Exploration using a commercially available database for the identification of video images scenes

Horia Ilie[1], Alain April[1] , Harald Kosch [2], Christian Hofbauer [2], and Pierre Bourque[1]
[1] École de Technologie Supérieure, 1100 Notre-Dame West, Montreal, Canada
[2] Institute of Information Technology, University Klagenfurt, Austria
Ilie.Horia.1@ens.etsmtl.ca, aapril@etsmtl.ca, harald.kosch@itec.uni-klu.ac.at, hchofbaue@edu.uni-klu.ac.at , pbourque@etsmtl.ca

***Abstract*- This paper investigates image vector attributes weights in order to detect a scene change in a video. Another objective is the development of an MPEG-7 library for annotating and presenting MPEG-7 documents. The contributions of this paper are: an exploration of the limits of scene identification technique, using the Oracle 10g database, describing an annotation technique to generate MPEG-7 annotations.**

I. 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 libraries.

Thanks to recent research efforts in metadata standardization, the management of multimedia content is becoming a reality. It is within this context that the new ISO-IEC MPEG-7 Multimedia Description Standard ([2], [3]) is offered as a means to represent and search multimedia by its content. For instance, MPEG-7 allows for the description of images at the following levels:

- Low level attributes (color, texture, form);
- Source (title, author, date, format, etc.);
- Conditions of use (royalties, etc.);
- Design features (format, coding, etc.);
- High level concepts (people, objects, etc.);
- Space and temporal structure (positions, displacements, etc.).

During MPEG-7 development and following its release, many experiments were published on the analysis and description of images and video. All these experiments have demonstrated the potential use of parts of the MPEG-7 standard. Some tools are lacking due to the use of non-standardized MPEG-7 elements, while others rely on special input formats only. To the best of our knowledge, the presentation and annotation aspects have not yet been considered in any single framework.

This paper is organized as follows: Section I presents the MPEG-7 concepts that are used to describe multimedia content, some examples of publications that have used these concepts and the overview of the project. Section II presents an overview of the project objectives, technology and development approach.

Section III presents the design choices, as well as the way in which each module operates. It describes the experimentation to assess the performance of the library components developed. Section IV discusses the experimentation results, and, finally, section V presents conclusions and directions for future work.

II. PROJECT OVERVIEW

The long term objective of the project is to develop libraries which are able to create and present MPEG-7 standard documents. In order to achieve this, we placed a multimedia database at the core of our project with a view to addressing the various phases of using and processing its metadata using MPEG-7 descriptive data. A first database stores annotated videos and images, and proposes interfaces for retrieving and presenting documents. Figure 1 presents an architectural overview of this first software library component. First, a video is processed with the goal of detecting a change of scene. Second, a sample image is chosen as a candidate to reference each segment of the video. This step includes video annotation using the MPEG-7 standard. The third and final step is presentation and visualization by the end-user.

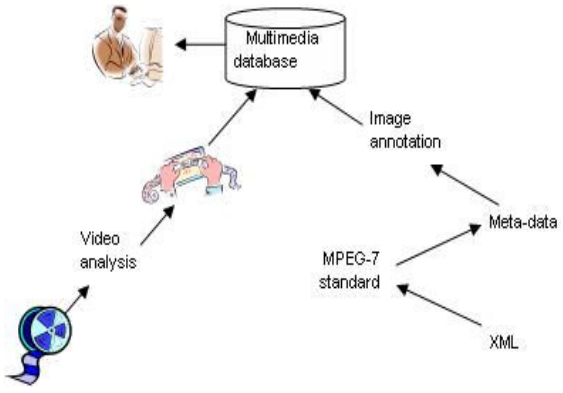

Fig. 1. Overview of the Video Processing Steps

A. 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).

The choice of Java is justified by its portability, availability, robustness, security and object-oriented use, its JDBC features and its XML libraries (XML parser, with SAX (Simple API for XML) and DOM (Document Object Model) [4, 5].) The multimedia database management system, Oracle 10g Intermedia [6], 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.

