

# USER INTERFACE DESIGN FOR KEYFRAME-BASED BROWSING OF DIGITAL VIDEO

*Hyowon Lee, Alan F. Smeaton, Noel Murphy, Noel O'Connor and Sean Marlow*

Centre for Digital Video Processing

Dublin City University, Glasnevin, Dublin 9, Ireland.

Tel: +353 1 7005433; fax: +353 1 7005508

e-mail: [murphyn@eeng.dcu.ie](mailto:murphyn@eeng.dcu.ie)

## ABSTRACT

In this paper we describe a structured approach for the development of user interfaces for the Físchlár video browsing system, a web-based system for recording, browsing and playback of TV programmes. The user interface to the system was originally designed for desktop use with a large screen and a mouse and we are currently developing versions suitable for mobile device (PDA) access to the system. We review a design framework for video browsing interface formats and some of the formats developed for desktop and PDA use, including interfaces for the Psion Revo and Compaq iPAQ PDAs. This work is driven by the need to investigate how best to include the user in the content specification and retrieval loop and how to find the various balance points between user interaction and system automation.

## 1 INTRODUCTION

The user interface is key to the acceptance of a media-related product in the marketplace. All the technology components can be in place and well integrated, but their effectiveness for an individual user will be unrealised if the user interface is unsuitable. For this reason, in the development and implementation of the Físchlár video browsing system, a web-based digital video retrieval system for TV programmes, we put substantial effort into a structured approach for the development of user interfaces. We have also rolled the system out to campus-resident users in order to get a broad spectrum of usage feedback. The user interface to the system was originally designed for desktop use with a large screen and a mouse. However, we are currently developing versions suitable for mobile device (PDA) access to the system to record and browse video content.

Físchlár is a web-based community-access digital video system with over 600 users within the campus environment in Dublin City University. The system allows the user to record broadcast TV programmes, and facilitates browsing and playback of the recorded programmes on a web browser. The user can easily browse eight terrestrial TV channel schedules for today and tomorrow, arranged in channel, genre, favourites

or personalised recommender form. By simply clicking on a programme, they can set the recording. The system then encodes the programme in MPEG-1 format when the broadcast time comes. The encoded programme is subsequently subjected to automatic camera shot and scene boundary detection to extract representative keyframes. These are the visual medium for the user's interface with video retrieval functions. The web-based interface allows the user to select one of the several browsing methods we have developed to see the keyframes. Clicking on any of the keyframes will pop up a new window which starts streamed playback of the video from the clicked keyframe onwards. The video database system is capable of delivering about 150 independent video streams.

The Físchlár system is a testbed for our technology development, wherein any implemented techniques such as various shot/scene boundary detection algorithms [1], integration with a programme recommender system [2], mobile application for video browsing and playback [3], and various user interface ideas [4] are easily plugged in to the system and the outcomes visibly demonstrated to our current user base. Users of the system are an important element of our work, as they provide new ideas from their own, real, usage context. The Físchlár system is further described in O'Connor *et al* [5] and in [6].

In recognition of the diversity of users' preferences and task contexts, we have developed a design framework for video browsing interfaces that allows us to come up with many different formats of browsing interface. Using this framework we have implemented 8 different browser formats suitable for a desktop environment. The user chooses and uses these different interface formats according to their preferences, and according to their retrieval objectives, which vary from time to time — even within a single user session — and vary from person to person. As we are presently working on the use of mobile devices to access the Físchlár system, porting the systems browsing/playback features to mobile devices has become an important concern for us. In this paper we review the application of the browsing interface design

framework to desktop applications and some of the formats developed for desktop Físchlár use are described. We then show how the framework can also be applied to design suitable interfaces for handheld personal digital assistant (PDA) type devices with their small, touch-sensitive screen and mobile environment use. The resultant PDA video browsing implementations have highly interactive interfaces, but require relatively less visual attention and focusing and can be comfortably used in a mobile situation to browse the multimedia content.

