FÍSCHLÁR: AN ON-LINE SYSTEM FOR INDEXING AND BROWSING BROADCAST TELEVISION CONTENT

N. E. O'Connor, S. Marlow, N. Murphy, A. F. Smeaton, P. Browne, S. Deasy, H. Lee and K. McDonald

Centre for Digital Video Processing
Dublin City University, Glasnevin, Dublin 9, Ireland.
oconnorn@eeng.dcu.ie
http://lorca.compapp.dcu.ie/Video

ABSTRACT

This paper describes a demonstration system which automatically indexes broadcast television content for subsequent non-linear browsing. User-specified television programmes are captured in MPEG-1 format and analysed using a number of video indexing tools such as shot boundary detection, keyframe extraction, shot clustering and news story segmentation. A number of different interfaces have been developed which allow a user to browse the visual index created by these analysis tools. These interfaces are designed to facilitate users locating video content of particular interest. Once such content is located, the MPEG-1 bitstream can be streamed to the user in real-time. This paper describes both the high-level functionality of the system and the low-level indexing tools employed, as well as giving an overview of the different browsing mechanisms employed.

1. INTRODUCTION

Applications and services based on digital video content are becoming more widespread. This trend is likely to continue as evidenced by the increasing use of intranet video streaming in the workplace, the introduction and subsequent take-up of DVD and digital TV, as well as the deployment of broadband telecommunications networks to the home. With the increasing amount of video information available, there exists a need for efficient management of this information on behalf of the provider and a complementary need for efficient access and navigation of the content on behalf of the end user.

The Centre for Digital Video Processing at Dublin City University is pursuing an on-going research effort to develop essential technologies required for efficient management of video content. The project concentrates on fully automatic video indexing processes addressing both shot-level and scene-level video segmentation. The Centre also addresses the provision of good video content navigation and browsing support for end-users, which we believe to be an equally important aspect of video management. The work of the Centre to date is demonstrated via the web-based Físchlár[1] system.

In this paper we describe the high-level system functionality of Físchlár, the low-level indexing processes and the various browsing/navigation interfaces we have developed in order to support this functionality. An overview of the entire Físchlár system is presented in Section 2 which also describes the user mechanisms for recording (i.e. video capture) and browsing. Section 3 describes the various visual indexing tools we have implemented in the system. The six different browsing interfaces we have developed are outlined in Section 4. Finally, our plans for future work with the system are presented in Section 5.

2. SYSTEM OVERVIEW

Físchlár is a web-based demonstration system which allows users to (i) browse today's and tomorrow's television listings, (ii) select programmes to be recorded, analysed and indexed, (iii) view the visual index created by the system's indexing tools and (iv) select content, based on the index, and have it streamed to them in real-time [1]. The video server used in the system can store approximately 400 hours of video content, whilst the streaming technology employed supports 100 concurrent users.

Users can select programmes from eight terrestrial public broadcast channels. Television schedules can be viewed by channel, programme genre (e.g. comedy, drama, sports, etc.) or day (i.e. today or tomorrow). Most recently, a personalised listing service was introduced in order to offer programme recommendations based on user feedback on previously recorded content [2]. When a programme is recorded, it is captured in MPEG-1 format and stored on

The work described in this paper was funded by the National Software Directorate of Ireland with additional support from the Research Institute for Networks and Communications Engineering (RINCE).

[1] The name Físchlár is derived from two words in the Irish language: *fís* meaning dream or vision and *chlár* meaning programme

0-7803-7041-4/01/$10.00 ©2001 IEEE

the system's video server. This MPEG-1 video bitstream is then analysed using a set of indexing tools in order to create a visual index for the content (see section 3).

Once the visual index has been created it can be presented to the user in the browse/playback section of Físchlár. In the browse/playback section, the list of recorded programmes currently stored by the system is displayed. The user can browse this list by date, channel or personalised recommendation. Once a programme is selected for viewing, its visual index is presented to the user for further browsing at the level of shots or scenes. The visual index for each programme consists of a set shot boundaries and associated keyframes, possibly grouped by scene or subject. A number of different interfaces has been developed, which allow a user to browse this visual index in order to locate video segments of particular interest (see section 4). Once such a segment has been located, the MPEG-1 bitstream for that part of the programme can be streamed to the user. An example of the browse/playback functionality of Físchlár is illustrated in Figure 1.

