Direct Space-Time Trajectory Control for Visual Media Editing

Stephanie Santosa   **Fanny Chevalier**   **Ravin Balakrishnan**   **Karan Singh**

Department of Computer Science
University of Toronto
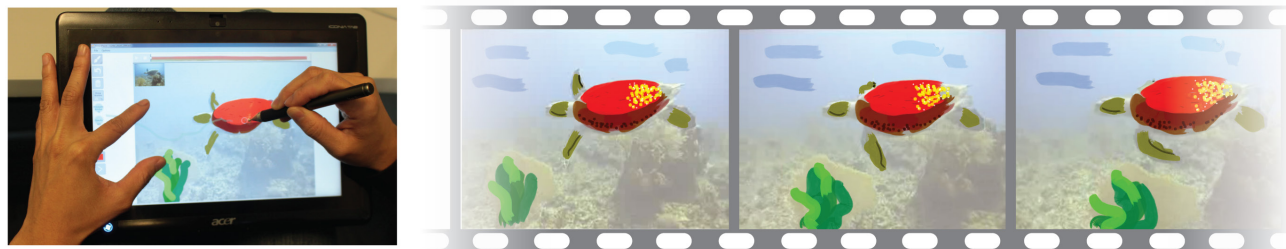{ssantosa, fanny, ravin, karan}@dgp.toronto.edu

Figure 1: DirectPaint implements pen-based interaction techniques for the fluid creation of free-hand painting in time and space.

ABSTRACT

We explore the design space for using object motion trajectories to create and edit visual elements in various media across space and time. We introduce a suite of pen-based techniques that facilitate fluid stylization, annotation and editing of space-time content such as video, slide presentations and 2D animation, utilizing pressure and multi-touch input. We implemented and evaluated these techniques in DirectPaint, a system for creating free-hand painting and annotation over video.

Author Keywords

Sketching; Pen-based interface; Pressure; Optical flow; Video navigation; Bimodal.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

INTRODUCTION

Around two decades ago Haeberli [9] demonstrated "Paint by numbers", a simple yet inexorably compelling interface that enabled the creation of artistic abstraction by painting over a base image. While static image tracing in itself is not a strictly digital affordance, having been used extensively by artists such as Andy Warhol, adding a temporal dimension poses unique challenges. However, this is not insurmountable, as animators commonly leverage live action video as a direct source or a guide to trace or paint over, albeit frame by frame. This technique, *rotoscoping*, intro-

duced in 1917 by Fleischer to create scenes in Betty Boop and Popeye animations, has been widely used since. However, sketching over video is not only a concern for professional animators. With the ubiquity of video capture in mobile devices, everyday users could benefit from easy space-time interactions for drawing, annotating, and adding filters to video.

This extension of visual composition into the temporal domain creates a challenging interaction problem. There is a big divide between spatial interaction that exploits the direct fluidity of sketch-based interfaces such as "Paint by numbers" [9], and temporal interaction that employs the cumbersome and indirect method of key-framing various visual attributes [12,20,25]. Existing tools supporting interpolated rotoscoping (e.g. Adobe After Effects [1] and Rotoshop [7]) are hard to master due to complex and diverse mechanics of space-time interaction.

In a similar vein, this difficulty in consolidating spatial and temporal control exists in vector graphics programs such as PowerPoint [18] or Flash [2] for creating animated presentations or cartoon animations. While they have varying levels of complexity in the spatial domain, the temporal workflow is similarly inadequately supported.

In current systems across these domains, the interaction for controlling the temporal dimension can be demanding because users need to be able to perceive and control visual behaviour across the timeline. In existing rotoscoping, video editing, or vector animation tools, this aspect of temporal control is manual and performed separately from the painting process. Because painting in space and navigating in time are disconnected, the interaction workflow requires a frequent mental shift between the two dimensions and does not aid the user in working with a single imagined spatio-temporal entity.
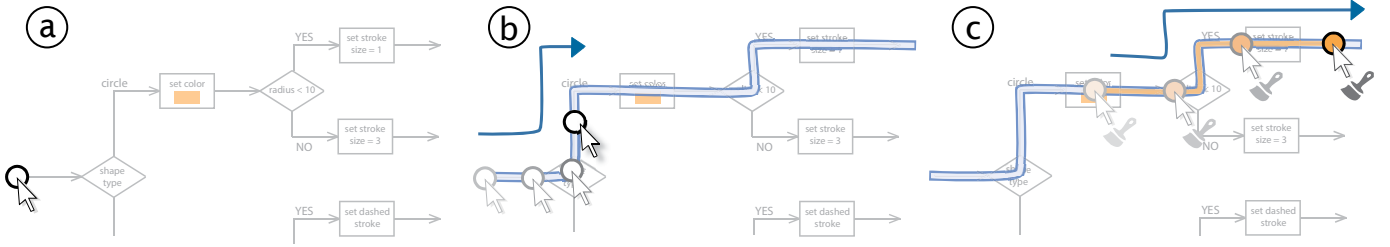
Figure 2: Example of attribute propagation in a slide presentation animation. Motion trajectories can be invoked for any animated object (a). The user can drag the object along to navigate through the animation (b), and apply and propagate attributes by directly specifying the propagation window along the object trajectory (c).
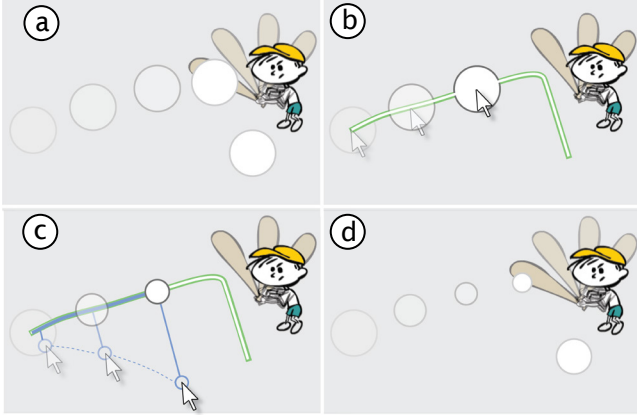

Figure 3: Size adjustment of an object in a vector animation. In the initial animation (a), the ball trajectory can be used for time navigation (b). While navigating, the size can be dynamically adjusted by moving further away from the trajectory (c) to create a more realistic animation (d).

