LIVE@WEB.COM – Using CBIR Technology in Interactive Web-TV

Felix Morsdorf and Stephan Volmer

Fraunhofer IGD, Rundeturmstr. 6, Darmstadt, Germany
{felix.morsdorf,stephan.volmer}@igd.fhg.de

Abstract. The increasing amount of internet based television broadcasts has lead to new approachs to interactivity in TV programs. We developed a system which is able to supply the viewer of the program upon interaction with information relating to the program, only based on the low-level visual content of the scene. This aim is achieved by comparing signatures describing the visual content of single frames of the video with a remote database of signatures derived from known videos. The database actually links the visual information contained in the signatures to some second-level information interesting for the user. Two main problems in extending CBIR technology to videos must be overcome, one is the extraction of the visual information out of the highly redundant video material, and the other is reducing the matching time of the system enough to allow for web-based interactivity.

1 Introduction

The steady improvements in performance of the Internet infrastructure and related technologies (e.g. data compression) now give the opportunity to broadcast films, video, live events (such as sports events, shows and conferences) and even TV programs over the Web with rapidly improving quality. Interactive TV over the Web is gradually becoming a reality and advertising over this new media is becoming increasingly attractive. Unfortunately, current approaches to Interactive TV lead to very high production costs, due to the need to create and maintain explicit links for each TV program with the content making the program itself *interactive*. LIVE@WEB.COM is a system that provides interactivity with the user based on *Content-based Image Retrieval* (CBIR) methods. The system is able to supply the user with additional information regarding the program he just watched (e.g. a URL of a product's webpage), only based on the visual content of the scene. In its first prototype version, the system is restricted on the recognition of commercials, as these are small videos and the interest in enhancing those through interactivity is higher than for normal TV-broadcast.

2 Previous Work

In [3], Kreyß et al. have proposed a system for video retrieval, using shot detection to segment the video material, and then constructing so called *mosaic* images containing all the visual information of the shot. On these static mosaiced images they applied traditional CBIR technology in order to retrieve the shots from a database. This approach is not applicable to the LIVE@WEB.COM scenario, as the mosaicing of the shot would have to be done on the client's computer. As this is a time consuming process, it would not allow for realtime interaction. In this paper a different technique is proposed, which is the use of cluster analysis and key frames, which only relies on the extraction of signatures from *single* frames. Other existing video retrieval systems have been established mainly for archiving and browsing in large video databases, but not for an interactive environment over the web, as the LIVE@WEB.COM scenario proposes.

3 System Architecture

The following three modules compose the LIVE@WEB.COM system:

Video Player This module includes the functionality needed to receive TV programs on the Web, accessible by the users via a standard Internet browser, for instance, as a plugin. It also enables the user to interact with the TV program to select frames in the video and to store them for later replay or further examination. The video player is shown in Fig. 1, with the bookmarks set by the user as thumbnails on the right.