In this project, the ORDImage and ORDImageSignature were used. Figure 2 shows how a direct connection between Oracle ORDImage objects and Java can be achieved through a JDBC connection.

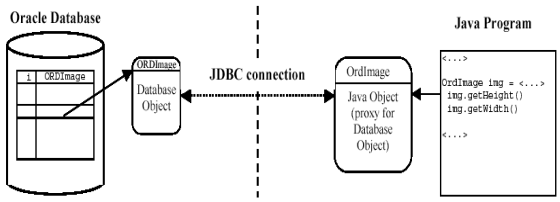

Fig. 2. ORDimage and Java connection [6]

B. Steps Towards Developing the Libraries

The work was separated in three modules: 1) video preparation; 2) image processing; and 3) the user interface. Activities were identified for each module (see Table I).

TABLE I
SEGMENTATION IN MODULE AND ACTIVITIES

Module	Activity
Image extraction (IEU – Image Extraction Unit)	1. Start up 2. Load video 3. Handle exceptions 4. Display handling
Image processing ITU (Image Treatment Unit)	1. Connect to DB 2. Create physical tables 3. Insert images 4. Create MPEG-7 descriptive data 5. Insert records 6. Detect scenes 7. Manage exceptions 8. Display management
User Interface VU (Visualization Unit)	1. Connect to DB 2. Choose image 3. Display image 4. Show MPEG-7 descriptive data 5. Handle exceptions

Use-cases were developed for each activity, identifying clearly the interactions between the user and the interface for each function. An example of a use-case is presented in Table II.

TABLE II
TYPICAL USE-CASE DEVELOPED FOR AN ACTIVITY

Summary description	The user clicks on the "MPEG-7" button. Display the contents of the XML file.
Primary actor	The user
Initiation rule	The user clicks on the "MPEG-7" button.
Process Description	Clicking the"MPEG-7" button launches a parameterized request (a user-selected value taken from the list on the right-hand side of the screen) to the multimedia data base. The generated XML file is displayed in a specific area of the user interface.
Termination Rules	The XML file is displayed. The user can exit the software.

III. DESIGN AND EXPERIMENTAL PROCEDURE

A. Design Overview

A modular design approach was chosen using the requirements expressed in use-cases. Each design module is further described as follows:

How the IEU module operates – Figure 3 presents the design of this module. A graphic interface allows the user to choose session parameters; for example, choosing the duration and sampling rate of the video. 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.

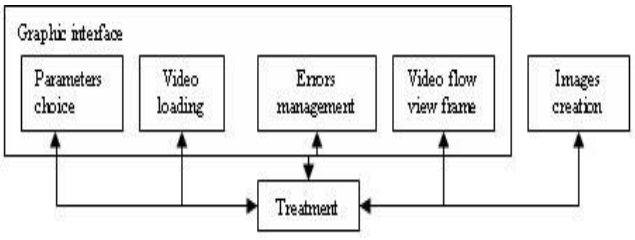

Fig. 3. Design of IEU (Image Extraction Unit)

How the ITU module operates – Figure 4 presents an overview of the design of 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.

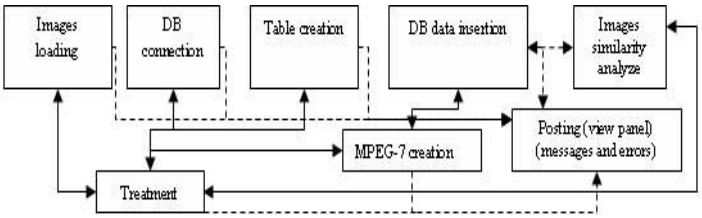

Fig. 4. Design of ITU (Image Processing Unit)