Motion trajectories have been successfully applied to video interactions for supporting direct-manipulation of video objects in navigation and for automating temporal aspects of annotation to provide simple interactions for creating them [6,8,13,15]. We extend this idea and explore trajectory-based interactions for temporal propagation of visual elements or actions. Figure 2 and Figure 3 illustrate examples of these controls.

We aim to reduce the burden of indirect keyframing required in current video and animation systems by bringing together space and time manipulation in a unified and coherent form. We first examine the design space for trajectory-based time control, then we introduce and evaluate 3 pen-based techniques spanning this space. We focus on pen-based interactions and investigate whether pressure information or bimanual input can be used to improve fluidity in the workflow.

As a platform for evaluating our techniques, we developed DirectPaint, an interactive system for free-hand painting and annotating over-top video. Designed for pen-based input – arguably the most appropriate input mechanism for drawing and painting applications, DirectPaint combines computer vision, interaction and visualization techniques for painting in both space and time (Figure 1).

Our contributions are the following: 1) introduction of a design space for trajectory-based space-time control; 2) design and implementation of a suite of novel pen interaction techniques for controlling the propagation of visual elements using feature-based optical flow; 3) implementation of a system introducing visualization and interaction approaches using our techniques to support rich time and space manipulation, and 4) usability evaluation of our techniques on various scenarios, including free-form painting using the whole system.

BACKGROUND
We consider prior work in the areas of temporal editing in animation, interaction techniques for video navigation and annotation, and optical flow-based video stylization.

Temporal editing of Graphical Objects
Editing visual attributes such as the colour, size, visibility, and opacity of graphical objects over time is a key task in creating digital animations. In professional animation tools like Flash [2], After Effects [1] or Maya [3], this temporal control typically relies on keyframing and interpolation. These tools also generally support scripts to automate property changes over time. A similar approach to scripts are found in tools such as PowerPoint [18], where simple animations are defined through the manual editing of timing parameters, including the type and duration of animation effects or triggering events.

Interpolation and scripting techniques have dramatically facilitated the creation of digital animations by alleviating the need for tracing each frame independently. Commercial software has been focused on empowering existing techniques with richer control of interpolation parameters. The underlying workflow for temporal control, however, has received little attention, and remains typically disjoint from the drawing process. We thus explore new pen-based interaction techniques for the temporal control of visual properties of graphical objects through direct manipulation of time on object trajectories.

Previous work has been done to define and re-time such object motion trajectories. GENESYS introduced the idea of sketch-based techniques to define motion trajectories for animation [4]. Davis et al.'s K-Sketch system records object dragging operations layered over the running animation to define and modulate motion trajectories [5]. More recently, Dragimation utilizes direct manipulation of objects
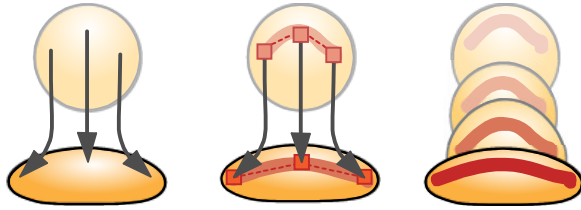
Figure 4: Pixel motion trajectories across frames (left) allow for the tracking of strokes' control points (middle) that we use to propagate and deform strokes over time (right).

to re-time their trajectories [24]. In these systems, new or revised motion trajectories themselves are the artefacts resulting from the interactions. Our focus is not to modify trajectories – rather, we utilize them as a pre-defined basis in our controls for editing visual properties.

Video Navigation and Interaction

Pen-based interactions for video control and annotation are explored in LEAN [21], where pressure widgets provide interactions using pen pressure as a continuous or discrete control. The use of pressure to input continuous values or act as a switch has been analyzed empirically [16,22]. We apply these ideas and findings in our treatment of pressure. LEAN also introduces a position-velocity time slider, where navigation speed is adjusted by dragging orthogonal to slider orientation. We use a similar affordance to control attributes simultaneously with time navigation.

The increased availability of pen and touch hardware has yielded explorations that combine pen and touch to create new input vocabularies [11] and applications [26], where the natural tendency of using a pen to write and touch to manipulate were identified. We explore this idea further with the use of touch gestures in our bimanual time propagation techniques.

Other work focuses on developing more direct methods for video navigation using optical flow algorithms. DimP [6], Trailblazing [15] and DRAGON [13] introduced direct manipulation for video navigation. Goldman et al. also apply this technique for navigation in a system that additionally supports annotation performed directly on video objects [8]. In these systems, video object trajectories are used as virtual sliders in navigation. We build upon the idea of direct object interaction to explore fluid space-time creation and editing of graphical objects.

While prior work has demonstrated the creation of spatial annotations that automatically propagate in time [8], we provide insights into the suitability of motion trajectory-based techniques to support the integrated control and editing of content in both space and time.

Optical Flow and Stroke Propagation

Previous research has produced several systems for creating stylized animations by automated stroke synthesis, sometimes guided by user input [10,12,17,20]. Notable systems include Kayaga et al.'s work [12], Anipaint [20], and Video Tooning [25]. We build on the ideas presented by these works in our system implementation and utilize optical flow to determine the spatial deformation of painted

strokes over time. While these systems produce high quality image abstractions, they do not address our problem of fluid interaction for control of time and space.

The SIFT-based computer vision algorithm applied in DimP [6,19] forms the basis for stroke point propagation we utilize in the context of painting over top of a video. We make use of feature trajectories computed and used by DimP to power navigation and to determine stroke point positions over time (Figure 4).