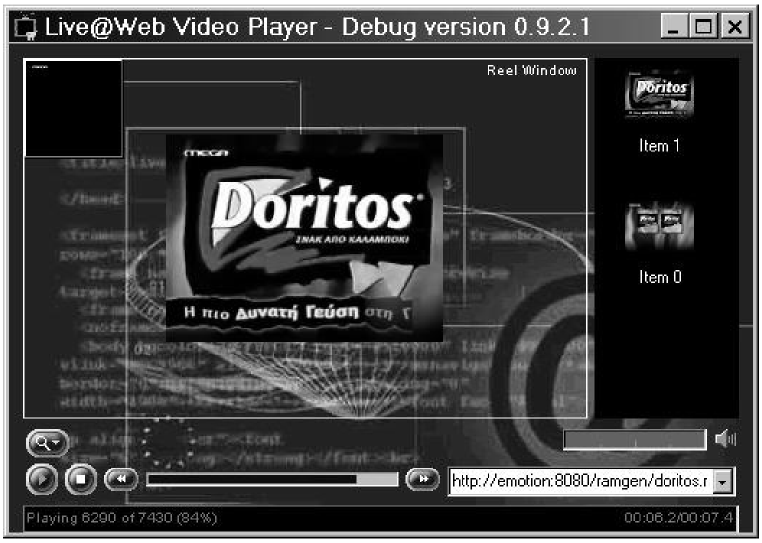

Live@Web Video Player - Debug version 0.9.2.1
Reel Window
Doritos
ΣΝΑΚ ΑΠΟ ΚΑΛΑΜΠΟΚΙ
Η πιο Δυνατή Γεύση στη Γ
Item 1
Item 0
http://emotion:8080/ramgen/doritos.r
Playing 6290 of 7430 (84%)
00:06.2/00:07.4

Fig. 1. The Video Player

Matching Engine This module leverages mainly upon image analysis technologies that extract significant features from the videos automatically. The main characteristic of this technology is its capability to represent those key features in an

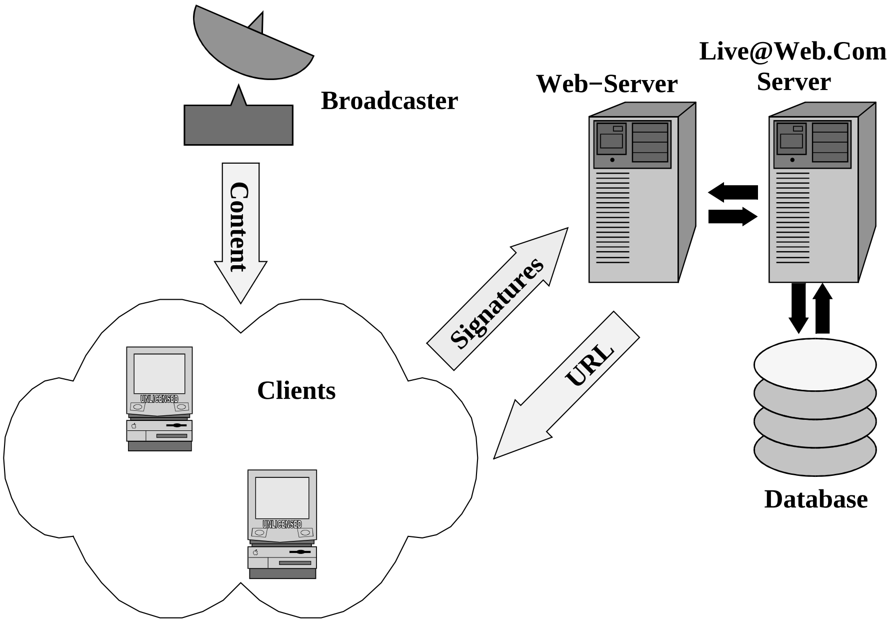


Fig. 2. LIVE@WEB.COM System Architecture

extremely compact signature. The Matching Engine also includes the algorithms which are needed to compare signatures in terms of their similarity concerning those visual features. That way, the video database can be searched by matching the signature of a query image with the signatures of the videos in the database.

Video Database The videos database contains all the videos that are to be linked via the LIVE@WEB.COM system with all the relevant information and links to relevant URLs. Together with each video, there are a number of different signatures stored in the database. Those signatures represent the visual features of the videos and make them distinguishable among all other videos in the database. The videos database comes along with a web-based administration module.

In Fig. 3 a sketch representing the LIVE@WEB.COM server architecture is shown. The Broadcaster provides the client's computers with the streaming video, and is completely independent from the rest of the system. The LIVE@WEB.COM Video Player is installed on the client's computer and provides the interactivity by extracting signatures on user interaction from the video stream and sending those to the LIVE@WEB.COM Server for comparison with the video database and linking of the information. The Video Player parses the information returning from the LIVE@WEB.COM Server, and opens, for instance, a web browser on the client's computer showing a relevant product information page.

