Summarizing Video Information Using Self-Organizing Maps

Thomas Bärecke, Ewa Kijak, Andreas Nürnberger, and Marcin Detyniecki

Abstract— **Facing a huge amount of multimedia information available today, it becomes inevitably necessary to develop efficient methods for accessing, searching, structuring, and representing it. Multimedia retrieval systems especially in the case of video should support users in all of these tasks. Therefore, specialized systems that focus on each of these aspects have been developed. However, an open research perspective issue is the development of a retrieval tool that integrates all user interactions in a single interface. In this paper, we present a system that focuses on the summarization of one single video. We consider the structuring and visualization components including reasonable user interactions to have the most significant influence on the systems usability. Our prototype is based on growing self-organizing maps. The emphasis is on intuitive hierarchical interaction for content visualization exploiting the features of the maps and integrating additional potentially useful information.**

I. Introduction

Extremely large databases with all types of multimedia documents are available today. Efficient methods to manage and access these archives are crucial, for instance quick search for similar documents or effective summarization via visualization of the underlying structure. Therefore, several research areas are involved in the improvement of today´s multimedia retrieval systems. In fact, databases are used to store the raw information. Text, image, sound and video processing techniques are used to extract significant features from the data. Data mining methods can be applied for structuring a collection. User interface design techniques help to provide advanced visualization and interaction options to the user.

The prototype presented in this paper covers to some degree all points mentioned above. However, we focus on the way information of a video is summarized in order to improve the navigation through its content. Our idea is to use unsupervised clustering algorithms in order to automatically group similar shots and to visualize the discovered groups in order to provide an overview over the considered video stream. A shot is a continuous video sequence taken from one single camera. One promising clustering approach that combines good clustering and visualization capabilities is the algorithm of self-organizing maps [1], [2]. In fact, it has been successfully used for the navigation of text [3], [4], [5], [6] and image collections [7], [8]. Furthermore, it is possible to incorporate user feedback in order to adapt the obtained categorization to specific user preferences [9]. The visualization capabilities of self-organizing maps provide an intuitive way of representing the distribution of data as well as the object similarities.

As most clustering algorithms operate on numerical feature vectors, video data should be described by means of a numerical vector. Thus we have to extract describing features from the video stream in order to characterize its content, e.g., by cutting the stream in shots, sequences or individual images and extracting features describing them. Automatic extraction of relevant features from multimedia documents is a wide and challenging research field. A variety of significant characteristics has been defined for text, image, sound, and video data [10]. Video information implicitly contains all types of multimedia information. Consequently, feature extraction algorithms from text, image, and sound processing methods can be applied additionally. However, combining several media is not straightforward and most of the applications rely on features extracted from one single media. From video documents, color histograms for describing the keyframes (images) are widely used [11], [12], [13], [14]. A keyframe is a representative still image taken from a video sequence. We also followed this simple approach, since our goal was to prove the viability of the summarization through visual similarity. We obtained satisfactory results using it as we discuss in the following.

In the following we first give a brief overview of related work in the field of summarizing multimedia contents (especially video) by means of clustering algorithms. Then, we explain the structure of our prototype. The next sections discuss the components. We begin with the preprocessing steps segmentation and feature extraction. Then, we present our structuring approach using growing self-organizing maps. Afterwards, a detailed description of the visualization component is given. We present each of the elements separately. Finally, the last section deals with the interaction possibilities of our system.

II. Related Work

Several of today´s systems apply clustering algorithms in order to summarize multimedia respectively video content. There are systems that perform clustering of shots and visualize the content based on keyframes like in our application. For instance the ShotWeave system [15] groups shots into scenes. It uses information about common continuity editing techniques to define a strict scene definition.

Thomas Bärecke is Ph.D. student at the Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, Paris, France (phone: +33 1 44 27 88 87; email: thomas.baerecke@lip6.fr).

Ewa Kijak is assistant professor specialized in video processing at the Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, Paris, France (email: ewa.kijak@lip6.fr).

Andreas Nürnberger is assistant professor (Juniorprofessor) for information retrieval at the Institute for Knowledge and Language Engineering, Otto-von-Guericke Universität, Magdeburg, Germany (email: nuernb@iws.cs.uni-magdeburg.de).

Marcin Detyniecki is a CNRS research scientist working on artificial intelligence applied to multimedia retrieval at the Laboratoire d'Informatique de Paris 6, France (email: marcin.detyniecki@lip6.fr).