DESIGN SPACE

At the core of this work, we focus on objects for which motion trajectory information is available (either user-defined, or computed through computer vision). We refer to tasks conducted in the spatial domain of a single frame, such as drawing a stroke or editing a visual attribute, as spatial manipulation. Conversely, we refer to tasks related to controlling where in time to affect them as temporal propagation. We identify three dimensions in the space-time control design space as aspects that characterize interactions for the trajectory-based propagation of spatial manipulations over time: *space-time workflow, time navigation*, and the *propagation window*. Columns in Figure 5 illustrate these parameters. We also consider how trajectories can be leveraged in interactions by specifying dynamic property adjustments along them.

Space-time workflow

When editing visual attributes over time, a first design consideration is whether temporal control is performed in a *simultaneous* or *sequential* workflow to spatial editing. Simultaneous drawing/editing and time propagation requires the use of a separate input modality – in addition to the spatial input – in order to manipulate both the temporal and visual parameters in parallel. Conversely, for the techniques that are based on sequential handling of space then time, spatial input can be leveraged for temporal propagation (e.g., specifying locations along the motion trajectory). While simultaneous space and time editing can be more challenging than sequential editing for accurate time propagation, its single-step interaction can be less disruptive of the task flow. In addition, it offers the ability to specify the propagation at a finer level of detail, as control points of a graphical object can each simultaneously receive temporal parameters on creation. This can yield richer space-time drawing effects in a single, fluid interaction.

Time navigation

A second design consideration is what spatial information is presented while controlling the time parameter. The interaction view may remain on the current frame or combine such control with time navigation. From a usage standpoint, both approaches have their pros and cons. The former is adapted to holistic drawing on a single frame, but requires feedback for temporal propagation (e.g., a preview window, a timeline), while keeping the current frame in focus. Conversely, moving through time allows for a more direct appreciation of the effect of the propagation over the different frames. However, it requires additional operation to return to the initial frame for further editing.

	SPACE-TIME WORKFLOW		TIME NAVIGATION		PROPAGATION WINDOW	
	SIMULTANEOUS	SEQUENTIAL	NONE	TRAVEL	CONTINUOUS	FRAGMENTED
WINDOW	✓		✓		✓	
TRAVEL		✓		✓	✓	
TRACE		✓	✓			✓

Figure 5: Classification of the Window, Travel and Trace approaches in our space-time editing design space.

Propagation window

Our final design consideration relates to the nature of the propagation window – the set of frames across which a property is to be propagated, which can be of two types. We distinguish between a continuous window, typically starting from the current frame and expanding either forward or backward in time; and a fragmented window comprising of frames (both backward or forward in time) that are not necessarily contiguous. While continuous forward propagation is most common, a discontinuous window supports more flexible control, which can be useful in creating special effects, such as blinking.

Dynamic editing

Two different approaches can be considered for editing visual attributes of graphical objects over time. A first approach consists of the *dynamic editing* of an attribute value while navigating in time. This can be performed by simultaneously manipulating time and property value, registering frame-value pairs while navigating over a trajectory. A second approach is *property propagating*, where the set of frames is specified, over which an attribute is to be applied.

INPUT MODALITIES FOR TEMPORAL CONTROL

In this work, we focus on pen-based interaction with tablets, a common and highly appropriate input device for digital drawing and painting. The main challenge in supporting propagation over time lies in reducing the amount of user input required to specify the temporal parameters while providing the user with a sufficient amount of control to create coherent animation.

Pressure capabilities of digital pens allow for an integrated control of an additional dimension alongside the 2D spatial position of the pen. Moreover, when working on a pen-based interface, the dominant hand is busy holding the pen and tracing, whereas the non-dominant hand is typically free for interaction.

We explore two different input modalities to support the propagation in time with minimal interruption of the drawing flow: *pressure*, and *bimanual touch + pen*. While other approaches can be considered (e.g. keyboard or mouse input), we focus on pen pressure and bimanual input as both modalities allow for all interaction to be performed directly on the tablet without the requirement and disruption of external devices or widgets.

Figure 6: Detail of pressure-based and bimanual interactions for the different propagation approaches.

PROPERTY PROPAGATION TECHNIQUES

To facilitate our exploration of the design space described above, we drill-down on the problem of drawing or painting strokes in time. We design a set of novel pen-based interaction techniques to specify the temporal propagation of each stroke. We introduce three main interaction approaches, that we adapt for both pressure and bimanual touch input modalities. Figures 5 and 6 show an overview of the different techniques, and their characterization in the design space respectively.

It is worth noting that while all combinations of the parameters described are possible, some permutations are arguably less applicable than others. For example, a simultaneous space-time workflow over fragmented propagation window intervals would require overly complex interaction mechanisms to be supported. As a first step towards investigating the design space, we focus on what we believe to be the three major permutations of the parameters for *property propagating*, and describe the corresponding interaction techniques we design for it in each pen modality. We later also introduce a technique for *dynamic editing* of a property over time.

WINDOW: Adjust the Propagation Window Size

Supporting the input of temporal information at the same time as drawing objects in space is arguably the least interruptive to the creation workflow. However, in order to keep simultaneous space and time interaction cognitively and practically manageable, control of the temporal dimension must remain simple. Here, we consider the input of an extra value during creation to define the propagation window through which the visibility of the stroke is propagated forwards. The fill in the stroke trajectory is representative of the propagation window to help visualize the propagation with respect to the object itself.
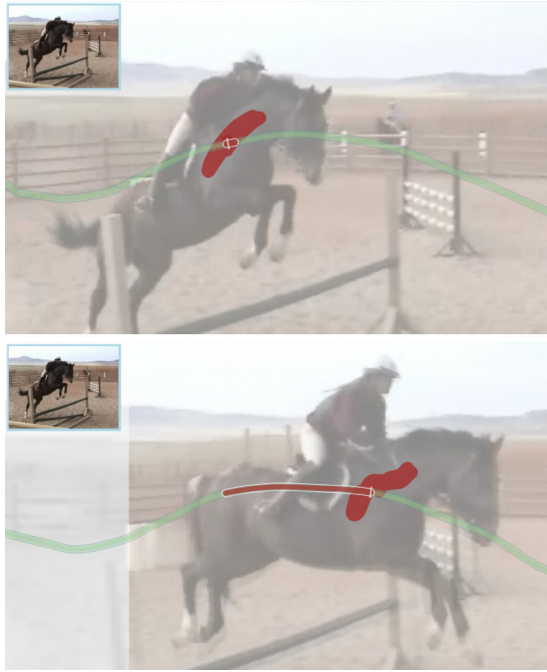
Figure 7: Stroke propagation with the TRAVEL technique.