Fig. 1. Browse and playback in Físchlár

3. INDEXING TOOLS

In this section, the different video indexing tools we have developed and integrated into Físchlár are described.

3.1. Shot-boundary detection and keyframe extraction

The core technology in any video indexing system is shot-boundary detection. We have investigated a number of different shot boundary detection algorithms [3, 4, 5]. The first algorithm investigated (and the algorithm currently employed in the "live" version of Físchlár) uses YUV colour histograms [3]. A histogram with 192 bins is computed for each image and compared with the previous image using the cosine distance similarity measure. A dynamic thresholding operation which adapts to the characteristics of the content being analysed is employed in order to detect shot boundaries. This approach works well for shot cuts but may lead to over segmentation in the case of fades or dissolves. For this reason, a shot boundary detection algorithm based on edge detection was investigated [4, 5]. A Sobel edge detector is applied to each decoded luminance image and the number of differing edge pixels between two successive images is calculated. Again, a thresholding process is employed in order to detect fades and dissolves. In an attempt to make the shot boundary detection algorithm as computationally efficient as possible, an approach based on counting MPEG-1 macro-block types was also investigated [5]. This approach detects when the number of Intra coded blocks rises above a pre-determined threshold signalling a shot boundary.

In order to aid our investigations, an evaluation base-line consisting of eight hours of manually indexed television content was employed. This base-line consists of different types of television content such as news programmes, soap operas, etc [3]. Every shot boundary detection algorithm we develop is applied to this base-line allowing their relative performance on a large test corpus to be evaluated. Using this baseline, work is already underway to investigate combining the three approaches outlined above into a unified approach [5].

Given shot boundaries for a programme, the next step is to extract a representative keyframe for each shot. The approach used selects a keyframe based on its similarity (using the cosine distance metric) to the average histogram calculated over the entire shot [3]. This approach was compared to approaches which simply select the first, middle or last video frame in a shot and was found to result in subjectively better representative keyframes, although this improvement is marginal.

3.2. Semantic boundary detection

Whilst extracting a key frame from each shot gives an overview of the contents of the video, typically this corresponds to a large amount of information which must be presented to the user. In general, people remember different events after viewing video content (and indeed think in terms of events during the information retrieval process) [6]. An event can be a dialog, action scene, news story or any other series of shots that are semantically related. For this reason we have developed a number of semantic boundary detection tools. A semantic boundary is defined as the boundary between two semantic units where a semantic unit is a series of consecutive shots that are related by some common theme or location [7].

In order to perform scene-level analysis of the content, a shot clustering algorithm has been developed. The algorithm we have implemented is based on the temporally constrained clustering approach of Rui *et al* [8]. The main

difference between our approach and that of Rui *et al* is the choice of features used for each shot. We use a single feature corresponding to the average histogram of the shot, rather than the multiple feature approach of Rui *et al*. We have found that this approach has worked well for our preliminary investigations but recognise that it will need to be extended in the future. The result of shot clustering is a set of groups consisting of visually similar shots. The relative temporal location of shots across groups is then analysed and temporal overlaps are detected in order to detect rudimentary scene boundaries [8].

The output groups of shots have also been used in a semantic boundary detection context in order to segment individual stories in Irish news programmes. The approach taken is to attempt to identify groups of shots corresponding to an anchor person. To this end, a number of heuristics based on the statistics of the groups are used. The statistics considered are the mean and standard deviation of the shot similarity measure, the mean and standard deviation of the temporal distance between shots, the number of shots and the mean shot length. Four rules are applied which successively eliminate groups as potential anchor person groups to finally settle on the set of groups which most probably contain an anchor person. This approach is designed to allow for news programmes with multiple news readers. The rules employed attempt to encapsulate the following characteristics of anchor person shots and groups: (i) anchor person groups tend to be larger than most other groups due to the fact that there are many similar shots contained within the entire news programme, (ii) anchor person shots tend to be longer than most other shots in a news programme, (iii) anchor person shots tend to have a global re-occurrence throughout a news programme whereas other shots are localised in time, (iv) anchor person shots tend to be extremely similar to each other. Some illustrative results of anchor person shot detection are illustrated in Figure 2.