A clustering method incorporating this definition generates the aggregation of the given shots into scenes. In [16] a hierarchical view of video documents is obtained by the means of fuzzy k-means clustering. The number of classes on each hierarchy level is fixed to five. Color histograms in the L*u*v color space are used to describe frames. Another system applying the k-means algorithm is proposed in [17]. It addresses particularly sports video documents and uses motion and color based features. Tensor histograms describe motion features. It has been applied to basketball and soccer videos.

The ADViSOR system [14] contains the Video Indexing Studio and the Video Exploitation System. The Video Indexing Studio includes methods for indexing video information while the Video Exploitation System provides retrieval functionalities. Shot boundaries are detected using the distance of color histograms. A specialty of this system is that it permits multiple keyframes for representing one shot. The extraction of keyframes is based on camera motion. A frame is considered keyframe when the camera motion is minimal. A threshold of camera motion is used to segment the shot into different units having different keyframes.

Apart from that, there are approaches using highlight sequences or skimming for visualization. In [18] a skimming approach is introduced that is based on fast playback and raises the speed of playback to the perceptual limit of the human observer. Different playback rates are used in segments of the video with different spatio-temporal complexity and motion to make sure that the user can perceive important pieces of information. A combination of hierarchical clustering with a visualization method using highlight sequences is discussed in [19]. There is a fixed amount of five layers starting from the raw video sequence, video shots, video groups, and video scenes up to clustered scenes. On each abstraction level a skimming method is applied in order to provide summaries with different detail levels to the user.

III. Prototype

A complete multimedia retrieval system should cover all user interactions implemented in the systems mentioned above. Thus it would typically consist of the following components [20]: feature extraction and indexing, querying, ranking, structuring, visualization, and optionally user modeling. If the system should only be used for structuring and visualization of multimedia data it can be simplified: In this case the querying, ranking and user modeling modules are not necessary. Consequently, our system is composed of a feature extraction, structuring, visualization, and user interaction components (see Fig. 1). It was developed on the basis of the prototypical image retrieval system presented in [6], [21]. In order to deal with video information a new feature extraction component was implemented. The index structure was extended while the structuring method itself, the self-organizing map was not significantly changed, but adjusted to fit the new data structure. The visualization and user interaction components were redesigned with the intention to propose intuitive content-based video browsing functionalities to the user. In the following four sections we will describe every system component and each processing step.

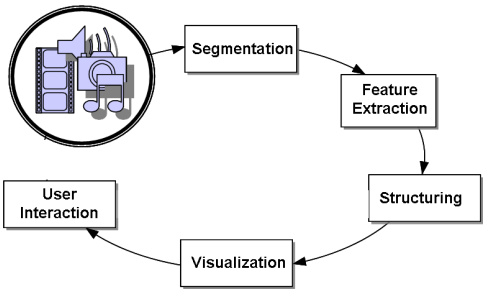

Fig. 1. The components of our prototype. This figure illustrates the data flow from raw multimedia information to visualization and user interaction.

IV. Video Preprocessing

The video preprocessing component supplies the self-organizing map with numerical vectors. Video processing techniques form the basis. The module consists of two parts, temporal segmentation and feature extraction.

A. Temporal Segmentation

For temporal segmentation, a shot boundary detection algorithm is needed. The currently mainly used shot boundary detection approaches have already been compared in several studies [22], [23], [24] (see also [25], [10] for general information). It is important to mention that a study [22] showed that the simpler algorithms outperformed the more complicated algorithms. The best performance with respect to combined accuracy and speed is obtained by histogram-based and compression-based algorithms.

We use a technique consisting of two steps. First, a shot detection algorithm based on color histograms is performed which operates with a single threshold. The colors are represented in the IHS space, because this is closer to the human perception than the RGB space. Another advantage is the independence between the three components: intensity, hue, and saturation. This allows treating them separately in one-dimensional structures instead of using a three-dimensional array representation. The time and memory complexity benefit from that. The histogram is created using a certain number of bins for each color component. The values of nine bins for the intensity, four for the hue, and three for the saturation offered good results. The difference between the histograms of two consecutive images is calculated and then it is compared to a threshold. If it exceeds the threshold a shot boundary is assumed.

