The Físchlár Digital Video Recording, Analysis and Browsing System

Hyowon Lee, Alan F Smeaton, Colin O'Toole
School of Computer Applications
Dublin City University
Glasnevin, Dublin 9
Ireland
{hlee, asmeaton, cotoole}@compapp.dcu.ie

Noel Murphy, Seán Marlow, Noel E. O'Connor
School of Electronic Engineering
Dublin City University
Glasnevin, Dublin 9
Ireland
{murphyn, marlows, oconnorn}@eeng.dcu.ie

Abstract

In digital video indexing research area an important technique is called shot boundary detection which automatically segments long video material into camera shots using content-based analysis of video. We have been working on developing various shot boundary detection and representative frame selection techniques to automatically index encoded video stream and provide the end users with video browsing/navigation feature. In this paper we describe a demonstrator digital video system that allows the user to record a TV broadcast programme to MPEG-1 file format and to easily browse and playback the file content online. The system incorporates the shot boundary detection and representative frame selection techniques we have developed and has become a full-featured digital video system that not only demonstrates any further techniques we will develop, but also obtains users' video browsing behaviour. At the moment the system has a real-user base of about a hundred people and we are closely monitoring how they use the video browsing/navigation feature which the system provides.

1 Introduction

The potential scope for applications which operate on digital video information is enormous and ranges from domestic consumer through to the workplace. Already we see DVD movies and intranet streaming in the workplace are common with the deployment of broadband telecommunications to the home set to broaden these applications. Having already got a wide user base in TV and VCR, the rate of uptake of these new applications should be quite fast.

One of the core technologies being investigated in the area of video indexing/browsing is shot boundary detection in which a video stream is automatically segmented into smaller, manageable units by analysing data from the video content such as colour, shape and motion. In the video indexing and browsing work at Dublin City University, we have been doing extensive research in developing various shot boundary detection techniques for accurate video segmentation along with the ways in which to evaluate their efficiency and this has led to a reasonably comprehensive evaluation reported elsewhere (O'Toole *et al.*, 1999). To demonstrate this work and to start work on actually browsing digital video content, we have developed a demonstrator system which we call Físchlár. This system is a full-featured, self-contained system and incorporates recording, shot boundary detection, representative frame selection, video browsing and playback components. It is also easily extensible to contain further techniques developed during the course of our future work.

This paper describes the Físchlár system, with the following section presenting an overview and functional description of the system – what it looks like, how a user interacts with it, and how it demonstrates our shot boundary detection and other techniques we developed to allow content-based browsing of video. Section 3 will briefly describe the technical aspects of the system – how it works, what resources are used, the file formats used for storing video data, video streaming, and so on. In section 4 we mention some of the future possibilities and improvements we are developing for the system and section 5 will conclude the paper.

2 The Físchlár Digital Video System

2.1 Overview

Our work is an on-going research effort to develop video indexing tools to allow efficient and effective video content browsing mechanism for users. Our main focus at present is to develop accurate shot and scene boundary detection and keyframe extraction techniques to index video content and use this output for providing a content-based video browsing/navigation user-interface for broadcast TV video. Commencing in November 1997, our work so far has developed five different methods of automatically detecting shot boundaries in video and we have compared their efficiency against an eight hour long *testbed video data suite* which we have encoded and manually logged the different types of camera cuts and scene changes. The Físchlár Digital Video browsing system demonstrates this work and gives us a vehicle for experimenting with video browsing interfaces and paradigms.
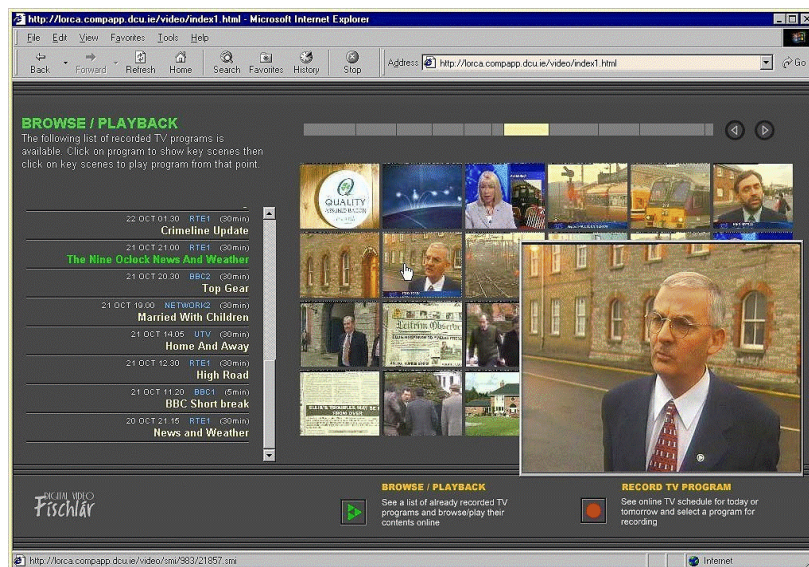