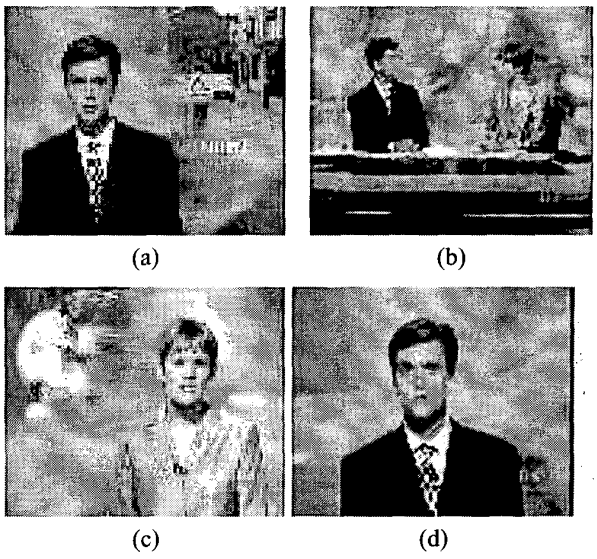


Fig. 2. Example of anchor shot detection in a news programme

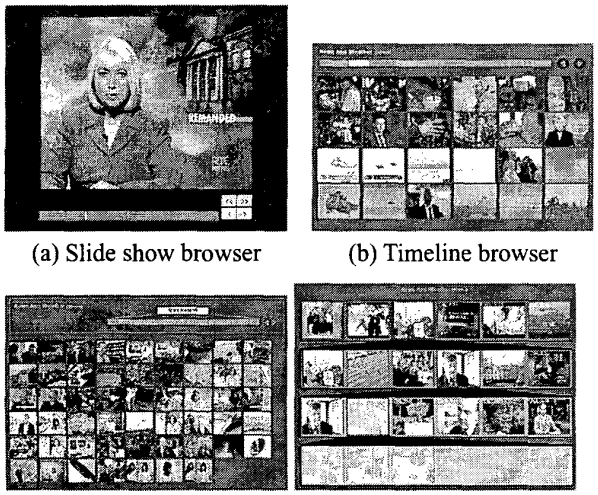
4. BROWSING INTERFACES

The design methodology employed in developing the various interfaces for browsing the visual index is introduced in [9] and described and discussed in detail in [10]. Examples of user feedback we have gathered on the various interfaces is provided in [10]. In this section we simply present a high-level overview of the interfaces.

In the scroll bar browser, the user simply scrolls up and down through all available keyframes which are arranged left to right, top to bottom in order of increasing temporal location in the programme. The advantage of this interface is that it is easy to use. However, such an approach can result in "information overload" for users due to the large number of keyframes associated with video content of any substantial length. In the slide show browser (see Figure 3(a)), keyframes are automatically displayed to the user one by one at rate of 2 per second (approx.). The user can also manually step forwards and backwards through the set of keyframes. A timeline indicator below the keyframes indicates the current temporal location in the programme. The main advantage of this interface is that it provides a summary of the content to the user. The main disadvantages are that typically this summary takes too long and that it is easy for a user to lose the context of what he/she is watching.