4 Innovation

It will be shown how to extend the functionality of existing CBIR technology to the recognition of streaming video at a time that allows for user interaction in an interactive television environment. This is a new approach to interactive TV, since the information that makes the program interactive does not have to be coded manually in the broadcasting stream. Simply applying CBIR technology to the data rich video material (ca. 1-10 MB/s depending on size and colordepth) would surely result in a large computational effort. Thus, methods have to be implemented to minimize the computational effort for this task. These methods can be classified in two fields:

- The visual information in the video material is highly redundant, and thus has to be reduced by some sophisticated means, as for instance by the extraction of key frames, which contain the dominant visual information of a scene.
- After the visual information is extracted, it has to be indexed in some way for storage and retrieval, as a linear search through a large database of the feature signatures would be too time consuming. This indexing makes the comparison time independent of the database size, which is crucial for the large databases of feature signatures we have for videos.

Using both of these methods allows for the implementation of a system that recognises videos in less than 1 second based on a first prototype database containing more than 100 videos. At this speed the system is interactive for an internet based environment.

5 The CBIR Methodology

A variety of visual features can be used for CBIR, based on previous work from [7] two different feature signatures are used in the LIVE@WEB.COM first prototype. One is a colour histogram, which resembles the global colour information, the second is a wavelet fingerprint, which contains the spatial information of the image.

As we need small signature sizes for storing and computation of similarities, the feature signatures for a frame of arbitrary size are restricted to a size less than 512 bytes. The largest advantage of using only small sized signatures for comparison is the ability of having this data transferred over the internet at very short times, even for low bandwidths. The comparison of the signatures of two different frames yields a score, telling how *similar* the particular frames are in respect to this specific visual feature. A detailed description on how the signatures are extracted from the frames, and how their similarity is computed, can be found in detail in [7] and more general in [5],[6].

6 Reducing the Video to Images

On interaction from the user, the LIVE@WEB.COM Video Player extracts feature signatures of n single frames, which were held in a special buffer. These frames need not be consecutive frames, as using a sample interval of approx. 1 second significantly increases the visual information contained in frames, as opposed to using 5 consecutive frames at a sample rate of $1/25$ of a second. The number and sampling of the frames is empirically selected. Thus, the problem of matching videos is reduced to matching $x \times n$ single signatures of the n frames to a database of signatures from the frames of the videos, where x is the number of used feature signatures. In order to keep the computational effort on the client's computer as small as possible (we have no specific information about the hardware on the client's side), no sophisticated method of selecting key frames and/or motion based features is used. The effort in reducing the large amount of data from the videos is solely accomplished on the LIVE@WEB.COM server side, by the means described in the following sections.

6.1 Shot Detection

The approach of recognising video sequences by using CBIR methods on single frames has one big shortcome which must be overcome: the large amount of data that has to be matched if all frames are stored in the reference database. As we pointed out before, the video material is highly redundant, and thus means have to be implemented in order to reduce the large amount of data. The first step is the partitioning of the video in scenes of similar visual content, as for instance via shot detection. This shot-detection is mostly accomplished by tracking the change of one global visual feature. As a comparison of two signatures generates a score determining how close two frames are to each other in the particular feature space, we can track the size of this score (a scalar) in order to detect scene changes. A threshold is used to detect changes in this property, which are candidates for a shot boundary. S_i denotes the signature of the ith frame, and δ is the function for calculating the similarity of two signatures as described in [7]. The similarity value SD_i is zero for identical values and increases with dissimilarity.

$$SD_i = \delta(S_i, S_{i+1}) \text{ with } SD_i \in \mathbb{R}_0^+ \tag{1}$$

A new shot is detected if the difference SD_i is larger than a given threshold. The selection of this threshold is the crucial point in shot detection and has to be optimized empirically. In Fig. 3 an example is given of how these similarity values look for one commercial. In the upper panel, the Color Histogram signature is shown, in the lower the Wavelet Signature. Large peaks indicate abrupt shot changes (*hard cuts*), the numbers mark shot change candidates determined by a threshold with twice the size of the standard deviation. As this method does not take into account any *salient* change in the visual content during the shots and fails to detect shot boundaries if fade effects and transitions are used, we propose the use of unsupervised clustering for key frame extraction, as presented by [10].

