Katika: An End-to-End System for Authoring Amateur Explainer Motion Graphics Videos

Amir Jahanlou
Simon Fraser University
Surrey, BC, Canada
amir_jahanlou@sfu.ca

Parmit K. Chilana
Simon Fraser University
Burnaby, BC, Canada
pchilana@cs.sfu.ca

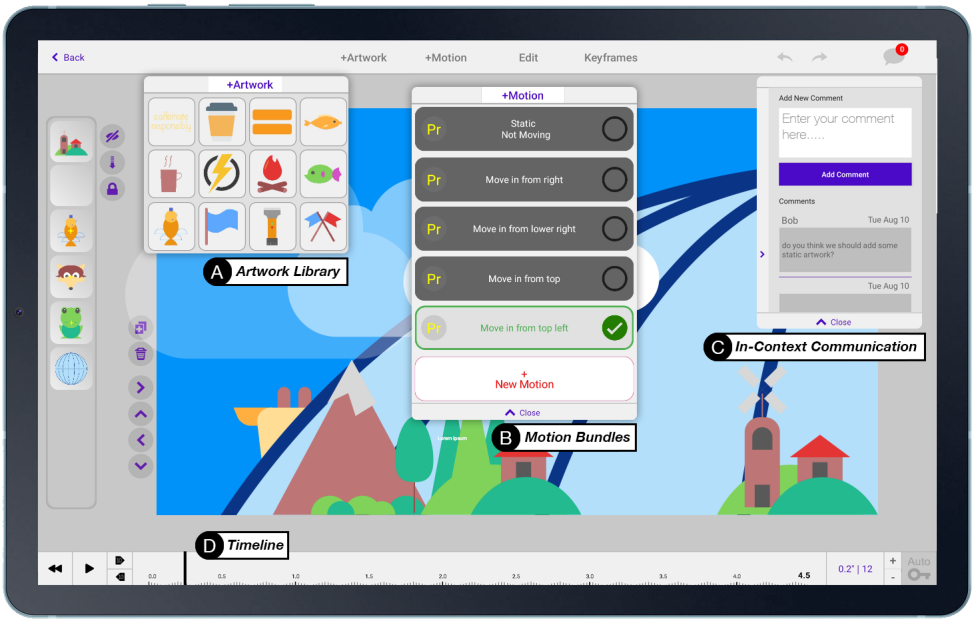

Figure 1: Katika is an end-to-end explainer motion graphics video authoring software that allows amateurs to understand and follow the key stages of motion design. This figure shows Katika's graphics editor view that includes: A) an embedded library that suggests artworks based on script , B) example motion bundles that can be directly applied on the artworks or further edited, C) in-context communication features and, D) a timeline for editing the animations.

ABSTRACT

Explainer motion graphics videos that use a combination of graphical elements and movement to convey a visual message are becoming increasingly popular among amateur creators in different domains. But, to author motion graphics videos, amateurs either have to face a steep learning curve with professional design tools or struggle with re-purposing slide-sharing tools that are easier to access but have limited animation capabilities. To simplify the process of motion graphics authoring, we present the design and implementation of Katika, an end-to-end system for creating shots based on a script, adding artworks and animation from a crowdsourced library, and editing the video using semi-automated transitions. Our observational study illustrates that participants (N=11) enjoyed using Katika and, within a one-hour session, managed to create an explainer motion graphics video. We identify opportunities for future HCI research to lower the barriers to entry and democratize the authoring of motion graphics videos.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**.

KEYWORDS

Explainer Videos, Motion Graphics, Infographics, Motion Design

ACM Reference Format:
Amir Jahanlou and Parmit K. Chilana. 2022. Katika: An End-to-End System for Authoring Amateur Explainer Motion Graphics Videos. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3517741

SIGCHI, April 30– May 06, 2022, New Orleans, LA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3517741

1 INTRODUCTION

Explainer motion graphics videos are short animated videos that demonstrate complex concepts using a combination of graphical elements, movement, and sound [3, 70, 78]. Unlike static forms of illustrations or infographics, these videos employ engaging artwork that can morph, shift, rotate, and transition in different ways to visually convey a message or a story [13]. Such videos can communicate topics in fields where videography is challenging (or perhaps even impossible) [11], making them popular in domains such as marketing, education, public health, and others [84, 96]. For instance, Figure 2 shows frames from an example explainer motion graphics video about vaccine production. These videos are usually authored by professional motion designers who have formal training in some aspect of animation, graphic design, or video production [9]. However, in recent years, the increasing demand for explainer videos —coupled with readily available online publishing platforms— has inspired more amateurs to tinker with tools of motion graphics video authoring [85]. Unfortunately, amateur motion designers often face a time-consuming and challenging learning curve with the currently available professional motion graphics tools [44, 57].

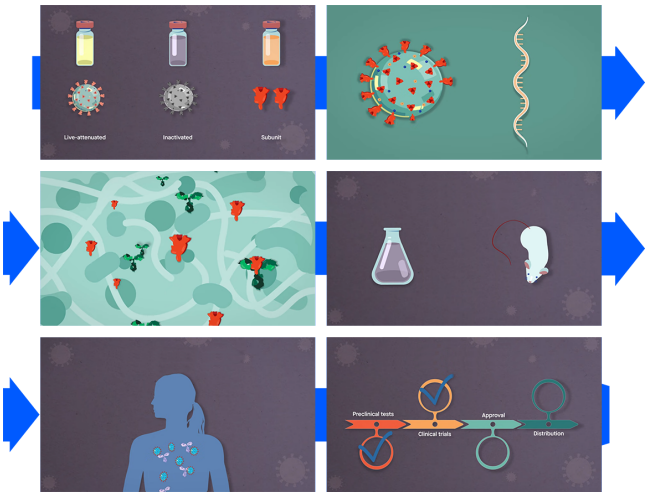

Figure 2: Sample frames from an explainer motion graphics video about the production and validation of vaccines. (Retrieved from: https://www.youtube.com/watch?v=uWGTciX795o)

There are several key reasons why motion graphics video authoring is particularly challenging for amateur motion designers. First, creating motion graphics videos involves multiple feature-rich tools across the pre-production to post-production stages that are usually disconnected from each other [44, 80]. Different tasks such as writing the initial script and dividing it into shots, finding the appropriate imagery and artwork, or combining the shots with transitions in a single video, are executed in separate software tools [44]. Moreover, unlike camera-recorded videos where the key content comes from the raw footage, motion designers face a blank canvas and have to populate the content of each frame using imagery or artworks, which can be challenging to make from scratch or locate online [9, 22]. Perhaps the most challenging aspect for amateurs is creating moving graphics that require competencies in animation techniques such as keyframing or interpolation [34, 44]. The combination of such barriers results in an environment in which amateurs get discouraged and even give up on their pursuit of creating explainer motion graphics videos.

Instead of working with feature-rich motion graphics software, some users may try to explore slide creation tools, such as Microsoft PowerPoint [74], Google Slides [81], or Apple Keynote [7], that have basic animation capabilities. However, the onus is on the user to repurpose such tools to create motion graphics videos and make decisions about how to segment their content into shots, locate relevant artwork externally, figure out if the desired motion effects are even possible, or assemble shots into a video. Moreover, slide creation tools mostly only offer animations in the form of transitions [20] that can only be applied to an artwork as a whole and there is limited room for editing the motions or adjusting timings.

