Fluid Interaction Techniques for the Control and Annotation of Digital Video

Gonzalo Ramos, Ravin Balakrishnan
Department of Computer Science
University of Toronto
bonzo | ravin @dgp.toronto.edu
www.dgp.toronto.edu

ABSTRACT
We explore a variety of interaction and visualization techniques for fluid navigation, segmentation, linking, and annotation of digital videos. These techniques are developed within a concept prototype called *LEAN* that is designed for use with pressure-sensitive digitizer tablets. These techniques include a transient position+velocity widget that allows users not only to move around a point of interest on a video, but also to rewind or fast forward at a controlled variable speed. We also present a new variation of fish-eye views called *twist-lens*, and incorporate this into a position control slider designed for the effective navigation and viewing of large sequences of video frames. We also explore a new style of widgets that exploit the use of the pen's pressure-sensing capability, increasing the input vocabulary available to the user. Finally, we elaborate on how annotations referring to objects that are temporal in nature, such as video, may be thought of as links, and fluidly constructed, visualized and navigated.

Keywords: Pen-based interfaces, fluid interaction techniques, annotations, video.

INTRODUCTION
Each day we interact with a rapidly growing amount of digital information, of various data types. The computer applications for viewing, manipulating, and annotating some of these data types, such as text and images, have become quite established among the average computer user. Video, however, is a data type that has only recently moved to digital form. The increasing availability, and ever lowering cost, of digital video capture equipment has resulted in the creation of videos moving beyond the realm of specialists such as filmmakers and TV producers into the broader consumer market. While the ability to capture raw digital video footage has become easy, affordable, and a popular pastime for many, the software applications for navigating and manipulating the resulting hours of footage remain relatively difficult to use, even for specialists. Currently available video manipulation and editing software tend to have user interfaces that mimic the style of old analog editing suites, with all their accompanying idiosyncrasies. Additional functionality afforded by the non-linear digital form is often buried within layers of menus, and many tasks often involve modal dialogues that disrupt the flow of the user's thoughts and actions. As a result, accomplishing even the simplest of tasks can take inordinate amounts of time and be rather frustrating. Current tools also do not easily allow for videos to be annotated or segments to be quickly linked to one another or to other data types. While these problems are not unique to video, much work has already gone into mitigating them for data types such as text and images, whereas comparatively little research has been done on fluid user interfaces for video. Moreover, unlike text or still images, video sets the pace at which it must be experienced, presenting unique interaction and visualization challenges given its nature as an object existing not only in space, but also in time.

In this paper we describe the design and implementation of a variety of fluid interaction and visualization techniques for navigating, segmenting, linking, and annotating digital video using a pressure-sensitive pen-based interface. These techniques are demonstrated within a concept prototype called *LEAN* (Figure 1). To motivate our interface designs, we first review the current practices of those who manipulate video and film in both physical and digital forms, as well as related systems and techniques. We then discuss the design philosophy behind *LEAN*, and details of its interaction techniques. We conclude with preliminary observations of users working with the system.

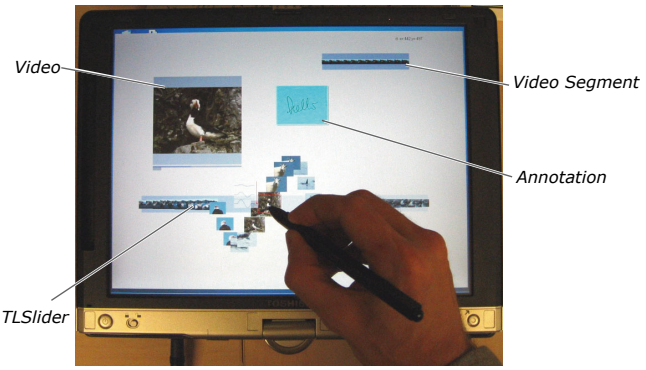


Figure 1: The *LEAN* system running on a TabletPC.

UIST '03 Vancouver, BC, Canada

TRADITIONAL VIDEO/FILM PRACTICES

During the design process, we conducted a number of interviews, along with task analyses, of five professionals who each manipulate video for very different purposes. These included the study and critique of film as an art form, the academic use of film/video as a record keeping medium, and the creation and editing of video in a production setting. These professionals were interviewed in their workplace. We solicited feedback on their methods, tools, and current practices. We also either demonstrated early versions of *LEAN* running on a TabletPC, or played a series of videos that demonstrated the interaction techniques afforded by the system. Our observations provided us with insight into the current tools and techniques used for interacting with video. They also enabled us to develop and refine our interaction techniques such that they leverage current best practices.

People involved in film and video production want to narrate a story. To that end, they manipulate and rearrange large quantities of film/video clips in order to arrive at the desired final product. When film is in digital form, *Non Linear Editors* (NLE) like Adobe Premiere or Final Cut Pro are the tools commonly used to cut, paste, and compose movie segments. Digital video allows for the reversible manipulation of its contents, and provides access to an assortment of compositing effects. However, NLEs do not offer the directness and fluidity in manipulations and interactions that are typical of physical film. For example, interviewees used to working with actual film appreciated being able to simply hold a film strip in both hands and to quickly move it back and forth in order to preview a segment. They are also used to holding it up to the light in order to view the contents of a single frame. In addition, these practitioners are accustomed to using a grease pen to make annotations directly on the film.

Scholars and students who study film as an art form analyze, critique, and communicate their views about a movie's context, history, features, and techniques. Interestingly enough, however, publications and articles in this field exist exclusively in the printed form. As a result, concepts and information relevant to those who study film have to be transmitted solely with the aid of static images, or at best a sequence of thumbnails accompanied by a textual explanation or transcript. Professors of film studies expressed their dissatisfaction with both the limitations of printed material and with the authoring tools at their disposal. They emphasized the need to be able to portray the dynamic nature of a particular movie scene, along with its relationship both to other scenes, and to the movie as a whole. Film students face challenges when they need to access and navigate a heterogeneous set of artifacts that includes film, tape, and digital media. For the non-technically savvy user, having to utilize different tools for media manipulation is a common source of frustration. It is not unusual for practitioners in this area to transcribe a movie clip into text or a log. Once in this form, the transcript becomes a representation of a movie that can then be accessed and manipulated using a set of tools (e.g. word processors) with which users are generally more familiar.

