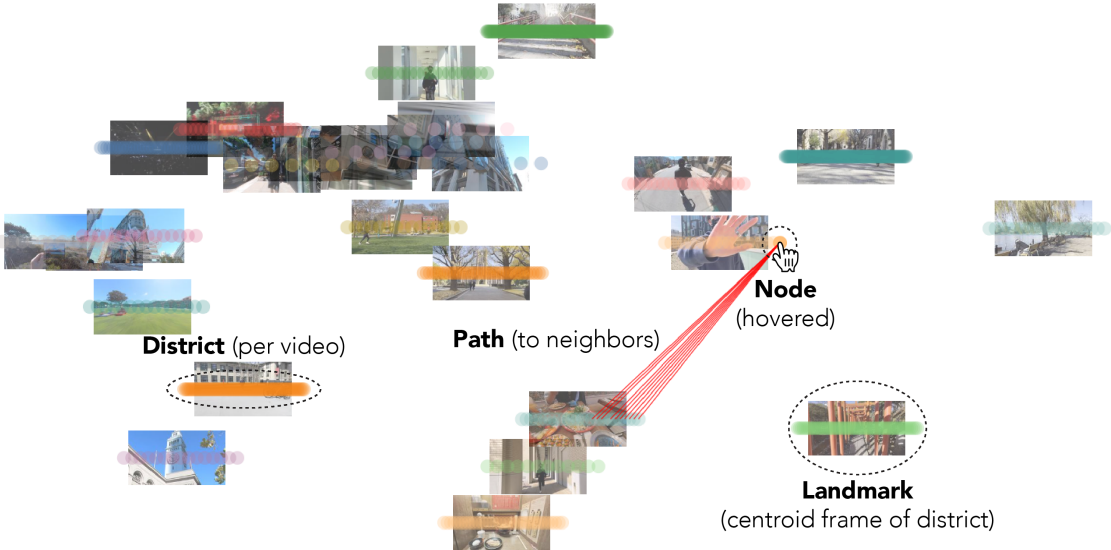


1 **VideoMap: Supporting Video Editing Exploration, Brainstorming, and**
2 **Prototyping in the Latent Space**

3
4 ANONYMOUS AUTHOR(S)



5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28 Fig. 1. VideoMap is a proof-of-concept video editing interface that operates on video frames projected onto a latent space, enabling
29 users to visually uncover patterns and relationships. We introduce map-inspired navigational elements (node, path, district, landmark)
30 to support users in navigating the map.

31
32 Video editing is a creative and complex endeavor and we believe that there is potential for reimagining a new video editing interface to
33 better support the creative and exploratory nature of video editing. We take inspiration from latent space exploration tools that help
34 users find patterns and connections within complex datasets. We present VideoMap, a proof-of-concept video editing interface that
35 operates on video frames projected onto a latent space. We support intuitive navigation through map-inspired navigational elements
36 and facilitate transitioning between different latent spaces through swappable lenses. We built three VideoMap components to support
37 editors in three common video tasks. In a user study with both professionals and non-professionals, editors found that VideoMap helps
38 reduce grunt work, offers a user-friendly experience, provides an inspirational way of editing, and effectively supports the exploratory
39 nature of video editing. We further demonstrate the versatility of VideoMap by implementing three extended applications.

40
41 CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; • **Applied computing** → **Arts and**
42 **humanities**.

43
44 Additional Key Words and Phrases: video editing, latent space, machine learning

45
46 **1 INTRODUCTION**

47
48 Video editing is a creative and complex endeavor. Consider a typical workflow: As the editor sifts through large amounts
49 of footage, they must first develop a comprehensive understanding of the material at hand and conceptualize a narrative.
50 The editor then needs to select the individual clips to use and identify connections between them to weave them
51

together with suitable transitions. As the editor refines the edit, they may constantly experiment with different editing ideas.

We observe that popular video editing interfaces used by video editors often feature a sequential editing timeline, with designs rooted in the metaphors of hand cutting and splicing film [4], supported by a grid-like asset management panel, where clips are listed by filename order or time of creation [7]. While these editing interfaces functionally support the task of assembling video frames in a specific order, they *do not have explicit mechanisms to support the more creative aspects of editing*, such as developing an understanding of the video footage, identifying connections between them, and experimenting with multiple editing ideas. We believe that there is an opportunity for building a new video editing interface that better supports this creative process. We aim to augment the way editors see and interact with their video clips during early stages of video editing, including exploration, brainstorming, and prototyping.

In this research, we take inspiration from latent space exploration tools that help users find patterns and connections within complex datasets [45]. Latent space exploration tools leverage data processing and recent machine learning techniques to transform data into vector representations and enable users to visually explore the data on a spatial interface. Building upon this insight, we ask if we can similarly project video frames onto a latent space and allow editors to edit videos within this space.

To investigate this possibility, we developed VideoMap, a proof-of-concept video editing interface based on latent space representations of video frames. In our system, we encode video frames onto latent spaces that are meaningful for video editors and support specific video editing tasks. For example, we created a latent space in which video frames containing similar shapes are near each other to help editors identify opportunities to link together scenes with similar shapes and create seamless shape-matching transitions. Editors may switch between different types of latent space maps that encode information meaningful to video to focus on different types of tasks, a concept we call “swappable lenses.” To facilitate more intuitive navigation of the map, we introduce a set of map-inspired navigational elements: nodes, districts, landmarks, and paths [47]. We then designed three components for VideoMap that utilize the latent space map to support three common video tasks. Specifically, we designed 1) Project Panel to help editors explore video footage, 2) Paths Explorer to help editors find suitable video transitions, and 3) Route Planner to help editors quickly prototype rough cuts and try out different editing ideas.

In a user study, we invited both professional and non-professional video editors to test VideoMap. We found that video editors were able to effectively use VideoMap’s components to perform the editing tasks listed above. Editors expressed that VideoMap provides a user-friendly editing experience, reduces tedious grunt work, enhances the overview capability of video footage, helps identify suitable video transitions, and enables a more exploratory approach to video editing. We further explored the design space of VideoMap by showcasing how VideoMap can be customized and extended to support additional applications, including summarizing videos, finding highlight moments within videos, and text-based video editing.

This research thus makes the following contributions:

- We introduce **VideoMap, a proof-of-concept video editing interface** that operates on video frames projected onto a latent space. We support intuitive navigation of the latent space through map-inspired navigational elements and facilitate transitioning between different latent spaces using swappable lenses.
- We built **three VideoMap components** to support editors in three common video tasks: exploring video footage, finding suitable video transitions, and quickly prototyping rough cuts.

- We demonstrate the effectiveness of VideoMap in supporting video editors complete creative editing tasks through a **user study** ($N=14$). Editors felt that VideoMap provides a user-friendly editing experience, reduces tedious grunt work, enhances the overview capability of video footage, promotes editing continuity, and enables a more exploratory approach to video editing.
- We further demonstrate the versatility of VideoMap by implementing **three extended applications**, showcasing how future developers may customize and extend VideoMap for additional use cases.

2 RELATED WORK

Our work is situated among literature on video editing tools, video browsing interfaces, and latent space exploration tools.

2.1 Video Editing Tools

VideoMap contributes to extensive literature on enhanced video editing tools. In this section, we discuss some past techniques, such as leveraging video metadata, pen input, tangibles, text-based editing, and automation.

Early research on video editing tools explored how data embedded within and associated with videos could be used to create new editing interfaces. One of the seminal works, SILVER [13], uses video metadata to support editors with additional panels to interface with and edit video footage, such as storyboards, editable transcripts, and timeline views. SILVER shows how the use of video metadata allows for smart editing capabilities beyond simple cutting, trimming, and arranging of video frames on the timeline. [25] analyzes video content for fast motion or zooming and computes a “suitability” score for each video frame to help editors decide which clips to include in the edit. This suitability score takes information about the video embedded in the pixels and makes it available as metadata for editing purposes, showing how metadata derived computationally from the frames can be leveraged to support creative editing work. VideoMap builds upon these works and leverages new kinds of metadata derivable with modern machine learning techniques.

Early researchers have also looked into alternative ways to arrange and trim videos. [14] proposes a way to edit video where the user gives voice comments over the video and the system creates a newly edited video based on the comments. Other research has explored augmenting video editing using pen-based technology, allowing editors to edit videos with direct manipulation [11, 12, 65]. Another thread of research explores using tangible interfaces to allow co-located users to edit videos collaboratively [50, 63, 75].

More recent research have mostly explored developing video editing tools that cater to a specific domain. A large thread of work focuses on building editing tools for text-focused videos, such as voiceover videos and talking head videos. [39] helps users edit dialogue-driven scenes by matching video clips to relevant dialogue. [24] lets users edit talking-head videos by editing a text transcript. [33] helps recommend b-roll video clips via interactive transcripts. Similarly, [69] lets users edit text and automatically recommends video clips they filmed to use in the edit. [40] helps users create videos by recommending matching images over text and transcripts through the notion of word concreteness (i.e., the extent to which a word describes something that can be visually experienced). [71] allows users to add images to podcasts by using natural language processing to identify important geographic locations and descriptive keywords. Today, several commercial, text-based video editing tools allow users to edit videos via a paired transcript, including Descript [1] and Type Studio [5]. In Section 7.3, we demonstrate how VideoMap may also be extended to support text-based video editing.

There have also been efforts to automate the video editing process using algorithmic techniques, such as by identifying important or highlight segments [30, 43, 70] or learning from datasets of professionally-created video examples [16, 31, 32, 52]. In addition, researchers have also explored automatically converting various forms of content into videos, such as web page to video [19], markdown to video [18], and physical demonstrations to video [20]. Although the main objective of VideoMap is not on automatic video editing, we designed a feature that supports editors in automatically generating an initial draft of a video (see Section 4.4), suggesting its potential to support new kinds of video editing automation.

