FUTURE-VIEWER: AN EFFICIENT FRAMEWORK FOR NAVIGATING AND CLASSIFIYING AUDIO-VISUAL DOCUMENTS

Marco Campanella, Riccardo Leonardi, Pierangelo Migliorati

Signals and Communications Lab - DEA University of Brescia, Brescia, Italy

ABSTRACT

In this paper we present an intuitive framework named *Future-Viewer*, introduced for the effective visualization of spatio-temporal low-level features, in the context of browsing and retrieval of a multimedia document. This tool is used to facilitate the access to the content and to improve the understanding of the semantics associated to the considered multimedia document. The main visualization paradigm employed consists in representing a 2D feature space in which the video document shots are located. The features that characterize the 2D space's axes can be selected by the user. Shots with similar content fall near each other, and the tool offers various functionalities for automatically finding and annotating shot clusters in the feature space. These annotations can also be stored in MPEG7 format. The use of this application to browse the content of few audio-video sequences demonstrate very interesting capabilities.

1. INTRODUCTION

In recent research works in the field of multimedia signal processing, the extraction of low-level features has been a point of crucial interest. Low level features are widely used for many applications; in particular for browsing, indexing and retrieval of text-based and multimedia documents. The MPEG7 standard has been therefore developed to define what these features represent and how they should be effectively described and organized. With the extraction from each document of these low-level features we obtain a large amount of useful information: what appears quite attractive is to use low-level descriptors in providing a feedback of the content of the described audio-visual programme. Experiments presented in [1] have shown that, by adequate presentation, 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 developed a tool that visualizes the features of an audio-visual document with various visualization paradigms. Specifically, the audio-video 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 its coordinates. The application displays this feature space in a 2D cartesian plane in which each of the two axes corresponds to a feature type (selected by the user), and the shots are drawn in this plane accordingly to these coordinates. Another 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 of the shots are displayed. Navigating jointly with these three windows improves the accessibility of the documents and the understanding of its semantics. Since in the cartesian plane video shots are positioned according to their feature values, shots with the same content will appear clustered in the same regions of the plane, and the application can automatically find these clusters and annotate them producing an XML file in MPEG7 format as output. The application's input are the features of the video documents represented by MPEG-7 descriptors. The paper is organized as follows. In Section 2 a set of examples from the literature is considered. In Section 3 the proposed application is presented in detail. In Section 4 the performance of the application are discussed.

2. VISUALIZATION OF LOW-LEVEL FEATURES: THE STATE OF THE ART

The idea of using multidimensional feature spaces to visualize multimedia documents has been already applied in some systems, typically related to multimedia analysis and to retrieval systems, and text-based searching and browsing systems. In [2] a multimedia retrieval system is described in which 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 looking to a list of results. In

This material is based upon work partially supported by the IST programme of the EU in the project IST-2001-32795 SCHEMA.

[3] 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 relevance the keyword has in that document. In [4] the system is improved with other visualization paradigms, such as representing documents on a circle with the aim of maintaining the same distances that the documents have in the feature space. In [5] 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 same idea that a graphical view of the content of a multimedia document can give a much more clear and intuitive information about the content than a list of numbers or a series of text captions or images. In [1] visualization is applied with the main 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 to provide a direct feedback on the program's content. Visualization can be used and has been used for more general purposes too. The tool proposed here is general: the applications of visualization are many. We are testing the performance of the tool as a classifier and annotator of multimedia documents. Another annotation tool is described in [6].

3. THE FUTURE-VIEWER FRAMEWORK

In this section the tool functionalities are described in more detail. The tool semantic 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. The features used are defined by the MPEG-7 standard and describe the color (Dominant color, Color layout, Scalable Color, Color Structure), the texture (Homogeneous Texture, Edge Histogram), the motion (Camera Motion) and the audio spectrum of the shots.

3.1. The GUI and its main visualization functions

In Figure 1 the GUI of the application 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 can be 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 is associated to 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 shots with red hue and little motion near the origin of the axes, meanwhile shots with high motion activity and violet hue will be put in the upper right corner of the cartesian plane. In the cartesian plane shots with similar feature values are drawn near each other, and by associating different feature types to the cartesian plane's axes it's possible to observe how the distances between the shots change. Instead of associating only two features to the plane's axes, it's possible to define a n-dimensional feature space with many features (whose MPEG-7 descriptors can consist of many coefficients). The n-dimensional space can be visualized in the cartesian plane reducing its dimension to two. This dimension scaling is performed by a linear transformation technique called *"Principal Component Analysis"* ([8]). This technique solves the dimension reduction problem 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 ([7]). This is a bar representing the video in a temporal domain: 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 corre-

