AN INTUITIVE GRAPHIC ENVIRONMENT FOR NAVIGATION AND CLASSIFICATION OF MULTIMEDIA DOCUMENTS

M. Campanella, R. Leonardi, P. Migliorati

Signals and Communications Lab - DEA University of Brescia, 25123, Brescia, ITALY

Abstract

In this work we propose an intuitive graphic framework for the effective visualization of MPEG-7 low-level features, in the context of classification and annotation of audio-visual documents. This graphic tool is proposed to facilitate the access to the content, and to improve a quick understanding of the semantics associated to the considered document. The main visualization paradigm employed consists in representing a 2D feature space in which the shots of the audio-visual document are located. In another window, the same shots are drawn in a temporal bar that gives the users also the information related to the time domain. In the main window, shots with similar content fall near each other, and the proposed tool offers various functionalities for automatically and semi-automatically finding and annotating shot clusters in the feature space. The use of the proposed system to analyze the content of few video sequences has shown very interesting capabilities.

1. Introduction

The extraction and exploitation of the significant low-level features has been a point of crucial interest in several recent research works. Low level features are widely used for browsing, indexing and retrieval of text-based and multimedia documents, and for many other applications. The MPEG7 standard has been therefore developed to define what these features represent, and how they should be described and suitably organized.

With the extraction from each document of these features we obtain a large amount of information, sometime difficult to be efficiently used. What appeared quite attractive was the direct use of these low-level descriptors to provide a quick feedback of the content of the considered audio-visual programme. The experiments presented in [2] have shown that, by adequate presentation, the low-level features carry instantly semantic information about the programme content, given a certain programme category, which may thus help the viewer to use such low-level information for navigation or retrieval of relevant events. This may be an attractive procedure with respect to using sophisticated search or navigation engines, especially if the program category is not adequately recognized.

Following this idea, we propose in this paper a graphic environment that allow the efficient visualization of the MPEG-7 low-level features with different paradigms. In more detail, the audio-visual documents are considered as sequences of shots, and for each shot some MPEG7 features are extracted and properly displayed. In this way, a shot becomes a point in the feature space, and the associated features represent its coordinates. The application displays this features in a 2D cartesian plane, in which each of the two axes corresponds to a specific feature type (selected by the user), and the shots are positioned in this plane accordingly to these coordinates.

A second window in which the shots are drawn in a temporal bar gives the users the information about the time domain that lacks in the cartesian plane. In a third window the key-frames associated of the current shots are displayed.

Navigating jointly with these three windows improves the accessibility of the documents and the quick understanding of its semantics. Moreover, the video shots with the same content will appear clustered in the same regions of the feature plane, and the application can automatically select these clusters and annotate them producing an XML file in MPEG7 format as output.

The rest of the paper is organized as follows. In Section 2 a quick overview of the related literature is considered. In Section 3 the proposed application is presented in more detail, whereas in Section 4 the performance of the application are evaluated and discussed. Concluding remarks are drawn in the final section.

2. A quick overview of the the state of the art

The use of a multidimensional feature spaces to visualize the content of a multimedia document has been already applied in some systems, typically related to multimedia analysis, retrieval applications and text-based searching and browsing systems.

In [3] a multimedia retrieval system is described in which

0-7803-9332-5/05/$20.00 ©2005 IEEE

the user can perform a query for an image and visualize the set of results in a 2D projection of the feature space instead of a 1D list of images ordered by similarity. This helps the user to understand the semantic relations between the images better than merely looking to a list of results.

In [4] visualization is used for a text document retrieval system. The documents retrieved during a query are displayed as points in a 2D space, keywords are displayed as points too. The closer a document is to a keyword, the higher is the relevance that the keyword has in that document. In [5] the same system is improved with other visualization paradigms, such as representing documents on a circle trying to maintain the same distances that the documents have in the feature space.

In [6] further solutions to the same problem are proposed. Visualization paradigms are implemented so as to provide an overall perspective to the results of a query, showing the general distribution of the documents in the feature space leading to potential clusters. All these efforts are moved by the idea that a graphical view of the content of a multimedia document can give a much more clear and intuitive information about the contents than a list of numbers or a series of text captions or images.