2.2 Video Browsing Interfaces

A related research area to video editing tools focuses on interfaces to support video browsing. Browsing and selecting videos from a large set of raw footage is an important first step in the video editing process. A large body of research investigates the use of direct manipulation techniques [34] for video browsing. Many works have explored video playback by allowing the user to directly click and drag objects within the video, such as clicking on a car and dragging it along the road to play the video over time [26, 36, 38, 51]. [57] creates a timeline with dynamic playback speeds to emphasize important content. Another thread of research supports users in browsing through videos by revealing additional information, such as additional thumbnails [35, 48], visualizing various types of metadata such as brightness intensity and hue histograms [66], enhancing the playback of software tutorial videos by leveraging crowdsourced data from previous user interactions with the tutorials, [73], and improving the playback of surgical videos by surfacing related surgical videos that perform similar surgical steps [37]. In VideoMap, we leverage the content embedded within videos to enable enhanced browsing of video clips and visualization of connections between video clips. This allows editors to explore videos based on their content, rather than merely viewing them as a list of files.

In addition to visual browsing methods, researchers have also explored adopting text-based browsing and navigation for content-heavy videos, such as for lecture videos [54, 72], movies [53], presentation videos [55], and how-to videos [15]. Such text-based browsing methods allow editors to quickly find the right moment in the footage, helping them focus more on composing the video than on searching for content. In VideoMap, we also leverage text-based search of features derived from a semantic latent space to allow editors to find content within large amounts of footage (see Section 4.2.1).

2.3 Latent Space Exploration Tools

With advancements in machine learning, researchers in visualization and interpretable machine learning communities have recently explored methods for visualizing the embeddings of neural networks. These embeddings are vector representations of data that have been learned by neural networks. A common initial step of such works involves reducing the dimensions of these vector representations, which are often high-dimensional, to just two dimensions. This allows for the data to be visually plotted using x and y coordinates, resulting in a 2D latent space. Various techniques can be used to reduce the dimension of embeddings while retaining as much information as possible. Some popular methods include t-SNE [67], HSNE [56], PCA [8], and UMAP [49]. In VideoMap, we adopt dimensionality-reducing preprocessing to develop a new interface for viewing and working with video footage.

Given a 2D latent space, researchers have investigated the development of interactive interfaces to support user exploration [62]. [45] develops an interface to help users interactively discover meaningful relationships among data points in the latent space. [23] allows users to discover structural relationships in data, such as the association of items within groups and the hierarchies of items between groups items. Several works have investigated interfaces

for visualizing semantic relationships of text data [44, 74]. [22] allows users to visualize a latent space of gestures. [27] support users in visualizing a latent space of human interpretable concepts, such as cars. Many researchers have also worked on tools for comparing multiple latent spaces, such as by highlighting visual changes [61] and through side-by-side comparisons [9, 42]. Most relevant to our work, [68] train video-specific autoencoders (i.e., autoencoder models overfitted to the task of regenerating the videos) to visualize video frames on a latent space, which could be used to perform video manipulations such as inpainting and creating video textures (i.e., looping videos). In VideoMap, we draw inspiration from latent space visualization interfaces to assist video editors in identifying patterns and connections within video footage. We create latent spaces that are meaningful for video editors, representing various aspects of video editing, such as concepts of semantics, color, and shape.

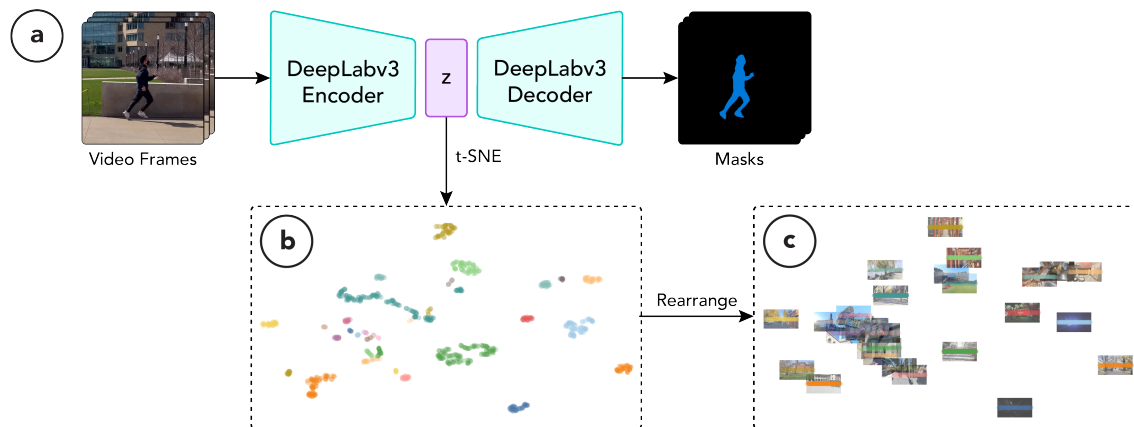


Fig. 2. The pipeline for creating the shape lens. We pass video frames through DeepLabv3, an image masking model, to extract shape-related vectors (a). We then apply t-SNE to reduce the vectors to two dimensions (b) and visually rearrange the vectors by video to make the videos more skimmable(c).

3 PRINCIPLES OF VIDEOMAP

We distill three key principles to ground the development of VideoMap based on theories and practices in video editing. These guiding principles include providing an overview of the video footage, maintaining editing continuity with seamless transitions between clips, and facilitating exploration and experimentation throughout the editing process.

3.1 Principle 1: Overview

As a time-based medium, video editing often requires editors to spend a considerable amount of time sequentially playing through and reviewing footage on an editing timeline to develop an understanding of the material they are working with. In VideoMap, we aim to overcome the sequential nature of browsing through videos by projecting video clips onto a 2D latent space, which offers a more comprehensive overview of the footage at a glance. Findings from Craft and Cairns [21] suggest the patterns and themes in data may only be seen from a vantage point that comprises the whole view. Map-like interfaces, such as plots on a 2D latent space, can provide such an overview of data and have been shown to allow people to better understand the thematic information across a dataset [29]. We posit that with an enhanced overview, editors will be able to more easily identify patterns and structure and conceptualize a narrative thread from a set of video clips. Furthermore, we designed “swappable lenses” for editors to easily switch between

different types of overviews (i.e., different latent spaces) to uncover different types of patterns within the footage, such as patterns in semantics, color, or shape (see Section 4.1.2). We designed the Project Panel component to test how VideoMap can support editors through an enhanced overview capability.

3.2 Principle 2: Continuity

Continuity is one of the key principles of video editing. In the 2012 Asia-Pacific Symposium on Creative Post-Production, Richard Crittenden, author of the book *Film and Video Editing*, comments: “*It remains true that good editing tends to be the art that conceals art*”. The goal of continuity is to connect clips seamlessly so that the edits are “invisible” to the viewer [59]. One editing technique that editors use to achieve continuity is called match cutting [3]. For example, the editor could cut from a video frame of a bagel to a video frame of a donut — using two different objects with similar shapes to transition between present and past scenes, for example. In VideoMap, we aim to assist editors in achieving continuity by recommending seamless video transitions. Since VideoMap operates in the latent space, we suggest video transitions for a video clip or frame by finding near neighbors in the latent space. Taking the shape-based match cut example: By creating a latent space where video frames containing similar shapes are organized near each other, the bagel video frame and the donut video frame will be near neighbors in the latent space. We designed the Paths Explorer component to test how VideoMap can support editors in achieving continuity in editing.

3.3 Principle 3: Exploration

Even with a well-defined narrative and a principle of continuity to follow, video editing remains a process that involves exploration and experimentation. Film director Ridley Scott likened editing to the process of painting: “*Editing is like painting, only with film. You are constantly making choices and trying to find the best way to tell the story.*” In VideoMap, we aim to provide editors with a wide range of exploration mechanisms. First, we created four map-inspired navigational elements to help editors explore the latent space: nodes, paths, districts, and landmarks [47]. Second, we implemented a semantic searching mechanism in the Project Panel component, allowing editors to quickly locate and discover clips using text prompts. Third, we recommend multiple alternative video transition options for editors in the Paths Explorer component. Finally, we designed the Route Planner component, which can automatically assemble video cuts from user-selected clips, enabling editors to quickly test and preview the results of different editing ideas.

4 IMPLEMENTATION

Our three principles are manifested in VideoMap and guide its implementation. The following outlines the implementation of our system. We first detail how we built the base layer latent space map that powers the VideoMap interface. We then go through how we designed VideoMap’s main components, including the Project Panel for exploring video footage, Paths Explorer for finding suitable video transitions, and Route Planner for prototyping video route cuts.

4.1 Building the Map

We implemented VideoMap’s base layer by first creating the 2D latent space. We then enable flexible reconfiguration of different latent spaces using swappable lenses, and support intuitive navigation of the map through map-inspired metaphors.

4.1.1 Creating the Latent Space. VideoMap is based on a latent space representation of video frames. Specifically, we encode video frames into vector representations such that they are meaningful for particular video editing tasks. For

example, we can create a latent space map to help editors find shape match cuts, a type of video transition which links together two scenes that contain similar shapes to create a smooth transition (see an example from Kubrick’s 2001: A Space Odyssey). To create this map, we pass video frames through DeepLabv3 [17], an image masking model, with a ResNet [28] backbone and extract a 512-dimensional vector representation right after the last visual layer (Figure 2a). Our insight is that due to the mask generation objective, which extracts the dominant shape(s) of an image, vector representations extracted from DeepLabv3 implicitly encode valuable shape-related information. In other words, video frames that contain similar shapes should have more similar vector representations.

Next, we project the vector representations of all video frames into a multi-dimensional latent space. To allow editors to browse through the latent space, we apply t-Distributed Stochastic Neighbor Embedding (t-SNE) [67] to reduce the latent space to two dimensions. In the first iteration of VideoMap, we visualized the vector embeddings directly according to the coordinates generated from the t-SNE (Figure 2b). However, in our pilot tests, editors felt that this was too messy. They also wished to discern groups of points that belong to the same videos more easily and be able to “play through” the videos. Therefore, in the current iteration of VideoMap, we visually rearrange the points on the map by video (Figure 2c). We first compute the centroid of all points for each video, then we then visualize frames as points horizontally across the centroid at equal distances. We also display the centroid video frames. Editors can thus play through each video by scrubbing through the points horizontally. From our user study, we found that editors considered this design to be easy to use (see Section 6).