Ethnographers are particularly concerned with the study and systematic recording of human cultures, and often use video to collect their observations and to analyze them at a later time. The analysis of these videos can involve tasks such as annotating portions of a clip, tagging frames, and organizing the scenes and data into collections.

Our observations and interviews strongly suggest that all the aforementioned practitioners would certainly benefit from tools that support casual and fluid annotation, linking, control, and dissection of one or more video streams. Furthermore, these tools should be as unobtrusive as possible, allowing users to perform their tasks without a surfeit of user interface widgets cluttering their data space. All interviewees expressed an intense interest in the early versions of *LEAN*. Even at the almost marginally interactive rates provided currently by the TabletPC hardware it was demonstrated on, the interviewees stated that *'...I could use a system such as this right now'*.

RELATED SYSTEMS AND TECHNIQUES

There are a number of pieces of related work that address the areas of fluid/non-intrusive interactions, navigation of video streams, and annotations, all of which have influenced our work. Fluid interactions using a pen as an input device are frequently showcased in whiteboard interfaces such as in Tivoli [15] and Flatland [14], or in the work done on large displays by Guimbretière et al. [4]. The Electronic Cocktail Napkin [3] is a pen-based environment that supports the abstraction, imprecision, and ambiguity of freehand diagrams made by users. The system parses the ink drawings and is able to recognize and disambiguate shapes, based on the drawing's context and structure.

The *XLibris* system [19] imitates paper by using a high-resolution pen tablet display that provides users with some of the affordances of paper. With XLibris, users can annotate and highlight pages of documents fluidly, with an ease approaching that of printed materials. XLibris departs from the traditional WIMP interface and follows the design principles of a transparent, minimalist user interface and modeless interaction.

Toolglasses [1] provide users with a bimanual, non-intrusive tool that does not distract their attention from the tasks at hand. Another non-intrusive technique is Marking Menus [8]. Marking Menus are transient widgets that allow users to have access to commands in a fluid manner. With Marking Menus, novice users can take advantage of a hierarchical radial menu structure, while advanced users can access commands by making a mark, or gesture, without having to wait for the menu to appear. FlowMenus [5], FaST sliders [12], and Control Menus [16] present quick, easy to learn, and transient controls that combine

menu selection and the adjustment of continuous values. In addition, FaST sliders allow users to quickly switch between different scale granularities when adjusting parameter values. In Snibbe et al. [21] users navigate a video sequence using a haptically actuated spinning wheel that takes advantage of the user's physical intuition.

SILVER [13] is a video-editing tool that presents a number of interaction and visualization techniques. Of particular interest to us is the system's Timeline View, which displays an explicit 3-level hierarchy that is defined when the user zooms down into a video segment This hierarchy is useful for navigating through the time-line of the video. Users can also add text annotations that span a portion of a video segment. Our system is similar in the way it handles the visualization of video segment relationships, but it does not have the limitation of allowing only a 3-level hierarchy.

The VANNA system [6] investigates how people manipulate and annotate temporal information. It supports a variety of input devices, e.g. mouse, keyboard, touch screen, and pen, all of which can be used to capture either on-line or off-line notes. The PhotoFinder system [20] addresses the complexity of a large collection of annotated images by allowing users to drag-and-drop labels from a scrolling list of attribute values to a particular place on a photo. The *Boom Chameleon* [22] introduces a specialized input and output device that allows users to navigate and annotate a 3-D environment. Annotations on this system are made by drawing directly on the surface of a virtual object, or by taking 2-D snapshots that capture the user's point of view at a given point in time.

In short, our review of the literature indicates that while many of the issues with which we are concerned – video, annotations, linking, fluid interactions, and uncluttered workspaces facilitated by transient widgets – have been explored individually by various researchers, they have yet to be explored in combination.

OVERVIEW and DESIGN PHILOSOPHY of *LEAN*

We developed a system called *LEAN* that serves as an exploratory platform for new visualization and fluid interaction techniques for navigating and controlling digital video. Our system targets the casual user, and in addition to various editing operations, allows for casual annotation and cross-linking of video streams. Its primary interface is a digitizer tablet with a pressure-sensitive pen. Our intention is to leverage users familiarity with pen-based interactions in the physical world, and the emerging tablet-based computers (*LEAN* runs on a TabletPC, although current TabletPC hardware is too slow to provide the interactive responsiveness we get with higher-end workstations equipped with digitizer tablets).

LEAN allows for the manipulation of a video stream by using a small set of gestures that lets users start, stop, and travel to any arbitrary point in time in the stream. Also, by using only simple gestures, users are able to select intervals, or segments, from the video. Besides allowing users to manipulate the video stream, the system also permits users to attach annotations – easily created by scribbling on the working area or over the video image – to video frames and segments. By connecting an annotation to a desired element on the working area, the user can provide it with a positional and temporal context. In addition, users can trigger at will visualizations that correspond to a complete video segment and that also allow for both the quick navigation of the video stream and the speedy location of the annotations situated within.

In designing *LEAN* we were particularly interested in creating techniques to enable users to navigate and annotate digital video with a fluidity and ease similar to navigating and making annotations on printed material using physical tools such as pens and post-it notes. Another goal was the design of appropriate visualizations for the subsequent retrieval and viewing of those annotations. In our design, we strove for a minimalist approach to the interface, both in the gesture set used, and in the visual aspects of the design, believing that an excess of visual decorations introduces noise to the task at hand and only serves to make the user acutely aware of the intrusive presence of the computer.

