
Organizing Multimedia Information with Maps

Thomas Bärecke¹, Ewa Kijak², and Marcin Detyniecki¹,
and Andreas Nürnberger³

¹ LIP6, Université Pierre et Marie Curie – CNRS, Paris, France,
thomas.baerecke@lip6.fr, marcin.detyniecki@lip6.fr

² IRISA, Université de Rennes 1, Rennes, France ewa.kijak@irisa.fr

³ IWS, Otto-von-Guericke Universität, Magdeburg, Germany
nuernb@iws.cs.uni-magdeburg.de

Summary. Semantic multimedia organization is an open challenge. In this chapter, we present an innovative way of automatically organizing multimedia information to facilitate content-based browsing. It is based on self-organizing maps. The visualization capabilities of the self-organizing map provide an intuitive way of representing the distribution of data as well as the object similarities. The main idea is to visualize similar documents spatially close to each other, while the distance between different documents is bigger. We demonstrate this on the particular case of video information. One key concept is the disregard of the temporal aspect during the clustering. We introduce a novel time bar visualization that reprojects the temporal information. The combination of innovative visualization and interaction methods allows efficient exploration of relevant information in multimedia content.

1 Introduction

A huge and ever increasing amount of digital information is created each day. The capacity of the existing manifold storage devices (for instance hard drives, optical disks, flash memories) increases continuously. Multimedia information in digital formats is, on the one hand, found everywhere in our everyday life, in devices such as portable media players, mobile phones, digital cameras. Thus, we already rely on the assistance of desktop search engines like Google Desktop, Beagle, or Spotlight for finding locally stored data.

On the other hand, the amount of publicly available information and its boost is even more impressive. Apart from classical media, the recent web 2.0 trend [1] of sharing user-created content is a major contributor. The blog scene as well as community websites like Flickr [2], MySpace [3], or YouTube [4] constantly continue to grow both in terms of users and the sheer amount of data. Facing this amazing amount of information, it has become extremely difficult and time consuming to filter and retrieve the relevant pieces.

A big challenge when dealing with multimedia is usually referred to as the Semantic Gap. It arises from the fact that there is a difference between the technical representation and the actual meaning of a given multimedia document. In other words, we cannot index multimedia information like numerical since there is no unique, well-defined semantic for a given document. Ideally, multimedia retrieval should be based on the meaning, but unfortunately, a computer is not able to identify it.

Multimedia retrieval systems [5] that provide satisfying interaction possibilities for all types of multimedia information are not yet available. A particular problem is the ambiguity of visual, audio, and audio-visual information. One question of crucial importance is: How can we efficiently organize personal and public multimedia collections in order to facilitate the user's access? From the user's perspective, in a multimedia retrieval system two tasks are of special interest: the search for a specific information and the exploration of a collection. In this chapter we focus on the latter. We are concerned with presenting the information in a convenient form to the user. We focus on organizing the data into a structured view. The main target is to present a comprehensive summary of a given collection to the user and to provide her with efficient browsing tools.

A major problem is the *curse of dimensionality*. For instance, the dimensionality of a simple text document, using a TF/IDF representation, equals the number of words in the dictionary. The RGB (and most other color spaces) description for digital images use three dimensions per pixel. Video information is even richer.

Organizing the data for convenient exploration has two requirements. On the one hand, the dimensionality has to be reduced in order to obtain a visualization in a human-interpretable space. On the other hand, similar data should be grouped together, reducing the total amount of data represented at once. We show that self-organizing maps can fulfill both. We illustrate this on the particular case of video browsing. We also introduce a new innovative user interaction tool, an enhanced time bar. In this chapter, we do not focus on the feature extraction process, but rather on the content organization and visualization once the features have been extracted.

The remainder of this chapter is organized as follows. In Sect. 2 we give an overview of related work. Then, we introduce the growing self-organizing maps and how they can organize multimedia data. Finally, we illustrate this by focusing on the particular case of video information.

2 Related Work

This chapter faces the challenge of content organization for efficient browsing. A still very common form of content visualization, used by all popular search engines like Google [6], is a simple ranking (by relevance, date, document name). The origin of this representation lies in the retrieval of textual

information using keywords as a query and computing relevance measures of a document for a given query. In the example of Google, the relevance is based both on text similarity, e.g. measured by TF/IDF, and source link-reputation, e.g. measured by PageRank.

However, particularly for large collections, it is more convenient to have similar documents grouped together. The user first browses the group index and then accesses only the documents classified in the group of interest. The main question is: How can we measure the similarity of multimedia documents?