4.1.2 Developing Swappable Lenses. We call the different methods of encoding video frames into meaningful vector representations as different *lenses*. In Section 4.1.1, we create a shape lens. Under the shape lens, video editors can identify shape-based match cut opportunities by looking at neighbors on the latent space. Similarly, one may develop new lenses by finding other ways of encoding useful vector representations for other use cases. In our proof-of-concept, we implemented a color lens (video frames organized by color) and a semantic lens (video frames organized by meaning). To create the color lens, we first extract 3D histograms from each RGB video frame using 8 bins per channel and normalizing with $range = [0, 256]$, then flatten the histogram to generate a 512-dimensional color-based vector. To create the semantic lens, we encode each video frame through the image encoder of CLIP [58], a neural network that has learned semantic concepts, to generate a 512-dimensional semantics-based vector. Rather than creating a one-size-fits-all map, we allow editors to flexibly swap between different lenses and reconfigure VideoMap according to different criteria. We demonstrate examples of how video editors creatively made use of the different lenses for different editing tasks in Section 5.4.3.

4.1.3 Navigating with Nodes, Paths, Districts, and Landmarks. We designed four map-inspired navigational elements to help editors navigate through the map (Figure 1). The four elements are nodes, paths, districts, and landmarks. These elements are inspired by Lynch’s seminal The Image of the City [47], which studied the elements people use to form mental maps of a city.

Nodes. The node is a point (i.e., a vector representation of a video frame) that the user hovers over with their cursor. We follow Schneiderman’s details-on-demand mantra [60] by expanding the node’s detailed information, such as its corresponding video frame, video filename, and timecode.

Paths. Paths are lines that connect from a user-selected node to its neighboring points on the latent space. When a user clicks on a node, we display ten paths that start from the node and connect to the ten closest neighboring points, which are the points with the smallest cosine distances to the node. We exclude neighboring points that belong to the same video as the node to ensure that the paths represent video transitions. To obtain an accurate representation of

neighbors, we calculate the distances based on the original vector representations rather than the rearranged coordinates on the visual map (see Section 4.1.1).

Districts. Districts are groups of points with common characteristics. Throughout most of this paper, we present the version of VideoMap where points that belong to the same video are grouped into a district. Note that paths connect across two different districts (i.e., video transitions). We color-code the points by district to visually separate them. We also consider alternative methods for defining districts, such as through groupings of semantically similar video frames, in Section 7.1.

Landmarks Landmarks are reference markers on the map. With landmarks, the user can obtain an overview of the map at a glance (Principle 1). We visualize the corresponding video frames of some points on the map as landmarks. Throughout most of this paper, we present the version of VideoMap where we generate one landmark for each district by visualizing the video frame of the point closest to the district’s centroid. We also explore the option of allowing users to define custom landmarks in Section 7.2.

4.2 Project Panel

The Project Panel is the first component that the editor can use on VideoMap (Figure 3). We designed Project Panel to enable editors to explore video footage more easily (Principle 1). Traditional video editing software typically feature a grid-like interface that displays all of the editor’s video files ordered by filename or time of creation. In VideoMap’s Project Panel, instead of displaying a standard grid, we rearrange the display of the video files based on the selected lens. For example, Figure 3 displays a collection of videos in the Project Panel under the semantic lens. The editor can swap between different lenses to rearrange the layout and discover new perspectives on the video footage. To reproduce the video thumbnails seen in traditional video editing software, we display the landmark for each video file. The editor may play through the video files by scrubbing the landmarks from left to right. The editor may also zoom in or out of the map by scrolling to view regions of the map with less clutter.

4.2.1 Prompts. The editor can filter videos with text prompts (Principle 3) (Figure 4). For example, by typing `torii gates`, the editor can highlight these videos and fade out the rest. We encode the text prompt using a CLIP text encoder and compare the text embedding to the image embeddings of video frames, encoded by a CLIP image encoder (i.e., the vector representations under the semantic lens). From our user study, we found that editors felt that this method handled prompt searches effectively (see Section 5.4.2).

4.3 Paths Explorer

Paths Explorer is the second component that the editor can use on VideoMap (Figure 5). We designed Paths Explorer to assist editors in finding suitable video transitions (Principle 2). Editors typically arrange videos in a sequential timeline, which can make it difficult for editors to identify patterns of video frames at different times and across different videos. With Paths Explorer, we overcome the inherent sequential nature of videos by arranging them on a 2D map, thereby removing the limitations of viewing video frames along a 1D timeline (Principle 1). While we only display each video’s landmark in Project Panel, we display each video’s individual video frames as points in Paths Explorer. We color-code the points by video (districts) and fade the landmarks to reduce visual clutter. When the editor clicks on a point, the interface draws 10 paths connecting to nearest neighbor vector representations from other video(s) (Figure 5b). These paths represent potential video transitions that the system suggests based on the selected lens. For example, Figure 5 displays recommended paths that connect between videos with similar color composition (having two prominent

Prompt

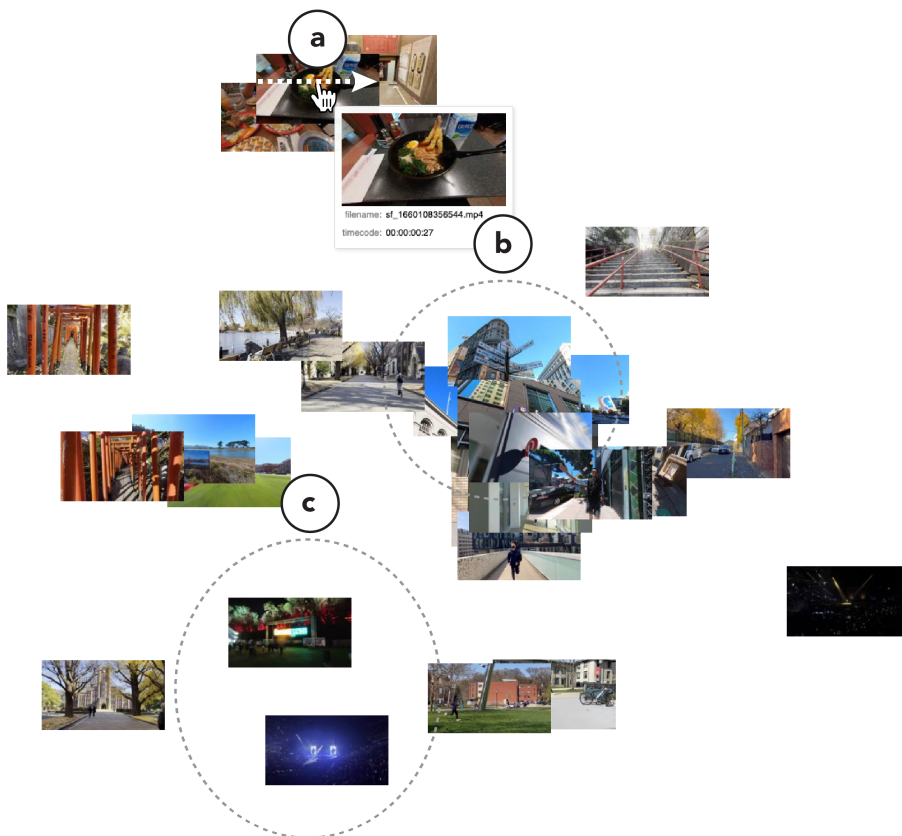


Fig. 3. VideoMap’s Project Panel component. The figure shows a collection of videos organized under the semantic lens. The editor can play through the videos by scrubbing the landmarks from left to right (a). Example semantic clusters of videos are shown in (b) for several videos containing streets and buildings and (c) for concert videos.

blocks of color divided at the horizon) under the color lens. The editor can switch between the lenses to discover different types of video transitions.

4.4 Route Planner

Route Planner is the third component that the editor can use on VideoMap (Figure 6). We designed Route Planner to help editors automatically generate rough cuts, which is an initial rudimentary edited version of a video, to quickly explore different video edits (Principle 3). With Route Planner, the editor can select several video clips and the system will automatically find an optimal route across paths (see Section 4.3) that connect these video clips, based on the selected lens. For example, Figure 6 displays an example route that connects across multiple videos containing the same running action under the shape lens. One can think of this component as being analogous to the Google Maps route planner [2]. The paths connecting video clips are like “streets”. Route Planner finds the fastest route that travels through the streets