The second step of our approach consists of a filtering process eliminating falsely detected shot boundaries from the first step. This can be caused by editing effects or noisy data. If a shot is identified with an insufficient number of frames (a value with good experimental results was 5) it is considered as false positive. Consequently, the shot boundary is deleted and the sequence is added to the next shot. Reasonably good results were obtained using this approach on news video.

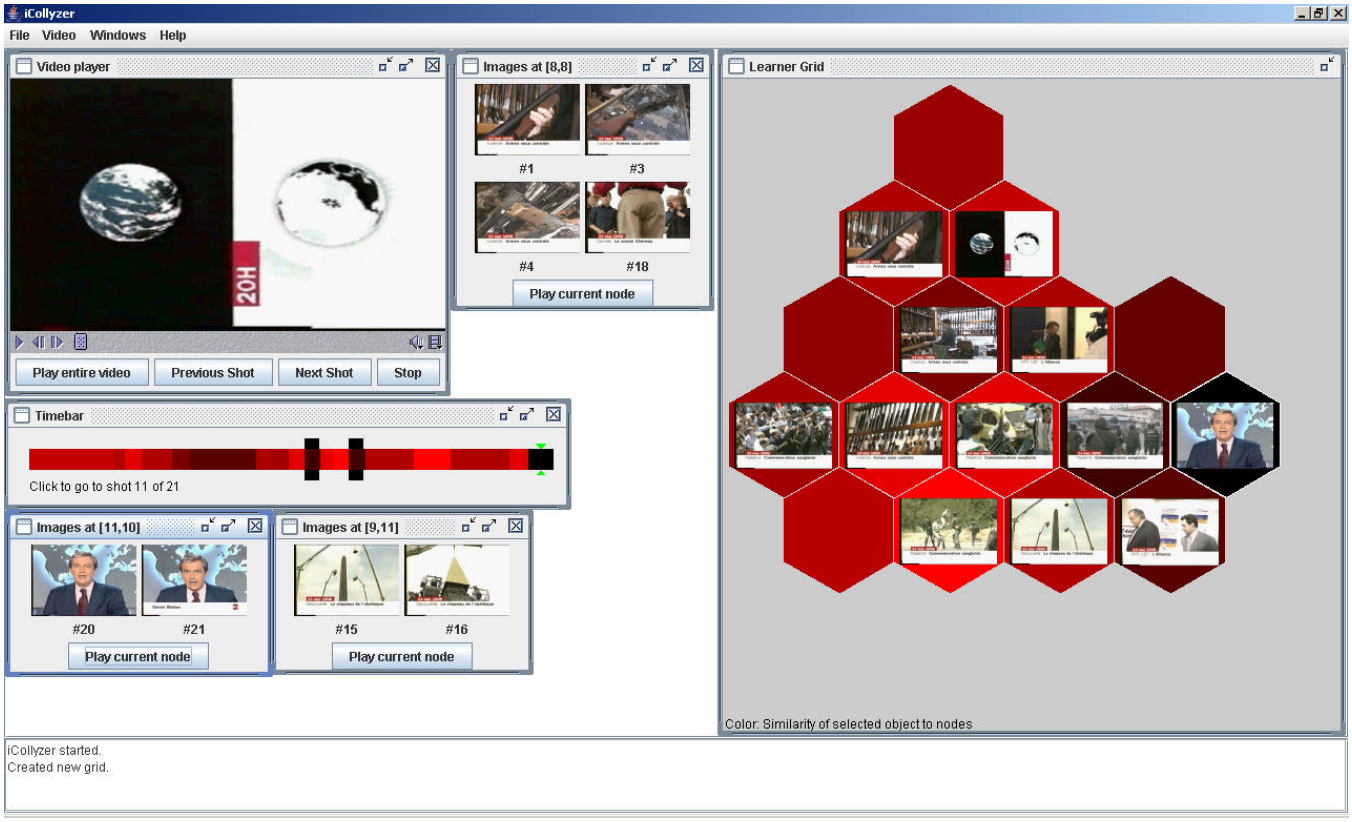

Fig. 2. Screenshot of our interface. The player in the top left corner provides video access on the lowest interaction level. The time bar and shot list provide an intermediate level of summarized information while the growing self-organizing map on the right represents the highest abstraction level.