Section 2 briefly explains the rationale for developing the design framework, and summarises the actual framework for keyframe-based browsing interfaces. In section 3, we show how it can be used to design a specific large-screen-and-mouse desktop browser. In section 4 we apply general interface design concerns for mobile devices to the design framework and demonstrate two example designs suitable for a PDA using the Psion Revo and the Compaq iPAQ. These, nevertheless, are primarily case studies. The design priority is to investigate how best to include the user in the content specification and retrieval loop, and how to find the various balance points between user interaction and system automation. Section 5 concludes with some future directions in video browsing interface design.

## 2 DESIGN FRAMEWORK FOR VIDEO BROWSING INTERFACES

One of the main problems in designing a user interface for a novel system such as a digital video browser is the lack of prior experience with any directly comparable type of human-computer interaction. Another is the fact that a single “optimised” interface cannot satisfy everybody, because people come to the system with different aptitudes, attitudes, preferences and task contexts. Furthermore, the current trend in technology is toward a diversification of devices using a single underlying system and sharing the same data, such as email software accessed from an office desktop PC, a PDA or a mobile phone. This results in the need to design different (though related) user interfaces for different devices that are suitable for different users and different contexts. To address these problems, there have been efforts to streamline and turn the fuzzy, unpredictable and ill-defined interface design approach into a more structured and formalised process, exemplified by “design space analysis” [7] and further adapted in various forms such as Stary [8]. In this approach, roughly the following steps are followed:

- analyse and identify important elements and alternatives in designing an interface, resulting in an exhaustive sets of possible design options, or design space,
- consider the particular environment where the interface in concern is to be used, and

- select a suitable set of options from the design space.

In this way, designing the functionality of an interface becomes less of an intuitive, artistic task and more of a concrete and simple decision-making process where the designer can come up with many different interfaces by selecting different combinations of options suitable for the target usage. A crucial part in this approach is the initial construction of the design space and the selection of the right set of options for the target usage. In designing video keyframe browsing interfaces for the Físchlár system, we identify three important design dimensions (layeredness, temporal orientation and spatial vs. temporal presentation) and several possible options or values for each dimension. The detail of the rationale leading to the selection of these particular dimensions and the selection of suitable values along each dimension are discussed in more detail in [9] and [10].

### 2.1 Layeredness

Keyframes extracted from a video programme are a useful way of providing an overview of the programme content, but the number of keyframes presented crucially affects the user’s browsing. A large number of keyframes allows detailed browsing, but is unsuitable for quick browsing. The *layeredness* dimension is concerned with the different possible levels of detail or granularity of the keyframe set and the transition between different levels of granularity. Some of the typical options for this dimension are:

**Single layer** Provides only a single set of keyframes, whether very detailed or selective;

#### **Multiple layer without navigational link**

Provides more than one set of keyframes in a browser, thus the user can select the granularity s/he wants in the browsing;

#### **Multiple layer with navigational link**

Provides more than one set of keyframes, and the user can jump between different sets of keyframes while maintaining the current point of browsing.

### 2.2 Temporal Orientation

The keyframes extracted from video are an ordered set of images in time. Thus an important concern is what kind of *time* information, if any, should be provided to the user when browsing the keyframe set. Some of the typical options for this dimension are:

**No time information** Provides no explicit time information regarding each keyframe;

**Absolute time** Provides exact time information in numeric form (for example, time-stamping a keyframe with “15 minutes 30 seconds into the video”);