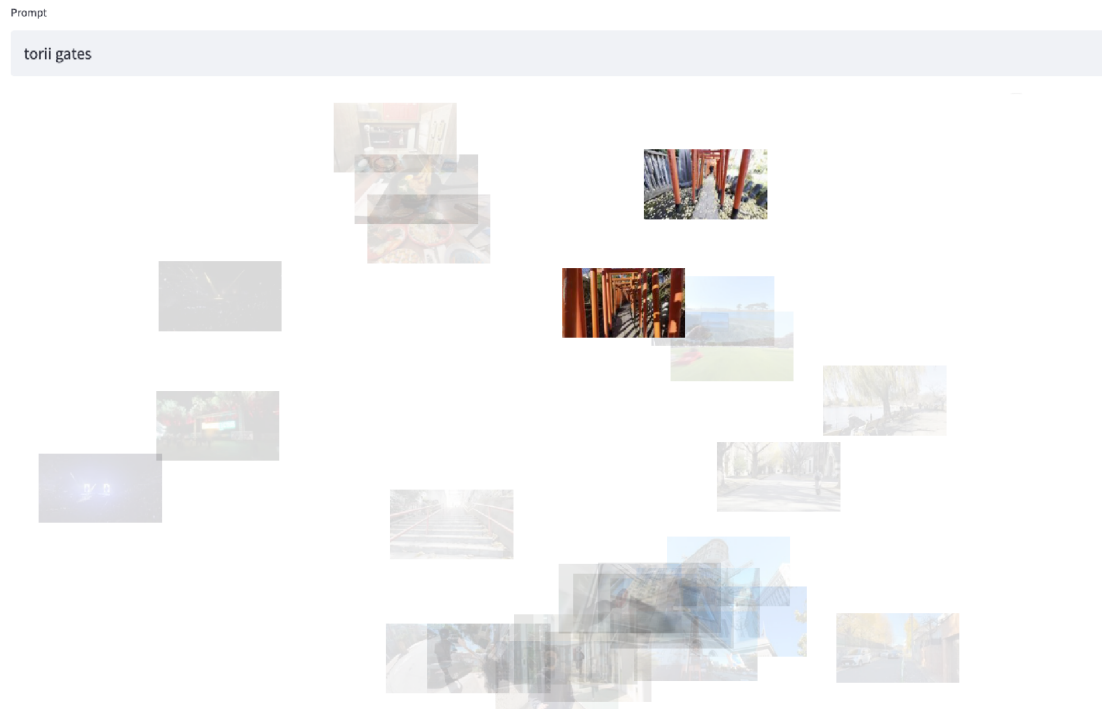


Fig. 4. The editor can search for videos using text prompts (e.g., torii gates). [See more prompt examples.](#)

and passes through each of the selected video clips once. Since the “distances” between video clips on the map are determined based on the selected lens, the editor can switch between the lenses to try out different rough cuts based on different criteria, such as switching to color to connect clips with similar color grading or semantics to connect clips depicting similar objects or activities.

To implement Route Planner, we first find the shortest path for all combinations of video pairs (i.e., optimal video transitions). These paths serve as the “streets” that connect across different videos. Next, we find the shortest route along the streets that passes through all videos (i.e., the Shortest Hamiltonian Path Problem) with a dynamic programming approach. Finally, we visualize the route on the interface and render the rough cut video by traversing through the route. Since the streets are undirected, some videos may be reversed during the traversal. This can lead to interesting results where videos are connected with one playing forward and one playing in reverse, such as a [foot transition](#) created by one of the editors in our user study.

5 USER STUDY

We conducted a user study to observe how professional and non-professional video editors might use VideoMap to complete creative editing tasks, including developing an overview of the video clips they have, finding unique transitions and cuts between clips, and crafting a rough cut of a video. We aimed to address three research questions with the study:

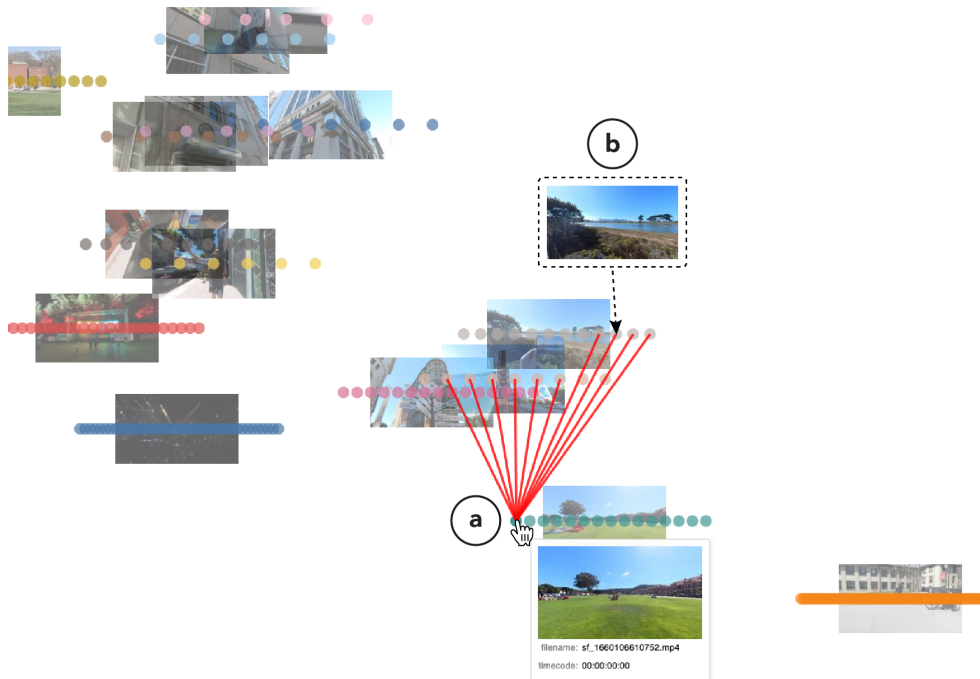


Fig. 5. VideoMap’s Paths Explorer component. The editor can select a video frame to display ten video transition suggestions based on the selected lens (a). For example, a video frame with a similar color composition is recommended under the color lens (b). See [example video result](#).

- RQ1. How can VideoMap assist editors in exploring video footage?
- RQ2. How can VideoMap assist editors in brainstorming suitable video transitions?
- RQ3. How can VideoMap assist editors in rapidly prototyping rough cuts?

5.1 Study Design

To address our three research questions, we asked video editors to perform the three video editing tasks using VideoMap. Specifically, we asked participants to use the Project Panel to explore and familiarize themselves with a collection of video footage, use Paths Explorer to find video transitions, and use Route Planner to create rough cuts. During the study, we asked participants to share their computer screens and perform a think-aloud [41] describing what they were doing and thinking while using VideoMap.

We collected data using three methods. First, we asked participants to complete a questionnaire where participants provided written descriptions on completing the tasks (e.g., a description of how the video clips are organized) as well as feedback on their experience using VideoMap in contrast to how they would perform the same task using a video editing software they are familiar with (see Questionnaire in Supporting Documents). Additionally, participants attached screenshots of the interface (e.g., a screenshot of a video transition shown in Paths Explorer that they found interesting) and video outputs from completing the tasks (e.g., a rough cut video generated by Route Planner). Second, we recorded the participants’ computer screens and transcribed their spoken audio throughout the sessions. Third, the first author silently observed participants working and noted down interesting insights, such as participants’ thoughts

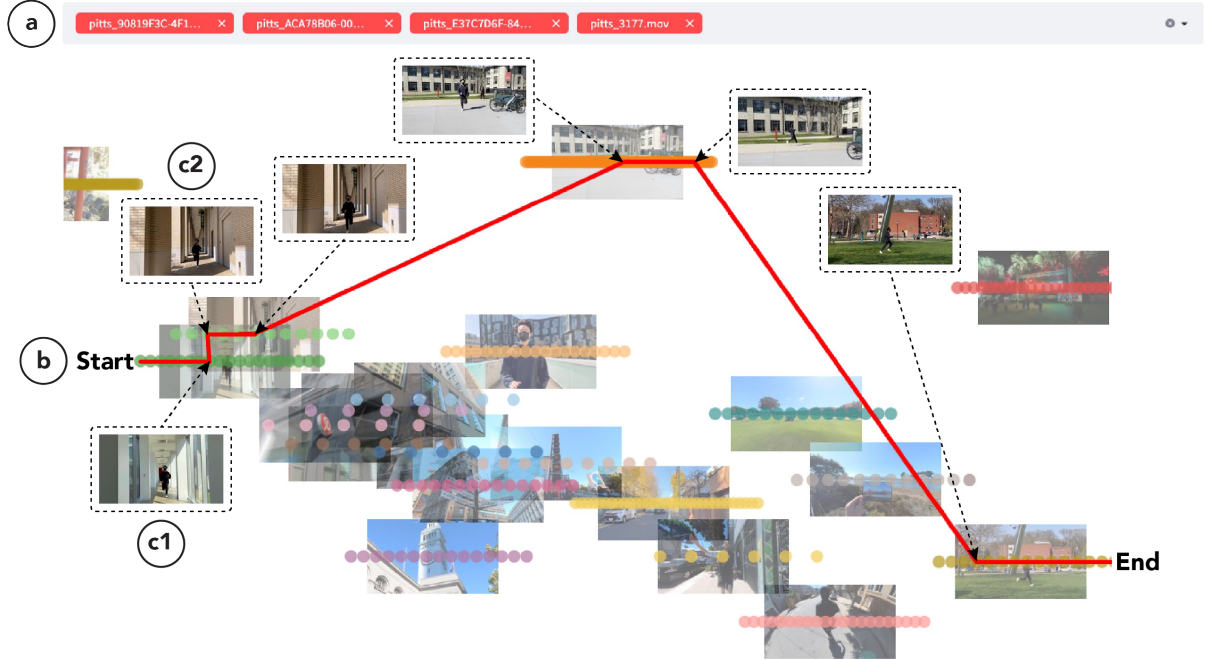


Fig. 6. VideoMap’s Route Planner component. The editor can select several video clips to automatically generate a rough cut video based on the selected lens (a). For example, the editor can create a video of a person running through various backgrounds under the shape lens (b). Route Planner automatically finds the optimal video transitions (c1 and c2). [See example video result.](#)

while using the system, interaction observations from the screen captures, and the creations participants made while completing the tasks. Overall, the data collected allowed us to gain rich insights into the participants’ experiences using VideoMap, as well as visual evidence of how VideoMap was used and could support editors in accomplishing fundamental video editing tasks.

5.1.1 Video Collection. We gathered a sample of video clips for participants to use during the editing tasks in our studies. This collection comprises 30 diverse videos, filmed by the authors in various locations including San Francisco, Pittsburgh, and Tokyo. The content covers a variety of categories, including objects like food and architecture, scenery like streets and grassy fields, and actions like running and scooting. The videos were captured using different camera equipment, such as mirrorless cameras, smartphones, and 360 action cameras. Figure 3 provides an overview of the sample video collection displayed in VideoMap’s Project Panel.