In this paper, we present the design and implementation of *Katika*, a novel end-to-end tool for creating explainer motion graphic videos that bridges the authoring activities of pre-production, production, and post-production and allows amateurs to learn and understand the various steps within each stage. We are using a human-computer interaction (HCI) approach to lower the barriers to entry for amateur motion designers by taking into account the actual workflows needed to produce a motion graphics video and offering a simplified walk-up-and-use interface. Katika addresses the key design challenge of end-to-end authoring by using a combination of semi-automated approaches (e.g., dividing a script into shots, assembling shots into a cohesive video) and community-based practices (e.g., creating content through examples shared by other users in an embedded library). Furthermore, Katika provides an in-context collaboration feature which makes it easy for amateurs to seek help and feedback at any stage of the process.

Another key challenge that Katika tackles is empowering amateurs to create animated graphics using a novel *motion bundles* approach. Current techniques of animation mainly depend on the notion of keyframing [9]. Such methods, however, are tricky for amateurs as they require an underlying understanding of concepts such as time/space integration, channels, frame rate, or interpolation [53, 78]. It can be particularly difficult for amateurs to anticipate the next state (such as position, shape, color, etc.) and animate the artwork between the current and the new state. With our design and implementation of motion bundles in Katika, motion is treated as an independent entity that can be shared across various artworks. This technique allows the animation to be applied to the artwork, a sub-object of the artwork, or copied from one artwork (or object) to another. Katika's motion bundles allow users to easily browse a library of example motions, apply a motion to their artwork and immediately view the animated artwork without struggling with keyframing or envisioning the possibilities. Moreover, users can also view the beginning and end of a motion on a timeline which provides a visual context for working with and adjusting the animation. Furthermore, users can create, edit, and share their own motions with others to grow Katika's example motion library.

To evaluate the concept of end-to-end explainer video authoring using motion bundles, we ran an observational study with 11 participants with varying degrees of familiarity with motion graphics, explainer videos, and video production. We found that within a

one-hour session, all participants (even those without any previous experience) could understand the steps of the authoring process and create an amateur explainer motion graphics video. Furthermore, during the post-task questionnaire and interviews, participants appreciated the freedom that Katika offered for exploring different designs and animations and were keen to re-use the system to produce their own explainer videos for presentations, conferences, and other similar contexts.

In this paper, we make three new contributions:

- Katika as an end-to-end tool for authoring motion graphics videos that facilitates the process of understanding and creating a motion graphics video without requiring external applications. Its minimal interface supports key workflows for creating motion graphics videos and includes built-in guidance for generating shots, selecting artworks based on a script, adding and editing example animations, and producing the final video. Katika also allows users to see what steps they still have to complete, minimizing the number of decisions that beginners have to make to produce their motion graphics videos.
- A novel approach to end-user animation using modular motion bundles that abstract away keyframing or interpolation and simplify the process of adding motion to artworks with a single click. These motion bundles can be applied to artworks as a whole or their constituent parts. They can be viewed and edited on a corresponding timeline (or curve editor), enabling highly detailed animations while not raising the usability threshold. Furthermore, Katika provides a library of motions that can grow with community contributions and allow for greater creative expression.
- Insights from an observational study demonstrating how people could learn the motion graphics design process and create amateur explainer videos in less than an hour using Katika and how they found the tool to be useful and intuitive.

Although recent works in HCI have been drawing attention to the importance of motion graphics in various informational domains [5, 44, 80], it is challenging to democratize the creation process unless we lower the barriers for amateurs. This paper argues that a walk-up-and-use interface (with embedded content, example motions, built-in communications, and automated features) is one way in supporting amateurs to better understand and engage in the authoring process. We envision a future in which users from different backgrounds and skill levels can use such interfaces to create explainer motion graphics videos with ease.

2 RELATED WORK

This research builds upon prior work related to amateur video production, the authoring of informational videos, and techniques for simplifying animation authoring.

2.1 Innovations in Amateur Video Production

The increasing demand for videos in various domains [12, 24, 32] has encouraged researchers to explore interfaces for improving video productions [79]. Much of the literature focuses on simplifying post-production efforts, such as video editing, where the main content comes from raw camera footage. However, such footage may not always be readily available. To tackle this, innovations have emerged in generating screen recordings [17], supporting home video creation [35], and supporting reuse of content [59]. While such methods focus on the initial video footage, a related challenge is the issue of sequencing the video content [64]. While almost all state-of-the-art tools (such as iMovie [43] or Adobe Premiere [75]) provide advanced editing timelines, amateurs have difficulty dividing their content into separate scenes and finding the appropriate order between them [39, 44]. To address such challenges, systems have been developed to suggest candidate next scenes [39, 67], assemble a video based on examples [69], and edit recorded narratives [88]. While such approaches are helpful, they usually target only one aspect of video production, and it can be difficult for beginners to contextualize the entire pre- to post-production processes.

Another emerging innovation in amateur video production is the automation of content creation. For example, researchers have explored automatic event timelines [14], video creation using recommendations [52], or automating text-to-video content creation [38, 77, 82]. Many of these approaches rely on algorithms such as word concreteness [56] or content analysis [39, 94, 95]. While such systems automate the entire process by mining the text transcript for keywords, some argue that there should still be a balance between automation and user control, especially in the case of authoring informational motion graphics videos [35].

Unlike prior innovations, Katika provides end-to-end authoring for amateurs by bridging pre- to post-production processes within the same user interface. Although Katika relies on some semi-automated techniques to assist users with content creation, the user has full control in exploring and adjusting the suggestions made by the system as they finalize their design decisions. Using walk-up-and-use approaches in Katika, our goal is to enable informational video authoring with the same ease as some of the entertainment authoring tools (e.g., TikTok [87]).

2.2 Authoring Informational Videos

One industry standard for authoring informational videos is the use of motion graphics. Recent works [5, 41, 80] illustrate how authoring a motion graphics video is a challenging task, and creating a minute-long video can take up to two days [71]. With more amateurs attempting to author motion graphics videos [9], researchers have been exploring different dimensions such as techniques for motion graphics data videos [4], methods of creating more engaging content [54, 61], or analyzing the space of motion graphics design [80, 85]. Trained individuals usually learn the intricacies of image/graphics editing or software applications [53] and the language and terminology used in such tools [34]. On the other hand, amateur motion designers are intrinsically different from expert motion designers [44] as they are often experts in different domains and expect that using the tools would not compete with their domain knowledge [58]. However, the state-of-the-art tools used in authoring these videos (Adobe After Effects [2], Autodesk Maya [63]) are not easy to use. Over several decades, these tools have evolved to serve high-end productions such as film or advertising [53], resulting in feature-rich applications with laborious learning curves [19]. To make matters worse, a combination of several such tools are often required to complete a motion graphics video [34],

and amateurs face a challenge with the lack of integrated authoring environments [44].

Moreover, informational video authoring often begins with blank canvases [76]. Techniques of template-based authoring [53] have been suggested to mitigate such blank canvases and tools such as Animaker [6], Vimeo [90] or Doodly [29] provide users with complete scenes. However, research shows that adjusting such templates to the specifications of informational videos (such as presentation style, imagery, icons, and the overall space) is not always as straightforward [44, 80]. Another challenge is the issue of collaboration [28, 73] and getting appropriate feedback. Despite research into collaborative video authoring in different environments [51, 83], such capabilities are yet to be widely available in the state-of-the-art video editing tools [28, 53].