Pressure. We propose mapping pressure to temporal control, wherein the more pressure applied to the stroke point, the longer it persists in the video. Here, we draw inspiration from an analogy of drawing on a stack of napkins with a pen which causes the stroke to penetrate across multiple sheets based on pressure. The propagation control being directly on the pen, it is seamlessly integrated with the main action of drawing and thus yields a fluid, nondisruptive interaction. However, this approach is sensitive to the precision with which the user is able to control pressure on the device. Moreover, using pressure to control a slider is not novel, and drawing applications usually exploit such a dimension to control stroke width, the alpha channel or other visual attributes of strokes. Therefore, window pressure may happen to be in conflict with such controls.

Bimanual. We propose leveraging the non-dominant hand to control stroke propagation in the temporal dimension with a pinch gesture: the greater the distance between the fingers, the longer the propagation in time. The bimanual method has the advantage of allowing the specification of precise time intervals over which the active stroke is to be propagated, whereas pressure can be difficult to maintain at a constant value. However, bimanual interaction may cause more of a split attention effect, whereas pressure control offers an in-place integrated contextualized interaction.

TRAVEL: Navigate to Propagate
While controlling the propagation window simultaneously with drawing has unique advantages in terms of workflow, it does not allow for visualizing the stroke's appearance over time. In this second approach, we consider the effect of the time navigation parameter in our design space in a sequential space-time workflow. After painting a stroke in space, trajectory-based navigation is applied – propagating

and previewing the resulting stroke across other frames. When TRAVEL is invoked, the object trajectory appears. In addition to providing contextual feedback, the trajectory is used as a control in this technique, and the user can trace over the path to navigate in time in a similar fashion as in DimP [6]. Upon release, the stroke is propagated from the frame it was created, forwards or backwards to the current frame. Figure 7 illustrates painting in this mode.

Pressure. We propose the use of a pressure threshold to switch between painting and navigating. When under the threshold, the user can paint and draw the stroke in the current frame. When the pressure is increased above a fixed threshold, the object trajectory appears and the user can then navigate-and-propagate.

Bimanual. We propose the use of a simple touch on the canvas as a delimiter between painting and propagating. While the touch is maintained, the object trajectory is visible and navigable. Upon release, the stroke is propagated, and the user can paint on the end frame.

The TRAVEL approach has certain advantages over the previous method, in that the user can track stroke deformations over the trajectory while adjusting the propagation window, as well as increased accuracy in specifying the exact frame to propagate to. However, it requires an explicit triggering of the propagation function – as opposed to a continuous control, resulting in a slower interaction time.

TRACE: Trace to Propagate
In both previous methods, propagation is constrained to a continuous window. To create special effects, or to repeat a stroke at particular points in time, it may be useful to support propagating a stroke across a disjoint set of frames. In this third approach, we consider propagation across a fragmented window by directly tracing on the trajectory, the portions where the stroke is to be visible. As in the TRAVEL approach, the user can switch from painting to propagation over the trajectory with TRACE. Upon this switch, the object trajectory appears, and the user can trace over top of it to define the fragmented propagation window.

Pressure. We propose a similar technique as TRAVEL, in that a pressure threshold is used as a delimiter to invoke the trajectory for tracing. A subsequent input point outside of a distance tolerance from the trajectory will remove it from view and complete the propagation so the next stroke can be drawn.

Bimanual. We propose a similar technique as TRAVEL in that a simple touch on the canvas reveals the object trajectory for TRACE. As with the previous technique, the stroke is completed when the touch is released.

DYNAMIC EDITING TECHNIQUES
There are a number of possible solutions for trajectory-based dynamic editing of visual attribute values. Different combinations of inputs from our proposed techniques can be applied to adjust and propagate an extra parameter value. For example, trace to propagate can combine pinching gestures to support additional input.
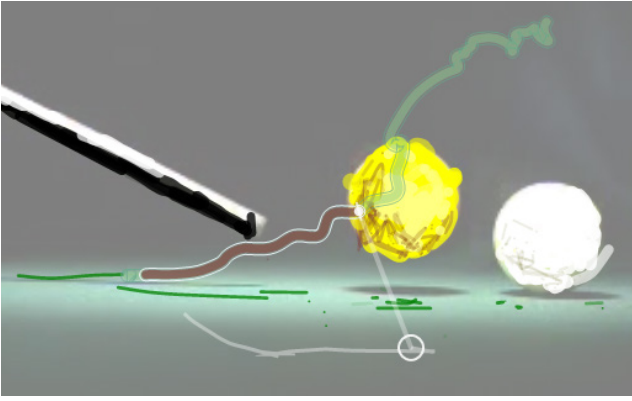
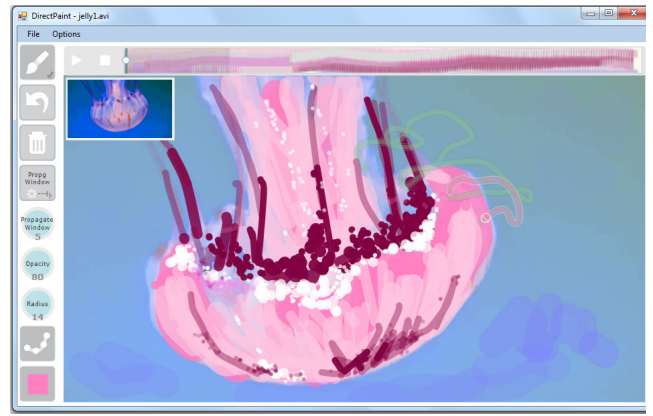**Figure 8: Orthogonal trajectory controller for dynamic editing of stroke opacity during propagation.**

Figure 9: DirectPaint interface. The timeline is augmented with a colour script, summarizing paint distribution in each frame. Input stroke propagation is shown on the timeline and trajectory; the thumbnail indicates its end frame.