5.2 Participants

We recruited 14 participants (8 male, 6 female) aged from 23 to 55 ($\mu=28.21$, $\sigma=8.78$). Seven of the participants are professional video editors recruited from Upwork [6], a platform for hiring creative freelance workers, and seven of the participants are non-professional video editors recruited at our institution via word-of-mouth and snowball sampling. We recruited both professional and non-professional individuals to test VideoMap’s ability to support professional editing work as well as make video editing more approachable for non-professionals. We conducted a background

survey with the participants before each study to assess their video editing experience. Professional participants have high self-rated familiarity with video editing ($\mu=5.86$, $\sigma=1.46$) (7-point Likert scale) and many years of experience ($\mu=8.07$, $\sigma=2.55$). Non-professional participants have moderate self-rated familiarity with video editing ($\mu=4.43$, $\sigma=1.27$) and several years of experience ($\mu=3.00$, $\sigma=1.15$). Participants use various video editing software such as Adobe Premiere Pro, Final Cut Pro, Movavi Video Suite, iMovie, CapCut, and Google Photos Movie Editor.

5.3 Procedure

We conducted the studies over video conference (Zoom). We asked participants to try out the three components of VideoMap (Project Panel, Paths Explorer, and Route Planner) and complete a video editing task using each component. The following details our step-by-step procedures.

After obtaining the participant's consent, we collected background information about their prior video editing experience. Next, the participant tried out VideoMap. First, they used the Project Panel to familiarize themselves with the sample video collection and think about how they could create a video montage using it. We also asked participants to try the different lenses (semantic, color, shape) and try searching for clips by typing different prompts in the prompts box. We asked the participant to write down three prompts that yielded interesting results. Second, the participant used Paths Explorer to find video transition opportunities. We asked the participant to record three transitions that they found interesting during their exploration. Third, the participant used the Route Planner to quickly create rough cuts for a video montage using the provided clips. Participants used the Project Panel to determine which video clips to select in the Route Planner for the rough cut. We asked the participant to record one rough cut that they found interesting. After trying out the three components of VideoMap, we asked the participant to complete a post-study written survey about their experiences. We also asked participants to contrast their experiences with how they would typically perform the same tasks using existing video editing software. The study lasted for approximately 1 hour.

5.4 Results

We analyzed our data, including observation notes, audio transcripts, and form responses, with inductive coding to identify common themes. The following presents our findings. We note direct quotes from professional participants as P and non-professionals as N.

5.4.1 Project Panel. Participants were able to use Project Panel to explore the video footage by familiarizing, grouping, and uncovering structure among the clips (RQ1). In addition, participants also noted additional use cases of Project Panel, such as supporting visual ideation and helping with planning new shots to take. As part of the first task, we asked participants to brainstorm narrative ideas that utilize the sample video collection. Some ideas from participants include a video starting from wide-angle shots and then showing details of single activities, a video that first shows campus life and then has a fancy transition to life during the holidays (traveling, concerts), and a video introducing various Japanese cuisines.

Familiarize with video footage. Participants were able to use Project Panel to familiarize themselves quickly with the video collection. P3 expressed that “*the beginning process of editing a video can be really daunting*” and Project Panel “*takes a lot of the time out of needing to familiarize yourself with the content that you have*”. N3 enjoyed using Project Panel as a “*good preprocessing step*” and expressed that it “*can save people time trying to eyeball videos manually themselves.*” Participants felt that Project Panel “*corresponds well with [their] mental model of how [they] would look for*

footage (e.g., grouping common themes) (N4)” and facilitates a “faster and more intuitive way to explore a collection of video assets (P4)”.

Organizing large projects. Participants noted that Project Panel could be especially useful for organizing large projects, such as being “a unified hub for large projects involving many team members (P1)”. In particular, many participants enjoyed that Project Panel helps save time and effort required to manually assign tags to video clips or bin video clips into folders (P1, P3, P5, P6, P7, N7): “It takes forever to tag clips. It’s either that or subfolders. It’s such a pain to go through a whole day’s worth of shooting. (P7)”

Uncover structure. Participants stated that Project Panel helps “reveal underlying structure within the recorded content (P4)”, especially those that “may not be entirely apparent at first” (P4). Participants enjoyed the “inspiration and surprise factor (N5)” and helps create “a fresh perspective on clips they might have passed up (P7)”: “Getting a new take or generating different ideas is what AI is great for and is how we can use tools like these to enhance our editing. (P7)” Participants were also able to use Project Panel to easily identify outlier videos (P2, N5): “This one color is definitely sticking out (P2)”.

Support visual ideation. Participants expressed that having the videos visually arranged in a 2D space helped them “have a better sense of the clips that they are working with and see the overall vision of what they want to create (P5)” (Principle 1). For example, some participants wanted to create a video that takes one scene from each cluster on the map (P1, P5) while others wanted to create a video that focused only on one cluster on the map (P6, N2). N1 noted that “seeing similar footage organized in the same place helps [them] figure out where to look if [they are] searching for certain elements (N1)”.

Plan the shots. Participants reported that Project Panel helped them with planning the (additional) shots that need to be taken (P5, N2). P5 stated that the “biggest pro [of Project Panel] was seeing the overall visuals of clips and the amount per grouping. I would use this at the very beginning of my workflow where I decide what other clips I need to take to complete the story. For example, if each scene of a trip is in there or if I need more concert clips.”

5.4.2 Prompts. Participants were able to make use of the prompt box in Project Panel for various purposes, such as quickly searching for clips, searching for similar clips, and searching for cinematic elements. As part of the first task, we asked participants to write down three prompts they found the most interesting during their explorations. The average length of the prompts is around 2 words ($\mu=2.24$, $\sigma=2.00$). The most popular prompt was food, which was included by 5 participants. We grouped the prompts into 6 categories. Overall, participants searched for objects 13 times (e.g., bike), scenery 10 times (e.g., outside in a grassy field), location 6 times (e.g., restaurant), action 7 times (e.g., cooking), time 3 times (e.g., daytime), and cinematography elements 3 times (e.g., establishing shot). Some example prompts from participants include videos of people (P2), running (N2), blue sky (P6), autumn colors (N5), motion blur (N5), and establishing shot (P3).

Search for a clip (instantly). Participants used the prompt box to search for clips quickly: “I would be editing at the hotel after a day of filming. Suppose I went to Rome today and I remember I went to the Trevi fountain. It’ll be way easier to find it [by searching ‘trevis fountain’]. (P2)” P5 commented that they could use the prompt box to more easily search for “clips with triangular elements while not remembering where those clips were” and P3 commented that they could use it to “find good b-roll shots” for interview-style videos.

Search for similar clips. P7 commented that they could use the prompt box to search for similar clips: “There’s a concert clip. I wonder if there are any other concert-related clips.”

Search for cinematic elements. P3 and P4 commented that they could use it to find specific cinematic elements to use for their videos, such as an “*establishing shot to start off a video* (P3)” or a “*wide-angle shot to sprinkle in some drone footage* (P4)”.

5.4.3 Swappable Lenses. Participants found the Swappable Lenses concept useful and expressed that it helped shine a light on different perspectives and allowed VideoMap to be able to cater to people with different editing styles. Participants came up with many creative ways of utilizing Swappable Lenses. P4 used the color lens to find videos of similar seasons, such as autumn. P5 used the color lens to arrange videos chronologically, starting with the day and ending with night scenes. P7 used the color lens to identify videos with warmer and cooler tones. N5 used the color lens to identify black frames to serve as transition points. N6 expressed the potential of using the color lens to maintain a steady color grading scheme. P7 used the shape lens to find videos with similar horizon positions. N6 used the shape lens to identify videos that contain people running. P4 used the shape lens to find videos containing circular shapes. P5 used the shape lens to find videos that have similar angles and are rotating in similar motions.

Shine light on different perspectives. Participants expressed that Swappable Lenses helped shine light on new perspectives (N1, N4, N5, P7): “*They give different perspectives that focus on different traits of the footage* (N1)”. For example, the color lens helped reveal videos with the same color palette or aesthetic (P1, P3, P4, P6): “*In the color lens, I could see the videos that have warmer colors versus cooler colors, which I didn’t notice this in the semantic lens.* (P7)” In addition, participants found the shape lens useful for finding videos containing people doing similar actions (N1, N5, N6, P5, P7) or videos with similar compositions (P4, P7). P7 felt that “*the different lenses created subtle filters to the footage which inspired [their] decision-making process.*”

Something for everyone. Participants expressed that Swappable Lenses allow VideoMap to be flexible in supporting editors who might prefer different types of editing techniques: “*It also serves users’ preferences in different video editing techniques. Users can choose ways that work the best for them.* (N3)” Among the 14 participants, 8 participants preferred using the semantic lens the most, 3 participants preferred using the color lens the most, and 3 participants preferred using the shape lens the most.

5.4.4 Paths Explorer. Participants were able to use Paths Explorer to identify suitable video transitions by recognizing interesting match cuts with minimal time and effort (RQ2). In addition, participants also expressed that Paths Explorer also helped encourage and facilitate more exploration. As part of the second task, we asked participants to record three video transitions they found the most interesting during their explorations. Some video transitions from participants include using the shape lens to cut between scenes that have buildings with similar architectural designs, using the shape lens to cut between scenes with similar composition (both having two prominent blocks of color divided at the horizon), using the semantic lens to cut between food scenes, and using the color lens to cut between scenes featuring trees with similar autumn tones.

Identify match cuts. Participants expressed that Paths Explorer is a “*cool and unique way to conceptually look at footage and generate match cut ideas* (P7)”. Participants were able to easily link together different environments with similar vibes (Principle 2) (P1, P7, N1, N2, N3, N7): “*[Paths Explorer] helps find associated concepts across different videos more easily.* (N2)” Non-professional editors found that Paths Explorer changed how they think about video transitions. For example, N3 noted that Paths Explorer “*made [them] put more thought into transitions*” where they “*previously had not thought too much about them.*” N1 commented that Paths Explorer “*inspired a lot of new ideas about how a transition could be.*”

Save time and effort. Participants noted that Paths Explorer helped save time and effort in finding transitions (P3, P7, N1, N5): “Scrolling over footage and going back and forth can be very time-consuming and tedious (P7).” P3 commented that “having a lens that specifically found the paths between matching shapes shaved off hours of looking through footage.” N5 appreciated being able to see “the exact frames that can be connected together”.