Figure 1: Físchlár - browsable & playable

The Físchlár Digital Video system, with a web interface, is, to put it simply, a digital version of a VCR on the web. With the system a user can pre-select the recording of a TV broadcast programme as one does with a normal domestic VCR. When the time for broadcasting the programme comes, Físchlár automatically starts encoding the programme in MPEG-1 format and this then goes through our shot boundary detection module to be segmented into a number of camera shots with representative frames extracted and miniaturised into thumbnail size pictures and displayed on the system's web page (see Figure 1). Displaying the set of representative images extracted from the encoded programme so that the video content can be easily browsed is the main feature of the Físchlár system and is the immediate output of the shot boundary detection and keyframe selection methods we have developed. The user can browse through these images and by clicking on one pops up a new window and starts streamed video playback from the clicked point in the recorded programme. The user can watch the whole programme from start to finish if s/he wishes (whether it be a ten minute news programme or two hour feature film), or just randomly browse the extracted images and playback from any point of interest in the recorded programme. In this way we provide browsing (albeit linear browsing) from a recorded TV program.

The system is fully working and available online (http://lorca.compapp.dcu.ie/video/) since June 1999 on our departmental intranet, and since then on has seen several feature upgrades. Many of our ideas on how shot boundaries should be detected and how representative frames should be selected have been

integrated into the system and tested with various TV programmes and the results have been validated on the operational system.

2.2 Recording TV Programmes Online

To test and demonstrate the techniques we have developed and integrated into the system, we wanted to have some convenient way of encoding various kinds of video as our test material. Having decided to avail vast amount of all kinds of material on TV broadcast programmes, we were considering recording feature similar to home VCR: recording-setting by specifying date, start time, end time of a TV programme or inputting Video Plus code or browsing TV schedule online and selecting a programme.

Having considered the implementability and convenience in using, we took online TV schedule approach, similar to the TV schedule viewing features in DVB-VCR (SMASH, 1998), ReplayTV (Replay Networks, 2000) and TiVo (TiVo, 2000).

The Físchlár system displays the TV schedule for today and tomorrow on the screen for a user to browse and initiate recording. To record a TV programme, a user browses the online schedule and clicks one s/he wants to record (see Figure 2). This clicking on a programme title adds the title to our system's recording schedule.

Figure 2: Online TV schedule

Being able to easily record a TV programme has allowed us to easily test how our shot boundary detection work on different *types* of video (news, soap, cookery, documentary, with or without commercial breaks, etc.) and also gives us a real set of users from within our Department who use our system on an almost daily basis from their workplace desktops. The system also has a conventional style of setting the recording by specifying the date, start time and end time for a TV programme in order to make the recording task more flexible.

Because our system cannot record more than one channel at the a time at present, just as with a home VCR, the presentation of the online TV schedule was designed to reflect the current status of the recording schedule: the TV schedule is automatically updated every 10 minutes to allow only those programmes which are recordable to be clickable. For example, one should not be allowed to set a programme that has already been broadcast, so all of the already broadcast programmes are greyed out

(in Figure 2, first seven lines of RTE1 schedule are greyed out because they have already been broadcast). Also, if a programme is set for recording in one channel for example, the other channels' time zone for the clicked programme in a channel where there is to be a recording are all greyed out thus also unclickable (in Figure 2, last three grey coloured programmes are due to some other programme from some other channel being set for recording). Clicked programmes which are to be recorded are highlighted in red and are also unclickable (in Figure 2, the two red coloured programmes with icons). In this way, our TV schedule shows what programmes are set for recording and what programmes are still possible to record, thus preventing any recording schedule conflict and user error.