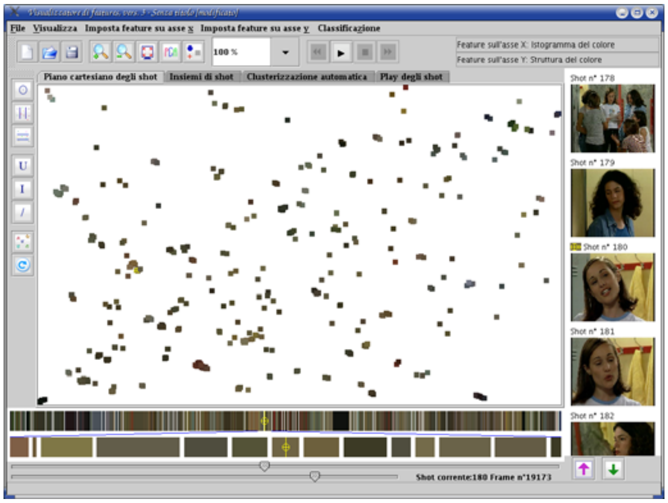
In [2] the visualization is applied with the aim of recognizing video program types. The video programs are divided in shots, and each shot is labelled with one of some visual classes, and one of four audio classes (silence, speech, music, noise). The classification is performed over the low-level features of the shots. A cartesian plane is displayed in which the X axis is the time axis, and on the Y axis the video or the audio labels are shown. In this graph the shots of the program and their associated labels according to the audio and visual classes they belong are displayed to provide a direct feedback on the program's content.

Visualization can be also used and has been used for more general purposes too. The application proposed here is general: it has been developed to study several possible applications of visualization. We are testing the usefulness of visualization for classifying and annotating multimedia documents. Another example of an annotation tool can be found in [7].

3. The proposed framework

In this section the main functionalities of the proposed framework, named FutureViewer, are described in some detail.

As previously mentioned, the elementary data unit is the shot, each feature is therefore referred to a single shot of a video sequence. Each shot is associated to a vector of feature values, one for each feature type that has been extracted. So, for example, in a video sequence each shot has its own value of spatial coherency, dominant color, intensity of motion, edge histogram, and so on. These feature values represent the coordinates that position the shot in the feature space, and each axis of this feature space is associated to a specific feature type.



Feature sull'asse X: Istogramma del colore
Feature sull'asse Y: Struttura del colore

Shot corrente:180 Frame n°19173

Visualizzatore di features, vers. 3 - Senza titolo [modificato]
File
Visualizza
Imposta feature su asse x
Imposta feature su asse y
Classificazione
100 %
Piano cartesiano degli shot
Insiemi di shot
Clusterizzazione automatica
Play degli shot
Shot n° 178
Shot n° 179
Shot n° 180
Shot n° 181
Shot n° 182

Figure 1: A screenshot of the proposed application.

3.1. The main visualization functions

In Fig. 1 the Graphic User Interface (GUI) of the visual environment is shown. This is divided into three main regions. The most important is the central region, in which a cartesian plane is displayed. The two axes of the cartesian plane correspond to the two feature types selected by the user.

The user can select the feature type that is associated to each of the two axes; for example, the dominant color can be selected and associated to the x-axis (colors are ordered by the hue in the red-violet chromatic scale), while the intensity of motion can be put on the y-axis. The program draws the shots in the cartesian plane as little squares filled with the dominant color of the shot; so, in our example, the user will see the shots with hue tending to red and little motion near the origin of the axes, meanwhile shots with high motion activity and violet-tending hue will be put in the upper right corner of the cartesian plane. Zoom in and zoom out functions are also implemented to allow to understand little portions of the cartesian plane. It's also possible to choose the number of shots to be simultaneously represented in the cartesian plane.

Observing the shots positioned in the feature space, the user can understand the semantic relationships between the shots, and can see at a glance where are the most significant shot clusters in the movie. Moreover, by associating different feature types to the cartesian plane's axes, it's possible to observe how the distances between the shots change. By selecting color or texture features, the user can, for example, see if some shots have similar texture patterns but different colors…

Instead of associating only two features to the plane's axes, it's possible to define an n-dimensional feature space with many features (whose MPEG-7 descriptors can consist of many coefficients). The n-dimensional space can be displayed in the cartesian plane reducing its dimension to two. This dimension scaling is performed by a linear transformation technique called *"Principal Component Analysis"* ([9]). This technique partially solves the problem of dimension reduction finding the globally optimal solution, where "optimal" means that the mean square error between the inter-shots distances in the n-dimensional space, and the distances in the two-dimensional space is minimized.