Encourage exploration. Participants felt that Paths Explorer helped encourage more exploration compared to traditional video editing software (Principle 3): “It’s a more exploratory approach than with the Premiere interface. I can discover relationships between scenes easier. It’s a great way to anticipate multiple possible transitions without committing to an idea. (P4).” N1 appreciated the ability to view multiple transition suggestions together: “Oh these are kinda cool. This is probably better. I’ll choose this one. (N1)” N2 enjoyed the ability to “simultaneously explore frames at different time points and not being constrained by the temporal ordering of video content.”

5.4.5 Route Planner. Participants were able to use Route Planner to quickly prototype rough cuts with automatic editing of clips (RQ3). In addition, participants also commented that Route Planner is able to help their editing workflows by generating ideas, facilitating quick previewing of ideas, and suggesting enhancements. As part of the third task, we asked participants to record a rough cut generated by Route Planner that they found the most interesting. Some rough cuts by participants include using the shape lens to generate a video of a person running with different backgrounds, using the shape lens to generate a video with a seamless foot transition, and using the color lens to generate a video that starts from daytime and ends with nighttime.

Automatic editing. Participants enjoyed being able to quickly generate edited rough cuts using Route Planner (P2, P3, N1, N2, N3, N6): “It reduced the hassle of trying to arrange clips in a way that makes visual sense. Using traditional software would have made this process take hours. (P3)” P2 and N6 commented that Route Planner could be helpful for publishing video content on social media platforms: “It could be great for platforms that need content generated quickly, like TikTok. It could easily help create content like a supercut or quick preview. (P2)” Participants felt that Route Planner generated quality cuts: “It’s a very cool video. That foot part didn’t even seem like a transition. It just seemed so seamless that I thought it was the same video. (N3) (see referred video)”

Generate ideas. Participants commented the Route Planner can be used for idea generation. For example, N1 commented that they could throw clips into Route Planner before jumping into editing to “quickly plan out an editing plan.” On the other hand, N5 commented that they would primarily use Project Panel and Paths Explorer and “use Route Planner if [they] run out of creative ideas.”

Preview ideas quickly. Participants felt that Route Planner helps editors “try out different ideas faster than the traditional software (N6)” (Principle 3). P7 commented that “[Route Planner] was a really good way for [them] to visually come up with and execute ideas. When using Route Planner, I had thought of many different ways the videos and clips could have been edited together.”

Suggest enhancements. Participants noted that Route Planner could be used as a tool for suggesting editing enhancements. P7 commented that they could first “manually do a rough cut, then give Route Planner the clips that [they] used and see if it could come up with a better cut.”

6 DISCUSSION

Participants generally felt that VideoMap offers a user-friendly approach to video editing (P3, P5, P6, N1, N2, N6). Specifically, non-professionals remarked that VideoMap “makes editing tasks much easier (N6)”, possesses a “smoother learning curve (N1)”, and offers a “more fun way to edit videos (N2).” Additionally, participants expressed that VideoMap

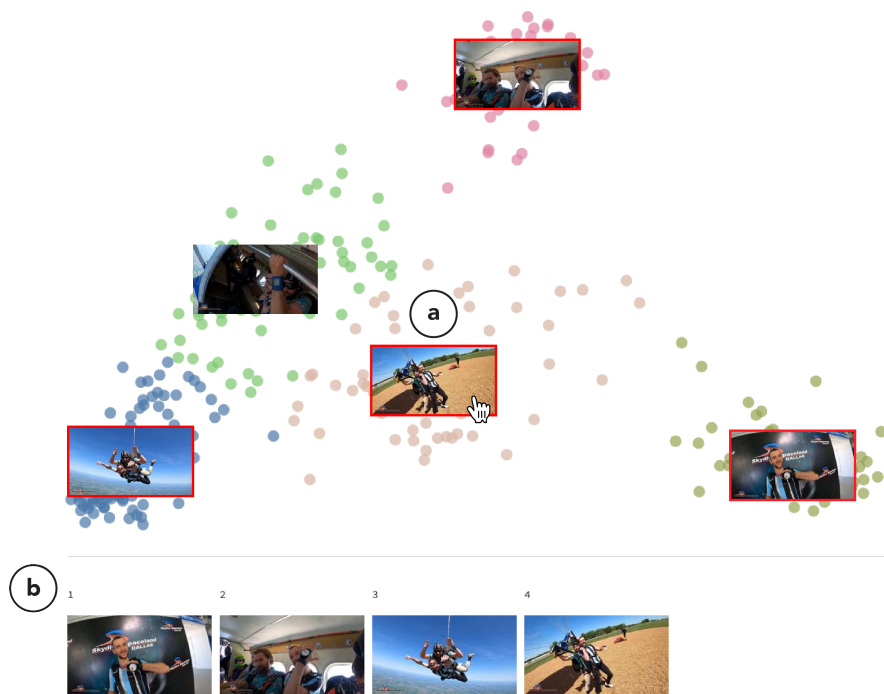


Fig. 7. VideoMap’s Project Panel can be extended to create summary videos. We automatically create “semantic districts” that approximately represent the main activities of a video using k-means clustering under the semantic lens. The editor can select several landmarks to specify the activities to include in the summary video. Selections are highlighted with red borders (a) and displayed as a storyboard (b). See [example video result](#).

helps reduce the non-creative grunt work involved in video editing, allowing for greater focus on creative aspects (P5, N5, N6): “Trying to write my process in Premiere made me realize how complicated it actually is to do this task, [and] how much manual searching and scrolling through frames I do. (P5)”. N6 explained that their workflow of organizing video footage using traditional editors involves “manually sorting the videos in a file browser, picking the videos that [they are] most interested in, then manually uploading them into the editing software (N6)”. With Project Panel, they could “have all videos uploaded at once, then filter with prompts using the ideas [they] have in mind (N6)”.

One clear benefit of VideoMap is its enhanced overview capability (Principle 1). By breaking the sequential nature of browsing videos, viewing videos across a latent space in the Project Panel helps editors explore raw footage, enabling them to better familiarize themselves with and uncover structure within the footage (RQ1): “[VideoMap] helped me get a more comprehensive overview of the materials more quickly (P4)”. P7 reflected: “I think the traditional video editor can leave much to be desired. [VideoMap] is a fresh take on the old concept, allowing for the clips to shine and visually show [me] what they are and how [I] might think to compose them.” N5 commented that “[VideoMap] is a more inspirational way of organizing video clips” and N3 noted that “[VideoMap] helps editors tell better stories with connected themes, color, and shape”. Given an enhanced overview, participants were also able to visually ideate narratives, such as creating a video that weaves together one clip from each cluster on the map (P1, P5), or conversely, creating a video that only focuses on clips from one cluster (P6, N2). In addition, P5 was also able to utilize the overview to “decide what other

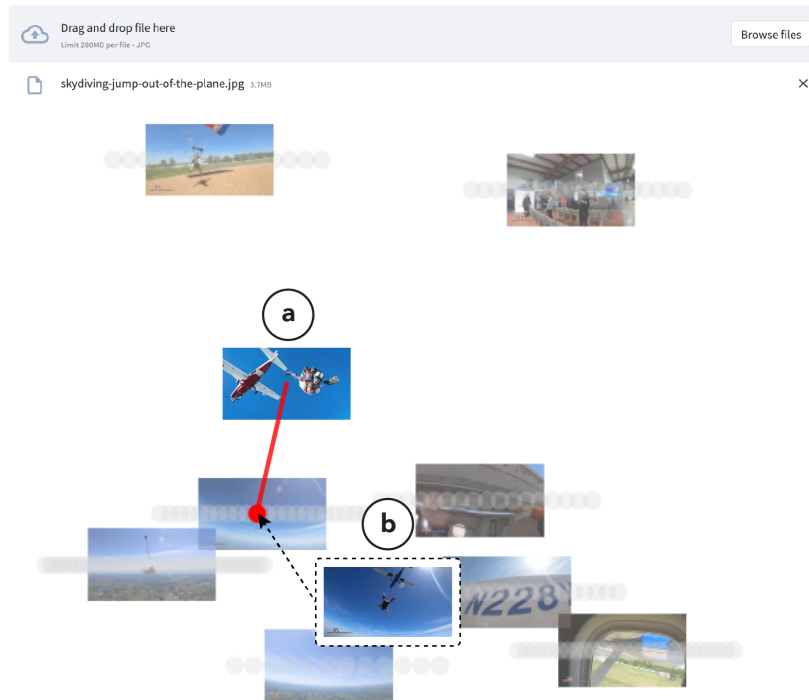


Fig. 8. VideoMap’s Paths Explorer can be extended to create highlight videos. The editor can upload a photograph (i.e., a custom landmark) depicting an activity (e.g., skydiving) (a). Our key insight is that photographs taken by photographers tend to capture the most highlight-worthy moments of an activity (e.g., when the skydiver jumps out of the aircraft). We then generate a highlight video using near neighbor video frames to the custom landmark in the semantic space (b). [See example video result.](#)

clips [they] need to take to complete the story”, such as capturing more concert footage after noticing relatively small semantic concert clusters on the map.

By operating on the latent space, VideoMap helps visualize semantic or visual neighbors, thereby incentivizing editors to maintain continuity in their edits (Principle 2). Paths Explorer assists editors in brainstorming opportunities to create continuous transitions through transition recommendations to latent space neighbors (RQ2): “[*Paths Explorer*] is a cool and unique way to conceptually look at footage and generate match cut ideas (P7)”. N5 explained that “using a traditional video editing software might involve jumping back and forth between the videos or overlaying frames on top of each other with a lower transparency to see if there are some common shapes”. P3 contrasted editing transitions using Paths Explorer with their typical workflow: “It really changes the process of editing. Usually, when you’re doing this [with traditional editing software], you already have everything onto a timeline, and you’re like, alright, now let’s find the transitions.” Furthermore, non-professionals felt that Paths Explorer “made [them] put more thought into transitions [where they] previously had not thought too much about them (N3)”, and helped “inspire a lot of new ideas about how a transition could be (N1)”.