6.2 Key Frame Extraction

After the partitioning, so called key frames are extracted from the shot, which are taken as a prototype for the visual content of the scene. Normally one would use for instance the first and the last frame of the scene. This method has the disadvantage of not taking

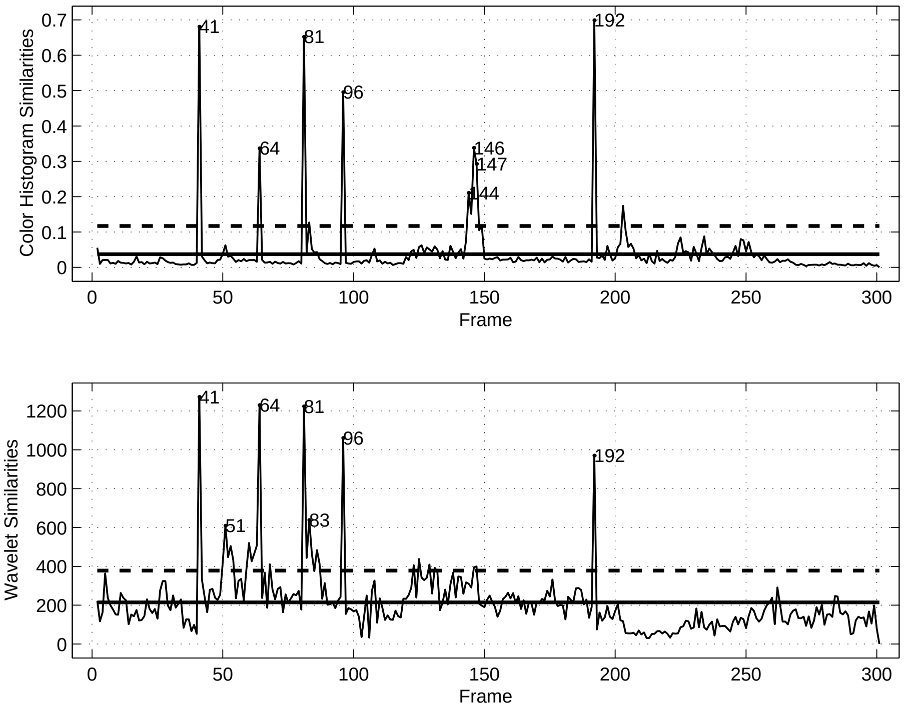


Fig. 3. Thresholding of similarities of consecutive frames for the Color-Histogram (top) and for the Wavelet Signature (bottom). The mean is indicated by the thick solid line, the standard deviation of this mean by the thick dashed line.

into account any changes of the visual content *during* the scene. Furthermore, the first and the last frames of a shot are mostly not stable, and thus not a good representative for the shot. Opposed to key frame extraction for archiving or browsing, where key frames should represent some important *semantic* aspect of the scene [4], the extraction of key frames for content based video retrieval needs to be tuned towards the capturing of the most important low-level visual content, as described by the feature signatures used for the retrieval. This means that the key frames need to cover the area in feature space spanned by *all* frames of teh video as well as possible. Thus, we propose the use of an adaptive key frame extraction using unsupervised clustering, as presented in [10]. In this approach, as many key frames from each shot are selected as needed to describe its visual complexity, for shots with no or a small amount of action this would be only one key frame, but for shots with a lot of action and/or global motion resulting in optical flow, several key frames are selected.

7 Indexing the Database

In [8] a method is proposed of efficiently indexing the database containing the signatures of the key frames, based on the visual features that the signatures represent, using a *k-Medians* clustering algorithm. The indexing yields a matching time of a query frame with the database, which is not dependent on the total database size. The matching time depends only on the size and number of the clusters, that are being used for the signature comparison. This enables the use of this system for arbitrarily large databases of signatures. For very large databases, there will be a trade off between comparison time and retrieval accuracy. This will have to be carefully evaluated as the database population of our first prototype grows.