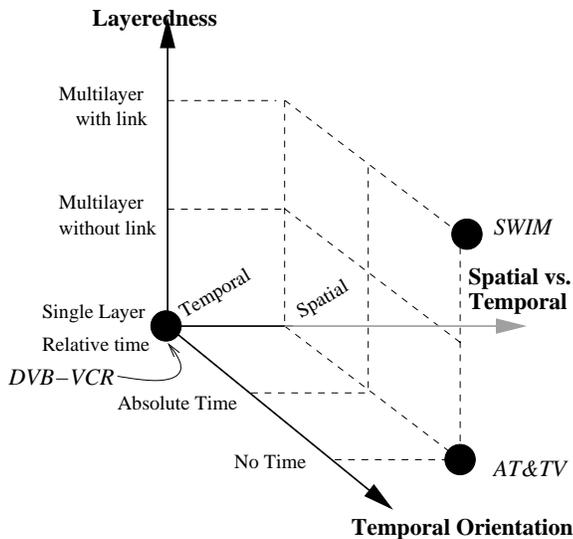


Figure 1: Diagram of a 3-D design space where each axis represents one of the design dimensions. The positions in this space of the SWIM hierarchical browser [11], DVB-VCR [12] and AT&TV [13] are shown.

**Relative time** Shows the time of the current browsing point in relation to the whole length of the video (for example, a timeline bar indicating the current viewing point).

### 2.3 Spatial vs. Temporal Presentation

There are two distinctive ways of presenting keyframes on the screen and the designer has to decide which one should be adopted for the interface in question:

**Spatial presentation** This displays many miniaturised keyframes side by side, allowing quick spatial browsing;

**Temporal presentation** This displays keyframes one by one in the manner of a slideshow.

Actually the two modes are not mutually exclusive and we have in some cases combined both modes in a single interface format. However, the different characteristics of the two modes become important in particular applications, so it is important to distinguish them.

## 3 SPECIFYING A DESKTOP BROWSING INTERFACE

The three dimensions and their typical options described above form a design space where the designer selects one (or more than one) option from each dimension. Because each design option represents a distinctive design decision in a dimension, different combinations of options result in distinctive browsing interfaces. This makes it possible to design all conceivable browsing interfaces within the constraints of the spatial dimensions.

The 3-dimensional space where each axis represents one of the dimensions described above is shown in Figure 1. In this space we can locate positions of several existing video browsing interfaces, for example, the SWIM hierarchical browser [11], DVB-VCR [12] and AT&TV [13]. These interfaces are then seen as choices within a systematic set of alternatives. By locating a selection of particular points in the space, it is possible to propose several further well-specified browsing interfaces that can then be evaluated against the design criteria for a particular target usage application or task.

An example of such an interface is the *Timeline Bar Browser* show in Figure 4. This involves presenting a fixed number of keyframes (24 shown) on one screen. Then as the user moves the mouse cursor over the timeline bar at the top, the screen of keyframes flips through to the next set of (24) keyframes. The timeline bar provides proper time orientation, as bar increments are proportional to the time covered by the set of 24 keyframes. Also, the “ToolTip” shows the exact time of the current screen. User feedback showed the initial implementation of this interface to be very easy to use, but suggested that it could be improved by having a “sticky” mouse pointer so that it is possible to concentrate on keyframes while moving the mouse over the timeline bar.

A selection of eight different interfaces implemented on the basis of this “space-filling” approach is illustrated by the icons in Figure 5. More details of the design criteria that each of these fulfils can be found in [9] and [10]. The Físchlár system for the campus-based “public” is currently running with a selection of five distinctive keyframe browsing interfaces. This reduced set is to avoid information overload of the users and to focus our user evaluations on a smaller of variables.

## 4 SPECIFYING A PDA BROWSING INTERFACE

In designing the browsing interfaces to the Físchlár system on a PDA, general guidelines and common sense can be used in this selection process. For example, on a small, low-resolution screen extensive spatial presentation is not suitable because each keyframe would be unrecognisably small.

However, apart from the physical limitations of PDA devices, the mobile PDA user environment is very different from that of the large-screen-and-mouse desktop. Well-established desktop GUIs are designed to keep the user looking at the screen with proprioceptive awareness of the mouse and/or fine hand-eye cursor control [14]. In the mobile environment (in a bus, on the street, on the metro) the user may be unable to keep focused on the screen, small visual details can easily be overlooked and only one hand may be available some of the time.