A simple approach that tries to extend text retrieval on other types of data is to index documents manually with keywords. For instance, the Yahoo!Directory and Flickr [2] are based on manually classified documents in hierarchically organized categories. A more recent approach, where this indexing task is performed by the users, are tag clouds with the underlying folksonomy concept. There are several problems with this approach: First of all, a lot of manual work is needed, even if it is distributed. Secondly, the granularity of the keywords is crucial for the performance (e.g. do we assign the keyword “car”, the more general keyword “vehicle” or the more specific keyword “sports car” to a given object?). Finally, not everybody would associate the same keywords to a given document. However, this approach has also become very popular. Researchers try to bypass the granularity problem by creating ontologies. Some scientific work has been dedicated to automatically associate labels with images. The great advantage of keyword-based search is that users are already familiar with it.

In the early days, content-based image retrieval systems were solely based on global low-level features, i.e. color, texture and shape descriptors. Some well-known examples are Virage [7], Photobook [8] from MITs Media Laboratory and IBM’s QBIC [9]. Later region-based systems have been introduced [10–12]. These capture local image properties and hence refine the retrieval process. Current state-of-the-art systems, like SIMPLIcity [13], try also to capture semantic concepts through high-level features. An automatic way to obtain high-level descriptors is to apply machine learning techniques to learn their associations with low-level features. Another popular approach is the use of relevance feedback [14, 15]. In fact, the user is required to evaluate the results of a query. The system refines the search based on these preferences.

Current content based video retrieval systems, like the IBM Video Retrieval System [16] and the MediaMill Systems [17], are also principally based on low-level features. Machine learning methods are then applied in order to learn associate high-level semantics to a set of low-level features. This high level feature extraction is still a major problem, addressed for instance in the TRECVID challenge [18]. Recently, it has been argued that for news video retrieval we need only a few thousand semantic concepts [19]. Thus, it is obvious that even if we are able to describe multimedia content with high level descriptors, the feature space will always remain very high-dimensional.

Manifold dimensional reduction methods are available, projection high-dimensional data into a lower dimensional space. For a survey we refer the reader to [20, 21]. Probably the most frequently used technique is principal component analysis (PCA), also referred to as singular value decomposition (SVD). PCA is a linear method aiming at identifying the directions with highest variance in the feature space. It usually starts with a normalization of each variable to mean zero and standard deviation one. Then, one applies a spectral decomposition on the covariance matrix. The principal components are then given by the eigenvectors with the highest eigenvalues associated to them. These form an orthogonal basis of the low-dimensional space. PCA is optimal in the sense that, when re-projecting the data into the original space, the mean squared error is minimal amongst all possible linear transformations. However, the main disadvantage of PCA and other spectral methods, e.g. multi-dimensional scaling (MDS) which tries to preserve pairwise distances instead of maximizing variance, is the computational complexity arising from the spectral decomposition of a large matrix.

Apart from projecting the data into a feasible space, clustering methods group similar items together and thus refine the structured view. In general, there are two main classes of clustering algorithms: hierarchical and partitional methods. In hierarchical clustering, larger clusters are either successively split into smaller clusters, or smaller clusters are successively merged. This results in a cluster hierarchy, the dendrogram. In order to obtain a given number of clusters, the dendrogram is cut off at the appropriate height. Partitional clustering directly tries to obtain k clusters, where k usually is a parameter. The k-means algorithm falls into this category.

Self-organizing maps (SOMs) [22] simultaneously provides both, a non-linear projection from a high-dimensional space, and a clustering of the data – including prototype vectors for each cluster – at the same time. Therefore, they are very well suited to the data organization task. In contrast to PCA, MDS and independent component analysis (ICA), which are globally tuned and attach more importance to large distances than to small details, self-organizing maps better preserve local neighborhood sets [23]. In fact, global relations may be visualized using coloring schemes as we will demonstrate later.

It has been shown, that SOMs can be effectively used for the organization of text [24–27], image [28–30], and music collections [31, 32]. In the following, we illustrate that they are also able to cover video information.

3 Organizing Information with Semantic Maps

3.1 The Self-Organizing Maps

Self-organizing maps (SOMs) [22] are artificial neural networks, well suited for clustering and visualization of high-dimensional information. In fact, they map high-dimensional data into a low-dimensional space (two-dimensional

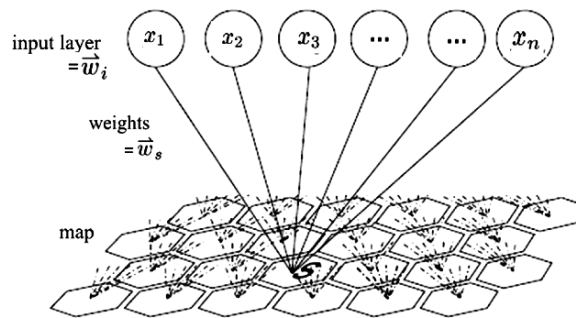


Fig. 1. Structure of a hexagonally organized self-organizing map: the basic structure is an artificial neural network with two layers. Each element of the input layer is connected to every element of the map.

map). The map is organized as a grid of symmetrically connected cells. During learning, similar high-dimensional objects are progressively grouped together into the cells. After training, objects that are assigned to cells close to each other, in the low-dimensional space, are also close to each other in the high-dimensional space. As most clustering algorithms, SOMs operate on numerical feature vectors. Its advantage is, that it is not limited to any special kind of data, since for all kinds of multimedia information well-studied numerical descriptors can be computed.