GESTURES, COMMANDS, and SCRIBBLING

Systems that use a pen as an input device for both commands and data input have to contend with the ambiguity that often results when interpreting the user's input actions. For example, an input stroke could have several meanings: a gesture intended to invoke a command, a simple scribble, or a simple pointer movement. Previous research systems have adopted different approaches to address these ambiguities. For example, Flatland [14] uses a button on the pen to divide the user's input into two modes: drawings and meta-strokes, and a tap gesture to invoke a pie-menu for command entry. DENIM [9] separates scribbles and commands by using a button on the pen, and also by using a tap gesture to invoke a pie-menu that then provides users with further commands. Guimbretière et al. [5] use a button on the pen to invoke a FlowMenu for command input. Another approach is to interpret the input strokes and classify them into either command gestures or raw scribbles.

We use a combination of these approaches. A small set of gestures is interpreted by parsing single-stroke inputs using Rubine's features [17]. The effect a gesture has depends on the context in which it was made, i.e. the object(s) upon which it was made. Table 1 summarizes this gesture set. The various gestures and their interpretations will be explained in detail as we proceed through the paper. With the exception of 'selecting' objects, we found that for the purposes of our initial research, it sufficed that the system distinguishes between scribbles and commands by a simple algorithm that tests a stroke's features such as space, time, speed, and pressure. Objects in *LEAN* are 'selected' in the

working area by using the pen's button, all without even having to touch the tablet's surface. Chosen objects reveal their links, and can be later moved over the workspace by moving the pen over the tablet's surface while simply keeping the pen's button pressed.

	Video Frame	Video Segment (VS)	Note	Link	Empty space
Flick Right	Invoke main *TLSlider*	n/a	Draw line	n/a	Create Note
Flick Left	Hide main *TLSlider*	Hide / Minimize *TLSlider* (1)	Delete Note	n/a	
Interval	Annotate frame	Create new Video Segment / *TLSlider*	Add scribble to Note	n/a	
Tap-And-Hold + Scrub	Control video flow	Navigate through timeline	Navigate through links	n/a	n/a
Tap-And-Hold + Pressure	Select beginning / end of new VS	Vary amplitude of Twist Lens	Pin / Unpin Note	Grab	n/a
Scribble Hello	Annotate frame	n/a	Add scribble to Note	n/a	Create Note
Tap	Start / Pause Video	Jump to Frame	n/a	n/a	n/a
Pen's Button	Select Object / Reveal Relationships				
Pen's Button + Move	Move Object				

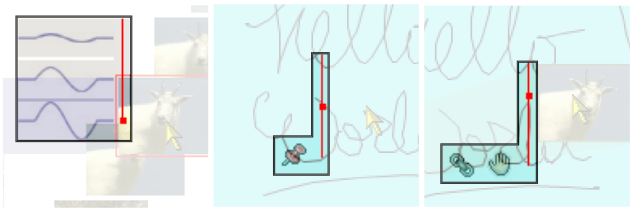
Table 1: Gesture grid that shows the basic set of gestures recognized by the *LEAN* system. The top row shows the object that gestures can be applied upon, while the leftmost column enumerates the basic set of gestures. Each cell in the grid describes the effect of a particular gesture on a certain object.

Our system also uses menus and widgets that are invoked by *Tapping-And-Holding* (TAH) the pen on the tablet's surface for a small period of time, after which the control appears or becomes active. This is similar to the way marking menus were invoked in [8]. An animated diagram, similar to the one found in the Apple Newton or in Windows CE 3.x, provides users with feedback regarding the initiation and completion of the TAH gesture.

PRESSURE and PRESSURE WIDGETS

Unlike the aforementioned previous research, our system uses the pressure information from the pen to expand the set of directly invokable commands available to the user. A pen's pressure is sometimes used in image manipulation programs like Adobe Photoshop to control some continuous parameters of a drawing tool, such as the thickness of a pencil or the opacity of a brush. However, traditional WIMP interfaces assume that a user's pointing device can only produce spatial x-y position coordinates and discrete clicks as input to a system. As such, their widgets are designed only for these two input types and do not take full advantage of the pen's pressure modality.

To leverage the capabilities of the pressure-sensitive pen, we developed visual *Pressure Widgets* (Figure 2) that help users become aware of the amount of pressure being applied, and the consequences of varying the pen's pressure (Figure 2). Discrete pressure widgets activate an action once a certain pressure threshold is exceeded, while continuous pressure widgets map pressure to the control of a continuous parameter. The key element of pressure widgets is the visual display of the effects of the changing pressure. For continuous pressure widgets, we use a series of icons that reflect the consequences of the user's actions (Figure 2a). For discrete pressure widgets, we use a single icon (Figure 2b), or set of icons (Figure 2c), displayed at the appropriate pressure threshold. Instead of employing complex icons to describe compound actions, we chose a small, simple set of icons that can be combined in what we call *sequential icons* (Figure 2c). We believe that sequential icons are likely to be simpler to learn than composite ones.



(a) (b) (c)

Figure 2: Pressure Widgets (background of this figure has been altered in order to emphasize the widget's appearance). a) Continuous control of the amplitude of the Twist-Lens. b) Discrete control for pinning a note to the workspace. The pinning action occurs after the pressure exceeds the displayed threshold. c) Discrete control for grabbing a link. A sequential icon indicates the action of grabbing and the item to be grabbed which is a link.

VIDEO CONTROL

The control of a video stream in most software is carried out using a VCR-like interface (Figure 3), with different buttons or widgets that play, pause, fast forward, or rewind the video. In addition, clicking on the timeline often directly positions the video at a particular point in time. Such an interface produces a separation between the video data with which users are engaged, and the widgets necessary to control it. This strategy of separating the controls from the data works with text documents and other types of non-temporal material, because of their static nature. In these cases, we expect (and are usually not disappointed) that a small switch in our attention from the document to the control and back will return us to the same view of the document. The same cannot be said about video – a media that, when engaged, changes as time passes. In video, this separation between controls and data forces users to play a 'game' of target acquisition, which we believe is unnecessary and quite avoidable in a properly designed video control interface.