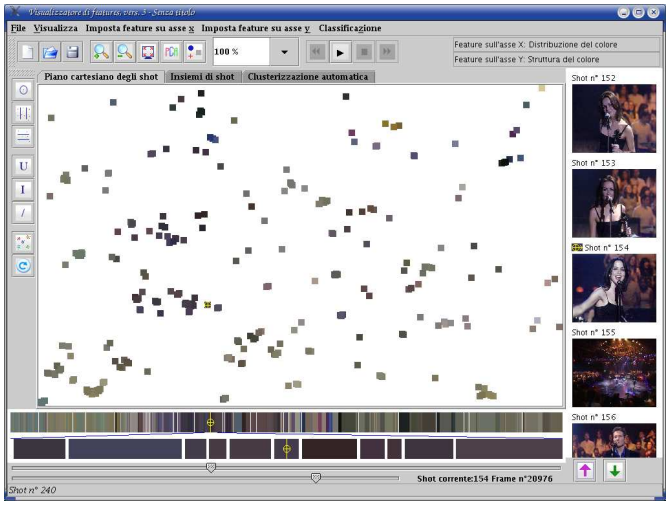

Fig. 1. Screenshot of the proposed application.

sponding shot. In the east region of the GUI the shots' key frames 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 to the user's actions like mouse clicks is designed to support a mixed navigation, watching the same multimedia document from three different points of view. In fact, if the user clicks 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 appears and indicates where is the shot in the other two windows. The main aim of these browsing funcionalities is to increase the accessibility the user has to the document and its semantic structure.

3.2. Clustering and annotation functions

While browsing the shots in the cartesian plane, one consideration is of particular interest: in most cases shots form some clusters in the cartesian plane. The application implements a function to recognize these shot clusters: the user, dragging the mouse, can draw a circle on the cartesian plane and see the shots that fall into it. The tool supports the possibility to save and manage the found similarities as shot sets. 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. The user can see 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 modify efficiently the shot collections. The application can also automatically find the shots' clusters in the feature space. The clustering process begins asking the user to select a feature space in which shots are going to be clustered. This feature space is composed of as many MPEG-7 low-level features as desided. The shots' temporal position too (measured in frames) can be considered as a feature and included in the feature space, in this case shots will be clustered accordingly to their temporal distance too. Shots are clustered in this feature space with an algorithm (taken and modified from [9]) that finds the optimal number of clusters according to two constraints: the first (δ_d) indicates the maximum standard deviation a cluster can have in the feature distances, the second (δ_t) indicates the maximum standard deviation of a cluster's shots in the temporal position. The calculated clusters are visualized mapping their centroids from the n-dimensional space to a 2D plane with the Principal Component Analysis. The clusters are represented as circles filled with the average dominant color of the shots in the cluster, the clusters' radius is proportional to the number of shots included. A clusters visualization is shown in Figure 2. In this figure we can see that the clusters

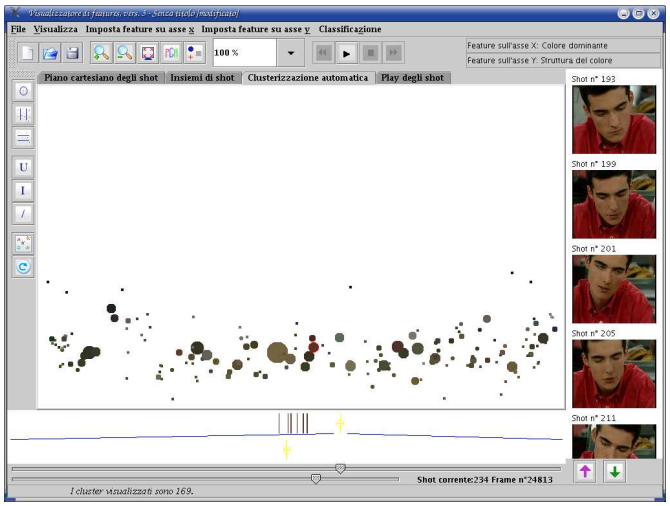

Fig. 2. Clusters visualization with Principal Component Analysis.

are arranged along an horizontal axis, which corresponds to the time axis. This is because the clusters'positions are calculated from their positions in the n-dimensional feature space, that includes the time too. Visualizing clusters with this paradigm intuitively provides the user with important information about the movie's semantic structure: the user can see which are the most important clusters in the movie and which are their relationships in the feature space. Moreover, by clicking on a circle the shots belonging to the corresponding cluster are shown in the cartesian plane, in the colorbar and in the key frames slide; as shown in Figure 2.