Beyond tools specific to motion design, some users may tinker with slide creation tools (e.g., PowerPoint or Keynote) to author motion graphics videos. However, to create compelling videos, users have to know how to repurpose these tools for the different pre- to post-production steps and will still need to use external tools to make artworks or export slides into a video. Such tools lack provisions to view the content at the current time, which means that the user cannot know when an animation begins or ends. Moreover, the nature of the animations (e.g., keyframes) cannot be modified in such tools.

In summary, options for creating motion graphics videos are at the two far ends of a spectrum: users have face a steep learning curves with specialized tools or struggle with re-purposing slide sharing tools that are easier to access, but have limited animation and end-to-end authoring capabilities. Our approach in Katika has been to design a simplified middle-level interface to ease the steps of making explainer motion graphics videos. It enables a broader class of amateurs to understand and follow the steps of motion graphics design and tell compelling stories using explainer videos.

2.3 Techniques for Simplifying the Animation Process

The widespread adoption of animation in fields such as feature film, motion graphics, or video authoring has motivated researchers to investigate ways to simplify its creation. Producing an animation, in essence, is the integration of time and space [53, 78] which is an inherently complicated task. To address such difficulty, K-sketch [26], for example, explores the idea of an animation library. However, this technique is confined to a specific narrative, limiting its application to a broader scope. Moscovich et al. [66] employed hand gestures to record animations that depends on the user's understanding of the timing. Another popular method has been sketching [91], in which software-made frames expand a series of user-generated sketches. Other sketch-based attempts have explored defining the position of objects [27] or movement paths [76, 86]. Such methods have some application for directional (e.g., side to side) movements. Still, they have an evident deficit in the user's ability to draw and actively imagine the upcoming state of the animation.

Transferring motion is another popular approach in animation. Studies have explored transferring a pose from a stick-figure sketch [25] or using video footage [92] to convey a pose for a character. Kazi et al. [46] explored how an amplified motion can be an approach to generate an animation from a very basic input. These techniques introduce their own limitations, such as understanding the application of basic animation and amplified motions. Other works have also explored animation created from a series of pre-built content [68] tailored to a specific narrative or relational animation [45] to make the dynamics between various objects.

Most of these prior efforts in simplifying the animation process rely on the user's ability to envision the next stage of animation, which is still a difficult task. To relieve amateurs from such burden in creating their motion graphics videos, we introduce a new paradigm of *motion bundles* in Katika. These example-based animations allow users to experiment with predefined motions (e.g., moving an object, morphing a shape) and help produce immediate gratification. Moreover, users can edit these predefined motion bundles, add new ones, and share their motion bundles with a community of users, ensuring the longevity of this example-based technique.

3 MOTIVATION AND DESIGN CONSIDERATIONS

Our motivation in this research is to lower the barriers for amateurs to both understand the process of motion design and create their own explainer motion graphics videos. Current state-of-the-art tools are feature-rich motion graphics design software (e.g., After Effects, Maya) that enable high-quality productions, but have a steep learning curve. Although it may be easier for amateurs to get started with slide-creation tools with basic animation capabilities (e.g., PowerPoint, Keynote), such tools can be difficult to repurpose to support all of the different stages of motion graphics design. Amateur motion designers need walk-up-and-use tools that support motion design workflows, minimize cognitive load, and allow amateurs to focus on conveying their informational message [34, 44]. We propose the following five key design considerations (DCs summarized in Table 1) for designing a tool that supports the process of motion graphic design for amateurs:

(DC1) Facilitate End-to-End Authoring: Producing any video consists of a three-stage process of pre-production, production, and post-production [33]. However, state-of-the-art video production tools often have limited or no provision for pre-production [9, 44]. As a result, creators have to rely on ad-hoc solutions to tackle pre-production tasks such as writing a script. Moreover, they need to employ separate feature-rich software applications [44, 53] for different stages of creating a motion graphics video, which increases the learning complexity and the cost of task switching. For example, a designer may write a script in a text editing tool, create artworks in an image editing software (Adobe Photoshop [72] or Illustrator [42] and similar) and assembles the artworks and animation in another application (e.g., After Effects). Integrating all of the key steps of producing an explainer motion graphics video within a single software application can help tackle such challenges.

(DC2) Minimize Complexity and Steep Learning Curves: Each stage of creating an explainer motion graphics video involves complex steps, such as dividing a script into separate scenes, coordinating color themes, adding content, or editing individual shots into a cohesive video [34, 44, 53]. New users (and even experienced designers) can face an overwhelming learning curve in completing

Table 1: Key considerations for designing a tool that supports the process of motion graphic design for amateurs.

No.	Design Consideration
DC1	Facilitate End-to-End Authoring
DC2	Minimize Complexity and Steep Learning Curves
DC3	Facilitate Access to Example Content
DC4	Lower the Barriers for Animating by Using Examples
DC5	Foster Easier Communication and Collaboration

such steps in feature-rich applications [44, 80]. The complexity of these steps can further discourage amateurs from experimentation and trial-and-error approaches for learning [49, 65]. To empower beginners, tools should embrace minimalism and simplify the user interface to lessen the difficulties (e.g., using automated techniques) while still offering control over various steps.

(DC3) Facilitate Access to Example Content: Explainer videos depend on elements such as icons, images, or pictograms [34]. However, amateurs are usually not trained to create such content, and beginning with a blank canvas can negatively affect their confidence in getting started [76]. While some pre-made content is available in online repositories, finding and editing such content is not always easy [44]. To alleviate such challenges and facilitate access to content, it could be helpful to include content (e.g., artwork) within the software tools. The built-in content can be further expanded to ensure longevity and diversity by using crowdsourcing or community-based approaches [18, 55].

(DC4) Lower the Barriers for Animating by Using Examples: Animating content is a tedious task that requires understanding keyframing or interpolation techniques [53, 78]. While beginners have difficulty with such practices, even trained animators need long sessions to create their animations [46]. While trained animators can at least envision the animation, amateurs have difficulty imagining what to make in the first place. One approach to lowering the barriers to animating is using an example-based approach and offering predefined animations. In this technique, users could browse animation examples to learn what they could achieve. The durability of this approach can be expanded by using crowdsourcing or community-based practices for including further animation examples.

(DC5) Foster Easier Communication and Collaboration: When working on a video project, creators often need to communicate with others about the content or alternative designs. However, it can be difficult to use traditional communication methods (e.g., email or text) that are disconnected from their workspace [44, 48]. Amateurs can benefit from context-specific help or feedback but face arduous processes like exporting videos or sharing and seeking feedback on separate platforms. Similar to tools that allow collaborators to communicate directly within the application [18, 89], it can be helpful for amateur motion designers to have such access and eliminate the need for 3rd party applications.

4 KATIKA

We designed Katika, a novel approach for the end-to-end authoring of amateur explainer motion graphics videos based on the above design considerations. Katika offers an integrated authoring system for writing scripts and creating shots, browsing an embedded library of built-in and crowdsourced artworks, applying animations through motion bundles, and collaborating in context. Moreover, we included various automated features such as dividing a script into separate shots, suggesting artworks based on a shot script, and editing shots into a cohesive video using transitions. In what follows, we describe the user interface design of Katika, a sample usage scenario, and the key features of Katika that tackle the challenges outlined above.