With properly set values for the number of bins used in the histogram, the threshold and the minimum number of frames, all shot boundaries were successfully detected. In addition the algorithm is very fast, i.e. the most time-consuming part is the construction of the histogram. However, sharp changes in the illumination of a scene still lead to a small number of falsely identified shot boundaries. A good example is the changes caused by the flash of some photographs. The elimination of this effect would imply a much more complex algorithm. Fortunately, this occurs rarely and is not very important for the summarization of the data as considered in this paper.

B. Feature Extraction

A reasonable representation of the video segments is the next step. A video sequence can be divided into (still-) image, audio, and motion information. In our prototype we only use still image features so far. Therefore, one representative keyframe was extracted from each shot. Then we extract color histograms using a specified color space. The system supports the IHS, HSV, and RGB color models. Apart from a global color histogram, histograms for certain regions of the image are also extracted. Four regions are defined, the top, bottom, left, and right rectangles of the image. Every histogram is described by a numerical feature vector. The resulting description for a video sequence is a set of vectors. In order to be able to train a self-organizing map with this set of vectors, we simply concatenate all histogram vectors into a single vector that is then used to define each sequence.

V. Structuring with Self-Organizing Maps

Self-organizing maps [1], [2] are artificial neural networks, well suited for clustering high dimensional information. In fact, they map high-dimensional data into a low dimensional space by progressively grouping similar objects together in one cell. More precisely, objects that are assigned to nodes close to each other, in the low-dimensional space, are also close to each other in the high-dimensional space. This does not mean that objects with a small distance in the high-dimensional space are necessarily assigned to cells separated by a small distance on the map. The map is organized as a grid of cells with constant distances. We choose a two dimensional topology in which the nodes are organized in hexagonal form, because in this case the distances between adjacent nodes are always constant on the map. In a rectangular topology the distance would depend on whether the two nodes are adjacent vertically (or rather horizontally) or diagonally.

Although the application of SOMs is straightforward, a main difficulty is defining an appropriate size for the map. Indeed, the number of clusters has to be defined before starting to train the map with data. Since the objective is

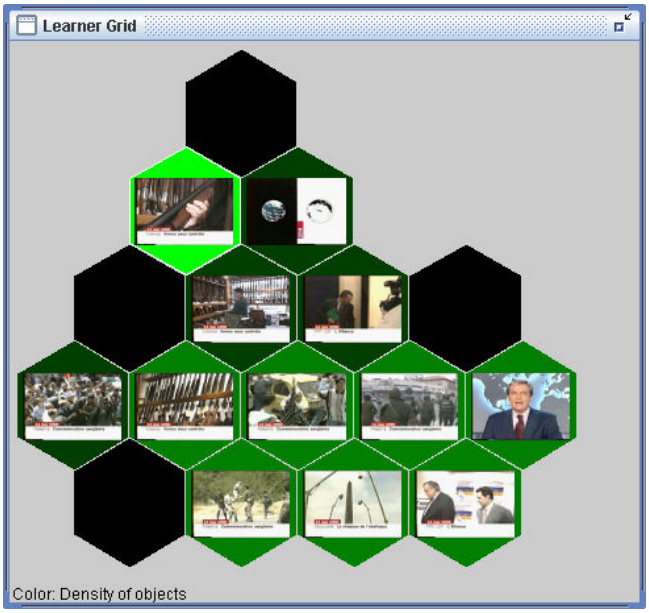

Fig. 3. Growing self-organizing map after training. The brightness of a cell indicates the number of shots assigned to each node (in the interface we use the color green). On each node the keyframe of the shot with the smallest difference to the cluster center is displayed.