The neuronal network structure of SOMs is organized in two layers (Fig. 1). The neurons in the input layer correspond to the input dimensions, here the corresponding feature vector. The output layer (map) contains as many neurons as clusters needed. All neurons in the input layer are connected with all neurons in the output layer. The connection weights between input and output layer of the neural network encode positions in the high-dimensional feature space. They are trained in an unsupervised manner. Every unit in the output layer represents a prototype, i.e. here the center of a cluster of similar documents.

In the traditional rectangular topology the distance depends on whether two cells are adjacent vertically (or rather horizontally) or diagonally. Therefore, our maps are based on cells organized in hexagonal form, because the distances between any two adjacent cells are always constant on the map (see Fig. 1).

Before the learning phase of the network, the two-dimensional structure of the output units is fixed and the weights are initialized randomly. During learning, the sample vectors are repeatedly propagated through the network. The weights of the most similar prototype w_s (winner neuron) are modified such that the prototype moves toward the input vector w_i . The Euclidean distance or scalar product is usually used as similarity measure. To preserve the neighborhood relations, prototypes that are close to the winner neuron in the two-dimensional structure are also moved in the same direction. The strength of the modification decreases with the distance from the winner

neuron. Therefore, the weights w_s of the winner neuron are modified according to the following equation:

$$\forall i : w'_s = w_s + v(c, i) \times \delta \times (w_s - w_i), \tag{1}$$

where δ is a learning rate. By this learning procedure, the structure in the high-dimensional sample data is non-linearly projected to the lower-dimensional topology.

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. Therefore, the size of the map is usually too small or too large to map the underlying data appropriately, and the complete learning process has to be repeated several times until an appropriate size is found. Since the objective is to organize multimedia information, the desired size depends highly on the content. An extension of self-organizing maps that overcomes this problem is the growing self-organizing map [27].

3.2 The Growing Self-Organizing Map

The main idea is to initially start with a small map and then add new units iteratively during training, until the overall error – measured, e.g. by the inhomogeneity of objects assigned to a unit – is sufficiently small. Thus the map adapts itself to the structure of the underlying data collection. The applied method restricts the algorithm to add new units to the external units if the accumulated error of a unit exceeds a specified threshold value. This approach simplifies the growing problem (reassignment and internal-topology difficulties) and it was shown in [27] that it copes well with the introduction of data in low and high-dimensional spaces. The way a new unit is inserted is illustrated in Fig. 2. After a new unit has been added to the map, the map is re-trained. Thus, all cluster centers are adjusted and the objects are reassigned to the clusters. This implies that objects may change clusters and can

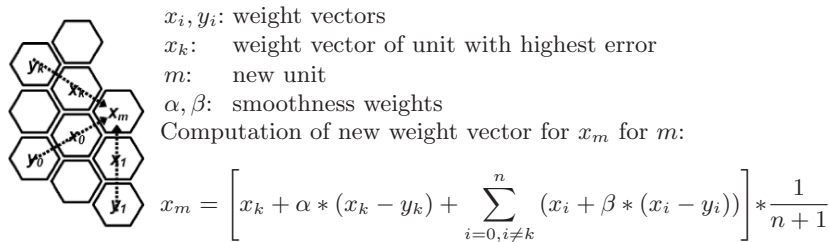


Fig. 2. Insertion of a new unit: when the cumulated error of a cell exceeds a threshold, a new unit x_m is added to the map. It is placed next to the unit with the highest error at the border of the map.

cause the emergence of empty clusters, i.e. clusters which “lost” their former objects to their neighbors. This might happen especially in areas where the object density was already small.

3.3 Visualization

Most of the problems in visualizing multimedia content come from the vast amount of information available. Users need a lot of time to search for specific information by conventional browsing methods. Providing several connected views at different abstraction levels allows a significant time reduction. The basic idea of using self-organizing maps is to provide the user with as much information as possible on a single screen, without overwhelming him. The SOM itself serves as an overview over the entire content. It is a very powerful tool for presenting a structured data summarization to the user.

Indeed, if we deal with visual information, on each of its cells the most typical element of the cluster can be displayed. The user then needs methods to refine his search on a lower level, which is established by the visualization of the content of a cell, on demand.

The background colors of the SOM’s grid cells are used to visualize different information about the clusters. After learning, shades of green indicate the distribution of elements: the brightness of a cell depends on the number of documents assigned to it. Later, the background color indicates the similarity of the cluster to a selected object. For a thorough discussion of coloring methods for self-organizing maps we refer to [33].