4.1 Overview of Katika's User Interface Design

During the design process of Katika, we continually sought the opinion of several interface designers, HCI specialists, and motion designers. As a result, we opted for lowering the number of the "actionable" views to three (that reflect the industry-standard video authoring of pre-production, production, and post-production [33]). These include: the script editor (Figure 3-left), a graphics and motion editor (Figure 3-middle) and a video editor (Figure 3-right). To describe the workflow of creating an explainer motion graphics video using Katika, imagine Emily, a journalist who has written a short script about electronic waste and wants to convert it into an explainer video. In a walk-up-and-use fashion, Emily opens Katika, starts a new project and can now enter a script for her video. She can then use Katika's built-in feature to divide the script into separate shots with a single click (Figure 3-left). This shot list contextualizes the project through thumbnails and allows Emily to choose a color theme that helps produce cohesion throughout the video. Next, Emily selects a shot and the canvas (Figure 1-A) provides her a wide array of editing tools and makes it easy for her to add artworks by browsing the embedded library and assembling a collage that statically represents the concept of this shot. Emily then chooses to animate (some of) the artworks. She explores different movements for each artwork by selecting motion bundles from the library (Figure 1-B). Emily repeats this process for all the shots and proceeds to the video editing (Figure 3-middle), where all shots are automatically connected using transitions, and a timeline represents the entirety of the video. In this view, she can also add background music, sound effects or record a voiceover for the video and view the video emerged from all shots. Finally, Emily can choose to view or export the video.

4.2 Key Features and Functionality of Katika

Katika addresses the key design challenge of end-to-end authoring by using a combination of semi-automated and community-based approaches. It further innovates on amateur animation techniques by introducing a novel concept of motion bundles and provides in-context communication and collaboration features.

4.2.1 Script Automation and Generation of Shots. With a single click, Katika divides a script into separate shots with continuous yet independent messages (Figure 3-left). These shots are automatically assigned a length (in seconds) and the timeline (Figure 1-D) adapts itself to their duration. The user can adjust the text of each shot, and Katika automatically re-calculates the length of that shot. Moreover, since color plays a significant role in producing consistency, Katika automatically applies a background color to each shot of

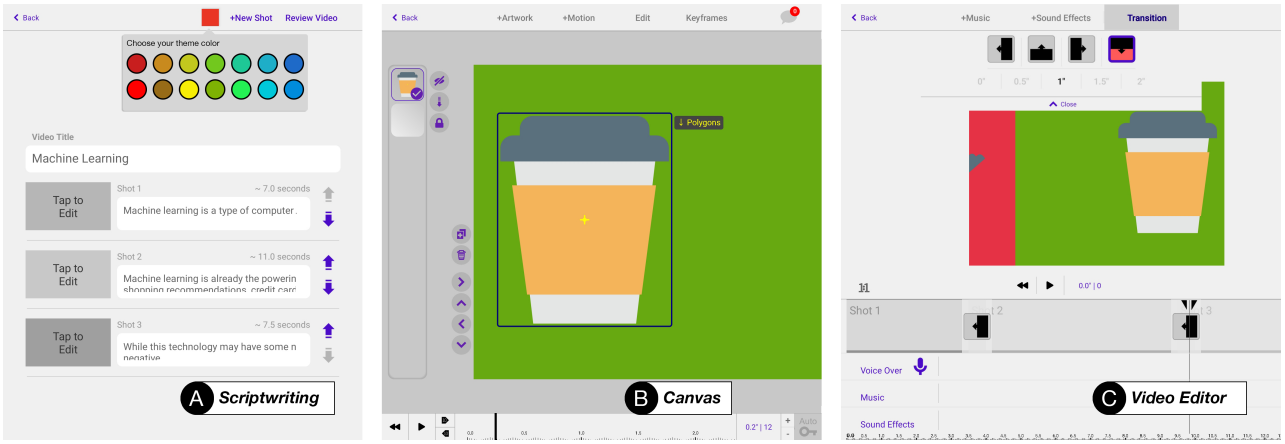

Figure 3: Katika's main panels consists of A) a scriptwriting editor and shots view where users can enter the story of their videos in text and automatically generate shots (left); B) a main graphics and motion editor that includes various features for adding and editing artwork and motions (middle, shown in Figure 1 in more detail); and, C) the video editor that automatically assembles a timeline of all the shots with different transitions while still allowing the user to make adjustments (right).

the video. The user selects the main theme color (Figure 3-left), and the software applies a color (by 15% variation from the central color) to all shots while still allowing the user to override each shot's color independently (DC2). Finally, a video is assembled by adding various shots using transitions. These transitions vary in style and length and are assigned automatically between adjacent shots. The user, however, can choose to adjust their style or length individually (Figure 3-right).

4.2.2 Embedded and Scalable Artwork. Previous research [34, 44, 53] and our own informal analysis of a corpus of more than 50 explainer motion graphics videos on YouTube and Vimeo, suggests that to author a motion graphics video, access to an extensive repository of editable artworks is necessary (DC1). We embedded a library of artworks in Katika (DC3) based on a freely available repository [62]. Being scalable vector graphics (SVGs), these artworks (Figure 1-A) allow a high degree of editability. Moreover, users can choose to import any of their own SVG files as artworks or benefit from artworks added and shared by a community of users. We further tagged and enabled search for the artworks. Katika automatically runs a search against the shot's keywords and suggests relevant artworks (DC2). These keywords are extracted from the shot's script and, similar to previous systems [39, 56, 94, 95], connect the video content to the script. However, Katika only offers the artworks as recommendations and the user has control over choosing other artworks. This approach is essential as previous works [9, 22] have illustrated that beginning with a blank canvas negatively affects the confidence of video creators. It is worth noting that Katika's approach of examples differs from the templates approach [85]. While template-based authoring relies on adjusting large templates, Katika depends on small modular pieces. Such components can be assembled in different contexts or merged to produce new ones.

While having examples simplifies the authoring process, there are many times when the user would need to copy content from another project. In Katika, any content, such as artwork, polygon, motion, keyframe, shot, or even the whole project, can be cloned in its entirety (DC2). Using this approach, if the user finds something that could help with their project, they can clone it to their own library and adjust it to represent their intention.

4.2.3 Modular Motion Bundles. In our first iteration of Katika, we had followed the standards of adding animation, such as changing time, adding keyframes, and adjusting interpolations. Once we had a stable prototype, we observed four users, who had no experience with motion design, use Katika informally. We found that although the end-to-end nature of the tool was empowering, the process of animating was a significant bottleneck. The difficulty with the animation illustrated two crucial factors: 1) the animation approach should require minimal effort or input from users (DC4); and 2) while beginners face technical challenges with animating, they also find it difficult to imagine how a different state or duration can be used to represent a concept (DC2).

Such observations made us realize that we need a new approach to animating for beginners. Investigating the previous animation methods (such as animation sketching [27], motion amplifiers [47], and motion transfer [92]), we learned that such approaches are either dependent on user input or are challenging to scale to a broader scope. We then reflected on the practices of experienced motion designers [34] that illustrate how they continually watch examples created by others in pursuit of broadening their horizons. Combining this strategy with the requirements of the amateur creators, we invented the concept of motion bundles to enable users to explore animations by examples.

How Motion Bundles Work: We designed our novel approach of motion bundles such that the animation of an artwork (including those of the various hierarchy levels) is consolidated in a single entity (Figure 1-B). Upon request, this bundle can be applied to another artwork or its constituent parts. Using this strategy, users can apply a motion to artwork and animate it accordingly. There is no limitation to the complexity of the animation in this approach, and even advanced morph animations with hundreds of keyframes can be bundled.