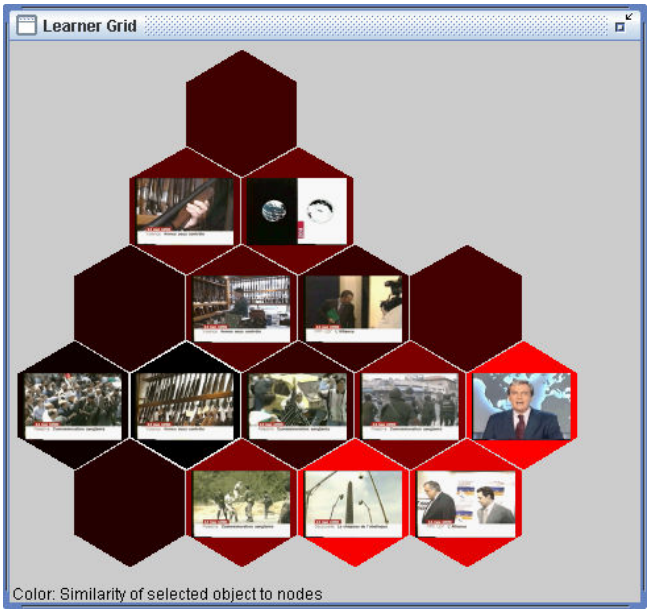

Fig. 4. Growing self-organizing map after a shot has been selected. The brightness of a cell indicates the distance between each cluster center and the keyframe of the chosen shot (in the interface we use the color red). Notice that sequences in adjacent cells are similar as intended.

to structure the video data, the desired size depends highly on the content. An extension of self-organizing maps that overcomes this problem is the growing self-organizing map [6], [9]. The main idea is to initially start with a small map and then add during training iteratively new units to the map, until the overall error - measured, e.g., by the inhomogeneity of objects assigned to a unit - is sufficiently small.

We model the difference of two video sequences by the distance of two vectors of significant features that were extracted from the video. However, this distance does not necessarily correspond to a perceived distance by a human. In fact on the one hand, these features represent only a small part of the video content. On the other hand, selecting relevant features is difficult. In any case, there remains a semantic gap between the video content and what we see on the map being a clustering of abstract objects that our system derived from the original video sequences.

VI. Visualization

The visualization component (see Fig. 2) is the only part of the whole system with which the user interacts. Since the other components of our system are not directly visible to him, it is considered to have a special importance. The problem in visualizing video content is the vast amount of information available. Due to the temporal aspects users need a lot of time to search for specific information by conventional browsing methods. This time can be reduced significantly by summarizing the content. Our system represents a video shot with a single keyframe and constructs higher level aggregates. The user has the possibility to browse the content in several ways. The basic idea is to provide as much information as possible on a single screen. Therefore, we combined elements providing information on three abstraction levels as illustrated in Fig. 2. First, there is an overview of the whole content. The self-organizing map window, described in subsection VI-A, provides this feature. The keyframe of the shot that is the nearest to the cluster center is displayed on the corresponding node. The second level consists of a content-based and a combined time-based visualization. A list of shots is provided for each grid cell (see subsection VI-B). A control element derived from the simple time-bar control (see subsection VI-C) helps to identify content that is similar to the currently selected shot. The main idea is to visualize distance information on the time scale since video is a time-based media. Therefore the bar is shaded with colors corresponding to the similarity. The color schemas follow the ones of the self-organizing-map. They are described, together with the map, in the following subsection.

A. Self organizing map window

The self organizing map (see figures 3 and 4) can be found in the Learner Grid window of our application (see Fig. 2). It contains a visual representation of a two-dimensional self organizing map where the clusters are represented by hexagonal nodes. The keyframe of the shot that is nearest to the center of the cluster is displayed on each node. If there are no shots assigned to a special node no picture is displayed. Due to this fact, empty nodes are easily visible. The background color of the grid cells directly after learning is green (see Fig. 3). The intensity of the color indicates the density of shots in this cell, i.e. the number of them. More precisely, there is a direct relation between the brightness of a cell and the number of shots contained in it. A click on

a cell opens a list of shots assigned to the specific cell (see subsection VI-B).

The user can then select a specific shot from the list. As a result, the color of the map changes to variations of red (see Fig. 4). Here, the intensity of the color depends on the distance between the cluster center and the currently selected shot. For example if the distance is very high the color would be bright red. The color changes towards a darker red if the distance is smaller.

The interaction possibilities within the map are limited to select nodes and to communicate cluster assignment and color information to the time bar. Nevertheless it is a very powerful tool which is especially useful for presenting a structured overview of the video data to the user.

Fig. 5. The video player is used to perform basic navigation operations. For instance playing the entire video, a shot, or shot by shot.

B. Player and Shot List

The player (Fig. 5) is an essential part of every video browsing application. The basic functionalities includes play, stop, fast forward, rewind buttons, and a slider control for determining the current temporal position within the video. Since the video is segmented into shots, two buttons were added especially for the purpose of playing the previous and the next shot. When one of these controls is used, the player searches automatically the first frame of the previous or next shot and starts to play there. When it reaches the last frame of the shot it automatically stops.