In the south region of the GUI a color-bar is drawn. This is a bar representing the video in the temporal domain, where the leftmost regions of this bar represent earlier shots in the video, and rightmost represent later shots. In this bar each portion corresponds to a shot and is drawn as a color stripe with width proportional to the temporal duration of the corresponding shot. These stripes are filled with the dominant color of the corresponding shot. So, in a sport video, soccer shots will be drawn as green stripes, whereas swimming shots will be drawn as blue stripes. The color-bar offers an intuitive view of the whole video sequence, and it's a powerful low-level features representation tool, as discussed in [8]. By clicking on a shot of the color-bar, a pointer appears in the cartesian plane indicating the square that corresponds to the selected shot.

In the east region of the GUI, the key frames related to the shots are represented. The user can scroll these key frames in temporal order. The cartesian plane, the color-bar and the key frames panel represent the same semantic units, the shots, with three different visualization paradigms. Their behavior associated to the user's actions (e.g.: mouse clicks) is designed to support a mixed navigation, watching the same audio-visual document from different points of view. In fact, if the user clicks on a shot in one of the three windows (a little square in the cartesian plane, a color stripe in the color-bar, a key frame in the key frames slide) a coloured pointer will appear and will indicate where is the shot in the other two windows. So it's possible to identify the temporal position on the color-bar of a given square representing a shot on the cartesian plane, or which is its key frame between that displayed in the key frame panel; or, vice versa, given a position on the color-bar, it's possible to find out the corresponding shot in the features plane. The principal aim of these funcionalities is to increase the accessibility of the user to the document and its semantic structure, by displaying at the same time the inter-shot relationships in the time domain and in the feature space domain.

3.2. The clustering and annotation functions

While browsing the shots in the cartesian plane, one consideration is of particular interest: in most cases shots form clusters in the cartesian plane. The application implements two functions to recognize and annotate these clusters. The first function consists of the following: by right-clicking on a little square in the cartesian plane, a menu pops up so that the user can choose to see the shots nearest in the plane to the clicked shot within a relative distance. These shots are displayed by showing their key frames in the key frames panel, ordered by distance. In this way the user can see if a cluster in the cartesian plane contains semantically similar shots. The same aim can be achieved with the second function: the user, dragging the mouse, can draw a circle on the cartesian plane and can see the shots that fall into it.

These similarities can be easily managed in the framework: whenever the user finds a group of shots that have semantic similarities they can be saved as a collection with a name and a description. Moreover, the user can browse and modify all the created shot collections . These shot collections are saved in the form of MPEG7 annotation descriptors, in which all the shots belonging to a given shot set receive the same annotation. Moreover, set operations like union, intersection and difference between shot sets are supported. To speed up the annotation process the application can automatically find the shots' clusters in the feature space. Shots are clustered in this feature space with an algorithm taken and modified from [10]. The calculated clusters are visualized by mapping their centroids from the n-dimensional space to a 2D plane with the Principal Component Analysis. An example of clusters visualization is shown in Figure 2. Visualizing clusters is another way to provide the user with the information about the documents semantic structure. Looking at Fig. 2, we can see that clusters are mainly organized along an horizontal axis. Looking inside the clusters, we can see that this axis coincides with the temporal dimension, while the distribution that the cluster have around this axis is caused by the inter-cluster distances in the audio-visual features domain.

Watching at this representation, the user gets a rapid idea of the principal clusters in the movie and of their position in the time domain and resemblance with respect to the features.

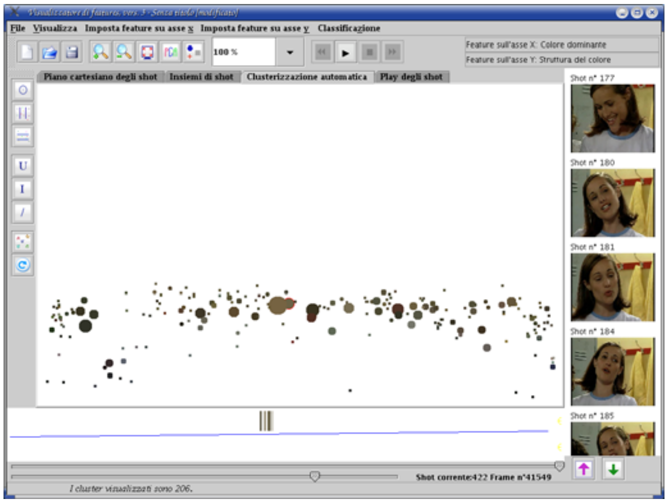
4. Performance assessment

The proposed environment has been implemented in Java and needs Java SDK 1.4.2 to be executed. The tool has been tested with some video documents of 40-50 minutes: a daily TV news program, a music program, a cartoon, a quiz program, a drama series, a sport program, a film.