In our system, we extend **TRAVEL** to support dynamic editing: the user can modify either the opacity or the positional offsets from the trajectory by using an orthogonal offset control, where the orthogonal distance to the trajectory parameterizes the attribute over time. With the opacity control, moving further away from the trajectory would cause the stroke to be faded out, whereas moving closer to it would cause it to be more opaque. The controller provides feedback showing its history trail. Figure 8 shows an example of opacity control on a stroke.

The ability to control two separate aspects of the stroke simultaneously and preview the effects in place during navigation is advantageous, however, there are limitations based on the trajectory shape, as with the **TRAVEL** and **TRACE** propagation techniques. Additionally, users can inadvertently scrub the video with movement that is not orthogonal to the trajectory at the point of intersection.

VISUALIZATIONS FOR USER FEEDBACK
We combine different visual indicators to provide feedback and support the propagation techniques: the visual trajectories (Figures 7-8), a thumbnail preview (Figure 7) and a colour script timeline (Figure 9).

Visual Trajectories
Motion trajectories are visualized and augmented to provide the user with in-place visual feedback on the stroke propagation window (see Figures 7-8). As the user interacts with the propagation control, the corresponding portion of the trajectory is dynamically emphasized with the colour of the active stroke so that the results of the current action are presented in context rather than only being shown on a separate widget such as the timeline slider. The latter though, may be relevant for some tasks, and we propose to augment it with colour script, as described later.

Thumbnail Preview
A thumbnail preview is given at the corner of the canvas to provide the user with a visual indicator of the edge frame of the stroke propagation window (see Figure 7). In the **WINDOW** and **TRACE** approaches, this thumbnail contains the image of the current edge frame in the adjustment. Conversely, for the **TRAVEL** approach, the canvas displays

the active edge of the propagation window, so the thumbnail displays the initial frame to provide the user with a reminder of the starting image.

Colour Script Timelines
We apply the idea of colour scripts used in animation to produce a representation of the colours used in the painted strokes over time. The timeline is augmented with a 'colour script' based on the painted strokes in the video, as showed in Figure 9. This provides the user with information on the strokes and paint density in each frame. The implementation here is a coarse representation where each frame is projected to a vertical strip, in a similar way as History-Flow [23], with each stroke indicated as a blob at the average vertical position of its points.

This script is dynamically updated as each stroke is drawn. Upon selection, all unselected strokes are faded out to emphasize the selection. During painting, the propagation window is also highlighted on the timeline. This provides the user with feedback on the new stroke's propagation relative to the others. This is particularly useful in cases when the stroke trajectory is not visually clear.

DIRECTPAINT SYSTEM
As a proof of concept, we implemented DirectPaint, a full system integrating the propagation and visualization techniques introduced above. DirectPaint applies trajectory-based direct manipulation and can be used for video scrubbing as in prior systems [6,8,13,15]. When painting, the user creates strokes and propagates them through time using one of the proposed interaction methods. DirectPaint supports several additional features, such as opacity, colour, stroke width, stroke style, and textured brush rendering in addition to flat line drawing to allow for rich artistic expression. Figure 9 shows the graphical user interface of the DirectPaint prototype.

DirectPaint also supports selection of a stroke or group of strokes on the canvas and the propagation window can then be re-adjusted for the entire group in using the **TRAVEL** approach, with the orthogonal control also available.

EVALUATION

We evaluate the suite of temporal propagation techniques discussed to address one important question: does the integration of temporal propagation into the spatial drawing workflow presented in DirectPaint come at the cost of ease or accuracy in the painting process? Specifically, we compare the **WINDOW**, **TRAVEL**, and **TRACE** techniques that we implemented for the **PRESSURE** and **BIMANUAL** input modalities on a set of various task scenarios.

Video painting with DirectPaint raises many other empirical questions. These include questions about the mental shift between space and time dimensions and its effects on user perception of a single imagined spatio-temporal entity; as well as questions on interaction design, such as the optimal time navigation widgets and visual feedback for space-time manipulations, and their effect on user motor planning. These questions are beyond the scope of this paper, but would benefit from future exploration.

Experimental design and procedure

We control two independent variables in our study: the input modality (**PRESSURE**, **BIMANUAL**) and the interaction technique (**WINDOW**, **TRAVEL**, **TRACE**). Half of the participants were evaluated under the **PRESSURE** condition, while the other half were in the **BIMANUAL** condition. Participants of each group performed the same set of tasks using all three interaction techniques for the input modality. The order of presenting the techniques to participants was counter-balanced across participants in the same group according to a Latin square.

Tasks

We designed a series of tasks with the goal of covering diverse painting scenarios and objectives. Table 1 provides a summary of the five tasks, differing in terms of trajectory quality and propagation constraints that require varying degrees of accuracy (e.g., tasks 1-3 involve event-based temporal constraints). We also included painting on temporarily stationary objects, for which the trajectory circles in place (3). Some tasks were well matched with a fragmented window propagation approach (1-2), while others were better suited to continuous window propagation (3). Finally, two more loosely structured tasks (4-5) were given to encourage the participants to explore capabilities of the different techniques in a free-form fashion.

Procedure

After filling out a background questionnaire, participants were asked to perform all tasks as described in Table 1, for all 3 techniques in either the **PRESSURE** or **BIMANUAL** condition. For each technique, participants were given instruction followed by a practice session on a short video segment. A one page manual describing the technique usage was also given for reference.

Participants proceeded through all tasks in order for each technique before switching to another. Each task started with the presentation of a sample video demonstrating the expected result. Participants were given paper instructions of the tasks to be completed, indicating the objects to paint, the colours to use, and the delimiting events for time propagation. We instructed participants to focus on temporal accuracy when propagating strokes, rather than spatial accuracy of stroke visuals. No time limit was given—when satisfied with their result, participants notified the instructor to start on the next task. For each task, we logged stroke creation and deletion counts, and the completion time and propagation window of each stroke.