2.3 Browsing and Playback

Once the user sets a recording of TV programme, the system checks record list against current time and starts encoding TV signal when it is time for programme to be broadcast on TV. The encoded MPEG-1 file is then fed to our shot boundary detection module, segmenting the video into each camera shot, and selecting a frame from each of the segmented shots which best represents the shot's content (for more detail on shot boundary detection see section 3).

One of the biggest advantages of digital video over conventional video is that it can provide the viewers with different ways of video content querying/browsing. As discussed in previous work (Mills *et al.*, 1992; Arman *et al.*, 1994; Elliott & Davenport, 1994; Aigrain *et al.*, 1995; Taniguchi *et al.*, 1995), the video medium by its nature provides only sequential access to its content – rewind the video to the beginning, or just playback until it finishes. Even so some devious ways of playback take advantage of the nature of recordable VCR tape and are available in modern VCRs such as FF (Fast-Forward) / REW (Rewind) buttons but these are still limited to the sequentiality nature of VCR tapes, thus do not allow efficient searching or birds-eye viewing of video content.

The shot boundary detection and representative frame selection technique incorporated in the Físchlár system opens up the random access feature by allowing the user a content-based video browsing/navigation tool that overcomes video's sequentiality. Currently available digital video indexing/browsing tools today either in commercial use or in experimental use mostly feature similar kinds of random access by displaying representative images (also known as 'keyframe display', 'storyboard' or 'filmstrips') extracted from the video in some way or another. The Físchlár system's browsing screen is shown in Figure 3 below.

Figure 3: Browse recorded programme's content

On the left of the screen (see Figure 3 above) the system displays the list of TV programmes that have been recorded, encoded and shot boundary detection completed, and are thus ready for users to browse

and playback. A new entry to this list comes at the top. Each entry shows the encoded programme's date, start time, channel name, programme length and the full title (see Figure 4 for close-up).

Figure 4: Recorded programme list

When a full title is clicked with a mouse, the set of miniaturised representative frames for that programme is displayed on the main part of the screen.

Currently the programme entries stay online and in the list for five days: encoded programmes that are older than five days are automatically deleted from the list. This is a way for us to manage the limited disk space available to our system – encoded video files are generally huge in its size even in MPEG-1 format with 30 minute programme taking up approximately 250 Mbytes.

On the main part of the screen the system displays the first 24 miniaturised representative frames extracted from the encoded programme's MPEG file. Displaying a large number of small, thumbnail size images on a screen has been very popular in many picture database applications where a user has to search for a particular image from among a large number, and well claimed for its usefulness allowing the user to quickly browse a large number of images easily (Besser, 1990; Cawkell, 1993; Enser, 1995). Showing thumbnail size images extracted from video scenes can have similar effects, allowing the user to quickly browse the video content in the same manner but the problem with doing this for a set of video thumbnails is the number of them for a long, or even a short program. This could be of the order of hundreds of images for a 30 minute broadcast and simply putting these on a scrollable palette wouldn't make them very accessible so we remove keyframes for shots less than a second or so and divide the remainder into "pages" of 24 thumbnails. A timeline bar on the top right of the screen as in Figure 3 indicates that the set of 24 keyframes currently being displayed on screen are from the start of the programme. The entire timeline bar corresponds to the length of the recorded programme, and each segment length is proportional to the proportion of elapsed time for the portion of the video it is representative of. Clicking on any segment or "page" of the timeline will bring up another set of keyframes on screen corresponding to that particular segment of the video. When the user brings the mouse pointer over the timeline bar, a small tooltip box pops up showing the exact time the bar segment represents (see Figure 5 below).
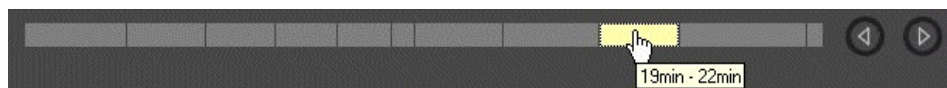

Figure 5: Timeline bar

The length of each segment on the timeline bar is irregular because of the irregular length of the TV programme's camera shots: a timeline segment that contains many long shots becomes a long segment, and vice versa. Using the timeline bar and corresponding miniaturised frames, a user can immediately move to the middle, end, or any time of the programme, quickly assimilating the content of the video. Because many short camera shots are found in TV commercials, we have noticed that short segments on the timeline bar quite often contain commercial breaks (for example see the sixth segment in the above timeline bar – that part of video is a commercial break in the middle of a news programme). Thus a user can roughly guess that those short segments in the timeline bar more likely contains commercial contents. The system does not have the functionality to automatically identify

commercial breaks by analysing the length of camera shots – if we are to implement automatic filtering of commercial, we will have to consider including more cues than simply measuring shot length data, such as more thorough analysis of total distribution of keyframes in the entire programme recorded (Dimitrova *et al.*, 1997).