When the user selects a specific object, the color of the map changes to shades of red. Here, the intensity of the color depends on the distance between the cluster centers and the currently selected document and thus is an indicator for its similarity. For instance, if we select a document that has the characteristics a and b , all the nodes with these characteristics will be colored in dark red and it will progressively change toward a brighter color based on the distance. This implies in particular that the current node will be automatically colored in dark red, since by construction all of its elements are most similar. In fact, objects that are assigned to cells close to each other, in the low-dimensional space, are also close to each other in the high-dimensional space. However, 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. For instance, we can have on one side of the map a node with documents with the characteristic a and on another the ones with characteristic b . Then in one of both, let’s say a -type, a document with characteristics a , but also b . According to the visualization schema presented above, when choosing a document that has characteristics a and b , located in a node A, we will easily identify nodes in which all the documents are rather of type b . This improves significantly the navigation possibilities provided by other clustering schemes.

4 Example: Organizing Video Data

We present a prototype that implements methods to structure and visualize video content in order to support a user in navigating within a single video. It focuses on the way video information is summarized in order to improve the browsing of its content. Currently, a common approach is to use clustering algorithms in order to automatically group similar shots and then to visualize the discovered groups in order to provide an overview of the considered video stream [34, 35]. The summarization and representation of video sequences is usually based on key frames. They are arranged in the form of a temporal list and hierarchical browsing is then based on the clustered groups. Self-organizing maps [22] are an innovative way of representing the clusters.

Since SOMs necessitate numerical vectors, video content has to be defined by numerical feature vectors that characterize it. A variety of significant characteristics has been defined for all types of multimedia information. From video documents, a plethora of visual, audio, and motion features is available [36, 37]. We rely on basic color histograms and ignore more sophisticated descriptors, since our goal is to investigate the visualization and interaction capabilities of SOMs for video structuring and navigation.

Our system is composed of feature extraction, structuring, visualization, and user interaction components (see Fig. 3). Structuring and visualization parts are based on growing SOMs that were developed in previous works and applied to other forms of interactive retrieval [27, 38]. We believe that *growing* SOMs are particularly adapted to fit video data. The user interface was designed with the intention to provide intuitive content-based video browsing functionalities to the user. In the following, we describe every system component and the required processing steps.

4.1 Video Preprocessing/Feature Extraction

The video feature extraction component supplies the self-organizing map with numerical vectors and therefore it forms the basis of the system. This process

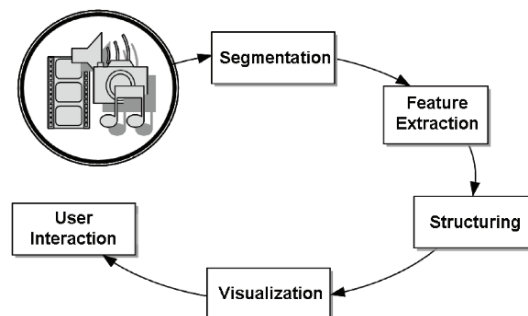


Fig. 3. The components of our prototype. This figure illustrates the data flow from raw multimedia information to visualization and user interaction.

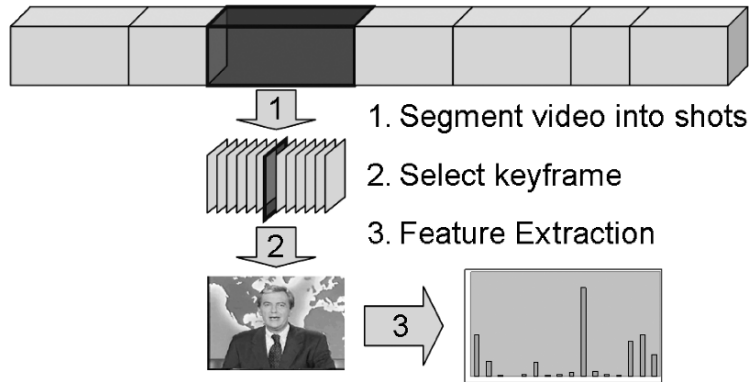


Fig. 4. Video feature extraction.

is shown in Fig. 4. The module consists of two parts, temporal segmentation and feature extraction.

Temporal Segmentation

The video stream is automatically segmented into shots by detecting their boundaries. A shot is a continuous video sequence taken from one single camera. We identify shot boundaries by searching for rapid changes of the difference between color histograms of successive frames, using a single threshold. In fact, transitions from one shot to another are usually associated with significant changes between consecutive frames while consecutive frames within a shot are very similar. Other properties that allow distance estimation between images include texture, and shape features. It was shown in [39] that the approach performs rather well for detecting cuts. We use the (intensity, hue, saturation) IHS color space, because of its suitable perceptual properties and the independence between the three color space components.