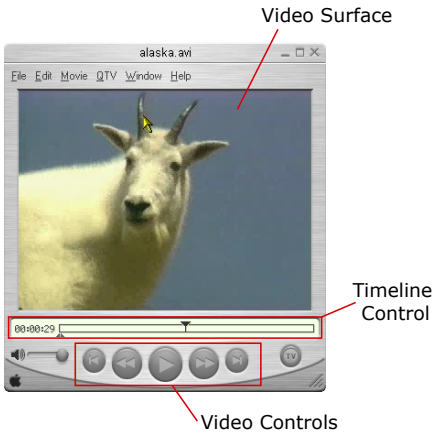


Figure 3: A typical video player with a VCR-like media control widget. This interface separates the data (video surface) and the widgets that control it (timeline and video controls).

Position+Velocity Sliders

We incorporated a number of interaction techniques into a 'one-stop shopping' solution for the non-intrusive control of a video stream. Users can start and stop a video by tapping on the video surface. Fast forward and rewind functions are performed by using a novel, unobtrusive transient position+velocity slider widget, called the *PVslider*. The *PVslider* (Figure 4) is a hybrid position+velocity control that allows users to drag across the tablet's surface in order to move within the vicinity of the current frame using position control, or to move forwards or backwards in the stream at a variable rate using velocity control. The *PVslider* is invoked when the user taps and holds over the video, a gesture that defines the *point of origin* (*PO*) of the control. The control looks like a horizontal line segment, which follows the pointer in the vertical dimension and remains connected to *PO* with a line, or 'rubber-band', linking the pen's position and *PO* (Figure 4b).

The *PVslider* is divided into a *Position Region* and a *Velocity Region*. The *Position Region* is the horizontal line the user sees. It is mapped to an interval on the video stream centered around the frame where the control was invoked. The size of this interval is directly proportional to the vertical distance between *PO* and the current pen's position. As such, the interval's size can be changed by moving the pen in the vertical direction (Figure 4a,b). Moving the pen in the horizontal direction within the boundaries of the *Position Region* allows the user to scrub through the frames in the given interval. The user fluidly enters the *Velocity Region* by dragging the pen horizontally beyond the ends of the *Position Region*. Here the *PVslider* acts as a velocity control allowing the user to move through the video stream at a velocity proportional to the length of the rubber-band, i.e., the farther away the pen moves from *PO*, the faster the user moves across the video stream in that direction. Thus, users can fast forward or rewind the video by dragging to the right or left of *PO*. Note that the transition from position to velocity control is completely seamless, with no explicit mode switch. Rather, the switch is implicit, based simply on the distance of the cursor from the *PO* in the horizontal direction. Also, the *PVslider* constantly provides visual feedback indicating its current status as either a position or velocity control, along with the magnitude of the speed at which the user moves through the video stream (Figure 4c,d).

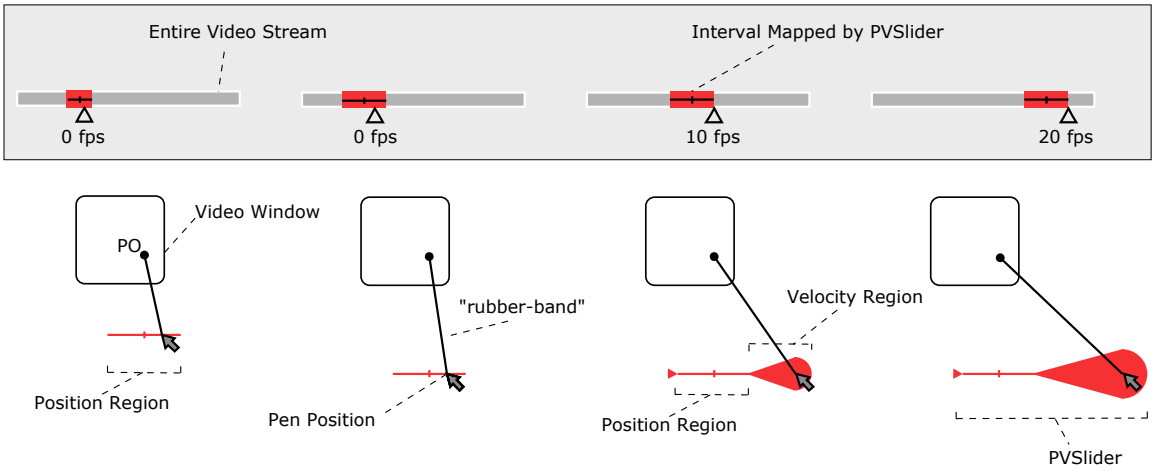


Figure 4: The *PVslider* widget and features. a) The *PVslider* is connected to the *point of origin* (*PO*), and mapped to an interval of the video stream. Note: the grey box above it is not part of the interface; it is here for illustrative purposes. Also the frames-per-second (fps) values are illustrative and do not correspond to real data. b) As the pen's vertical distance to *PO* changes, the size of the interval mapped changes. c,d) Moving the pen beyond the *Position Region* takes it into the *Velocity Region*. The farther away the pen is from the starting point in the horizontal direction, the faster the users move through the video stream. The size of the *Velocity Region* cone provides visual feedback on the magnitude of the current speed.

Twist-Lens Sliders

Although the *PVslider* offers users an absolute position control, this region does not map to the whole length of the video stream the same way slider controls on VCR-like interfaces do. With this in mind, we developed a novel interaction and visualization technique based on fish-eye lenses called the *Twist Lens slider* or *TLslider*. Using a flick right gesture (Table 1), a user invokes the *TLslider*, which provides a visualization of the complete video stream as a sequence of thumbnails. Once a user taps and holds on the *TLslider*, it acts as an absolute position control for the portion of the video stream to which it is mapped.