Participants were then invited to create an artistic painting of their own, on the video segment they used for practicing. Participants were allowed to use the techniques of their choice, and all the functionalities of the user interface. No time limit was given. Overall, participants took 10 to 40 minutes to create their piece. Finally, participants were asked to fill out a post-experiment form asking for qualitative feedback. The total experiment took about 90 minutes.

Apparatus

For the **PRESSURE** condition, we used a Toshiba Portégé M700 tablet PC using a 2.20GHz Intel Core2 Duo CPU, and at screen resolution of 1280x800 pixels. Single point and pressure input was provided by a built in touch screen and a pressure sensitive Wacom digitizer supporting 255 pressure levels. For the **BIMANUAL** condition, we used a multi-touch Acer tablet using a 1.00 GHz AMD C-50 dual core CPU, and at a screen resolution of 1280x800 pixels. The pen and touch input was provided via a capacitive multi-touch display supporting 4 touch points.

Participants

Twelve unpaid participants (3 females), aged 24-34 (mean 30) took part in the study. Four of the participants had

Sequence	Id	Task	Strokes	Window type	Trajectories
Spot the thief	1	Give the thief a blinking highlight (3 times) in red as soon as he walks to the car.	1	Fragmented	Good
Pool: Ball energy transfer	2	Paint the ball with a yellow colour from the beginning until it get hits by the cue, and after it hits the white ball.	3 horizontal	Fragmented	Good
		Emphasize the ball with a purple colour, while it holds the main action.	3 vertical	Continuous	
	3	Paint the ball with a while colour until it gets hit by the yellow ball.	3 horizontal	Continuous	Stationary
		Emphasize the ball with a purple colour, while it holds the main action.	3 vertical	Continuous	
Sunset: The whimsical sunset	4	Paint the sun with several strokes of varying lengths of time and colour, with a general tone gradually changing from yellow to orange to more red tones.	Free form	Free form	Fairly good
	5	Create a "feathery" effect in the sky by painting several strokes of varying size and lengths of time, at various points in the video.	Free form	Free form	Inconsistent

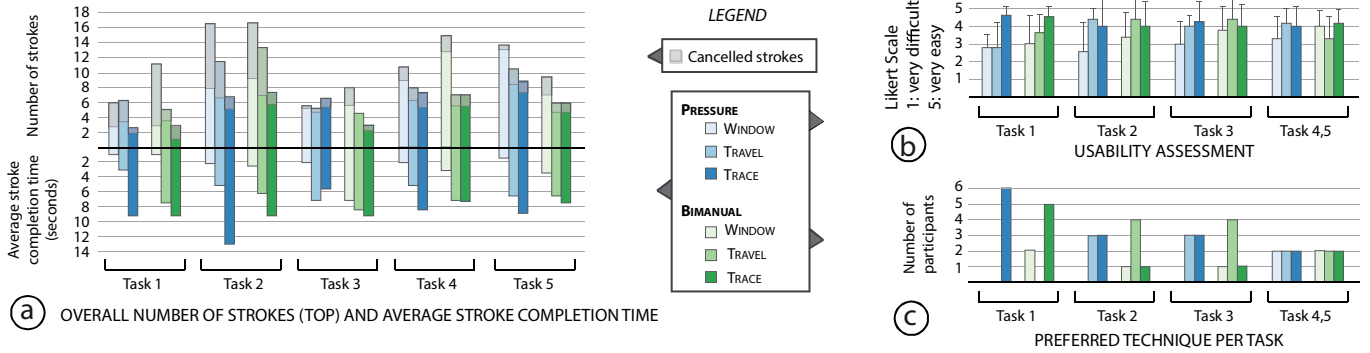**Table 1: Description of the tasks used in our experiment.**

Figure 10: Summary of the results of our study. (a) The average stroke completion time and average number of strokes based on log data for each technique and input mode, grouped by task. (b) The perceived difficulty of each technique and input mode for the tasks. (c) The number of times a technique was reported as the preferred choice for each task and input mode.

good experience in digital drawing and computer animation, while eight had no or very little experience. Ten of the participants rarely or never used pen input tablets. Only one participant of the pressure condition group used pen pressure input on a regular basis, the five other participants in that group never or rarely used pressure-sensitive pen. Four participants in the bimanual condition used multi-touch tablets on a regular basis (iPad, Android), the two others in that group used multi-touch tablets only rarely. All participants were right-handed.

RESULTS

Due to the limited number of participants, we did not collect enough data to perform statistical analysis. We report here on general observations on quantitative measures as well as qualitative feedback from participants. Figure 10 shows a summary of the results.

Completion time and error

Figure 10.a shows the average stroke creation process for each task and condition. The top bars depict the average number of strokes created (full bar) and cancelled (grayed part) in each task. The bottom bars correspond to the average stroke completion time. Our results across techniques under the same conditions suggest an inverse correlation between the stroke completion time and the number of strokes created. Naturally, a shorter completion time encourages the creation of more strokes, as we observe in the free form tasks (4,5). However, this is to the detriment of accuracy, as suggested by the inverse proportion of stroke deletion and completion time in tasks requiring accuracy (1-3). This points to how the varied underlying mechanisms of our techniques trade-off differently between completion time and error tolerance. Overall, the window technique privileges fast, but less accurate creation of strokes, while the trace techniques allow for better accuracy, but longer stroke completion time.

Propagation accuracy

We use the number of blink highlights in Task 1 as an indicator of accuracy, as we observed that several participants failed at making the thief blink three times due to a lack of precise control over the time dimension. All participants successfully completed the task with the **TRACE**

technique. Half of the participants even created extra blinks with this technique. In contrast two participants with the **TRAVEL** technique, two other participants with the **WINDOW** techniques, and one participant with both techniques created only two blinks. Not surprisingly, the **TRACE** technique provided the best results on this task as it allowed for a holistic, all-at-once selection of the fragments of the trajectories where the thief is to be highlighted.