Falsely detected shot boundaries can be caused for example by more sophisticated editing effects, such as fades or dissolves, or noisy data. A simple filtering process allows the reduction of the number of false positives, i.e. a set of two successive frames which belong to the same shot although the difference of their color histograms exceeds the given threshold. Our filter deletes shots with an insufficient number of frames (usually less than 5) and adds these sequences to the next actual shot. However, the number of false positives does not have a great influence on our approach, since similar shots will be assigned to the same cluster, as described in the following.

Feature Extraction

In order to obtain a good clustering, a reasonable representation of the video segments is necessary. For each shot, one key frame is extracted (we choose the

median frame of a shot) along with its color histograms. Apart from a global color histogram, histograms for the top, bottom, left, and right regions of the image are also computed. The self-organizing map is trained with a vector merging all partial histogram vectors, which is then used to define each shot.

Similarity Between Shots

As in any clustering algorithm the main problem is how to model the similarity between the objects that are going to be grouped into one cluster. We model the difference of two video sequences by the Euclidean distance of the two vectors that were extracted from the video. However, this distance does not necessarily correspond to a dissimilarity perceived by a human. In addition, these features represent only a small part of the video content. Also, there remains a semantic gap between the video content and what we see on the map.

We are mainly interested in organizing the video data. For this purpose, SOMs assist the user by structuring the content based on visual similarity. However, we can not guarantee that the shots are grouped semantically.

4.2 Visualization

Additionally to the general problem of the vast amount of information available, video information includes a temporal aspect that makes traditional search and browsing even less effective. Our system represents a video shot by a single key frame and constructs higher level aggregates of shots. The user has the possibility to browse the content in several ways. We combine elements providing information on three abstraction levels on a single interface as shown in Fig. 5. First, there is an overview over the whole content provided by the self-organizing map window. On each cell, the most typical key frame of a cluster, is displayed. The second level consists of a combined content-based and time-based visualization. Furthermore, a list of shots is provided for each grid cell and a control derived from the time bar control helps to identify content that is similar to the currently selected shot.

Self-Organizing Map Window

The self-organizing map window (see Fig. 6) contains the visual representation of the SOM. The clusters are represented by hexagonal nodes. The most typical key frame of the cluster, i.e. the key frame which is closest to the cluster center, is displayed on each node. If there are no shots assigned to a specific node no picture is displayed. These empty clusters emerge during the learning phase as described earlier.

After this first display, a click on a cell opens a list of shots assigned to the specific cell (see Sect. 4.2). The user can then select a specific shot from the

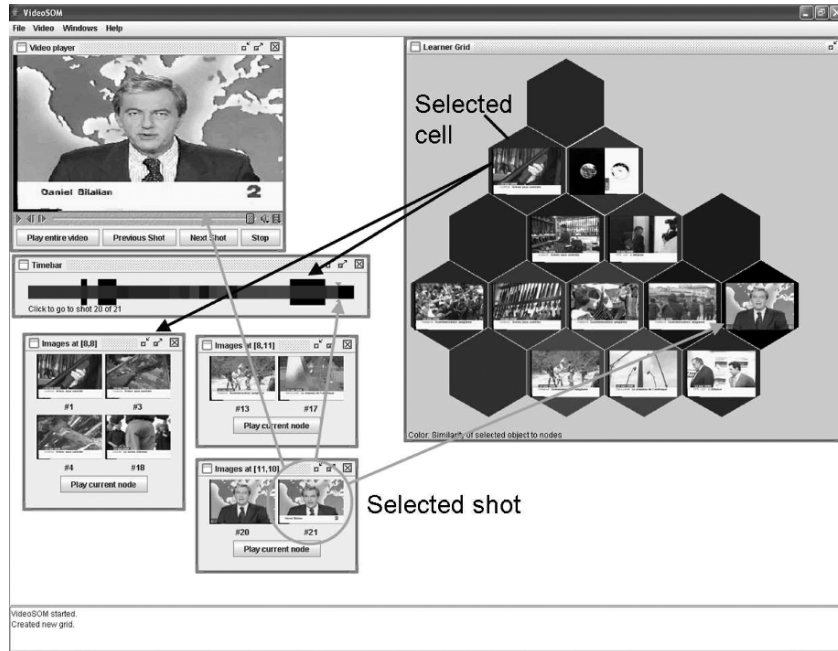


Fig. 5. Screenshot of the interface: the player in the top left corner provides video access on the lowest interaction level. The time bar and shot list provide an intermediate level of summarized information while the growing self-organizing map on the right represents the highest abstraction level. The selected shot is played and its temporal position is indicated on the time bar whose black extensions correspond to the content of the selected cell (*marked with black arrows*).

list. In other words, from user interaction perspective the map is limited to the following actions: select nodes and 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 summarization of the video to the user.