The timeline browser (see Figure 3(b)) presents a fixed number (24) of keyframes on one screen. The user can move between screens, and thus browse different sets of keyframes by selecting the associated temporal segment on the timeline bar. The timeline bar provides temporal orientation for users since it is segmented in proportion to the time spanned by a set of keyframes. A ToolTip indicating the exact start and end time of each segment is also provided. Feedback indicates that our users have found this interface attractive and easy to use. The initial screen of the overview/detail browser displays a small number of significant keyframes (see Figure 3(c)). A more detailed view of the video can be obtained on the second screen of this browser which presents the timeline browser to the user. The overview keyframes are selected based on the results of the scene-level analysis in the generic case, and on the results of anchor person detection in the specific case of news programmes. In the hierarchical browser, keyframes are grouped into a hierarchical tree structure which the user can navigate by moving up or down levels in the hierarchy (see Figure 3(d)). The highest level consists of a small set

of keyframes representative of the entire programme. The selection of these keyframes implicitly defines a temporal segmentation or grouping of the set of keyframes. Subsequent levels contain further segmentations of the previous level. This approach has previously been presented in [11]. Currently in Físchlár, the grouping which forms the temporal segmentation at each level is pre-defined and is not based on the results on semantic boundary detection.



(a) Slide show browser (b) Timeline browser

(c) Overview/detail browser (d) Hierarchical browser

Fig. 3. Browsing interfaces

5. CONCLUSIONS AND FUTURE WORK

The Físchlár system is currently used by a small set of technically oriented users. Preparations are underway to extend this user group to include both technical and non-technical users, corresponding to undergraduate and postgraduate students in the University. This would constitute a more representative user group and facilitate rigorous usability studies of our system.

To date, all indexing tools employed in the system work purely on the visual aspect of the video content. This is usually sufficient for tasks such as shot boundary detection and keyframe extraction. However, semantic boundary detection would benefit considerably from some analysis of the audio signal. For this reason, it is intended to develop a set of audio analysis tools which can be combined with our existing tools in order to perform scene-level and eventually event/object-level analysis with a view to aiding the detection of semantic boundaries. Tools such as silence detection, speech vs music classification and speaker segmentation are already being developed.

6. REFERENCES

[1] H. Lee et al, "The físchlár digital video recording, analysis, and browsing system," in *Proc. Content-based Multimedia Information Access (RIAO'2000)*, Paris, France, 12-14 Apr. 2000.

[2] B. Smith and P. Cotter, "A personalized television listings service," *Communications of the ACM*, vol. 43, no. 8, pp. 107–111, 2000.

[3] C. O'Toole et al, "Evaluation of automatic shot boundary detection on a large video test suite," in *Proc. The Challenge of Image Retrieval - 2nd UK Conference on Image Retrieval (CIR'99)*, Newcastle, UK, 25-26 Feb. 1999.

[4] A. Smeaton et al, "An evaluation of alternative techniques for automatic detection of shot boundaries in digital video," in *Proc. Irish Machine Vision and Image Processing Conference (IMVIP'99)*, Dublin, Ireland, 8-9 Sep. 1999.

[5] P. Browne et al, "Evaluating and combining digital video shot boundary detection algorithms," in *Proc. Irish Machine Vision and Image Processing Conference (IMVIP'2000)*, Belfast, Northern Ireland, 2000.

[6] A. Hanjalic J. Biemond, R. Lagendijk, "Automatically segmenting movies into logical story units," in *Proc. of the Third International Conference VISUAL '99*, Amsterdam, Netherlands, June 1999, pp. 229–236, Springer-Verlag.

[7] D. Petkovic P. Aigrain, H. Zhang, "Content-based representation and retrieval of visual media: A state-of-the-art review," *Multimedia tools and applications*, vol. 3, pp. 179–202, 1996.

[8] S. Mehrotra Y.Ru i, T.S. Huang, "Constructing table-of-content for videos," *Multimedia Systems*, vol. 7, pp. 359–368, 1999.

[9] H. Lee et al, "User-interface issues for browsing digital video," in *Proc. 21st Annual Colloquium on IR Research (IRSG 99)*, Glasgow, UK, 19-20 Apr. 1999.

[10] H. Lee et al, "Implementation and analysis of several keyframe-based browsing interfaces to digital video," in *Proc. 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000)*, Lisbon, Portugal, 18-20 Sep. 2000.

[11] H. Zhang et al, "Video parsing, retrieval and browsing: an integrated and content-based solution," in *Proc. of ACM International Conference on Multimedia '95*, New York, 1995, pp. 15–24.