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;
- Creation of a new database: The status is shown when creating a new database;
- Database image insertion: Invoke specific ORDImage and ORDImageSignature classes to display the image and its signature, as well as OrdImage object creation, with entry flow performed by getBinaryOutputStream() and exit flow coming from images (FileInputStream);
- Creation of MPEG-7 files describing an image: The Java (JAXP library) classes are used to obtain an MPEG-7-compatible document. The *TransformerFactory* class is invoked for this purpose. It is based on the *trans.transform(source,result)* class, 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 similarity analysis: 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 image attribute vector (for example: "color=0.3 texture=0.5 shape=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. Figure 5 shows an example of an XML document that is presented to the user:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001
Mpeg7-001.xsd">
<Description xsi:type="ContentEntityType">
    <MultmediaContent xsi:type="ImageType">
        <image name="ph00000.jpg">
            <MediaLocator>
              <MediaURi>
              fichier:c://testimag//ph00000.jpg
              </MediaURi>
            </MediaLocator>
            <VisualDescriptor
            xsi:type="DominantColorType">
              <SpatialCoherency>31</SpatialCoherency>
              <Value>
                <Percentage>31</Percentage>
                <Index>27 25 22</Index>
                <ColorVariance>0 0 0</ColorVariance>
              </Value>
            </VisualDescriptor>
        </image>
    </MultmediaContent>
 </Description>
</Mpeg7>
```

Fig. 5. MPEG-7 File Example (*DominantColor* Characteristic)

How the VU module operates – Figure 6 presents an overview of the design of this module, which allows database image visualization through a graphical interface.

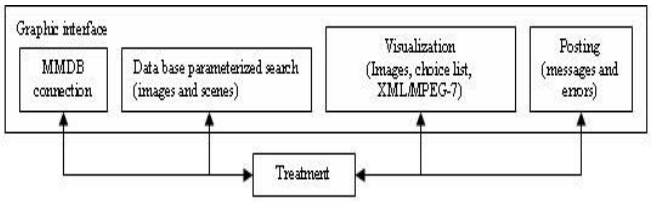

Fig. 6. Design of VU (Visualization Unit)

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 7 shows an example of an image associated with the beginning of a new scene. The VU graphical interface shown in Figure 7 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 beginning of the chosen scene;
  - right sub-section: display MPEG-7 descriptive data associated with the image displayed in the center sub-section.

Fig. 7. VU Interface – MPEG-7 File Visualization

The section displayed on the left of Figure 7 is used as a visual confirmation for scene change 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.

B. *Tests and Experiment*

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 iterative experiment process used to find the images that best represent a scene transition in a video:

- 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 identification);
- identification steps, which is composed of:
  - experiment measures: precision, recall, noise and silence calculation;
  - qualitative data collection: direct answer observation (semantic analysis);
  - image attribute vector scenarios: 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 is similar to that used by students in an undergraduate multimedia database course at the École de Technologie Supérieure [11] who:

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

IV. Interpretation of Results

A. *Interpretation Context*

Managing and handling multimedia documents pose several challenges, at both the technical and interpretation levels, and require a multi-level approach [11] 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], [8] and [9], 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 an automatic identification 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.