A shot list window (Fig. 6) is added to the interface every time a user selects a node from the map. Multiple shot lists for different nodes can be opened at the same time representing each shot by a keyframe. These keyframes correspond to the actual selected node in the self-organizing map which is described in section VI-A. The player also reacts when clicking on one of the keyframes. In that case our system searches the corresponding shot in the video and plays it. The button for playing the current node is a special control which results in a consecutive play operation of all shots corresponding to the selected node, starting with the first shot. After it has reached the last frame of it, it jumps to the position in the video where the first frame of the next shot is located and resumes playing from there and so on. This

Fig. 6. The shot list box displays summaries of the shots assigned to a specific node. Several shot lists windows for different cells can be open at the same time.

adds another temporal visualization method to the segmented video.

Fig. 7. The time bar control provides additional information. The brightness of the color indicates the distribution of similar sequences on the time scale. They correspond to the projection of the colors of the self-organizing map on the time line. Around the time bar, black blocks visualize the temporal positions of the shots assigned to the currently selected node. Finally, the two arrows point out the actual player position.

C. Time bar

A time bar can be found already in many other video browsing applications. The time bar of our prototype (see figure 7) is more specific and provides additional information and some special interaction possibilities. Of course, it displays the current temporal position within the video. This is done by the double green arrow.

Apart from that it uses the same colors as the self organizing map for displaying the main bar. With this approach, it is possible to see within the same view the distance information and the corresponding temporal information. Additionally, there are black extensions on the time bar at the places where the corresponding shots of the selected node can be found.

Two interactions are possible with the time bar. The first is to click once on any position within it: The corresponding shot in the video is played. By clicking twice the self organizing map changes the currently selected node to the one corresponding to the chosen frame. Furthermore, the background color schema of the map is recomputed in order to visualize the new similarity distribution.

VII. User Interaction

As described in the previous sections, a structured view of the data is obtained through the four components of the view. They are integrated into one single screen (see Fig. 2). The methods for user interaction are hierarchically organized. The first (lowest) layer is represented by the video viewer. The

shot lists and the time bar visualize the data on the second layer. The self-organizing map provides the highest abstraction level. The time bar provides additional information to the user. In this section, the interaction possibilities between the components are described, starting from the highest to the lowest layer, in the order a user usually browses video content. Fig. 8 gives an overview.

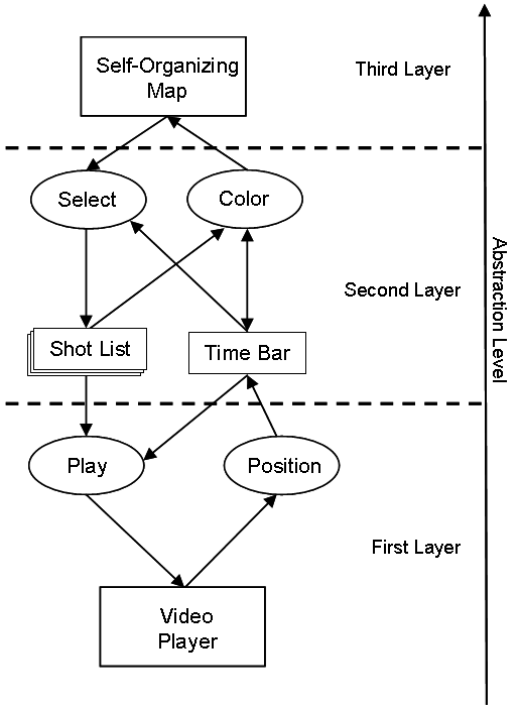

Fig. 8. This figure illustrates the main user interactions possible with our system. All operations can be performed at any time and the visualization components interact with each other. Due to the aggregated shot information it is not reasonable to provide functions for direct interaction between the first and last layer. All listed elements are visible to the user on one single screen thus providing a summarization on all layers at the same time.

The self-organizing map is situated in the third layer. The user can select nodes and retrieve their content i.e. the list of corresponding keyframes. The time bar is automatically updated by visualizing the temporal distribution of the corresponding shots when the current node is changed. Thus, a direct link from the third to the second layer is established. Furthermore the user views at the same time the temporal distribution of similar shots inside the whole video on the time bar, after a certain shot has been selected. In the other direction selecting shots using both the time bar and the list of keyframes causes the map to recompute the similarity values for its nodes and to change the selected node. The color of the grid cells is computed based on the distance of its prototype to the selected shot. The same colors are used inside the time bar. Once the user has found a shot of interest, he can easily browse through similar shots using the color indication on the time bar or map.