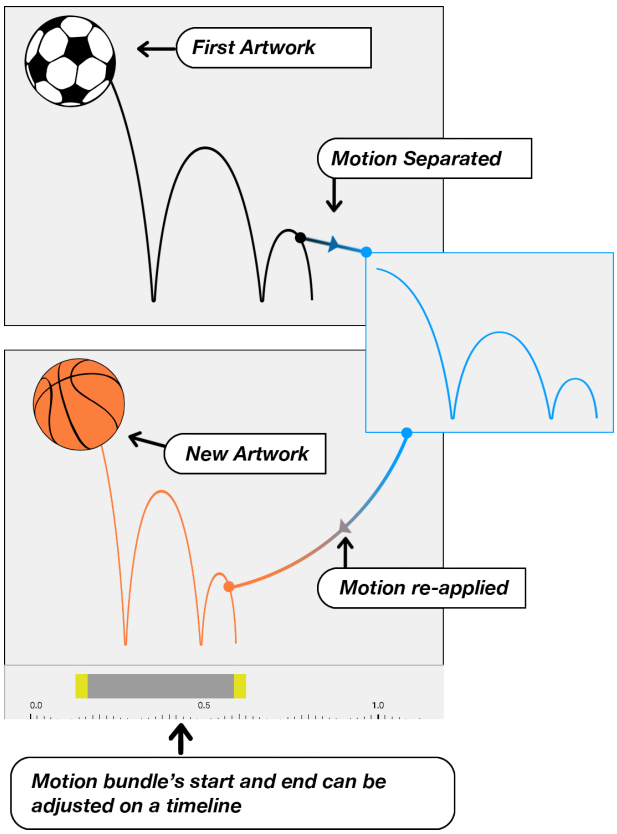

Figure 4: This example shows the process of separating the motion from one artwork (the soccer ball) and applying it to another one (the basketball). Users can choose to adjust the motion by simply dragging the motion (or its start/end) on timeline. This way, Katika treats the motion as an independent entity.

A simple example would be animating a bouncing soccer ball and then applying that same motion to a basketball (Figure 4). In such a process, the user still has complete control over the duration (or start and end) of a motion, and can choose to clone a motion bundle into their private library to edit each keyframe or channel in its entirety. While this approach depends on the use of keyframes, it abstracts away the intricate parts of the process. For beginner users, the task is limited to selecting the names of motion bundles until they find one that matches their intention (DC3 and DC4).

Built-in Motion Bundles: In designing the user interface options, we opted for terminology that could be more relatable for beginner users. This was done based on previous studies [1, 21, 44], highlighting the difficulty of understanding domain-specific terminology for beginners (DC1). For instance, instead of "alpha," we used "visibility," and instead of "stroke," we used "border." Below, we provide a list of the motion bundles offered by default in four categories of swipe, pop, travel, and shake:

- Swipe in from left, right, top, or bottom
- Appear by pop at top-left/right, bottom- left/right, middle-left/right, bottom-top/center

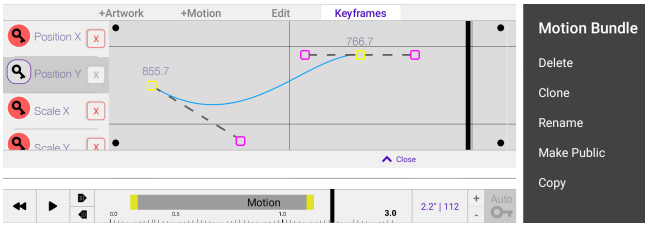

Figure 5: Katika's curve editor allows adjustments on keyframes such as changing their time, value and interpolation. Users can isolate various channels and work on them independently. Users also have the option to clone a motion bundle or make it public.

- Travel from left-right, right-left, bottom-top, or top-bottom (while choosing the starting point)
- Shake and appear in various locations

Katika's built-in motion bundles also make it possible to apply and merge multiple motions on the same artwork. This approach allows for a quick exploration of different ideas (DC2). Beyond using the built-in motion bundles, advanced users can also create their own motion bundles. To enable such creation, we have included various animate-able channels. Users can choose to make a new motion bundle (Figure 5-Right). Upon adding keyframes or adjusting curves (Figure 5-Left), they can decide to make their motion bundle public, which the system would make available to all users.

4.2.4 Built-in Collaboration and Communication . With DC5, our goal was to foster easier collaboration as sharing video projects is inherently a difficult task [28]. A video project often has numerous shots, artworks, transitions, and animations presented at different parts of the software tool. In Katika, everything is on a cloud server, and effectively all collaborators of a video are working on the same "live" project (DC5). This inclusive synchronized collaboration throughout all the steps is not presently available in any of the state-of-the-art software tools of motion design. Such a collaborative nature is beneficial for beginners who can delegate technically challenging tasks to others. Moreover, Katika allows users to add comments on various object levels such as artworks, motion, or shots and makes such comments discoverable in context (Figure 1-C). Katika provides non-intrusive notifications that inform users of the total number of adjustments or comments to make it easier to see responses or new comments.

4.3 Implementation

Katika was developed to work on tablet devices (with Android OS | SDK version 27 and above), which benefit from using various gestures. Doing so furthers the notion of amateur video authoring as many new creators may not have access to powerful desktop computers. We next describe the implementation of Katika's core features and functionality related to script-writing and shot breakdown, embedded artwork, modular motion bundles, and built-in collaboration.

4.3.1 Automatic Script Breakdown Algorithm. To implement the shot breakdown feature of Katika, we reflected on the practices of experienced motion designers [5, 44, 80]. We learned that they

assign the beginning and end of shots upon parameters such as the length of the script for each shot, or the presence, complexity, and the number of keyword terms (that would require specific artwork and animation), and the location of various punctuation. We wrote a simple algorithm that first divides the story into pieces based on the placement of the punctuation, such as comma or period. It then assesses the complexity of each segment based on the number of words and their complexity. We included libraries of four word classes of pronouns, prepositions, conjunctions, and interjections to assess complexity as is common in other natural language processing algorithms [15, 37]. The algorithm removes all instances of such word classes and counts the remaining keywords. If the number of words in a shot is 50% lower than the average, the algorithm joins it with the previous or the next one. If the number of the words is 50% higher than the average, the algorithm keeps the shot but prompts the user to consider dividing it into two. Upon generating all of the shots, Katika assigns a length (in seconds) to each shot by evaluating the total number of the words multiplied by an average time to speak a word.

4.3.2 Scalable Artworks Details. Using The keywords extracted from the script and assigned to each shot, Katika runs an implicit search within the built-in artwork library. Artworks are tagged and the search algorithm matches the script keywords with them to suggest relevant artworks based on the highest similarity scores. To develop this artwork library, we considered several approaches such as image libraries, icons, PNG files with transparent backgrounds, or simple shapes (such as those in tools like PowerPoint or Keynote). After much consideration, we settled on highly customizable vector graphics and developed a parser that converts SVG files into an internal format to enable high editability. The implementation considered matters such as SVG groups and objects, referred to as "artwork" and "polygons." In this approach, the separate pieces of the SVG are represented as polygons (with corners and including body and stroke color/width). In essence, each artwork is a higher-level object that consists of a combination of one or more polygons. The parser converts a stream of SVG files from a freely available repository (Magicons) into editable artworks and extracts the related tags. Moreover, the implementation trickles down an artwork's transformations (position, scale, rotation, transparency) on its polygons, while each polygon can have independent shape, color, stroke color, or stroke width. Since the artworks are editable, users can modify artworks (Figure 6) and share them back as new ones. Beyond this built-in and growing library, users can also choose to import any of their own SVG files as artwork.