Player and Shot List

The player is an essential part of every video browsing application. Since the video is segmented into shots, functionalities were added especially for the purpose of playing previous and next shots.

A shot list window showing all key frames assigned to a cell (Fig. 5) is added to the interface every time a user selects a node from the map. Multiple shot lists for different nodes can be open at the same time representing each shot by a key frame. These key frames correspond to the actual selected node in the self-organizing map, as described in Sect. 4.2. When clicking on one of the key frames, the system plays the corresponding shot in the video. The

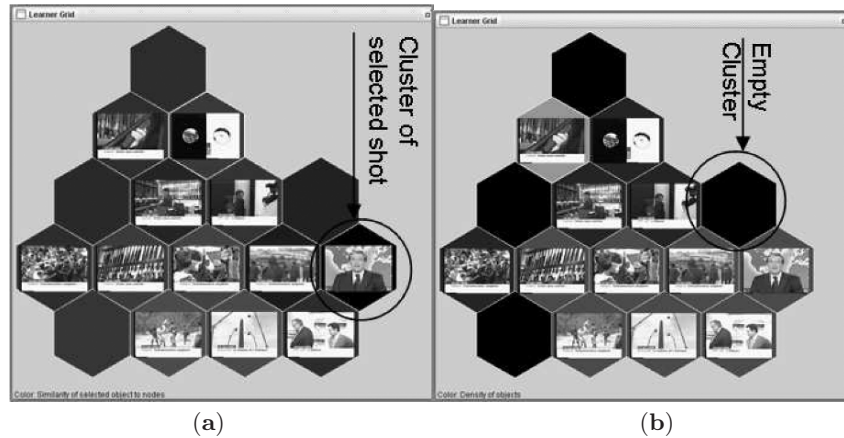


Fig. 6. Growing self-organizing map. **(a)** After training. The brightness of a cell indicates the number of shots assigned to each node. On each node the key frame of the shot with the smallest difference to the cluster center is displayed. **(b)** After a shot has been selected. The brightness of a cell indicates the distance between each cluster center and the key frame of the chosen shot. Notice that sequences in adjacent cells are similar as intended.

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. This adds another temporal visualization method of the segmented video.

Time Bar

The time bar of our prototype (Fig. 7) reintroduces the temporal aspect into the interface, which is ignored by the SOM. The colors of the self-organizing map are projected into the temporal axis. With this approach, it is possible to see within the same view the information about the similarity of key frames and the corresponding temporal information. A green double arrow displays the current temporal position within the video. Additionally, there are black extensions on the time bar at the places where the shots of the selected node can be found. This cell can differ from the cluster of the currently selected shot, in which case the black bars correspond to the selected cluster while the color scheme is based on the selected shot from another cluster. This enables the comparison of a family of similar shots with a cluster.

There are two interaction possibilities with our time bar. By clicking once on any position, the system plays the corresponding shot. Clicking twice forces the self-organizing map to change the currently selected node to the one corresponding to the chosen frame. And therefore, the background color schema of the map is recomputed.

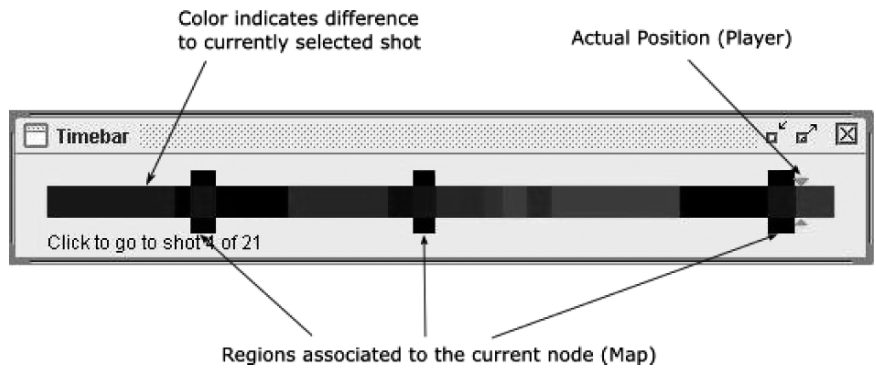


Fig. 7. The time bar control provides additional information. The brightness of the color indicates the distribution of similar sequences on the time scale. Around the time bar, black blocks visualize the temporal positions of the shots assigned to the currently selected node. Finally, the two arrows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