VideoMap enables a more exploratory approach to editing through various components and mechanisms, such as automatic rough cuts with Route Planner, map-inspired navigation elements, prompt searching in Project Panel, and

recommending multiple possible transitions in Paths Explorer (Principle 3). By automatically generating rough cuts, Route Planner helps editors quickly test and preview editing ideas (RQ3). Participants utilized the generated rough cuts for idea generation (“*quickly plan out an editing plan* (N1)”), previewing the results of editing ideas (“*I had thought of many different ways the videos and clips could have been edited together* (P7)”), and suggesting enhancements (“*see if it could come up with a better cut* (P7)”). Participants commented positively on various interactions and map-inspired navigation elements, such as scrubbing landmarks to play through videos (P5, P6, N6) and selecting nodes to visualize paths to neighbors (P4, N1, N2). Participants enjoyed using prompts in the Project Panel to semantically search for clips in mind or discover new clips: “*It is like the difference between using Google Search and searching information from books* (N2).” Additionally, P4 commented that “*Paths Explorer helped [them] discover and plan possible transitions much faster*” and felt like a “*more exploratory approach than with [a traditional editing interface]*”, given the ability “*to anticipate multiple possible transitions without committing to an idea*”.

Finally, we envision a vast and versatile design space for VideoMap, with future developers enhancing and extending its capabilities. Our current proof-of-concept enables editors to edit video through notions of semantics, color, and shape. We plan to open-source our code so that developers can build new lenses to support additional editing tasks. Furthermore, we hope that VideoMap can be used as a building block to support the development of new creative video editing applications. In the following section, we demonstrate some examples.

7 EXTENDED APPLICATIONS

We implemented three extended applications of VideoMap to demonstrate how it can be customized for additional use cases. These applications include (1) video summarization with semantic districts, (2) video highlights with custom landmarks, and (3) text-based video editing.

7.1 Video Summarization

We show how VideoMap Project Panel can be extended to help editors create summary videos (Figure 7). We first project the source video’s frames into a semantic latent space. Instead of creating districts per video, we apply k-means clustering to automatically partition the video frames into “semantic districts.” Since we are working in a semantic space, each district approximately represents a main activity of the source video. We determine the number of districts (i.e., k value) using the elbow method [64]. We color-code the districts and visualize the centroid frame of each district as semantic landmarks. To create the summary video, the editor can add semantic districts (i.e., main activities) by clicking on the landmarks, in the order they want them to appear. Selected landmarks are highlighted with red borders (Figure 7a) and displayed as a storyboard at the bottom (Figure 7b). We generate the summary video by taking three seconds of video centered around the landmark for each semantic district (see [example video result](#)).

7.2 Video Highlights

We show how VideoMap Paths Explorer can be extended to help editors create highlight videos (Figure 8). The editor can first upload a photograph depicting an activity (e.g., skydiving). We then project the photograph into the latent space under the semantic lens and visualize it as a “custom landmark” (Figure 8a). Next, we draw a path from the custom landmark to the nearest video frame in the latent space (Figure 8b). Our key insight is that photographs taken by photographers tend to capture the most highlight-worthy moments of an activity (e.g., when the skydiver jumps out of the aircraft). Thus, we take the photograph’s neighboring video frames as the highlight video. We generate the highlight video by taking five seconds of video centered around the nearest video frame (see [example video result](#)).

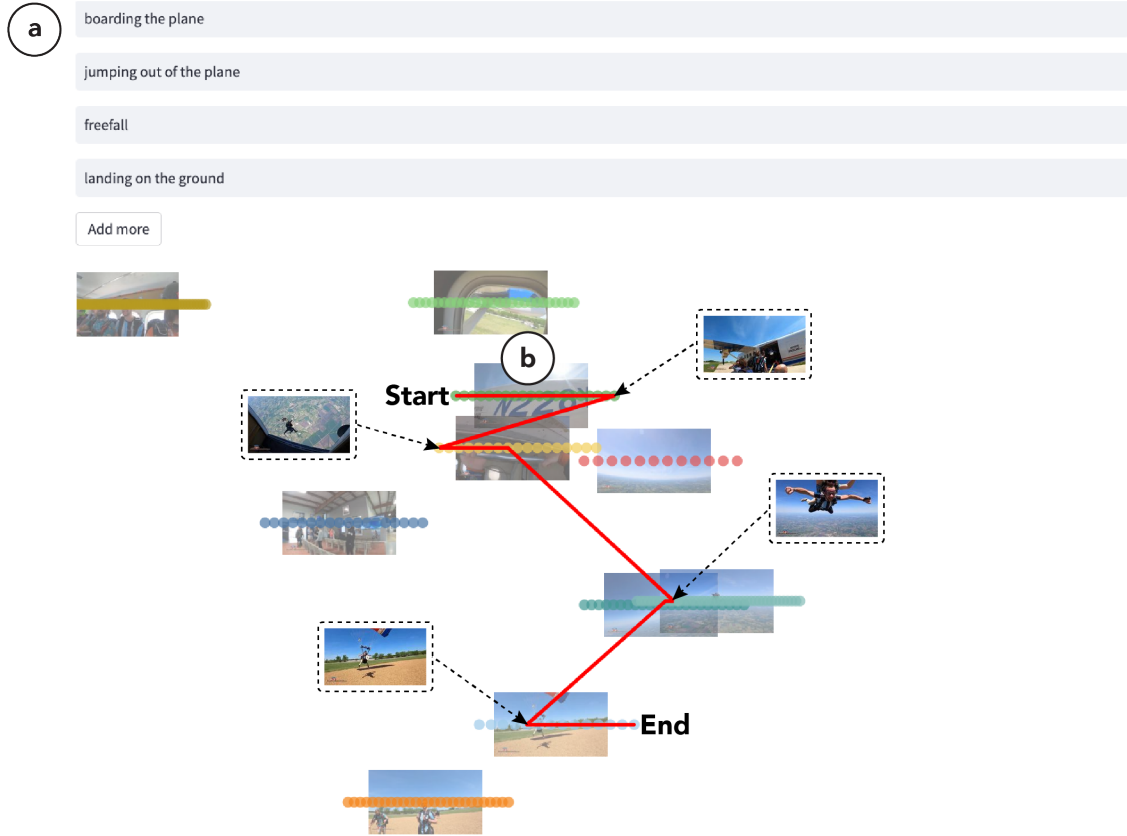


Fig. 9. VideoMap’s Route Planner can be extended to edit videos using text. The editor can describe a desired video using descriptive sentences, like writing a story (a). We then match each sentence to the closest video clip in the semantic space and generate a video by finding the shortest route along the clips (b). [See example video result.](#)

7.3 Text-Based Video Editing

We show how VideoMap Route Planner can be extended to support editors in text-based video editing (Figure 9). The editor can first describe the video they want to create using descriptive sentences, like writing a story (Figure 9a). For each sentence, we match the closest video in the semantic space, using the same CLIP-based method for comparing text and image embeddings as detailed in our implementation for Prompts (see Section 4.2.1). We then create the shortest route along the videos using the same dynamic programming approach as detailed in Route Planner (see Section 4.4) with an additional ordering constraint (i.e., in the order of the story) (Figure 9b) ([see example video result](#)).

8 FUTURE WORK

While VideoMap was positively received in our user study, there are several avenues for improvement that we plan to address for future work. First, VideoMap currently supports three types of lenses that facilitate the formation of narratives (semantic) and the creation of visual transitions (shape and color). In our study, participants suggested additional lenses, such as a cinematography lens to cluster videos by different shot types (e.g., by training a shot type

classification model), an Image Signal Processor (ISP) lens to cluster videos by white balance, focus, and exposure values (e.g., by utilizing image analysis algorithms), and a motion lens to cluster videos by distinct camera movements (e.g., by computing motion vectors with optical flow [46]). For future work, we plan to open-source the pipeline code for developers to build new lenses by generating new latent spaces that are meaningful for video editing. This pipeline could involve extracting embeddings using foundation models [10] or training custom models tailored to specific video editing objectives. Moreover, we could also explore the creation of “hybrid lenses.” For instance, we could create a lens with weightings of 80% shape and 20% semantics to enable the discovery of shape-matching cuts that also possess related semantics. Hybrid lenses could provide editors with the flexibility of blending together the properties of multiple lenses, allowing them to fine-tune the latent space according to their editing goals.

9 CONCLUSION

This research takes a step towards video editing exploration, brainstorming, and prototyping based on latent space representations of video frames. We distilled a set of principles grounded on editing practices and theories to guide our solution. Based on these principles, we built VideoMap, a proof-of-concept video editing interface that operates on video frames projected onto a latent space. We facilitate navigation of the latent space through map-inspired navigational elements and enable switching between different latent spaces using swappable lenses. We implemented three VideoMap components to assist editors in three editing tasks: exploring video footage, brainstorming suitable video transitions, and rapidly prototyping rough cuts. In a user study with both professionals and non-professionals, we found that VideoMap provides a user-friendly editing experience, reduces tedious grunt work, enhances the overview capability of video footage, promotes editing continuity, and enables a more exploratory approach to video editing. Finally, we implemented three extended applications to demonstrate how future developers may customize VideoMap for additional editing use cases. We hope that VideoMap can be enhanced and extended over time through the creation of new creative lenses and that it can be used as a building block to support the development of future video editing applications.