The playback feature is important when the user wants to watch the recorded video as the final goal of using the system. From the end-user's point of view, this is the feature where the user ends up sitting on his/her desk, passively looking at the screen while the video is being played back. The system allows video playback from any point in video immediately: clicking with a mouse on any of the representative frames will pop up RealPlayer software plug-in and start playback from that clicked point in video onwards (as a result, the screen becomes just like Figure 1). This does not require the user having to wait until the video file is downloaded – a streaming video server allows us to playback while video is being streamed to the web browser.

It is possible to browse sets of keyframes while playing back, which can be useful to compare the current scene being played with frames selected as a result of shot boundary detection. It is also possible to get the playback in full-screen, this being a feature in RealPlayer, effectively watching a TV screen. The quality of the recorded MPEG files when playing back on a full-screen is currently high enough to enjoy, though careful viewers might notice small squares through the screen. This is a matter of our encoding configuration and when the time comes when we consider the full-screen quality of more importance we will start encoding in a higher MPEG-2 resolution at the cost of taking more disk space but still using MPEG-1 for analysis.

In a sense, our video analysis and browsing features using a timeline bar and sets of representative frames provides accurate access point from which the user can initiate playback. Thus a person's video watching behaviour can take the following form:

(1) Choose a programme from the digital library of recorded and available ones
(2) Browse through the major scenes in the video to locate the point you want to watch, then
(3) Playback from that point

where browsing provides a good overview of the part or whole of video and allows content navigation in a random manner, then playback allows the user to watch the content. When digital video systems equipped with convenient random access and content-browsing features become widely available at home in near future, people's video watching behaviour will likely be changed with more frequent jumping from one point of video to another as it becomes easier to do so. This is similar to the frequent channel swapping behaviour of today's TV watchers as it became easier to do with a remote controller in one's hand. Having a fully-working, coherent digital video demonstrator with real users doing real browsing and playing allows us to test out our ideas on users' video browsing behaviour that will be affected by the kind of features in newer systems.

We are deploying the Físchlár system within our department to up to 100 postgrauzdate students and staff in the University so that they can freely record TV programmes and watch them using our system, during which we can observe how they actually use the system and get feedback through interaction logging, questionnaire and voluntary comments.

3 Technology Incorporated in Físchlár

This section briefly describes the technical aspects of the Físchlár system, covering the core techniques of shot boundary detection and others.

The Físchlár system's online TV schedule is automatically updated with the source being updated twice a day and the TV recording schedule status updated checking for recorded programmes to be removed, overlaps with other channels, or new ones clicked for recording, every ten minutes. The system gets its TV schedule source from the PTV web site (Changing Worlds, 2000), which is parsed and stored in our system to be used until the next source update.

The system makes use of two computers, the main one being SUN Enterprise server with large external disk space for storing MPEG files (currently more than 80 Gbytes of disk space is available for storing encoded TV programmes), performing shot boundary detection and also serving as our web server (see Figure 6 below). A separate, dedicated PC with a TV tuner card and MPEG encoding board is used for receiving TV signals and encoding this into an MPEG file. The PC's TV tuner card is connected to a cable TV source in order to receive signals from eight TV channels. Once encoding is finished and the MPEG file generated on the PC, this is FTPed to the Enterprise server, and all the rest of the processes such as storing, shot boundary detection, web page and thumbnail generation, and so on are done on the server.
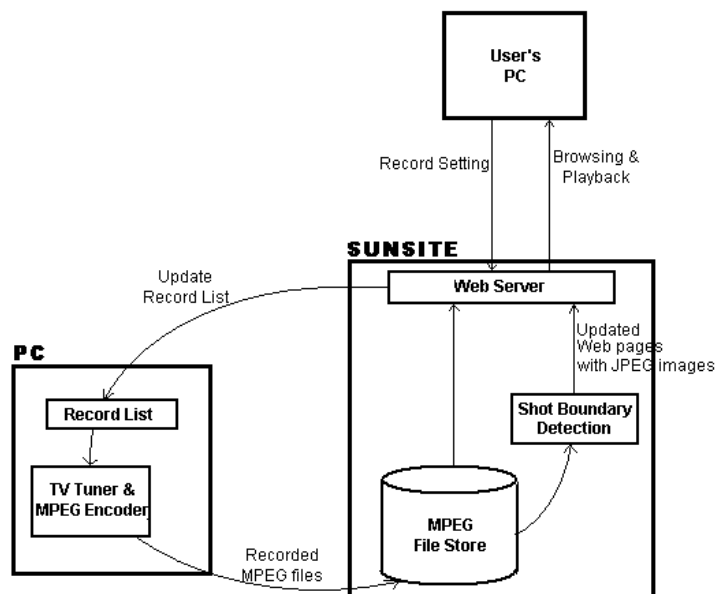