When the *TLslider* becomes active, the user can drag across the control with the pen and the result is that the fish-eye view expands the area centered at the location of the pointer. While the visualization of the *TLslider* enables the frames of interest to be expanded visually, our design does not expand the targets in the motor domain because of the issues regarding target acquisition that have been studied in detail by McGuffin in [11]. As discussed in [11], in a widget with multiple targets expanding in the motor domain, the motor location of the targets typically shifts as the targets change size, making them difficult to acquire. Such an effect can be seen in the 'dock' in the Mac OS X interface. Instead, we keep the mapping between the video frames and the space defined by the *TLslider* constant. However, this design choice presents another challenge: the frames visually expanded by the fish-eye view partially occlude their neighbors, or context (Figure 5). We overcome this problem in two ways. First, the thumbnail that is the focus of attention shows not an enlarged version of the closest key frame, but the actual frame corresponding to that particular point in time. Second, we morph the linear layout to an s-shape (which gives this technique its name) that depends on the pressure applied by the user's pen on the tablet's surface (Figure 6).

A continuous pressure widget (Figure 2a) provides a visual preview of the results of varying the pressure. By showing the precise frame at a particular point in time, instead of a static thumbnail representing an interval, we allow users to accurately preview moving through the timeline.

By smoothly morphing the slider into a sinusoidal shape, we create sufficient space to eliminate occlusion among thumbnails. We found that this distortion technique has the added bonus of providing a visualization that is not occluded by the user's hand as is often the case in devices that integrate display and digitizer (e.g. Wacom CintiQ or TabletPC), and that can also accommodate, by mirroring its shape, both right-handed or left-handed users (Figure 1).

Video Segments

In our system, the *TLslider* is also a particular instance of a more generic object, a *Video Segment*. *Video Segments* are sections of the video stream that the user can define simply by selecting an initial and final frame, or by using a gesture to select an interval from an existing *Video Segment*. *Video Segments* also indicate the progress of the video stream, by changing over time the color of its background border from grey to blue as the video is played. Unlike typical progress bars found in most video players which are spatially separate from the associated video stream, ours does not divide the user's attention. This feature allows users to see at a glance if the segment has already been played, is currently being played, or hasn't been played yet. In order to unclutter the workspace users can, if they wish, collapse a *Video Segment* into an iconic representation with a simple flick gesture (Table 1).

We also support the user's need to see relationships – for example, if a *Video Segment* is fully or partially contained in another. When a user grabs a segment, the system automatically displays its immediate relationships to other segments via a series of semi-transparent 'large-base' arrows, as shown in Figure 8. *Video Segments* can be used to structure a video stream into different pieces that can then be used to support tasks such as the analysis of film and the navigation through a video stream. In a sense, this is analogous to the traditional practice of using a pair of scissors to cut film into strips that we observed during our user interviews and task analysis.

Figure 5: This partial view of the *TLslider* shows how a regular fish-eye approach that keeps a fixed target size may present occlusion problems in the vicinity of the focus.

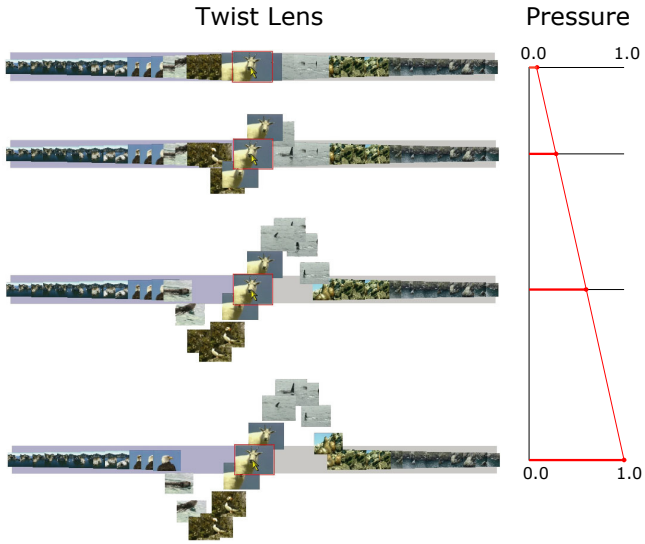


Figure 6: *TLslider*. The figure shows from top to bottom how the amplitude of the lens changes with the pen's pressure, which is displayed on the right.

ANNOTATIONS and LINKS

Apart from providing fluid controls for video navigation and segmentation, another primary goal of our work was to research techniques for annotating video. Because of its widespread use and undeniable fluidity, active reading on paper is used as a model to study, and from which to generalize, the practice of annotation [10], or as a metaphor for systems and interface design [19]. To a certain extent, we follow this approach and let users create explicit annotations by writing directly into the empty area of the screen. They can then connect the resulting note to a movie frame or a *Video Segment*. Users can also scribble on top of a video frame in order to leave 'in-place' markings on a particular frame.

From Marshall [10] we learn that annotations have both form and function. One of the most significant attributes of an annotation's form is its location. A note on the margin of a book, for example, has a location near some printed text that is likely to be related to what was hand-written. In addition, the portion of a photograph where a circle was drawn, or the moment at which a voice comment was made, also demonstrates the importance of an annotation's location, regardless of the type of media. An annotation only becomes useful because of its location and its relationship with the surrounding context. When dealing with printed material, a mere visual inspection can reveal both the annotation and its context. However this is not the case with a video stream, where the context can be not only space, but also time. When the context of an annotation is temporal, a person must experience the media through time until the moment when the annotation was actually made occurs. The nature of temporal context does not allow us to experience the previous and future moments that surround an annotation's place with a quick glance, unlike the way we experience spatial context.