4. PERFORMANCE EVALUATION

The proposed tool has been tested with five video documents of 40-50 minutes: a daily TV news program, a music program, a cartoon, a quiz program, a drama series. We annotated these programs by performing automatic clustering of the shots and correct wrong clusters when needed. In order to assure a quick annotation process it's very important a good performance of the automatic clustering process. A clusterization is good if the clusters reflect the semantic and logical structure of the movie. In order to quantify the goodness of the automatic clusterization we segmented by hand the movies in *Logical Story Units* and calculated to what extent the clusterizations are consistent with this segmentation. A Logical Story Unit (LSU) is "a series of contiguous shots that communicate a unified action with a common locale and time" ([10]). LSUs are widely used in literature and there are various method to automatically extract them, in particular an algorithm has been defined in [11] to extract logical story unit from a sequence of shots, once the shots

have been clustered. We use this algorithm to extract the LSU structure from an automatically generated clusterization and compare these LSUs with the ground-truth LSUs. To perform this comparing we used a standard method proposed in [10]. This method consists in calculating two parameters, *coverage* and *overflow*. Coverage reflects to what extent shots in the same ground-truth LSU belong to the same automatically generated LSU. Overflow measures to what extent shots belonging to the same automatically generated LSU are shared between more ground-truth LSUs. So, the optimal clusterization should give coverage = 1 and overflow = 0. Comparing the LSUs automatically generated by FutureViewer and the ground-truth LSUs, we found the results plotted in Figure 3. These results indicate a perfor-

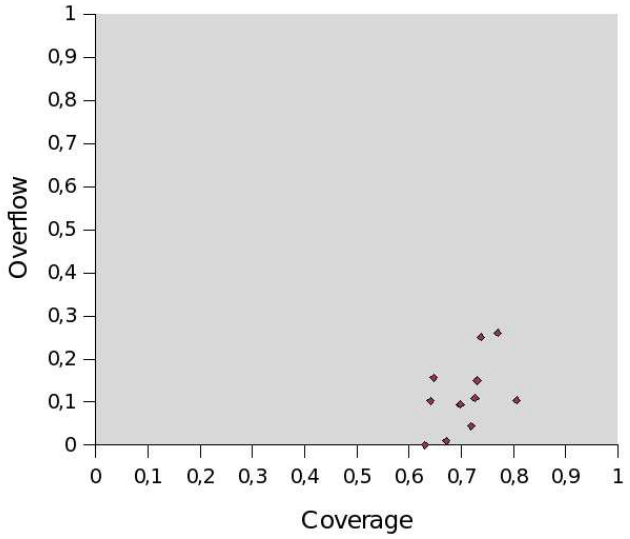

Fig. 3. Automatic clustering results.

mance at the same level of the state of the art. Annotating all the TV programs tested with the aid of automatic clustering is far easier and faster than using a by-hand annotation approach. Tests show that the easiest programs to cluster were the news program and the quiz program, the hardest the cartoon.

5. CONCLUSIONS

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

6. 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 August 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 Conference on Information and Knowledge Management (CIKM 1996), November 12-16 1996, Rockville, Maryland, USA.

[5] M. Carey, D. C. Heesch, S. M. Ruger, "Info Navigator: a Visualization Tool for Document searching and Browsing", Proc. of DMS '2003), September 24-26 2003, Miami, Florida, USA.

[6] Ching-Yung Lin, Belle L. Tseng, John R. Smith, "VideoAnnEx: IBM MPEG7 Annotation Tool for Multimedia Indexing and Concept Learning", Proc. ICME 2003, July 6-9 2003, Baltimore, Maryland, USA.

[7] Mauro Barbieri, Gehrard Mekenkamp, Marco Ceccarelli, Jan Nesvadba, *The color browser: a content driven linear video browsing tool*, proceedings of 2001 IEEE International Conference on Multimedia and Expo, August 22-25, 2001, Tokyo, Japan.

[8] J. Edward Jackson, *A User's Guide to Principal Components*, John Wiley & Sons, Inc. 1991 pp. 1-25.

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

[10] Jeroen Vendrig, Marcel Worring. 2002. *Systematic Evaluation of Logical Story Unit Segmentation*. IEEE Transactions on Multimedia, vol.4, no. 4, December 2002, pp. 492-499.

[11] Minerva M. Yeung, Boon-Lock Yeo, *Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 5, Ottobre 1997, pp. 771-785.