Figure 6: Físchlár system components

Once an encoded MPEG file is FTPed to the server, it is fed to the shot boundary detection module and starts shot detection and representative frame extraction.

What our shot boundary detection module is doing is basically attempting to detect camera cuts in the encoded video stream and mark those cut points. Currently we use a colour histogram comparison method where each frame's colour data is analysed and compared with adjacent frames. A big change in adjacent frames' colour data indicates that a camera cut has likely happened at that point in the video – this frame-by-frame colour comparison allows the system to automatically segment a video stream into small camera shots (O'Toole *et al.*, 1999). However, camera shots are not always clear cut at their start and end as camera effects such as fade-in/fade-out, wipe and dissolve make it difficult for a colour histogram method to detect camera shots accurately.

Recognising the method's limitation we have worked on alternative ways of shot boundary detection, including edge detection (by detecting the appearance of intensity edges in a frame that are fixed distance away from the intensity edges in the previous frame), colour moments (by segmenting each frame into blocks and comparing colours on the basis of statistical intensity from each block of a frame to each block of a previous frame), motion-vector analysis (by extracting motion vector from MPEG stream and comparing this frame by frame) and macro-block type analysis (by looking at MPEG picture types' attributes on forward/backward prediction changes over adjacent frames), and compared these with the colour histogram method (Smeaton *et al.*, 1999). Currently we are finalising evaluation and comparison of these methods, to characterise each method in its strength and weakness,

then see how they can be used together to produce more accurate shot detection results. Once this part of the work is complete, we will combine the methods and integrate into the Físchlár system.

Once a video stream has been broken into small segments, each segment corresponding to a camera shot in the programme, we select from each segment one frame that can best represent the shot's video content. As a preliminary step in selecting a representative frame from each shot, we filter out among detected camera shots the ones that last for less than one second – this is a crude but simple way of compensating some of the false drops of the shot boundary detection (often due to aforementioned camera techniques such as fade-in/fade-out, wipe and dissolve) and also a way of getting rid of not-so-representative shots (a camera shot shorter than one second will be in most case visually insignificant anyway) thus further compacting the resultant frame display.

After this filtering-out of short shots, the selection process for representative frames from within each camera shot starts. In many commercial and experimental systems at the moment, this is simplified by selecting the first frame in the shot, as used in an earlier version of Físchlár and also by others (Simpson-Young & Yap, 1996; Chang *et al.*, 1997; Virage, 2000), or selecting the middle frame in the shot (Deng *et al.*, 1998). More intelligent ways of selecting representative frames are also found in systems such as SWIM (Zhang *et al.*, 1995), WebSEEk (Smith & Chang, 1996) and Informedia (Christel *et al.*, 1997).

In the current version of the Físchlár system we calculate an average colour histogram from every frame in the shot, and compare this average histogram against each actual frame's histogram from the same shot, and we select the one that is closest to the average histogram. Compared to earlier versions of Físchlár which simply selected first frame from each shot, this more intelligent selection method proved far 'better' in representing shot content. For example, it is often the case that the first frame of a shot contains simply blank screens, a black image or an image before the camera becomes focused properly, or a person's face at the corner of screen before fully emerging into the centre. Since we upgraded the system's frame selection method to the average histogram selection method, the browsing screen visibly improved, with each frame more recognisable and more significant than when we used first frame selection.