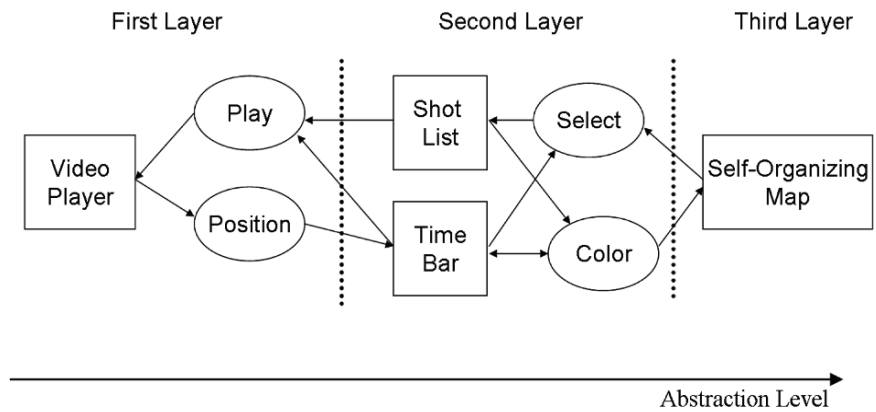


Fig. 8. User interactions. All listed elements are visible to the user on one single screen and always accessible thus providing a summarization on all layers at the same time.

4.3 User Interaction

The four components presented above are integrated into one single screen (Fig. 5) providing a structured view of the video content. The methods for user interaction are hierarchically organized (Fig. 8). The first layer is represented by the video viewer. The shot lists and time bar visualize the data on the second layer. The self-organizing map provides the highest abstraction level.

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 key frames. 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, after a certain shot has been selected, the user also views the temporal distribution of similar shots inside the whole video on the time bar. In the other direction, selecting shots using both the time bar and the list of key frames 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.

Notice that 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 actualized usually 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, key frames from the list or from the time bar, or even nodes from the time bar by double-clicking.

5 Conclusions

The organization of multimedia information is a complex and challenging task. In this chapter, we proposed the use of growing self-organizing maps to assist the user in his browsing and information retrieval task. In the one hand, self-organizing maps efficiently structure the content based on any given similarity measure. In the other hand, although, no perfect (semantic) similarity measure for multimedia documents exist and although this uncertainty remains under any form of visualization, coloring schemes for self-organizing maps allow to easily localize similar documents to a given query example.

We illustrated the efficiency of SOMs with a prototypical content-based video navigation system. Our interface allows the user to interact with the video content from two perspectives: the temporal as well as content-based representations. In fact, ignoring the temporal aspect during clustering enhances the quality of the organization by similarity distribution. The temporal aspects are visually re-linked using similar colors. Three hierarchically connected abstraction levels facilitate the user's navigation.

The combination of innovative visualization and interaction methods allows efficient exploration of relevant information in multimedia content.

References

1. O'Reilly, T.: What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software. <http://www.oreillynet.com/> (last visited April 5, 2007)

2. Flickr. <http://www.flickr.com/> (last visited April 5, 2007)
3. MySpace. <http://www.myspace.com/> (last visited April 5, 2007)
4. YouTube. <http://www.youtube.com/> (last visited April 5, 2007)
5. Bade, K., De Luca, E.W., Nürnberger, A.: Multimedia retrieval: Fundamental techniques and principles of adaptivity. *KI: German Journal on Artificial Intelligence* **18** (2004) 5–10
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Computer Networks* **30** (1998) 107–117
7. Bach, J.R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., Shu, C.F.: Virage image search engine: an open framework for image management. In Sethi, I.K., Jain, R.C., eds.: *Proc. SPIE. Volume 2670* (1996) 76–87.
8. Pentland, A., Picard, R., Sclaroff, S.: Photobook: content-based manipulation of image databases. *International Journal of Computer Vision* **18** (1996) 233–254.
9. Flickner, M., Sawhney, H.S., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. *IEEE Computer* **28** (1995) 23–32
10. Carson, C., Thomas, M., Belongie, S., Hellerstein, J., Malik, J.: Blobworld: A system for region-based image indexing and retrieval. In: *Third International Conference on Visual Information Systems*. Springer, Berlin Heidelberg New York (1999) 509–516
11. Omhover, J.F., Detyniecki, M., Bouchon-Meunier, B.: A region-similarity-based image retrieval system. In Bouchon-Meunier, B., Coletti, G., Yager, R., eds.: *Modern Information Processing: From Theory to Applications*. Elsevier, Amsterdam (2005)
12. Natsev, A., Rastogi, R., Shim, K.: WALRUS: A similarity retrieval algorithm for image databases. *IEEE Transactions on Knowledge and Data Engineering* **16** (2004) 310–316
13. Wang, J., Li, J., Wiederhold, G.: SIMPLiCity: semantics-sensitive integrated matching for picturelibraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23** (2001) 947–963
14. Rui, Y., Huang, T., Mehrotra, S.: Content-based image retrieval with relevance feedback in MARS. In: *Proceedings on International Conference on Image Processing* (1997)
15. Kim, D., Chung, C.: QCluster: relevance feedback using adaptive clustering for content-based image retrieval. In: *Proceedings of ACM SIGMOD International Conference on Management of data, New York, NY, USA, ACM Press* (2003) 599–610
16. Campbell, M., Haubold, A., Ebadollahi, S., Joshi, D., Naphade, M.R., Natsev, A., Seidl, J., Smith, J.R., Scheinberg, K., Tesic, J., Xie, L.: IBM Research TRECVID-2006 video retrieval system. In: *NIST TRECVID-2006 Workshop* (2006)
17. Worring, M., Snoek, C., de Rooij, O., Nguyen, G., Smeulders, A.: The mediamill semantic video search engine. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (2007)
18. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and trecvid. In: *MIR '06: Proceedings of the Eighth ACM International Workshop on Multimedia Information Retrieval, New York, NY, USA, ACM Press* (2006) 321–330