REFERENCES

- [1] 2023. *Descript — All-in-one video & podcast editing, easy as a doc*. Retrieved March 10, 2023 from <https://www.descript.com>
- [2] 2023. *Get directions and show routes*. Retrieved March 10, 2023 from <https://support.google.com/maps/answer/144339>
- [3] 2023. *Match Cuts and Creative Transitions with Examples — Editing Techniques*. Retrieved March 10, 2023 from <https://www.studiobinder.com/blog/match-cuts-creative-transitions-examples>
- [4] 2023. *Surprising Facts on The History of Video Editing*. Retrieved March 10, 2023 from <https://www.videoeditinginstitute.com/surprising-facts-on-the-history-of-video-editing>
- [5] 2023. *Type Studio — Edit Your Video By Editing Text*. Retrieved March 10, 2023 from <https://www.typestudio.co>
- [6] 2023. *Upwork*. Retrieved March 10, 2023 from <https://www.upwork.com>
- [7] 2023. *Working in the Project panel*. Retrieved March 10, 2023 from <https://helpx.adobe.com/premiere-pro/using/customizing-project-panel.html>
- [8] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.
- [9] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. 2022. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. In *27th international conference on intelligent user interfaces*. 746–766.
- [10] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).
- [11] Diogo Cabral and Nuno Correia. 2012. Videoink: a pen-based approach for video editing. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*. 67–68.
- [12] Diogo Cabral and Nuno Correia. 2017. Video editing with pen-based technology. *Multimedia tools and applications* 76 (2017), 6889–6914.
- [13] Juan Casares, A Chris Long, Brad A Myers, Rishi Bhatnagar, Scott M Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett. 2002. Simplifying video editing using metadata. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*. 157–166.

- [14] Renan G Cattelan, Cesar Teixeira, Rudinei Goularte, and Maria Da Graça C Pimentel. 2008. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 4, 4 (2008), 1–24.
- [15] Minsuk Chang, Mina Huh, and Juho Kim. 2021. Rubyslippers: Supporting content-based voice navigation for how-to videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [16] Boris Chen, Amir Ziai, Rebecca S Tucker, and Yuchen Xie. 2023. Match Cutting: Finding Cuts with Smooth Visual Transitions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2115–2125.
- [17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).
- [18] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. 2021. Automatic Instructional Video Creation from a Markdown-Formatted Tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 677–690.
- [19] Peggy Chi, Zheng Sun, Katrina Panovich, and Irfan Essa. 2020. Automatic video creation from a web page. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 279–292.
- [20] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. 2013. Democut: generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 141–150.
- [21] Brock Craft and Paul Cairns. 2005. Beyond guidelines: what can we learn from the visual information seeking mantra?. In *Ninth International Conference on Information Visualisation (IV'05)*. IEEE, 110–118.
- [22] Hai Dang and Daniel Buschek. 2021. GestureMap: Supporting Visual Analytics and Quantitative Analysis of Motion Elicitation Data by Learning 2D Embeddings. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [23] Klaus Eckelt, Andreas Hinterreiter, Patrick Adelberger, Conny Walchshofer, Vaishali Dhanoa, Christina Humer, Moritz Heckmann, Christian Steinparz, and Marc Streit. 2022. Visual exploration of relationships and structure in low-dimensional embeddings. *IEEE Transactions on Visualization and Computer Graphics* (2022).
- [24] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. 2019. Text-based editing of talking-head video. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- [25] Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox. 2000. A semi-automatic approach to home video editing. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. 81–89.
- [26] Dan B Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M Seitz. 2008. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 3–12.
- [27] Nicolas Grossmann, Eduard Gröller, and Manuela Waldner. 2022. Concept splatters: Exploration of latent spaces based on human interpretable concepts. *Computers & Graphics* 105 (2022), 73–84.
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [29] Marius Hografer, Magnus Heitzler, and Hans-Jörg Schulz. 2020. The state of the art in map-like visualization. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 647–674.
- [30] Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. 2003. AVE: automated home video editing. In *Proceedings of the eleventh ACM international conference on Multimedia*. 490–497.
- [31] Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. 2019. Learning to film from professional human motion videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4244–4253.
- [32] Yuzhong Huang, Xue Bai, Oliver Wang, Fabian Caba, and Aseem Agarwala. 2021. Learning Where to Cut from Edited Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3215–3223.
- [33] Bernd Huber, Hijung Valentina Shin, Bryan Russell, Oliver Wang, and Gautham J Mysore. 2019. B-script: Transcript-based B-roll video editing with recommendations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [34] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human-computer interaction* 1, 4 (1985), 311–338.
- [35] Dan Jackson, James Nicholson, Gerrit Stoeckigt, Rebecca Wrobel, Anja Thieme, and Patrick Olivier. 2013. Panopticon: A parallel video overview system. In *proceedings of the 26th annual ACM symposium on User interface software and technology*. 123–130.
- [36] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. 2008. Dragon: a direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 247–250.
- [37] Jeongyeon Kim, Daeun Choi, Nicole Lee, Matt Beane, and Juho Kim. 2023. Surch: Enabling Structural Search and Comparison for Surgical Videos. (2023).
- [38] Don Kimber, Tony Dunningan, Andreas Girgensohn, Frank Shipman, Thea Turner, and Tao Yang. 2007. Trailblazing: Video playback control by direct object manipulation. In *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 1015–1018.
- [39] Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph.* 36, 4 (2017), 130–1.
- [40] Mackenzie Leake, Hijung Valentina Shin, Joy O Kim, and Maneesh Agrawala. 2020. Generating Audio-Visual Slideshows from Text Articles Using Word Concreteness.. In *CHI*, Vol. 20. 25–30.
- [41] Clayton Lewis. 1982. *Using the "thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

- [42] Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. 2018. Embeddingvis: A visual analytics approach to comparative network embedding inspection. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 48–59.
- [43] David Chuan-En Lin, Fabian Caba Heilbron, Joon-Young Lee, Oliver Wang, and Nikolas Martelaro. 2022. Videogenic: Video Highlights via Photogenic Moments. *arXiv preprint arXiv:2211.12493* (2022).
- [44] Shusen Liu, Peer-Timo Bremer, Jayaraman J Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. 2017. Visual exploration of semantic relationships in neural word embeddings. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 553–562.
- [45] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. 2019. Latent space cartography: Visual analysis of vector space embeddings. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 67–78.
- [46] Bruce D Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, Vol. 2. 674–679.
- [47] Kevin Lynch. 1984. *Reconsidering the image of the city*. Springer.
- [48] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Swifter: improved online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1159–1168.
- [49] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
- [50] Allison Merz, Annie Hu, and Tracey Lin. 2018. ClipWorks: a tangible interface for collaborative video editing. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*. 497–500.
- [51] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2013. Direct manipulation video navigation in 3D. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1169–1172.
- [52] Alejandro Pardo, Fabian Caba, Juan León Alcázar, Ali K Thabet, and Bernard Ghanem. 2021. Learning to cut by watching movies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6858–6868.
- [53] Amy Pavel, Dan B Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 181–190.
- [54] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video digests: a browsable, skimmable format for informational lecture videos.. In *UIST*, Vol. 10. Citeseer, 2642918–2647400.
- [55] Yi-Hao Peng, Jeffrey P Bigham, and Amy Pavel. 2021. Slidecho: Flexible non-visual exploration of presentation videos. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–12.
- [56] Nicola Pezzotti, Thomas Höllt, B Lelieveldt, Elmar Eisemann, and Anna Vilanova. 2016. Hierarchical stochastic neighbor embedding. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 21–30.
- [57] Suporn Pongnunkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. 2010. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. 139–142.
- [58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- [59] Karel Reisz and Gavin Millar. 1971. The technique of film editing. (1971).
- [60] Ben Shneiderman. 2003. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*. Elsevier, 364–371.
- [61] Venkatesh Sivaraman, Yiwei Wu, and Adam Perer. 2022. Emblaze: Illuminating machine learning representations through interactive comparison of embedding spaces. In *27th International Conference on Intelligent User Interfaces*. 418–432.
- [62] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B Viégas, and Martin Wattenberg. 2016. Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469* (2016).
- [63] Tomas Sokoler, Håkan Edeholt, and Martin Johansson. 2002. VideoTable: a tangible interface for collaborative exploration of video material during design sessions. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. 656–657.
- [64] Robert Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (1953), 267–276.
- [65] Ruo-Feng Tong, Yun Zhang, and Meng Ding. 2011. Video brush: A novel interface for efficient video cutout. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 2049–2057.
- [66] Yoshinobu Tonomura, Akihito Akutsu, Kiyotaka Otsuji, and Toru Sadakata. 1993. Videomap and videospaceicon: Tools for anatomizing video content. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*. 131–136.
- [67] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [68] Kevin Wang, Deva Ramanan, and Aayush Bansal. 2021. Video-Specific Autoencoders for Exploring, Editing and Transmitting Videos. *arXiv preprint arXiv:2103.17261* (2021).
- [69] Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, Ariel Shamir, et al. 2019. Write-a-video: computational video montage from themed text. *ACM Trans. Graph.* 38, 6 (2019), 177–1.
- [70] Tinghui Wang, Andrew Mansfield, Rui Hu, and John P Collomosse. 2009. An evolutionary approach to automatic video editing. In *2009 Conference for Visual Media Production*. IEEE, 127–134.

- [71] Haijun Xia, Jennifer Jacobs, and Maneesh Agrawala. 2020. Crosscast: adding visuals to audio travel podcasts. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*. 735–746.
- [72] Haojin Yang and Christoph Meinel. 2014. Content based lecture video retrieval using speech and video text information. *IEEE transactions on learning technologies* 7, 2 (2014), 142–154.
- [73] Saelyne Yang, Jisu Yim, Aitolkyn Baigutanova, Seoyoung Kim, Minsuk Chang, and Juho Kim. 2022. SoftVideo: Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data. In *27th International Conference on Intelligent User Interfaces*. 646–660.
- [74] Zijun Yao, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the eleventh acm international conference on web search and data mining*. 673–681.
- [75] Jamie Zigelbaum, Michael S Horn, Orit Shaer, and Robert JK Jacob. 2007. The tangible video editor: collaborative video editing with active tokens. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction*. 43–46.