8 Rejection of Unknown Videos

The concept of similarity yields only the return of the *closest* match on comparing a query frame with the database. Thus, if we want to avoid misleading answers from the LIVE@WEB.COM Server on presentation of an *unknown* video, we need to define a criterion to differ known videos from unknown ones. One way would be the use of the information on *how* close the match was based on the similarity computation. Technically, this can be achieved by using a hardcoded threshold on the similarity values returned from the comparison; if the value is below that certain threshold, the queried frame is match, otherwise it is declared as not known. Unfortunately, the choice of a threshold is crucial and has to be optimized in two opposing directions, as can be seen from Fig. 4.

The recognition reliability for unknown videos (*recognition = classified as unknown*) decreases with increasing threshold size, whereas the recognition reliability for known videos increases. Thus, a compromise in setting the threshold has to be made. If both cases (known video and unknown video) had the same probability, one would choose the intersection of the two curves. But for our scenario, we would in the best case know only 20% of the TV program, the commercials, and hence the threshold has to be pushed to the left to acommodate for the four times higher probability of unknown videos. The selected threshold is indicated by the dashed vertical line. In the first prototype with its relatively small database, this is not a big issue, but as the population of the database will grow, methods have to be implemenented to overcome this shortcome. As we use not one, but n frames for querying, we propose the use of some heuristics, which takes the consecutive order of the frames into account.

9 Results and Conclusion

The first prototype of the system is capable of recognising known videos from a small database of 100 videos in less than a second, and is able to supply the user with the relevant information. The layout of the video player frontend was shown in Fig. 1. During the broadcast, the user is able to click several times in order to set *bookmarks* (thumbnails on the right), and after the broadcast ended (or whenever he desires), he may send the signatures to the LIVE@WEB.COM Server. On correct identification of the query