We also measure accuracy in Tasks 2-4 by measuring the average distance between the frame the strokes were propagated to, and what we consider to be the interval of tolerance for success (20 frames around the key event frame). All participants succeeded at propagating strokes accurately with the **TRAVEL** and **TRACE** techniques. However, for **WINDOW**, we observed stroke propagation of up to 21 frames away from the interval of tolerance, i.e., when the key event has clearly not happened, or is over. Participants in the **BIMANUAL** group performed better with the **WINDOW** technique on these tasks (8 frames distance in average) than the **PRESSURE** group (15 frames distance in average).

Participants from both groups were generally confused with the dynamic control of the **WINDOW** technique, and had difficulty understanding the underlying mechanisms. We observed that many typically attempted to control pressure only after finishing tracing the stroke in the **PRESSURE** group for all tasks. The resulting propagated strokes were therefore typically reduced to a single point, corresponding to the last control point to which the propagation through pressure was applied. For the **BIMANUAL** group, the challenge was due to the difficulty assessing the adequate propagation value associated with the pinching gesture. In our implementation, the stroke propagation is triggered at the detection of a pinch gesture, and the first pinch value is applied to all of stroke control points traced until then, with subsequent points applying on new pinch values. Both cases leave little room for error.

Qualitative feedback

Figure 10.b and Figure 10.c depict the average usability assessment measured on a 5-point Likert scale (1: very difficult to 5: very easy to use to complete the task), and preferred technique respectively. Overall, all the techniques

were well received by our participants who indicated that they would use them for painting and annotating in time. We do however, observe variations in the usability assessment across the tasks and techniques.

Most notably, the **WINDOW** technique revealed challenges in tasks requiring accuracy, especially in the **PRESSURE** condition; however, one participant noted it to be "*easier to control when objects were not moving*". Participants found the technique to be "*fast but hard to control*" and "*found the correct pressure difficult to control, offering little room for error*". For the above reasons, four of the participants in the **PRESSURE** group disliked the technique and one strongly disliked it, while only one participant in the **BIMANUAL** group disliked it. In the latter group, one participant reported that "*it was easier to grasp the Travel and Trace techniques. [She] was slightly confused with how the Window worked, but [she] likes how this technique is controlled by the left hand rather than switching control back to the pen like in other techniques.*"

The **TRACE** technique was best received. Six participants liked it, five really liked it, and one had a neutral opinion. Participants found that "*the visual nature of the technique was helpful*", and generally found that it offered "*best control of the timeline.*" Finally, the **TRAVEL** technique was also well received, as it was "*easy to propagate through time to the exact desired moment*". Five liked it, three really liked it, and three were neutral.

These observations are reflected in the varying preferences for techniques across tasks, and support our observations on quantitative results previously described. Overall, the results suggest that all three techniques are complementary, and equally useful for different scenarios.

Visual feedback
We asked our participants about the usefulness of the visual feedback for the different techniques. The most commonly provided answer was that all of the trajectory, thumbnail and colour script timeline were useful after learning how to best take advantage of their complementary functions. Many participants really liked the colour script timeline as a feedback of their edits. Nevertheless, three participants in the **PRESSURE** group found neither one of the visualizations helpful in solving problems posed by the simultaneous space-time control in the **WINDOW** technique.

Time navigation
Participants quickly grasped the power of the time navigation as introduced in DimP [6] as they became familiar with the trajectory-based manipulations. Toward the end of the evaluation, they tended to navigate using the trajectory more systematically, and used the timeline only when the trajectories were not well defined. One participant expressed that "*the trajectory and timeline are complements to each other*". Three participants found navigation through direct manipulation of the trajectory strongly useful, eight found it useful, and one was neutral.

Free form composition
Participants found the free form composition was "*fun and gave the freedom to use the best technique according to where thought it would work best*". DirectPaint as a whole was well understood: "*I developed a good grasp of the basic interface and workflow. I was able to switch colours, navigate through time, and switch propagation modes.*" one says. Participants reported a good overall satisfaction of their own stylized animation in the free form painting.

DISCUSSION
Participants expressed overall satisfaction with using the system and its suite of propagation techniques. Results indicate that overall, they were able to successfully understand the mechanisms involved with each of them. The consistency between trajectory-based interactions across navigation and drawing proved to form an effective direct manipulation system for space-time painting.

The main source of difficulty we observed was in grasping as well as performing the **WINDOW** technique. We also note the limitations of our system particularly due to the fidelity and performance of the computer vision techniques that support it. These limitations are consistent with those identified in other trajectory-based video navigation systems [6,8,13,15]. We note, however, that these are implementation rather than conceptual limitations, and will undoubtedly diminish as vision algorithms improve over time. Another issue inherent in trajectory-based controls is the limitation associated with stationary or ambiguous trajectories. In our system, exact control within a stationary section of the trajectory is not supported, however, solutions that address this limitation have recently been proposed in Draglocks [14].

We observed that the tradeoff between the costs and benefits of the different techniques were fairly balanced across them, indicating that they are complementary to each other. The context has a high influence on the users preference of technique. For high accuracy tasks, the **TRAVEL** technique received the highest rating, while for animations involving some repetition, as well as those demanding a moderate degree of accuracy, the **TRACE** technique was the clear winner as completion time was generally lower than **TRAVEL**. Also, despite the difficulty, participants had with the **WINDOW** technique, several still found it favourable for performing tasks where accuracy was of lesser concern, due to the quick interaction time it offers. It was also valuable for situations where the trajectory was not a clearly defined path.

Visual feedback offered in DirectPaint was highly valuable to users. Manipulating the temporal dimension where only one frame can be viewed at a time creates a need for these types of visual indicators to assist users. As one participant stated: "*When I learned to combine it [the trajectory] with the timeline and thumbnail, I really liked it… when used with the two other navigation methods, it gave a full picture of how I was moving in terms of linear time, visual movement, and placement.*"

The response generated towards the system indicates that casual users were able to quickly learn and perform space-time manipulations without requiring specialized knowledge or animation skills. One participant expressed: "*The techniques and tools were enjoyable and inspirational to use. Moreover, the technique themselves inject new stylistic opportunities based on how the technique works. This contributed largely to my satisfaction.*"

CONCLUSION AND FUTURE WORK

Spatio-temporal manipulations are challenging and tedious for applications such as animating visual attributes, creating annotations on video, and stylizing video, due to the lack of cohesion between the space and time domain interactions. Object motion trajectories offer the potential of improving space-time interactions by providing spatial context to the temporal domain. We introduce the design space for the problem of trajectory-based space-time manipulation, and in our exploration of the area, we design and evaluate 3 different techniques spanning the space, implemented with two different modalities each.