19. Hauptmann, A., Yan, R., Lin, W.H.: How many high-level concepts will fill the semantic gap in news video retrieval? In: Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR (2007)
20. Fodor, I.K.: A survey of dimension reduction techniques. Technical Report, Lawrence Livermore National Laboratory (2002)
21. Burges, C.J.: Geometric methods for feature extraction and dimensional reduction: A guided tour. Technical Report, Microsoft Research (2004)
22. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin Heidelberg New York (1995)
23. Kaski, S.: Data Exploration Using Self-Organizing Maps. PhD thesis, Helsinki University of Technology (1997)
24. Lin, X., Marchionini, G., Soergel, D.: A selforganizing semantic map for information retrieval. In: Proceedings of the 14th International ACM/SIGIR Conference on Research and Development in Information Retrieval, New York, ACM Press (1991) 262–269
25. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paattero, V., Saarela, A.: Self organization of a massive document collection. *IEEE Transactions on Neural Networks* **11** (2000) 574–585
26. Roussinov, D.G., Chen, H.: Information navigation on the web by clustering and summarizing query results. *Information Processing & Management* **37** (2001) 789–816
27. Nürnberger, A., Detyniecki, M.: Visualizing changes in data collections using growing self-organizing maps. In: Proceedings of International Joint Conference on Neural Networks (IJCNN 2002), IEEE (2002) 1912–1917
28. Laaksonen, J., Koskela, M., Oja, E.: PicSOM-self-organizing image retrieval with MPEG-7 content descriptors. *IEEE Transactions on Neural Network* **13** (2002) 841–853
29. Koskela, M., Laaksonen, J.: Semantic annotation of image groups with self-organizing maps. In: Leow, W.K., Lew, M.S., Chua, T.S., Ma, W.Y., Chaisorn, L., Bakker, E.M., eds.: Proceedings of the Fourth International Conference on Image and Video Retrieval (CIVR 2005). Volume 3568 of Lecture Notes in Computer Science, Berlin, Springer-Verlag, Berlin Heidelberg New York (2005) 518–527
30. Nürnberger, A., Klose, A.: Improving clustering and visualization of multimedia data using interactive user feedback. In: Proceedings of the Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (2002) 993–999
31. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. In: MULTIMEDIA '02: Proceedings of the Tenth ACM International Conference on Multimedia, New York, NY, USA, ACM Press (2002) 570–579
32. Knees, P., Schedl, M., Pohle, T., Widmer, G.: An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web. In: ACM Multimedia, Santa Barbara, CA, USA (2006)
33. Vesanto, J.: SOM-based data visualization methods. *Intelligent-Data-Analysis* **3** (1999) 111–26
34. Lee, H., Smeaton, A.F., Berrut, C., Murphy, N., Marlow, S., O'Connor, N.E.: Implementation and analysis of several keyframe-based browsing interfaces to digital video. In: Borbinha, J., Baker, T., eds.: LNCS. Volume 1923 (2000) 206–218

35. Girgensohn, A., Boreczky, J., Wilcox, L.: Keyframe-based user interfaces for digital video. *Computer* **34** (2001) 61–67
36. Marques, O., Furht, B.: *Content-Based Image and Video Retrieval*. Kluwer, Norwell, MA (2002)
37. Veltkamp, R.C., Burkhardt, H., Kriegel, H.P.: *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer, Norwell, MA (2001)
38. Nürnberger, A., Detyniecki, M.: Adaptive multimedia retrieval: From data to user interaction. In: Strackeljan, J., Leiviskä, K., Gabrys, B., eds.: *Do Smart Adaptive Systems Exist – Best Practice for Selection and Combination of Intelligent Methods*. Springer-Verlag, Berlin Heidelberg New York (2005)
39. Browne, P., Smeaton, A.F., Murphy, N., O'Connor, N., Marlow, S., Berrut, C.: Evaluating and combining digital video shot boundary detection algorithms. In: *Proceedings of Irish Machine Vision and Image Processing Conference*, Dublin (2000)