Our system uses the XIL Imaging and Video Library to extract still-images (in JPEG format) from the MPEG file. Extracted JPEG images are then included in newly-generated HTML pages with their playback time synchronised using SMIL (Synchronised Multimedia Integration Language: World Wide Web Consortium, 2000), so that when a user clicks one of the JPEG images on the screen, the system knows exactly which point in the MPEG file it should start streaming from.

For the playback of video when one of the keyframes is clicked, the system uses a RealNetworks video streaming server. A user's PC needs the RealPlayer software plug-in to playback streaming video through the web browser. Because we are using RealPlayer for playback of the recorded programmes, the playback feature of Físchlár is constrained by RealPlayer's control features – play, pause and stop buttons with time indicators. The RealPlayer also features double-size playback and full-screen playback which can be used together with the browsing features of the Físchlár system.

4 Further Improvements

The Físchlár system's current feature set reflects our work so far in this area. We are currently working on finishing up the evaluations of various shot boundary detection techniques and incorporating them into the system, possibly making it intelligently swap between techniques on the fly that best suits the type of video being analysed.

One of the drawbacks of the system's current browser interface based on shot boundary detection is that we show too many number of keyframes – many of them repeatedly showing similar keyframes. Consider two persons' conversation scene where each person's face alternately shows when s/he speaks. Our detection method takes each person's shot and generates keyframe for each, as a result showing person A, person B, then person A again, person B again,... Repeated keyframes are

redundant and only two keyframes are really necessary to summarise this conversation scene. Thus to make the user's browsing more concise and quicker, we will have to further reduce the number of keyframes especially redundant ones, by grouping several related camera shots into a larger, more topical, semantic unit, or 'scene.' It could be done by further comparing three to four adjacent keyframes to seek the repeatedness among them, or by using continuous audio information through shot cuts (a scene is likely to have the same, continuous background music throughout its component shots), as done in the VISION project (Li *et al.*, 1996). This higher-level segmentation will also be integrated into the Físchlár system as our ideas are formed and implemented, and will be reflected in the system's modified user-interface and interaction mechanism to accommodate higher-level segmentation, possibly a hierarchical display of representative frames showing scene-level as well as shot-level contents interactively, as in Hierarchical Video Magnifier (Mills *et al.*, 1992) and the video browsing tool in SWIM (Zhang *et al.*, 1995). It will be interesting and important to observe how new interactive ways of video browsing are accepted and used by people, and how this actually starts making shape people's video watching pattern.

5 Conclusion

The development of a digital video recording, analysis and browsing tool which operates on real TV broadcasts from eight TV channels and made available to real users who have real interest and motivation for using the system allows us to conduct real experiments in the area of browsing and navigation through digital video. Being able to look at the direct output of our video indexing techniques on the screen with test video material coming from our users' favourite TV programmes has been useful for seeing visible result of our techniques and as a presentable demonstrator of our ideas.

Instead of using a finite set of closed test video materials and only internally testing them against any techniques developed, our approach uses a more open source of video files supplied by recording TV programmes easily. The system has been providing us with numerous opportunities in developing digital video applications – directly testing the output of developed video segmentation methods and frame selection methods against different types of video contents, logging data provided on the output results of different types of video content, measuring the current status of web technology for the video medium, observing the end-user interaction with real people providing ideas on how the user-interface for the shot detection systems should be designed.

The Físchlár system as it currently stands is an implementation of video analysis techniques with a browsing and streaming front end and has a good deal of utility but in addition it is a starting point for further work on more elaborate indexing and browsing mechanisms.

Bibliography

Aigrain, P., Joly, P., Lepain, P. & Longueville, V. (1995). Representation-based user interfaces for the audiovisual library of year 2000. In *Proceedings of SPIE, 2417, Multimedia Computing and Networking* (pp. 35--45). Bellingham, WA: International Society for Optical Engineering.

Arman, F., Depommier, R., Hsu, A. & Chiu, M-Y. (1994). Content-based browsing of video sequences. In *Proceedings of ACM International Conference on Multimedia '94* (pp. 97--103). New York, NY: ACM.

Besser, H. (1990). Visual access to visual images: the UC Berkeley image database project. *Library Trends*, 38(4), 787--798.

Cawkell, A.E. (1993). Picture-queries and picture databases. *Journal of Information Science*, 19(6), 409--423.