4.3.3 Motion Bundles Technical Implementation. We use the term motion bundle due to the multiple layers of motions in play. In Katika, each object (e.g., an artwork or an underlying polygon) has a "motion" attached to it. Depending on the type of object (artwork or polygon), the motion will include different animate-able channels. For artworks, channels include position (on x and y axes), rotation (on the z-axis), scale (on x and y axes), and transparency (as a float value between 0 and 1). On the other hand, the polygon channels consist of shape (represented by an array of corners and control points each represented on x and y axes), body and stroke color (represented in red, green, blue, and alpha floating values), and a stroke width (as a float value equal or above to 0).

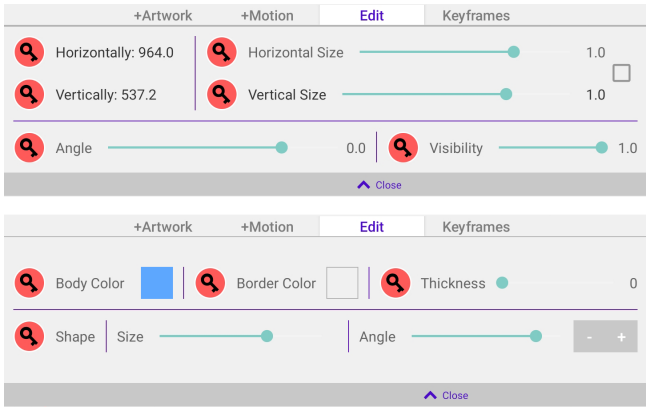

Figure 6: Katika offers various options for users to adjust the graphics. Beyond the on-screen adjustment (such as move or rotate), users have access to panels for artworks and polygons to edit specific attributes such as an artwork's scale, rotation or transparency (top), or a polygon's body or stroke color, stroke thickness, size, or rotation (bottom). Both panels also allow users to add keyframes on different attributes.

In each channel, users can add as many keyframes, edit the value or the interpolation of the keyframes and position the keyframe against time. As the video plays, in each frame, the motion of the artwork is first calculated. The resulting values (formatted as x and y axes based on the combination of position, rotation, scale, and transparency multiplier) are then applied as a coefficient to all the underlying polygons. After that, each polygon is drawn based on its corners, the color/stroke values, and the parent coefficient. There is no limitation to the complexity of the animation, and combining motions on various levels allows for more complex animations such as morph animations with hundreds of keyframes.

4.3.4 Collaboration and Communication Cloud Backend. In most applications, the project file maintains a link to the various artwork on a hard disk and how these files co-relate in the video. In Katika, we use Firebase [31] backend so that all pieces of a project are stored on the cloud and, as such, eliminate the notion of saving or loading projects. The working file of a project is a JSON file that maintains the relationship between resources. Having everything on a cloud server means that effectively all collaborators are working on the same "live" project (DC5). As several collaborators might work simultaneously, the software implements a first-in-first-out strategy that presents the latest adjustment at all times (similar to technologies such as Google Documents or Sheets).

5 EVALUATION

We ran an observational study to assess how end-users could use Katika to create amateur explainer motion graphics videos. In this initial evaluation, we focused on observing the feasibility of our end-to-end authoring approach and whether users found Katika's features, such as motion bundles, to be helpful and intuitive.

5.1 Participants

We recruited 11 participants (6 female) aged 21 to 43 (average: 30.8) through personal and campus mailing lists. Among these participants, three individuals (P2, P3, and P10) had some previous experience with motion design or video production, and eight were entirely new. While we focused on how beginners could learn and use Katika to make an amateur motion graphics video, we included more experienced participants for comparison. We explored how they perceive Katika's more advanced features for creating new artworks or motion bundles.

5.2 Procedure

5.2.1 Initial Setup and Training: We ran the study in person using an Android Tablet (SDK version 27). The study began with an initial demographic survey followed by an explanation of the goals of this study. Before starting the tasks, participants had the opportunity to watch a three-minute training video that illustrated how an explainer motion graphic video gets created. We prepared this video training so that viewers could gain a high-level understanding of concepts such as scripts, shots, artworks, animation, video editing, or transitions. The concepts were demonstrated at a high level and the software was abstracted away. After the training, we gave our participants five initial tasks to get familiar with Katika's user interface. In these tasks, they had to a) find the final video of a completed project, b) divide a script into separate shots, c) add artworks, d) browse motions for an artwork, and e) respond to comments from a hypothetical colleague. Within the 15 minutes allotted time, all participants managed to finish these tasks.

5.2.2 Video Authoring Task: After the initial tasks, we invited participants to complete the main task, replicating a 30-seconds public announcement explainer video (Figure 7) inspired by an online video [36]. Producing such a video would require common elements of an explainer motion graphics video, such as multiple shots, using artworks, animations, and editing the final video. This task had 25 minutes allocated to it. Upon its completion, we followed up with a post-task questionnaire by investigating metrics such as the tool's intuitiveness, the collaborative feature's efficiency, and the motion bundles. We then proceeded to the semi-structured interview that probed into the usefulness of the various components and the level of confidence participants felt upon completing the authoring.

5.2.3 Analysis: Study sessions were audio and screen recorded while we also took notes. We analyzed participants' approaches of interacting with the tool and completing the tasks. We then explored our data to gain insights about the usage of the application, synthesize the strengths of the system, and discover difficulties. We used an inductive analysis [23] approach to analyze the interview data and explored the themes around our main research question. To ensure the validity of the coded data, the primary author performed an open coding pass and consulted with the other author to develop the list of codes. To determine the quality of the generated videos, we devised a matrix of completeness, use of artworks and animation, number of shots, and the overall length of the video. Using this matrix, we then asked an external examiner (who was not a member of the research group) to compare the produced videos against the sample video and provide a score out of 5.

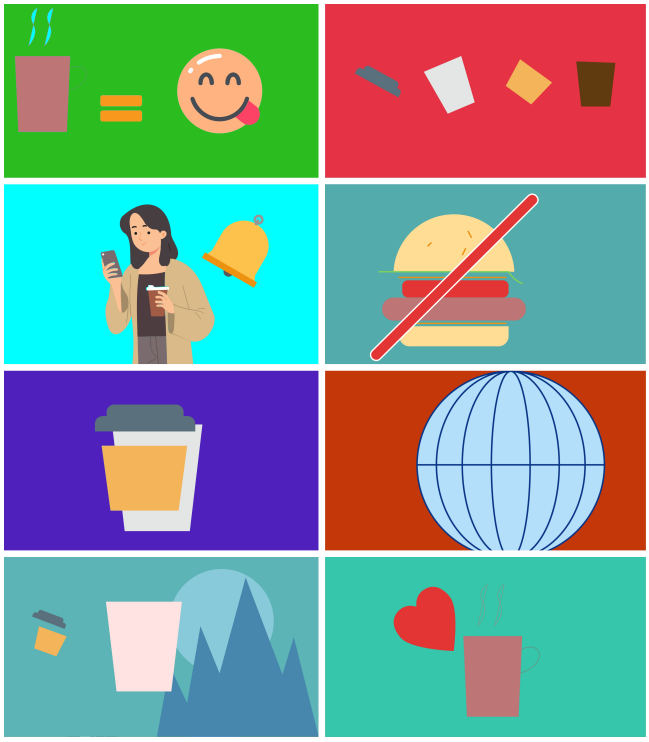

Figure 7: The gallery shown is extracted from the motion graphics videos authored by participants in our usability study of Katika. These participants could utilize the different features and functionality of Katika to put together videos using a built-in library of artworks and animation.