B. *Results Obtained*

As a result of executing the experiment steps described in the previous section, we have collected data based on definitions formulated in [11] and [12] concerning:

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

A suggested image for results interpretation is as follows:

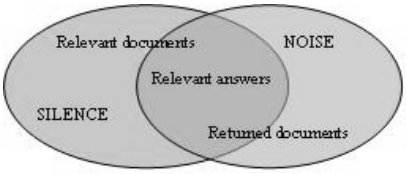

Fig. 8. Concepts of Noise and Silence [8]

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 scene" connected in time and space by their semantic content ([1], [10] and [11]).

A 30-second video was analyzed manually and 12 images were chosen as representative of the beginning of a new scene. These were used as reference images to be found, it was hoped, by the identification process. The identification values obtained are presented in Table III. Figure 9 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).

TABLE III
DETAILED EXPERIMENT RESULTS

Case	color	texture	form	localization	threshold	results	relevant results	precision	recall	noise	silence
1	0.3	0.7	0	0	10	7	6	0.8571	0.5	0.143	0.5
2	0.1	0.9	0	0	10	3	3	1	0.25	0	0.75
3	0.1	0	0.9	0	10	32	8	0.25	0.6667	0.75	0.3333
4	0.1	0	0	0.9	10	63	11	0.1746	0.9167	0.825	0.0833
5	0.5	0.5	0	0	10	21	9	0.4286	0.75	0.571	0.25
6	0.7	0.3	0	0	10	26	9	0.3462	0.75	0.654	0.25
7	0.4	0.6	0	0	10	18	9	0.5	0.75	0.5	0.25
8	0.35	0.65	0	0	10	10	7	0.7	0.5833	0.3	0.4167
9	0.35	0.25	0.4	0	10	32	10	0.3125	0.8333	0.688	0.1667
10	0.7	0	0	0.3	10	43	12	0.2791	1	0.721	0
11	0.7	0	0.3	0	10	38	12	0.3158	1	0.684	0
12	0.7	0	0.3	0	15	28	10	0.3571	0.8333	0.643	0.1667
13	0.7	0	0.3	0	20	22	8	0.3636	0.6667	0.636	0.3333
14	0.7	0	0	0.3	15	25	9	0.36	0.75	0.64	0.25
15	0.7	0	0	0.3	20	18	9	0.5	0.75	0.5	0.25
16	0.7	0.3	0	0	15	21	9	0.4286	0.75	0.571	0.25
17	0.7	0.3	0	0	20	8	7	0.875	0.5833	0.125	0.4167
18	0.7	0.3	0	0	19	13	7	0.5385	0.5833	0.462	0.4167
19	0.5	0.5	0	0	15	8	7	0.875	0.5833	0.125	0.4167
20	0.5	0.4	0.1	0	15	13	8	0.6154	0.6667	0.385	0.3333

Some observations can be made about Figure 9:

- 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 irrelevant images).
- Precision and silence curves take similar forms; therefore, the most precise answers reduce the quantity of relevant information returned.

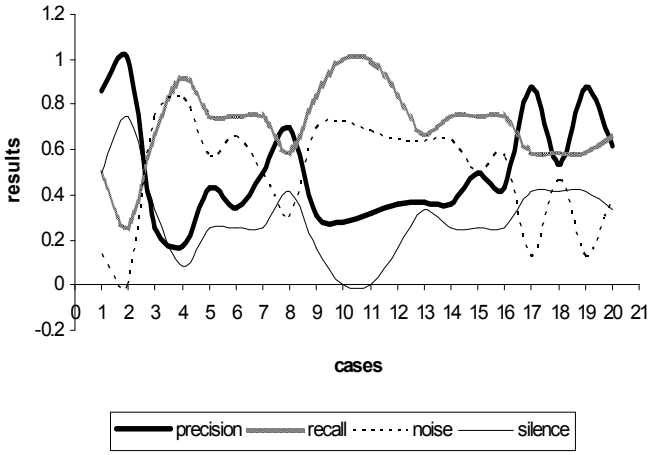

Fig. 9. Precision and Recall

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 9 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. An important conclusion of this research is that these weights and threshold value have been found to be give the best performance using Oracle 10g across parts of six different music videos we have tested.

The color-texture ratio seemed to have the most impact on the relevance obtained. Experimental results support the statement, as did [1, 3, 12], that the great majority of CBR databases, and up to 90% of CBIR functionality, are founded on the color characteristic.

C. *Limits and Possible Improvements*

Main processing – A color-based analysis gives satisfactory results, but such an approach, based on global treatment, also brings with it several problems [10]. One of the most frequent of these is related to the inability to process camera movement in a relevant way (see Figure 10). For example, the system identified this camera movement as a scene change.

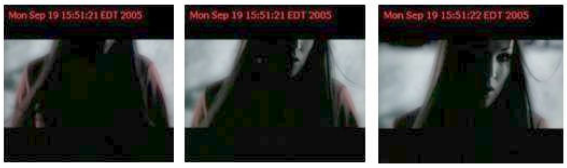

Fig. 10. Camera movement interpreted as a scene change

Second, false positives occur with a change in luminosity. Figure 11 presents the case of image 00010, often considered as the beginning of scene because of the presence of a lightning flash in the background.

Fig. 11. Changing Luminosity