We implemented and evaluated these techniques within DirectPaint, a system targeted to casual users that utilizes computer vision, interaction, and visualization techniques in a novel way to support quick and simple freehand drawing and annotation over video. In our evaluation, we found that our techniques offer promise in forming space-time interactions that are easily accessible to casual users. These techniques operate in a complementary manner, with each offering advantages in different contexts.

There are numerous avenues of future work in this area. The techniques presented here represent an initial step towards trajectory-based space-time manipulations. They can be extended to different aspects of the design space, and applied to other applications, such as integration in vector-based animation systems. Optimizations of the optical flow algorithm can be explored, as well as methods of automatic segmentation of different motion regions. Finally, visualizations present another area for future development, as they were shown to be essential to providing appropriate feedback for challenging temporal tasks. This indicates that augmenting visual feedback further could improve interactions for space-time manipulations.

Our findings indicate that trajectory-based interactions supported by visualization for feedback offer a powerful alternative to the traditional keyframe-based method for propagating spatial manipulations over time. We believe that the design space we present here will help characterize these interactions for further development and eventual integration into applicable tools.

ACKNOWLEDGMENTS

REFERENCES

1. Adobe. After Effects CS6. http://www.adobe.com/products/ aftereffects.html, (2012).
2. Adobe. Flash Professional CS6. http://www.adobe.com/products/ flash.html, (2012).
3. Autodesk. Maya. http://usa.autodesk.com/maya/, (2012).
4. Baecker, R. Picture-driven animation. *AFIPS 1969 (Spring)*, ACM Press (1969), 273-288.
5. Davis, R. C., Colwell, B., and Landay, J. K-sketch: a 'kinetic' sketch pad for novice animators. *CHI 2008*, ACM Press (2008), 413-422.
6. Dragicevic, P., Ramos, G., Bibliowitcz, J., Nowrouzezahrai, D., Balakrishnan, R., and Singh, K. Video browsing by direct manipulation. *CHI 2008*, ACM Press (2008), 237–246.
7. Flat Black Film. Rotoshop. http://www.flatblackfilms.com/ Flat_Black_Films/Rotoshop.html, (2012).
8. Goldman, D. B., Gonterman, C., Curless, B., Salesin, D., and Seitz, S. M. Video object annotation, navigation, and composition. In *IEEE UIST 2008,* ACM Press (2008), 3–12,
9. Haeberli, P. Paint by numbers: abstract image representations. In *ACM SIGGRAPH*, 24(4): (1990), 207–214.
10. Hertzmann, A. and Perlin, K. Painterly Rendering for Video and Interaction. *NPAR 2000*, (2000), 7-12.
11. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., et al. Pen + touch = new tools. *UIST 2010,* ACM Press (2010), 27-36.
12. Kagaya, M., Brendel, W., Qingqing Deng, Kesterson, T., Todorovic, S., Neill, P. J., and Zhang, E. Video Painting with Space-Time-Varying Style Parameters. In *IEEE TVCG*, 17(1): (2011), 74-87.
13. Karrer, T., Weiss, M., Lee, E., and Borchers, J. DRAGON: a direct manipulation interface for frame-accurate in-scene video navigation. *CHI 2008*, ACM Press (2008), 247–250.
14. Karrer, T., Wittenhagen, M., and Borchers, J. DragLocks: handling temporal ambiguities in direct manipulation video navigation. CHI 2012, ACM Press (2012), 623-626.
15. Kimber, D., Dunnigan, T., Girgensohn, A., Shipman, F., Turner, T., and Yang, T. Trailblazing: Video playback control by direct object manipulation. *IEEE ICME 2007*, (2007), 1015-1018.
16. Li, Y., Hinckley, K., Guan, Z., and Landay, J., Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces, *CHI 2005*, ACM Press (2005), 461-470.
17. Litwinowicz, P. Processing Images and Video for an Impressionist Effect. *SIGGRAPH 1997*, ACM Press (1997), 407-414.
18. Microsoft. PowerPoint 2010. http://office.microsoft.com/en-us/ powerpoint/, (2012).
19. Nowozin, S. autopano-sift: Automatic panorama stitching package. http://www.nowozin.net/sebastian/tu-berlin-2006/autopano-si, 2012.
20. O'Donovan, P. and Hertzmann, A. AniPaint: Interactive Painterly Animation from Video. In *IEEE TVCG*, 18(3), (2012), 475-487.
21. Ramos, G. and Balakrishnan, R. Fluid interaction techniques for the control and annotation of digital video. *UIST 2003*, ACM Press (2003), 105–114.
22. Ramos, G., Boulos, M., and Balakrishnan, R., Pressure Widgets. CHI 2004, ACM Press (2004), 487-494.
23. Viégas, F. B., Wattenberg, M. and Dave, K. Studying cooperation and conflict between authors with history flow visualizations. *CHI 2004,* ACM Press (2004), 575-582.
24. Walther-Franks, B., Herrlich, M., Karrer, T., Wittenhagen, M., Schröder-Kroll, R., Malaka, R., and Borchers, J. Dragimation: direct manipulation keyframe timing for performance-based animation. *GI 2012*, ACM Press (2012) 101–108.
25. Wang, J., Xu, Y., Shum, H.-Y., and Cohen, M. F. Video tooning. *SIGGRAPH 2004*, ACM Press (2004), 574–583.
26. Zeleznik, T., Bragdon, A., Adeputra, F., and Ko, H. Hands-On Math: A Page-based Multi-touch and Pen Desktop for Technical Work and Problem Solving. *UIST 2010*, ACM Press (2010), 17-26.