The first layer cannot be accessed directly from the third layer. Different play operations are activated by the time bar and shot lists. The player itself gives feedback about its current position to the time bar. The time bar is usually updated when the current shot changes.

All visualization components are highly interconnected. In contrast to other multi-layer interfaces, the user can always use all provided layers simultaneously within the same view. He can select nodes from the map, keyframes from the list or from the time bar, or even nodes from the time bar by double-clicking.

VIII. Conclusions

In this paper we presented an approach to structure and in this way summarize a video using a self-organizing map as core technology. It can be a main component of a multimedia retrieval system. The basic components: feature extraction, indexing, structuring and visualization are covered by the proposed system while query processing and ranking are subject to further development. An independent ranking component is not inevitably required because the structuring by means of the self-organizing map also provides some kind of ranking. It differs from the usual representation in the form of a list of ranked documents with matching probabilities. Most users are already familiar with the list visualization. Due to this fact, this kind of representation as an extra option could help users to easily understand their interaction with the system. The main target was to provide an adaptive tool especially for structuring and visualization of video documents. Therefore, the feature extraction component uses only a small part of currently available feature detection techniques, even though great importance was directed towards the expandability with more features.

The visual feature used in this paper is color histograms derived from each keyframes of the video document. First, a shot boundary detection technique is applied to segment the video into basic units, i.e. camera shots. This method is based on a single threshold for the distances between two consecutive frames. It is enhanced by a second step eliminating falsely detected shot boundaries by defining a minimum length for any shot. This approach with properly set values for the two thresholds enabled an almost perfect automatic shot detection in the case of video news. The only weakness that occurred during the evaluation phase was the sensitivity to suddenly occurring brightness changes in one shot, e.g. caused by flashes of photo cameras. Then, one keyframe per shot is extracted and a color histogram of this frame is computed. The application proposes different color spaces for color histogram representation. Besides a histogram for the complete frame, partial histograms can also be computed for different areas of the frame. In addition, different distance measures can be used for computing histograms distances.

Once, a numerical vector is provided by the feature extraction step, a partitioning of the shots is obtained by a self-organizing map based system which provides already good visualization properties. The visual representation of video content was structured by multiple linked layers. The self-organizing map represents the highest one and gives a summary of the whole video information. A time bar displaying also distance information provides a temporal

overview depending on the selected object while a video player and lists of similar shots are used to represent the information on the lowest level of abstraction.

IX. Future Works

As mentioned before, a point that can be extensively enhanced is the variety of features offered. Features describing sound, motion, visual spatial relationships, objects in a keyframe, etc. can be added to the system. Advances are also possible in the field of feature extraction itself. For example the detection of semantic objects in an image and based on it advanced methods of motion detection still offer a research field. Furthermore, it is possible to incorporate user feedback into the clustering result by applying the algorithm proposed in [9].

Special attention has to be paid to the usability of the system. Novice users could get overwhelmed with a high variety of options. Thus, advanced options should be hidden from them but provided to experienced users. This emphasizes the requirement of an appropriate user modeling component. This component can contain information on the knowledge or experience of the user as well as his interests. In general, the interests of a user can not be discovered directly. His past behavior, i.e. past searches and interactions with the system, gives an indication of what his interests could be. Besides providing a personalized view of the system, a user model can also influence the results of a query, for example based on past queries.

Acknowledgment

This work was partially supported by the German Academic Exchange Service (DAAD) and the German Research Foundation (DFG).

References

[1] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin Heidelberg: Springer-Verlag, 1993.

[2] ——, *Self-Organizing Maps*. Berlin Heidelberg: Springer-Verlag, 1995.

[3] X. Lin, G. Marchionini, and D. Soergel, "A selforganizing semantic map for information retrieval," in *Proc. of the 14th International ACM/SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM Press, 1991, pp. 262–269.

[4] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paattero, and A. Saarela, "Self organization of a massive document collection," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 574–585, May 2000.