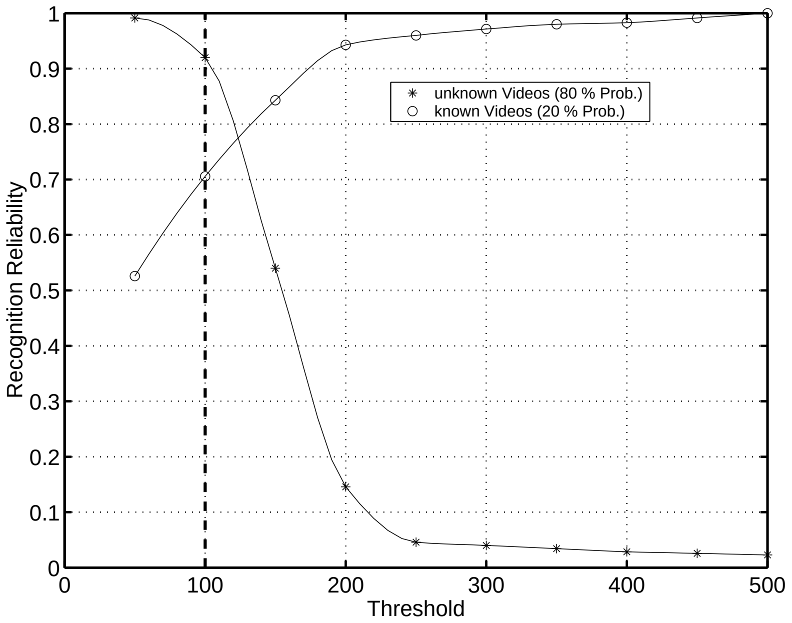


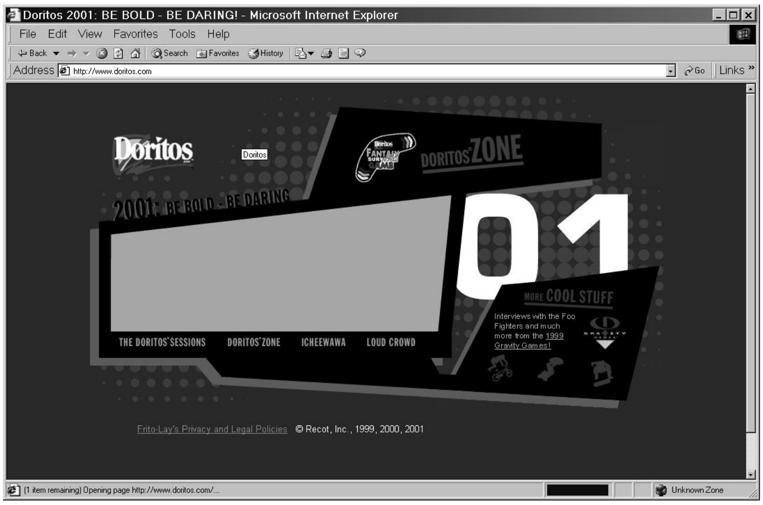
Fig. 4. Recognition Reliability in Respect to Threshold Value

frames, a response containing a URL is sent back to the client's machine, where the LIVE@WEB.COM Video Player opens a web browser window with this specific URL (Fig. 5).

As the database population grows, the above cited methods are becoming more and more important and some fine tuning will be important in order to achieve the projected aims of having a database with more than 5000 videos. We have shown that it is possible to extend CBIR technology to video material without defining special video features, that would for instance include motion based algorithms. The key task in setting up such a system is managing the large amount of data in the video, and this can be achieved by the methods we proposed. First, key frame extraction in the feature domain via clustering guarantees an optimal compromise in reducing the data *and* keeping the relevant visual information. Second, an indexing of the feature signature database allows for fast comparison times applicable for interactive enviroments.

Acknowledgements

This work was supported by the European Commission with the IST Project #12277 LIVE@WEB.COM .



Interviews with the Foo Fighters and much more from the 1999 Gravity Games!

© Recot, Inc., 1999, 2000, 2001

(1 item remaining) Opening page http://www.doritos.com/...

Fig. 5. Web Browser with Product Homepage

References

1. Aigrain, P., Zhang, H.J.,Petkovic, D.: Content-based Representation and Retrieval of Visual Media: A State-of-the-Art Review. *Multimedia Tools and Applications*, 3(3):179–202, July 1996.
2. Herzog, O. , Miene, A. , Hermes, T. , Alshuth, P.: Integrated Information Mining for Texts, Images, and Videos. *Computers & Graphics*, 22(6):675–685, December 1998.
3. Kreyß, J. , Röper, M. , Alshuth, P. , Hermes, Th. , Herzog, O.: Video Retrieval by Still Image Analysis with ImageMiner. In *Storage and Retrieval for Image and Video Databases V*, vol. 3022, pp. 36–44. SPIE, San Jose, USA, February 1997.
4. Macer, P., Thomas, P.: Browsing Video Content. *Proceedings of BCS HCI Group Conference, The Active Web*, Staffordshire University, UK., January 1999
5. Pass, G., Zabih, R.: Comparing Images Using Joint Histograms. *ACM Journal of Multimedia Systems*, 7(3):234–240, May 1999.
6. Swain, M.J., Ballard, D.H.: Colour Indexing. *Int. Journal of Computer Vision*, 7(1): 11-32,1991
7. Volmer, S.: Tracing Images in Large Databases by Comparison of Wavelet Fingerprints. In *Proc. of the 2nd Int'l Conf. on Visual Information Systems*, pp. 163–172, La Jolla, USA, December 1997.
8. Volmer, S.: Buoy Indexing of Metric Feature Spaces for Fast Approximate Image Queries. To appear in *Proc. of the 6th Eurographics Workshop on Multimedia*, Manchester, UK, September 2001.
9. Zhang, H.J., Kankanhalli, A., Smoliar, S.W.: Automatic Partitioning of Full Motion Video. *Multimedia Systems*, 1(1):10–28, June 1993.
10. Zhuang, Y., Rui, Y., Huang, T., Mehrotra, S.: Adaptive Key Frame Extraction Using Unsupervised Clustering. In *Proc. of IEEE Int'l Conf. on Image Processing*, pp. 866–870, Chicago, USA, October 1998.