In order to provide the user with a similar type of contextual awareness that occurs with annotations made in space, we have developed an approach that visually blends a linked note (or annotation) smoothly in and out of the environment as the moment (or time interval in the case of a *Video Segment*) when the annotation was made approaches (Figure 7a,b,c), and then passes (Figure 7c,d,e). This is similar to the techniques used in HyperVideo [18], where *hypervideolinks* or 'opportunities' fade in and out of a running video sequence. But while the aforementioned work in HyperVideo separates creators and users, ours blurs the distinction between 'readers' and 'writers' of an annotated video stream. Other visual cues are provided in the form of animated markers on the side of the video frame being played. These markers have a size and position directly related to both the number of annotations and the moment a particular annotation was made. Users also have the ability to 'pin' a note into the workspace using a discrete pressure widget, making it visible at all times (Figure 7). Notes connected to a frame have an associated thumbnail that can be seen on all *Video Segments* containing the annotation's temporal context. Notes made directly over a frame have an associated mark also seen on the relevant *Video Segments*. These thumbnails and marks can be used as visual landmarks or bookmarks that help users to navigate the video stream to reach defined points of interest. A note attached to *Video Segments* has the same behavior, except that its thumbnail is displayed on the right of the segment (Figure 8).

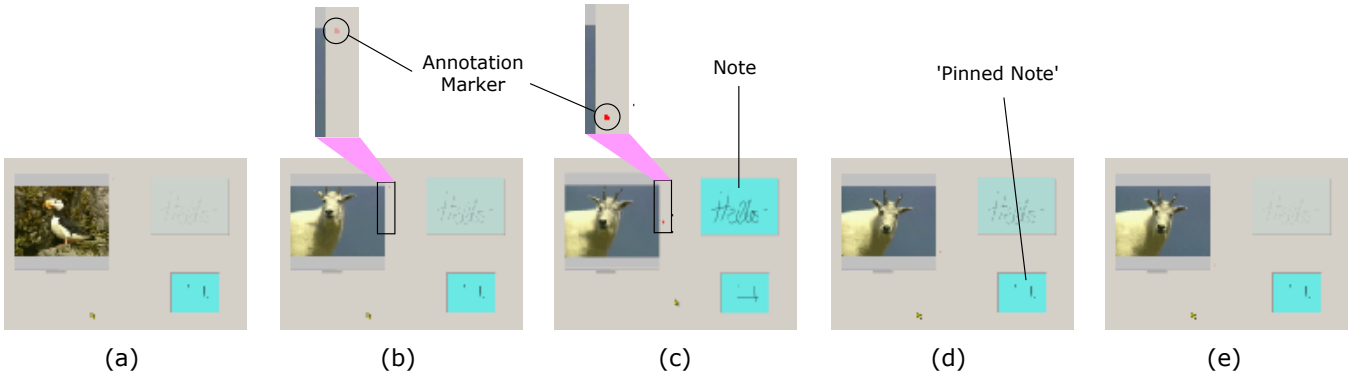


Figure 7: A sequence demonstrating the contextual visualization of an annotation. From a) - c) A note fades into the workspace, while an annotation marker – zoomed in b) and c) – provides further information. From c) - e) A note fades out of the workspace, while the annotation marker keeps providing information. a) through e) A pinned note remains visible at all times, regardless of the current frame being displayed.

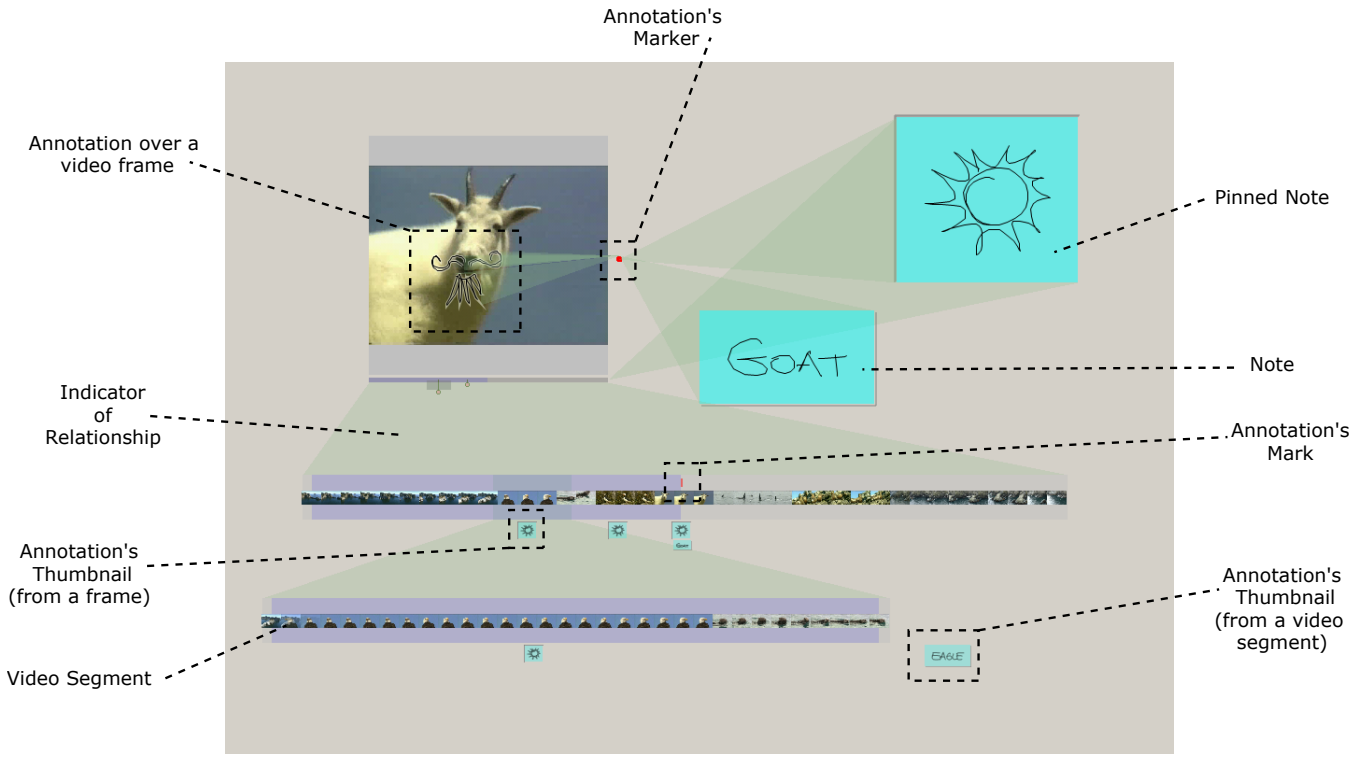


Figure 8: An example of a typical session with *LEAN*. The figure identifies the different elements on the screen