We tried to identify the semantic structure of these movies by browsing and annotating them with FutureViewer. In particular, we identified in the documents the logical story units and the clusters.

A Logical Story Unit (LSU) is “a series of contiguous shots that communicate a unified action with a common locale and time” ([11]). In order to segment a movie into its LSUs, one must know the movie’s main story, events and characters. Using the shots’ representation in the cartesian plane, the mixed navigation and the clusters’ visualization it’s possible to know in a straightforward way the events and the characters that are repeated with the highest frequency in the document. So, classifying and annotating the TV programs tested with the aid of visualization, is far easier and faster than using a by-hand annotation approach.

Another aid to the movies’ classification is the automatic clustering. We evaluated the goodness of automatically clustering by finding from them the LSUs (using the algorithm defined in [12]) and comparing these to some ground-truth LSUs. To perform this comparison we used a standard method proposed in [11]; the obtained results are similar to those described in the state of the art. So, the visualized clusters represent efficiently the real logical structure of the considered audio-visual document.


Visualizzatore di features, vers. 3 - Senza titolo [modificato]
File
Visualizza
Imposta feature su asse x
Imposta feature su asse y
Classificazione
100 %
Piano cartesiano degli shot
Insiemi di shot
Clusterizzazione automatica
Play degli shot
Shot n° 177
Shot n° 180
Shot n° 181
Shot n° 184
Shot n° 185

Feature sull'asse X: Colore dominante

Feature sull'asse Y: Struttura del colore

Shot corrente:422 Frame n°41549

I cluster visualizzati sono 206.

Figure 2: Clusters visualization with Principal Component Analysis.

5. Conclusions

In this paper we have presented an intuitive tool for an effective visualization of MPEG-7 low-level audio-visual features. With this tool the user can explore graphically how the basic segments of a audio-visual sequence are distributed in the feature space, and can recognize and annotate significant clusters and their structure. Annotating documents with the aid of the proposed visualization paradigms is easy and quick, because the user has a fast and intuitive access to the audio-video content, even if he or she hasn’t seen the document yet. We are currently assessing the complete potentialities of FutureViewer.

References

[1] R. Leonardi, P. Migliorati, ”Semantic Indexing of Multimedia Documents”, IEEE Multimedia, vol. 9, no. 2, pp. 44-51, April-June 2002.

[2] B. Moghaddam, Qi Tian, T. S. Huang, “Spatial Visualization for Content-Based Image Retrieval”, Proc. ICME 2001, 22-25 Aug. 2001, Tokyo, Japan.

[3] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, J. G. Williams, “Visualization of a Document Collection: the VIBE System”, *http://Itl13.exp.sis.pitt.edu/Website/Webresume/ VIBEPaper/VIBE.htm (1992)*.

[4] J. Cugini, C. Piatko, S. Laskowsky, “Interactive 3D Visualization for Document Retrieval”, Proc. ACM CIKM 1996, Nov. 1996, Rockville, USA.

[5] M. Carey, D. C. Heesch, S. M. Ruger, “Info Navigator: a Visualization Tool for Document searching and Browsing”, Proc. of DMS ’2003, Sept. 2003, Miami, USA.

[6] C-Y Lin, B. L. Tseng, J. R. Smith, “VideoAnnEx: IBM MPEG7 Annotation Tool for Multimedia Indexing and Concept Learning”, Proc. ICME 2003, July 2003, Baltimore, USA.

[7] M. Barbieri, G. Mekenkamp, M. Ceccarelli, Jan Nesvadba, *The color browser: a content driven linear video browsing tool*, Proc. of ICME’2001, Aug. 2001, Tokyo, Japan.

[8] J. Edward Jackson, *A User’s Guide to Principal Components*, John Wiley & Sons, 1991.

[9] Tou, Julius T. and Rafael C. Gonzalez. 1974. *Pattern Recognition Principles*. Addison-Wesley Publishing Co.

[10] J. Vendrig, M. Worring. 2002. *Systematic Evaluation of Logical Story Unit Segmentation*. IEEE Transactions on Multimedia, Vol.4, No. 4, Dec. 2002, pp. 492-499.

[11] Minerva M. Yeung, Boon-Lock Yeo, *Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content*, IEEE Trans. CSVT, Vol. 7, No. 5, Oct. 1997, pp. 771-785.