Chang, S., Chen, W., Meng, H., Sundaram, H. & Zhong, D. (1997). VideoQ: an automated content based video search system using visual cues. In *Proceedings of ACM International Conference on Multimedia '97*, (pp. 313--324). New York, NY: ACM.

Changing Worlds Ltd. (2000). *On The Box.* http://ptv.ucd.ie/ (visited 5 February 2000)

Christel, M., Winkler, D. & Taylor, R. (1997). Multimedia abstractions for a digital video library. In *Proceedings of ACM International Conference on Digital Libraries '97* (pp. 21--29). New York, NY: ACM.

Deng, Y., Mukherjee, D. & Manjunath, B.S. (1998). NeTra-V: towards an object-based video representation. In *Proceedings of SPIE, 3312, Storage and Retrieval for Image and Video Databases VI* (pp. 202--213). Bellingham, WA: International Society for Optical Engineering.

Dimitrova, N., McGee, T. & Elenbass, H. (1997). Video keyframe extraction and filtering: a keyframe is not a keyframe to everyone. In *Proceedings of 6th International Conference on Information and Knowledge Management* (pp. 113--120). New York, NY: ACM.

SMASH Project (1998). DVB-VCR. In SMASH (Storage for Multimedia Applications Systems in the Home) Project final report. Eindhoven, The Netherlands: Philips Research Labs.

Elliott, E. & Davenport, G. (1994). Video Streamer. In *ACM CHI'94: Proceedings on Human Factors in Computing Systems (ACM CHI'94): "Celebrating Independence"* (pp. 65--66). New York, NY: ACM.

Enser, P.G.B. (1995). Pictorial information retrieval. *Journal of Documentation*, 51(2), 126--170.

Li, W., Gauch, S., Gauch, J. & Pua, K. (1996). VISION: a digital video library. In *Proceedings of ACM International Conference on Digital Libraries '96* (pp. 19--27). New York, NY: ACM.

Mills, M., Cohen, J. & Wong, Y. (1992). A magnifier tool for video data. In *ACM CHI'92: Proceedings on Human Factors in Computing Systems* (pp. 93--98). New York, NY: ACM.

O'Toole, C., Smeaton, A., Murphy, N. & Marlow, S. (1999). Evaluation of automatic shot boundary detection on a large video test suite. *The Challenge of Image Retrieval (CIR'99): 2nd UK Conference on Image Retrieval.* Newcastle, UK: Institute for Image Data Research, University of Northumbria at Newcastle.

Replay Networks (2000). *ReplayTV.* http://www.replaytv.com/ (visited 5 February 2000)

Simpson-Young, B. & Yap, K. (1996). FRANK: trialing a system for remote navigation of film archives. In *Proceedings of SPIE International Symposium on Voice, Video and Data Communications* (pp. 18--22). Bellingham, WA: International Society for Optical Engineering.

Smeaton, A., Gilvarry, J., Gormley, G., Tobin, B., Marlow, S. & Murphy, N. (1999). An evaluation of alternative techniques for automatic detection of shot boundaries in digital video. *Iirish Machine Vision and Image Processing Conference.* Dublin, Ireland: School of Electronic Engineering, Dublin City University.

World Wide Web Consortium (2000). *Synchronised Multimedia.* http://www.w3.org/AudioVideo/ (visited 2 February 2000)

Smith, J. & Chang, S-F. (1996). Searching for Images and Videos on the World-Wide Web. *Technical Report # 459-96-25.* New York, NY: Center for Telecommunications Research, Columbia University.

Taniguchi, Y., Akutsu, A., Tonomura, Y. & Hamada, H. (1995). An intuitive and efficient access interface to real-time incoming video based on automatic indexing. In *Proceedings of ACM International Conference on Multimedia '95* (pp. 25--33). New York, NY: ACM.

TiVo Inc. (2000). *TiVo.* http://www.tivo.com/ (visited 5 February 2000)

Virage Inc. (2000). *The Virage VideoLogger.* http://www.virage.com/products/videologger.html (visited 5 February 2000)

Zhang, H., Low, C., Smoliar, S. & Wu, J. (1995). Video parsing, retrieval and browsing: an integrated and content-based solution. In *Proceedings of ACM International Conference on Multimedia '95* (pp. 15--24). New York, NY: ACM.