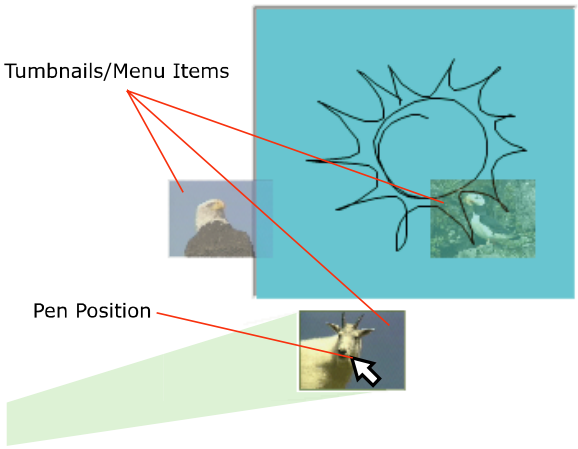


Figure 9: Frames connected to a Note are visualized as thumbnails that can be used as a menu to visit these annotated frames. The thumbnail under the pen is emphasized and an indication of relationship connects it to the point in the video stream where it can be found.

Link Navigation and Manipulation

Our system regards annotations as links between two data objects, links that can be traveled in any direction. If an annotation is visible, a user is able to quickly find the two objects participating in it. In general, and as was described in the case of *Video Segments*, selecting an object on the workspace reveals the object's direct relationships with other entities on the workspace (Figure 8). For example, selecting a visible note reveals the links (annotations) in which the note participates. The user can then tap-and-hold the note to reveal a set of thumbnails that corresponds to the frames to which the note is connected. These thumbnails also function as a menu from which the user can select a frame (i.e., a point in time) to be visited (Figure 9). Users can also grab these thumbnails in order to unlink a note from a frame (deleting the link), or in order to move the link's endpoint to another note.

DISCUSSION and USER FEEDBACK

In developing *LEAN*, we strove to follow a simple set of design rules and interaction principles, including maintaining a minimalist interface without a surfeit of decorative elements, unobtrusive fluid visualizations and interactions, and a small easily understood set of meaningful gestures.

Through our design process, however, we found that tradeoffs between these principles needed to be considered. For example, there is the tension between the desire to have a minimalist interface and the nature of the available input / output devices. When there are no explicit widgets or controls available, an object should provide the affordances that suggest how it should be operated upon. In the physical world, people can use sight and touch to quickly scan for an object's affordances. However, with objects behind the glass of a computer screen this task is not so easily accomplished. Hence the use (and misuse) of controls and

decorations in many graphical user interfaces. We believe that the techniques demonstrated in *LEAN* have provided examples of how to achieve such minimalist interfaces.

Six users have informally tried *LEAN* on a desktop platform. After a 5-minute guided tour of the system, they were asked to explore the system freely and were encouraged to engage in tasks that involved navigating and annotating a video clip. Only some of these users had previous experience with pressure sensitive digitizer tablets, and all of them considered themselves novice or inexperienced users of video editing systems. Although not a formal study, observing these users provided us with the opportunity to gather valuable feedback that helped us to fine-tune the interaction techniques presented in this paper. Our observations can be summarized as follows:

Pressure Control: When using the *TLslider*, people initially exhibited difficulty in controlling the amount of pressure they were applying with the pen. However, we also observed that after a few minutes of practice, they became aware of the consequences of varying levels of pressure and then developed better pressure control. Users also consistently reported that the pressure widgets provided useful feedback when they were using the pen.

Tap-And-Hold Gesture: Users' responses to the TAH gesture were mixed. While some were comfortable with a delay of *3/4* of a second, others found this waiting time excessive and preferred a *1/2* second delay instead. This last group made frequent use of the navigation controls and found it unacceptable to have to wait for their operation to be started. Regardless of their timing preferences, all users found the animated feedback provided while performing the TAH gesture useful.

Mode Errors: It was common for users to try to use the *PVslider* directly, without first making a TAH gesture. This behavior revealed a mode error in which users scribbled on top of the video frame instead of moving through its timeline. In a sense, this observation helps to demonstrate that the *PVslider* provides an intuitive and useful media control that users liked. On the other hand our observations may indicate that users did not perceive the gesture as a whole, but rather as two separate phrases [2]. Buxton's work on 'chunking and phrasing' [2] leads us to believe that it could be possible to abandon the TAH gesture in favor of one that leverages the user's kinesthetic tension (i.e., the pen's pressure) instead of time. By doing so, we can create a continuous 'statement' that combines the invocation and use of a control that itself incorporates both kinesthetic (pressure) and visual (rubber-band) tension [12].

Unforeseen Functionality: After 15 minutes of use, all users easily became familiar with the features of the *LEAN* system, and even used it in ways that we had not previously anticipated. For example, one person started using the system as if it were a story-boarding authoring tool by making notes appear and disappear while a video was played. Furthermore, this user seemed more interested in the dynamic nature of the notes, than in the contents of the video. In general, users during their first session were able to create what can be best described as 'pop-up videos' with surprising ease.