In our work to date we have designed PDA interfaces for the Psion Revo and Compaq iPAQ. The Revo has a 480 × 160-pixel landscape touch-sensitive screen with 16

shades of grey. Figure 6 shows one of our designed interfaces. The list of available TV video content is displayed on the right side of the screen, with a scroll bar for right thumb manipulation, while holding the device with the same hand. Below the keyframe from the selected programme displayed on the left side of the screen there are two buttons (previous/next) for the user to flip through keyframes one-by-one, using the left thumb while also holding the device with the left hand. A timeline bar beside the buttons shows the current point of browsing in relation to the whole programme. Automatically flipping through keyframes (true temporal presentation) would be possible, but it would force the user to keep concentrating on the screen. Requiring a high degree of interaction (repeated tapping on the previous and next buttons) should be okay with only two interaction objects (buttons) where these are always under the user's thumb. Note that it is possible to interact with the device using only one hand at either stage of interaction. When both hands are available, the user can use the right thumb for scrolling and selecting a TV programme and the left thumb for flipping through the selected programmes keyframe content.

Another interface for the Revo is shown in Figure 7 below. This interface is designed for browsing the keyframes of a single programme, with multiple layers of keyframes available. With the two buttons on the right side (up/down buttons), the user can jump between 6 different layers, while the layer indicator beside the buttons shows the currently selected layer. The top layer has ten selected keyframes providing an overview of the whole programme; the bottom layer has the full camera shot-level set of keyframes (usually 300-700 keyframes); With the two buttons on the left side (previous/next buttons), the user can flip through the keyframes in the currently selected layer. The current temporal position in the programme is indicated with the timeline bar above the buttons. The layers have navigational links between them, meaning that when the user jumps up or down a layer, the current point of browsing is maintained. This browser is meant to be used with both hands holding the device and continuously tapping buttons in a highly interactive manner as if playing a pocket video game console.

Examples of the same sort of design approach applied to the Compaq iPAQ with its more detailed screen and 4,096 colours are shown in Figures 2 and 3. The different screen format dictates a different layout, but all the features met with the Revo are present: a small number of simple-to-use buttons, single or two-handed use, scroll-bars and level indicators, textual presentation used where necessary, but sparingly, and a clear visual presentation.

In the PDA interface designs, the user has full control over the displayed information on the screen with the widgets being very obvious and always in easy reach. This style makes it acceptable and natural for the user

to casually take attention away from the screen and a few seconds later focus back on the screen. The interfaces were designed in such a way that the user need not pay careful visual attention or point at a small area in the middle of the screen, unlike the majority of desktop application interfaces. The Revo optionally provides "tick" sound as aural feedback whenever a screen is touched. However, mapping the "virtual buttons" on the above interfaces to physical buttons on the device would enhance the tactile feedback for the user.

## 5 CONCLUSION

In this paper, designing keyframe browsing interfaces for video in a desktop environment and a PDA environment is considered, with a specially constructed design framework as a base. The commercial and research community are more and more aware of the importance of recognising people's individual differences and personal preferences. In the user interface design field, attempts to cater for the diversity makes it difficult to have a single user interface for a system which supports everybody's needs. Furthermore, the diversification of different devices for very different environments makes it impossible to stick to a single interface to support these different environments. Identifying all possible interface elements and specifying an interface from this list can be a good starting step for heading toward realising universal access which supports potentially all users and their circumstances. This is the first step toward designs which automatically identify each individual user's preferences and needs at the time of use, and assemble suitable interface elements to provide this to the user dynamically.

Mere technological progress does not guarantee a wide acceptance of usage of that technology in the end product. Numerous failures in usability are found in small, handheld devices because the same interface paradigm for the so-far dominant desktop systems were used without further elaborate consideration. It is thus important to consider in depth the context of the use of the particular interface in concern.