Improvement is possible [10] if the image is cut into several blocks and each is analyzed independently. In this approach, only one block is affected, thereby decreasing the influence on global image assessments.

Essential human control – Exclusively based on temporal discontinuity detection, the system uses global descriptors (color, texture, etc.) and leaves semantics-related aspects untreated. This deficiency must be corrected by a human observer.

Subjectivity – The value of the similarity threshold is established following successive iterations, while analyzing results and applying corrective feedback. This method gives approximate results, because, in reality, a sequence image membership must be dictated by an analysis performed on a succession of several images belonging to the same sequence. Improvement is possible [10, 12] by introducing an adaptive threshold, which would consider the noise and camera movements.

V. Conclusion and Future Work

This paper described the work of a master degree student experimentation using Oracle 10g multimedia libraries. This work experimented using Oracle 10g feature description weights for the identification of video image scenes. It also presents a prototype system developed to annotate and present MPEG-7 annotations of an image. The prototype is able to analyze images drawn from parts of music videos in order to identify the images that best represent a scene and to perform its annotation using MPEG-7 descriptions. Experimental results identified the Oracle 10g parameter-setting values for the best results in this annotation process for music videos.

We are considering two scenarios for future work:

- Development of libraries for a similarity-based image search: analyzing images, calculating values to associate MPEG-7 descriptors and comparing these values with those of corresponding MPEG-7 descriptors stored in a 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.

Acknowledgment

Thanks to Dr. Harald Kosch for providing expertise to the Montreal ETS Multimedia Laboratory.

References

[1] A. Mostefaoi, F. Prêteux, V. Lecuire, J.M.Moureaux. *Sirsale: un système d'indexation et de recherche de séquences audiovisuelles à large échelle*, Gestion des données multimédias, 283–306, 2004.

[2] H. Kosch, ***Distributed multimedia database technologies supported by*** *MPEG-7 and MPEG-21*, 2004, CRC press LLC, USA, ISBN 0-8493-1854-8.

[3] L. Dunckley, ***Multimedia Databases. An object-relational approach***, Pearson Education Ltd 2003, Great Britain, ISBN 0-201-78899-3.

[4] Sun Java Web page: http://java.sun.com/xml/

[5] Gardarin, Georges, *XML*, 2002, Dunod, Paris, 2002, ISBN 2100069330

[6] Oracle interMedia [On line]: http://www.oracle.com/technology/products/intermedia/index.html

[7] Web Master Hub, Noise and Silence, 2005 [On line]: http://www.webmaster-hub.com/publication/IMG/gif/bruit-silence.gif

[8] B.Ionescu, D.Coquin, P.Lambert, V.Buzuloiu, *Analyse et caractérisation de séquences de films d'animation,* ORASIS2005 - 9ème Congrès Jeunes chercheurs en Vision par Ordinateurs, May 2005.http://orasis2005.univ-bpclermont.fr/user/www/orasis/papiers/042.pdf

[9] P. Mulhem, J. Gensel, H. Martin, *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), 91-118, ISBN : 2-7462-0684-6

[10] L.Chen, Y. Chahir, *Indexation de la vidéo numérique*, Gestion des données multimédias, pages 306-334, Hermes, Paris, 2004, ISBN 2-7462-0824-5.

[11] A. April, *Multimedia DataBase Course****(GTI440)***, Département de génie logiciel et des TI, ÉTS, Université de Québec, winter 2006.

[12] C. Arsenault, Cours de *Recherche d'information avancée****(BLT6322)***, École de bibliothéconomie et des sciences de l'information, Université de Montréal, session hiver 2005.