6 RESULTS

Our study investigated how users could employ Katika to create an explainer motion graphics video. The external examiner determined that all of our 11 participants successfully completed the main task of the study within the allotted time. During the post-task interviews, some participants reflected on their previous experience and expressed that creating the video in Katika was much faster. P3, for instance, suggested: *"If I had to make this on my own, I would have to make the shots and artworks by myself. I think it would take me maybe 3 or 4 hours."* Furthermore, the independent examiner's assessment reveals that videos produced by participants had a high similarity score compared to the sample video (average of 4.2 out of 5). We next provide insights into the overall experience with the end-to-end authoring software, details of the usefulness of the motion bundles, and the perceptions of Katika in actual practice.

6.1 End-to-End Authoring of Explainer Motion Graphics

Overall, integrating all steps of authoring an explainer motion graphics video into one system was encouraging for our participants, with P11 suggesting: *"Having everything in one platform is great. I myself don't like to go to so many apps or websites."* The post-task questionnaire illustrates that all participants found the

user interface to be intuitive to some extent (very intuitive: 4/11, intuitive: 6/11, somewhat intuitive: 1), and it helped them in feeling more confident about experimenting with different features. One participant highlighted:
"The UX of this app is great. I could find everything that I wanted real fast. It really simplifies the process. It comes down to making animation now and finishing the video." (P9)
One of the challenges in authoring motion graphics videos is viewing the context of the different parts of the process [34, 44]. Our participants appreciated Katika's central view (where all shots are listed) as having "everything in one place" helped them contextualize the process and learn about what they should do next. P6, for example, commented: *"It [the process] was pretty obvious what I should do. Add the artwork, add motions and view the final video. It was very easy to know where to go next."* To convey the message within shots, our participants opted to use a variety of artworks (Figure 7) and even enjoyed exploring the built-in library of content. Participants with experience in motion design appreciated the built-in artworks as normally they would have to spend a significant amount of time in finding content when working with state-of-the-art tools:

"[in other tools], it takes much longer. I can't be bothered to look [for artwork] online. Finding them is usually a disaster. It takes a long time and then you have to clean them and import. This [having the artwork built-in] is amazing." (P3)

Moreover, our results indicate that the built-in collaborative process of Katika was empowering for different reasons. For example, P1 compared it to being as easy as using Google Drive's shared documents with in-context commenting. Other participants, such as P10, commented: "*We always prefer to work with tools that offer collaboration. It's difficult to work with people overseas. I think that makes collaboration a major factor.*" We also noticed some interesting and unexpected usage of the collaboration feature. For example, P7 viewed the commenting tool as a mechanism by which they could take their own notes in context and reflect on their progress: *"I think it's useful if you have the chance to review all comments. I have a problem remembering things. I guess I could use it as note-taking for myself."*

6.2 Utility of Motion Bundles

One of our key innovations in Katika was our motion bundles feature for adding animations. We observed that participants explored different animations quickly and foresaw various uses for motion bundles. P8, for instance, stated: *"I was thinking about making animations, and I think this [motion bundles] is something that helps me [think] creatively."* Another participant commented on how this tool could also be useful for animating their own characters:
"I've been looking for something like this. I would particularly like to use a [motion bundles] to animate 2D characters. There are a few apps that I use to create characters, and it would be great if I could also animate them." (P3)

We also observed that some participants were initially hesitant to explore many motions in our warm-up task. They would, for example, choose one of the motion bundles and continue with it for the remainder of the task. However, after attempting the main task of the study and trying different examples, participants organically learned about the low cost of the explorations and were, in fact, intrigued by the idea of examining different motions. Some participants considered every single motion bundle before settling on one that best represented their intended concept. Others even attempted to merge different motions to make advanced animations, such as changing colors and shapes while moving objects around. While participants were not expected to change the duration of a motion, several participants explored that and were encouraged by the simplicity of moving a motion or adjusting its size:
"[An] advantage of the [motion bundle] template is that they are just there and ready to use. I could use [the slider in timeline] to extend the time to make them faster or slower. That really helps." (P8)
Such results indicate that participants with some experience could also produce more complex animations using Katika (beyond simple click-and-apply). Overall, these results are encouraging as they illustrate the effectiveness of presenting users with predefined animations. For many beginners, such examples mitigate the challenge of envisioning animations and encourage them to think about different movements representing an idea. Others, such as P10, who attempted to create their own motions, commended the streamlined process: *"Adding (new) motion was simple. The curve editor helped. If I needed to, I'd make my own motions, but I think I could use existing ones for most work."*

6.3 Perceptions about Using Katika in Practice

During our study, we consistently observed that participants were not intimidated by Katika's various features or functionality and some participants even attempted to push the application features further within the allotted study time. In fact, our questionnaire results show that our participants enjoyed using Katika (6/11 strongly agree, 5/11 agree), and their interview responses highlight how Katika enabled them to explore different ideas. Several participants commented that they wished to re-use the tool once they got used to it for their projects. P11, for instance, commented that: *"I would use this [Katika] to potentially make presentations for my conference. It's great to show visual stuff in presentations."* Another participant (P5) spoke of how they would prefer to watch videos as opposed to read content, and a different participant further elaborated on the value of democratizing motion graphics video authoring:
"I think this is a great tool...especially in today's world, there are so many videos on YouTube...this way, many [people] can make videos on their own." (P3)

Our participants appreciated some of the technical approaches of Katika as well. For instance, P3 enjoyed the cloud-based strategy of saving content as in the past they had trouble with moving files around. Another participant commended Katika's minimalist approach and that even the less-experienced team members would be able to use it effectively:
"I think this is a very straightforward system. I could see how I would use this for making video tutorials. Because I know how to use Adobe, but I have colleagues in marketing and sales that for them it's very difficult. We could add PNGs of our software and just make the tutorial. For that, this would be ideal." (P10)
Our participants had a few suggestions for improvement as well. Most of these ideas were cosmetic (or engineering) suggestions

such as colors, size of the windows, or the type of interaction (such as drag and drop instead of click and act). However, we did receive some substantive suggestions for making Katika easier to use in practice. For instance, three participants suggested having a method to preview motion bundles. P2, for example, indicated that decision-making would have been easier with such an approach: *"Motions are great. They can be applied by a simple click. It would be great if I could see a thumbnail of them before applying them. It makes it really easy to quickly decide."* Other participants proposed that commenting should also be possible at a specific point in time to enable more substantive discussions.

7 DISCUSSION

Our paper contributes the design and implementation of Katika, an integrated tool for authoring amateur explainer motion graphics videos. Current practices of creating motion graphics videos are at the two far ends of a spectrum. Amateurs either have to face a steep learning curves with professional feature-rich tools (e.g., After Effects or Maya) or struggle with re-purposing slide sharing tools (e.g, PowerPoint or Keynote) that are easier to access, but have limited animation and end-to-end authoring capabilities. The key novelty of Katika is in providing a bridge for pre- to post-production activities of motion graphic design and in enabling end-user animation using motion bundles.

