RD=12). These were used as reference images to be found, it was hoped, by the automated identification process. Figure 3 shows an alternative, and more visual, presentation of the same data points, which facilitates their interpretation. The goal of iteratively changing the image vector weights is to obtain results as close as possible to those found by a human (which is subjective, because scene identification is a semantic interpretation of content). Some observations can be made about Figure 3:

- Recall and noise curves take similar forms; therefore, for approaches the ideal situation (finding all images that identify a given scene) we obtain a "noisy" situation (obtaining enough irrelevant information).
- Precision and silence curves take similar forms; therefore, the most precise answers reduce the quantity of relevant information returned.

The ideal situation is a balance between recall and precision. This is observed when the precision and recall values are greater than the noise and silence values. That means more relevant answers (from all relevant data) and good accuracy condition (less non relevant information is disturbing the result). These situations appear in Figure 3 in cases 8 and 15, and 17 to 20. Case 20 is considered better than case 8 in terms of the recall-precision/noise-silence compromise. Case 15 offers very good recall, but is less precise because of noise. Two cases (17 and 19) are offering similar good results, but 19 has a small value in the similarity threshold (it detected a false scene beginning). The best result was obtained in case 17, that offers the higher value for the precision and a good recall, the smallest value for the noise and an acceptable silence. That case was based on: 0.7 for color weight value, 0.3 for texture weight value and 20 for similarity threshold value. The color-texture ratio seemed to have the most impact on the relevance obtained. Experimental results support the statement, as did [1], that the great majority of CBR databases, and up to 90% of CBIR functionality, are founded on the color characteristic.

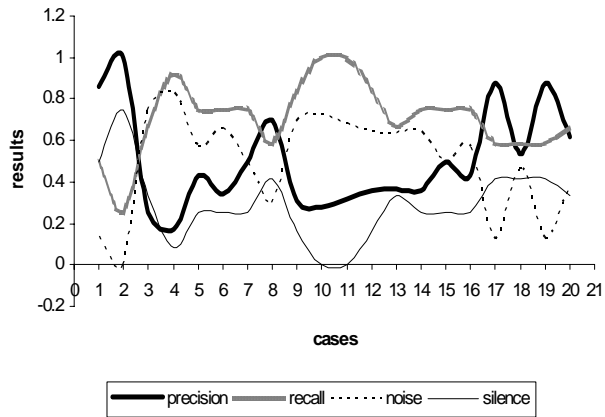


Figure 3. Experimental results

5. CONCLUSION AND FUTURE WORK

This paper described the development of several libraries to annotate and present MPEG-7 documents in connection with a multimedia database system. The prototype is able to analyze images drawn from a video in order to identify the inter-sequence descriptors and to perform effective annotation using MPEG-7 descriptions. Experimental results confirmed the Oracle 10g parameter-setting values for the annotation process. We are considering two scenarios for future work:

- Development of libraries for a similarity-based image search: analyzing the image model, calculating values for associated descriptors and comparing these values with those of corresponding descriptors for images stored in the MMDB (parsing existing XML documents, or re-analyzing the images themselves);
- Use of soundtrack in the video analysis, by adding software modules allowing its treatment. The image-sound cross-analysis has good potential to improve the annotation module.

ACKNOWLEDGEMENT

Thanks to Dr. Harald Kosch and Christian Hofbauer of the Institute of Information Technology, University Klagenfurt, Austria for their support and advice on this master degree student project.

REFERENCES

- [1] Mostefaoui, A. et al, 2004. *Sirsale: un système d'indexation et de recherche de séquences audiovisuelles à large échelle*, Gestion des données multimédias, Hermes publisher, Paris, France, pp.283-306.
- [2] Sun Java Web page, [On line] <http://java.sun.com/xml/>
- [3] Gardarin, G., 2002. *XML*, Dunod publisher, Paris, France, 400p.
- [4] Oracle interMedia [On line]: <http://www.oracle.com/technology/products/intermedia/index.html>
- [5] Ionescu, B. et al., *Analyse et caractérisation de séquences de films d'animation*, présentation à ORASIS2005 - 9ème Congrès Jeunes chercheurs en Vision par Ordinateurs, Mai 2005 [On line] <http://orasis2005.univ-bpclermont.fr/user/www/orasis/papiers/042.pdf>
- [6] Mulhem, P. et al., *Modèles pour résumés adaptatifs de vidéos - Bases de données et multimédia* (Ingénierie des systèmes d'information RSTI série ISI-NIS Vol.7 N° 5-6/2002), pages 91-118, ISBN : 2-7462-0684-6
- [7] Chen, L., Chahir, Y., 2004. *Indexation de la vidéo numérique*, Gestion des données multimédias, Hermes, Paris, pp. 306-334, ISBN 2-7462-0824-5.
- [8] April, A. Cours de *Bases de données multimédia (GT1440)*, Département de génie logiciel et des TI, ÉTS, Université de Québec, Montreal, Canada.