## References

- [1] P. Browne, A. Smeaton, N. Murphy, N. O'Connor, S. Marlow and C. Berrut, "Evaluating and Combining Digital Video Shot Boundary Detection Algorithms", *Irish Machine Vision and Image Processing Conference (IMVIP 2000)*, QUB, Belfast, Northern Ireland, 31 August - 2 September 2000.
- [2] B. Smyth, and P. Cotter, "A personalized television listings service", *Communications of the ACM*, 43(8), 107-111, 2000.
- [3] H. Lee, A. Smeaton, P. McCann, N. Murphy, N. O'Connor and S. Marlow. "Físchlár on a PDA: A Handheld User Interface to a Video Indexing,

Browsing and Playback System”, Poster presented in *6th ERCIM Workshop "User Interfaces for All."*, Florence, Italy, 25-26 October 2000.

- [4] M. Dunlop and K. Mc Donald, “Supporting Different Search Strategies in a Video Query Interface”, *RIA0 2000 - Content-based Multimedia Information Access*, Paris, France, 12-14 April 2000.
- [5] N. O’Connor, S. Marlow, N. Murphy, A. Smeaton, P. Browne, S. Deasy, H. Lee, and K. Mc Donald, “Fischlár: An on-line system for indexing and browsing of broadcast television content”, to appear in the *26th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT.
- [6] <http://lorca.compapp.dcu.ie/Video/>
- [7] A. MacLean, R. Young, and T. Moran, “Design rationale: the argument behind the artifact”, *Proceedings of the ACM Conference on Wings for the Mind (CHI '89)*, Austin, TX, pp.247-252, 1989.
- [8] C. Stary, “A structured contextual approach to design for all”, *Proceedings of the 6th ERCIM Workshop "User Interfaces for All"*, Florence, Italy, 83-97, 2000.
- [9] H. Lee, A. Smeaton, C. Berrut, N. Murphy, S. Marlow and N. O’Connor, “Implementation and analysis of Several Keyframe-based Browsing Interfaces to Digital Video”, *Proc. 4th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2000*, Lisbon, Portugal, 18-20 Sept., 2000.
- [10] H. Lee, *User Interface Design for keyframe-based Content Browsing of Digital Video*, PhD Thesis submitted to Dublin City University, January 2001.
- [11] H. Zhang, C. Low, S. Smoliar and J. Wu, “Video parsing, retrieval and browsing: an integrated and content-based solution”, *Proceedings of ACM International Conference on Multimedia 9 5*, San Francisco, CA, November 7-9, 1995, 503-512.
- [12] “DVB-VCR”, *SMASH(Storage for Multimedia Applications Systems in the Home) Project final report*, 1998.
- [13] M. Mills, J. Cohen and Y-Y. Wong, “A Magnifier tool for video data”, *Proceedings of ACM CHI '92*, 1992.
- [14] S. Kristoffersen and F. Ljungberg, “‘Making place’ to make IT work: empirical explorations of HCI for mobile CSCW”, *Proceedings of the ACM Conference on Supporting Group Work (GROUP '99)* Phoenix, AZ, 276-285, 1999.



Figure 2: An example browsing interface on a Compaq iPAQ.

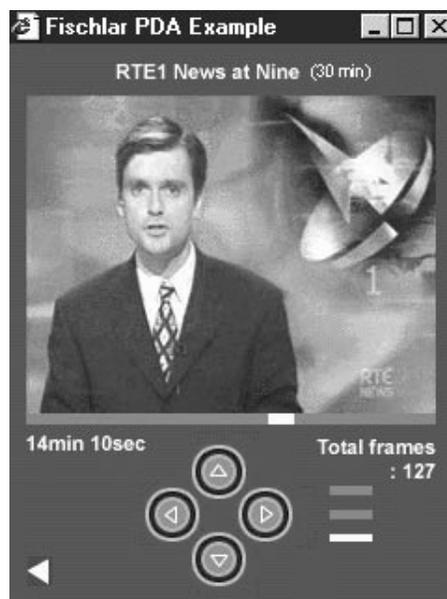


Figure 3: An example browsing interface on a Compaq iPAQ.



Figure 4: A Timeline Bar Browser screen shot with a playing video window superimposed.



Figure 5: The icons a Físchlár user can use to alter the video browsing format.



Figure 6: An example browsing interface on a Psion Revo.



Figure 7: An example browsing interface on a Psion Revo.