CONCLUSION and FUTURE RESEARCH

We have demonstrated both a system and a set of novel interaction techniques for the fluid navigation, segmentation, and annotation of digital video. Preliminary user observations indicate that the ability to freely annotate and link items in a workspace can be advantageous. However, our work has only begun to scratch the surface of our broader research agenda to create computational workspaces that enable the seamless annotation, linking, and manipulation of a variety of data types. We also note that some of the interaction techniques we have demonstrated, such as pressure widgets and the *TLslider*, can be more broadly applied to any application that uses pressure sensitive digitizing tablets. However, it is also clear that our ideas will need to be validated by extensive user observations. In addition to having users in the field actually utilize the system in a holistic way in their actual video processing tasks, we also intend to perform formal studies in order to evaluate the different interaction techniques contained in LEAN. We want to see if these present a significant improvement over traditional methods of video navigation, control and annotation. Also, in future implementations of LEAN we plan to incorporate scribble recognition techniques, like the ones encountered in SATIN [7] and the TabletPC SDK. Such a feature will allow both data and annotations in the system to be efficiently indexed and searched. We also intend to expand the vocabulary of possible annotations, by allowing in the workspace other types of data such as voice and text, and by allowing links between any two objects. This is unlike our current prototype, which at present only lets users connect a note with a frame or a segment.

At this point our system only handles videos in the order of a few minutes in length. It is not hard to imagine that the workspace in a system such as *LEAN*'s may become over populated with annotations that were made over a long video stream. Because of this, it still remains to be studied how the visualization and interaction techniques we presented in this paper scale in the presence of both a large number of annotations and *Video Segments*.

ACKNOWLEDGMENTS

We thank Microsoft Research for financial support; members of our Dynamic Graphics Project lab for their support and ideas; Garrick Filewood, Lila Pine, and Alex Bal from Ryerson University for valuable discussions on practices in the film industry; Kay Armatage, Charlie Kail, and Lisa Steele from the Cinema and Visual Arts Studies programs at the University of Toronto for discussions on practices in the analyses and critique of film.

VIDEO

A video demonstrating this system can be downloaded from www.dgp.toronto.edu/research/videointeraction

REFERENCES

1. Bier, E., Stone, M., Pier, K., Buxton, W., & DeRose, T. (1993). Toolglass and Magic Lenses: The see-through interface. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. p. 73-80.

2. Buxton, W., Chuncking and phrasing and the design of human-computer dialogues, in *Readings in human-computer interaction: Towards the year 2000*, R. Baecker, *et al.*, Editors. 1986, Morgan Kaufmann: San Fransisco, CA. p. 494-499.

3. Gross, M.D., & Do, E.Y.-L. (1996). Ambiguous intentions: a paper-like interface for creative design. *ACM UIST Symposium on User Interface Software and Technology*. p. 183-192.

4. Guimbretière, F., Stone, M., & Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. *ACM UIST Symposium on User Interface Software and Technology*. p. 21-30.

5. Guimbretière, F., & Winograd, T. (2000). FlowMenus: combining command, text, and data entry. *ACM UIST Symposium on User Interface Software and Technology*. p. 213-216.

6. Harrison, B.L., & Baecker, R.M. (1992). Designing video annotation and analysis systems. *Graphics Interface*. p. 157-166.

7. Hong, J.I., & Landay, J.A. (2000). SATIN: a toolkit for informal ink-based applications. *ACM UIST Symposium on User Interface Software and Technology*. p. 63-72.

8. Kurtenbach, G., & Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. *ACM CHI Conference on Human Factors in Computing Systems*. p. 35-42.

9. Lin, J., Newman, M.W., Hong, J.I., & Landay, J.A. (2000). DENIM: finding a tighter fit between tools and practice for web site design. *ACM CHI Conference on Human Factors in Computing Systems*. p. 510-517.

10. Marshall, C.C. (1997). Annotation: from paper books to the digital library. *ACM International Conference on Digital Libraries*. p. 131-140.

11. McGuffin, M., & Balakrishnan, R. (2002). Acquisition of expanding targets. *ACM CHI Conference on Human Factors in Computing Systems*. p. 57-64.

12. McGuffin, M., Burtnyk, N., & Kurtenbach, G. (2002). FaST Sliders: Integrating marking menus and the adjustment of continuous values. *Graphics Interface*. p. 35-42.

13. Myers, B.A., Casares, J.P., Stevens, S., Dabbish, L., Yocum, D., & Corbett, A. (2001). A multi-view intelligent editor for digital video libraries. *ACM/IEEE-CS Joint Conference on Digital Libraries*. p. 106-115.

14. Mynatt, E., Igarashi, T., Edwards, W., & LaMarca, A. (1999). Flatland: New dimensions in office whiteboards. *ACM CHI Conference on Human Factors in Computing Systems*. p. 346-353.

15. Pederson, E., McCall, K., Moran, T., & Halasz, F. (1993). Tivoli: An electronic whiteboard for informal workgroup meetings. *ACM CHI Conference on Human Factors in Computing Systems*. p. 391-398.

16. Pook, S., Lecolinet, E., Vaysseix, G., & Barillot, E. (2000). Control menus: Execution and control in a single interactor. *ACM CHI Conference on Human Factors in Computing Systems (Extended Abstracts)*. p. 263-264.

17. Rubine, D. (1991). Specifying gestures by example. *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. p. 329-337.

18. Sawhney, N., Balcom, D., & Smith, I. (1996). HyperCafe: narrative and aesthetic properties of hypervideo. *ACM Hypertext Conference*. p. 1-10.

19. Schilit, B.N., Golovchinsky, G., & Price, M.N. (1998). Beyond paper: supporting active reading with free form digital ink annotations. *ACM CHI Conference on Human Factors in Computing Systems*. p. 249-256.

20. Shneiderman, B., & Kang, H. (2000). Direct Annotation: A drag-and-drop strategy for labeling photos. *IEEE Conference on Information Visualization*. p. 88-98.

21. Snibbe, S.S., MacLean, K.E., Shaw, R., Roderick, J., Verplank, W., & Scheeff, M. (2001). Haptic techniques for media control. *ACM UIST Symposium on User Interface Software and Technology*. p. 199-208.

22. Tsang, M., Fitzmaurice, G.W., Kurtenbach, G., Khan, A., & Buxton, B. (2002). Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *ACM UIST Symposium on User Interface Software and Technology*. p. 111-120.