[5] D. G. Roussinov and H. Chen, "Information navigation on the web by clustering and summarizing query results," *Information Processing & Management*, vol. 37, no. 6, pp. 789–816, 2001.

[6] A. Nürnberger, "Interactive text retrieval supported by growing self-organizing maps," in *Proc. of the International Workshop on Information Retrieval (IR 2001)*, T. Ojala, Ed. Oulu, Finland: Infotech, Sep 2001, pp. 61–70.

[7] J. Laaksonen, M. Koskela, and E. Oja, "Picsom: Self-organizing maps for content-based image retrieval," in *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN'99)*. Washington, DC: IEEE, July 1999.

[8] A. Nürnberger and A. Klose, "Improving clustering and visualization of multimedia data using interactive user feedback," in *Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, 2002, pp. 993–999.

[9] A. Nürnberger and M. Detyniecki, "Weighted self-organizing maps: Incorporating user feedback," in *ICANN: International Conference on Artificial Neural Networks*, 2003, pp. 883–890.

[10] A. D. Bimbo, *Visual Information Retrieval*. Morgan Kaufmann, 1999.

[11] A. Girgensohn, J. Boreczky, and L. Wilcox, "Keyframe-based user interfaces for digital video," *Computer*, vol. 34, no. 9, pp. 61–67, 2001.

[12] H. Lee, "User interface design for keyframe-based content browsing of digital video," Ph.D. dissertation, School of Computer Applications, Dublin City University, 2001.

[13] H. Lee, A. F. Smeaton, C. Berrut, N. Murphy, S. Marlow, and N. E. O'Connor, "Implementation and analysis of several keyframe-based browsing interfaces to digital video," in *Lecture Notes in Computer Science*, J. Borbinha and T. Baker, Eds., vol. 1923, 2000, pp. 206–218.

[14] A. Miene, T. Hermes, and G. Ioannidis, "Automatic video indexing with the advisor system," in *Proc. International Workshop on Content-Based Multimedia Indexing*, Brescia, Italy, 2001.

[15] J. Zhou and W. Tavanapong, "Shotweave: A shot clustering technique for story browsing for large video databases," in *EDBT: International Conference on Extending Database Technology, LNCS 2490*. Berlin Heidelberg: Springer Verlag, 2002.

[16] D. Zhong, H. Zhang, and S.-F. Chang, "Clustering methods for video browsing and annotation," in *Storage and Retrieval for Image and Video Databases (SPIE)*, 1996, pp. 239–246.

[17] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, "On clustering and retrieval of video shots," in *MULTIMEDIA '01: Proceedings of the 9th ACM International Conference on Multimedia*. New York, NY, USA: ACM Press, 2001, pp. 51–60.

[18] K. A. Peker and A. Divakaran, "Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure," Mitsubishi Electric Research Laboratories, Tech. Rep., 2004.

[19] X. Zhu, J. Fan, and W. Aref, "Medical video mining for efficient database indexing management and access," in *Proceedings of the 19th International Conference on Data Engineering*, Bangalore, India, 2003, pp. 569–580.

[20] K. Bade, E. W. D. Luca, and A. Nürnberger, "Multimedia retrieval: Fundamental techniques and principles of adaptivity," *KI: German Journal on Artificial Intelligence*, vol. 18, no. 4, pp. 5–10, 2004.

[21] A. Nürnberger and M. Detyniecki, "Adaptive multimedia retrieval: From data to user interaction," in *Do smart adaptive systems exist - Best practice for selection and combination of intelligent methods*, J. Strackeljan, K. Leivisk, and B. Gabrys, Eds. Berlin: Springer-Verlag, 2005.

[22] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, Apr. 1996.

[23] P. Browne, A. F. Smeaton, N. Murphy, N. O'Connor, S. Marlow, and C. Berrut, "Evaluating and combining digital video shot boundary detection algorithms," in *Proc. Irish Machine Vision and Image Processing Conference*, Dublin, Ireland, 2000.

[24] A. Dailianas, P. England, and R. B. Allen, "Comparison of automatic video segmentation algorithms," in *Proc. SPIE: Integration Issues in Large Commercial Media Delivery Systems*, A. G. Tescher and V. M. Bove, Eds., vol. 2615, 1996, pp. 2–16.

[25] O. Marques and B. Furht, *Content-based Image and Video Retrieval*. Norwell, Massachusetts: Kluwer Academic Publishers, 2002.