Informed by actual motion design workflows, Katika provides a simplified, walk-up-and-use interface that helps amateurs understand the different stages of the motion graphic design process. This was inspired by dedicated authoring tools emerging in other design domains, such as user interface design and interactive prototyping. For example, although UI designers initially had to re-purpose and learn feature-rich image editing or slide sharing tools to create their prototypes, the industry standard now is the use of dedicated UI prototyping tools (e.g., Figma [30], Adobe XD [93], Axure [8]). Such tools not only provide a better user experience for designers as they have been informed by actual workflows of UI designers and UX researchers, but they also allow novices to learn the steps and terminologies of UI design. Our vision for Katika is similar in that we believe more designers will be able to learn about the process of motion graphic design and produce amateur motion graphics videos using our simplified end-to-end approach.

The initial results from our study demonstrate that amateurs from different backgrounds could use Katika to successfully create their first explainer motion graphics video within an hour. The responses from our participants also appear to confirm that the use of motion bundles simplified the process of animating and that they even enjoyed using the system. Furthermore, it was encouraging to see that our participants expressed interest in using Katika beyond the study for their own purposes (such as product demonstrations or conference presentations). These results suggest that Katika is complementary to other HCI approaches for simplifying video authoring for non-experts [35, 69], authoring of informational videos [5, 24], and streamlining animation techniques [46, 76, 86]. Finally, our participants were also enthusiastic about our open, community-shared content approach. Our vision is that user contributions for new artworks and motions will enable the creation of more advanced and creative motion graphics videos over time, ensuring the longevity of this authoring approach. Next, we reflect on our design and study insights to highlight opportunities for future HCI research.

7.1 Towards Complete Storytelling

A key novelty of Katika is that it integrates all steps of the motion graphics design process (breaking down a script into shots, adding graphics and animation, and video editing) into a single end-to-end system. Katika's underlying assumption is that a user would provide a story or a script on their own. While this may be possible for some users from creative fields, it may not necessarily be the case for other amateur creators. Given innovations in storytelling across various domains [32, 70], an exciting future direction would be helping users convert their ideas into stories. For example, we foresee a future in which a series of keywords can be utilized to develop the overall concept and use of semi-automated approaches to generate a corresponding script. This design can conform with our tool's current approaches, providing a starting color theme, shot breakdown, or video transitions. Using such as approach, a scientist, for instance, can generate their own visual stories and push information sharing through explainer videos a leap ahead.

7.2 Opportunities for Learning Video Production

A significant challenge in creating motion graphics videos is understanding the role of the smaller pieces in making the final video. During our study, we observed that some of our participants opted to go back and forth between the training video and Katika to better understand the subtleties. Given this behavior, future work can consider learning through reverse engineering as an approach for helping beginners dissect an example video to understand how it was created. Although prior works in learnability have explored techniques such as contextual help [18], on-demand support [16], and expert patterns [50], another direction could be to allow users to dig deeper into the different sections of a video and learn about the role of different building blocks within the video. We believe that having a way to learn by reverse engineering the video content will enhance the learnability of not only motion design, but other types of video productions as well. Moreover, in our initial studies, we learned that for many amateurs, envisioning the next state of animation was difficult or impossible. Using motion bundles, we supported them in browsing a library of animations. However, similar to prior work in example-based programming [10] or 3D design [49], we can use this technique for teaching animation even further as our participants were enthusiastic about using examples. In such an environment, beginners can learn about different (creative) approaches by merely watching and replicating examples on their own.

7.3 Quick and Dirty Video Prototyping

Our implementation of Katika is centered around video authoring for amateurs that binds scriptwriting with shot-by-shot breakdown, artwork, animations, collaboration, video transitions, and editing. This approach offers numerous advantages, including lowering the cost of task switching between different tools that can be useful for

professionals as well. Our approach is similar to the use of end-to-end and lightweight approaches in design fields such as UI/UX (e.g., XD or Figma), 3D printing [40], or CAD [60] to produce rapid prototypes. Moreover, during our usability study, participants with prior experience in motion design saw the possibility of using Katika as a lightweight tool for quick authoring tasks that required feedback and iteration. Taking inspiration from such work, we envision a future in which users can employ lightweight solutions such as Katika in realtime and leverage repositories of pre-made artworks, motions, and automated features, combined with built-in collaborations to prototype motion graphics videos rapidly. These "quick and dirty" prototypes could perhaps be utilized in environments such as classrooms or even professional meetings.

8 LIMITATIONS

This research has had a few limitations, and as such, we put forth our contributions with caution. The design of our tool currently only supports the two-dimensional style of motion graphics. It is worth expanding this approach into other motion graphics design styles. As with any qualitative study, the small sample size warrants further evaluation and future work can directly compare Katika with other software tools used to create motion graphics videos. The main challenge in doing such a comparative study is that the available tools are at either end of the spectrum (such as high-quality productions with After Effects or rudimentary content with tools such as PowerPoint or Keynote) and do not focus on the end-to-end production process as we have in Katika. In addition, our study's main task asked the participants to re-produce an existing video given a particular script and we did not evaluate the cognitive processes of storytelling and scriptwriting. This aspect should be further investigated in real-world deployments or longitudinal studies of Katika.

9 CONCLUSION

The interest in creating explainer motion graphics videos is increasing across several domains. Our research contributes the design and evaluation of Katika, an end-to-end software application that supports amateurs in authoring motion graphics videos. Katika offers an embedded library of artworks, a novel approach of motion bundles for animation, and built-in communication features. Moreover, it lowers the complexity of various tasks through automated solutions and uses terminology that is easier to understand for beginners. Our observational study illustrates how amateurs who had no prior experience with motion graphics could begin making explainer motion graphics videos using Katika within a few minutes. Overall, our insights provide fodder for compelling opportunities in HCI to lower barriers to entry for amateur motion graphics designers.

ACKNOWLEDGMENTS

The authors wish to thank Scott Ritchie, William Odom, and Sheelagh Carpendale. This work was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Mark S. Ackerman, Juri Dachtera, Volkmar Pipek, and Volker Wulf. 2013. Sharing Knowledge and Expertise: The CSCW View of Knowledge Management. *Computer Supported Cooperative Work (CSCW)* 22, 4-6 (Aug. 2013), 531–573. https://doi.org/10.1007/s10606-013-9192-8

[2] Adobe After Effects. 2021. Adobe After Effects. https://www.adobe.com/ca/products/aftereffects.html

[3] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Christophe Hurter, and Pourang Irani. 2015. Understanding Data Videos: Looking at Narrative Visualization through the Cinematography Lens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, Seoul, Republic of Korea, 1459–1468. https://doi.org/10.1145/2702123.2702431

[4] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Jason Leboe-McGowan, and Pourang Irani. 2018. Hooked on data videos: assessing the effect of animation and pictographs on viewer engagement. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces - AVI '18*. ACM Press, Castiglione della Pescaia, Grosseto, Italy, 1–9. https://doi.org/10.1145/3206505.3206552

[5] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Andres Monroy-Hernandez, and Pourang Irani. 2017. Authoring Data-Driven Videos with DataClips. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan. 2017), 501–510. https://doi.org/10.1109/TVCG.2016.2598647

[6] Animaker. 2021. Animaker. https://www.animaker.com/

[7] Keynote Aple. 2021. Apple Keynote. https://www.apple.com/ca/keynote/

[8] Axure Axure. 2021. Axure. https://www.axure.com/

[9] Niksa Babic, Jesenka Pibernik, and Nikola Mrvac. 2008. Media Study: Motion Graphics. (2008), 4.

[10] Lingfeng Bao, Zhenchang Xing, Xin Xia, and David Lo. 2019. VT-Revolution: Interactive Programming Video Tutorial Authoring and Watching System. *IEEE Transactions on Software Engineering* 45, 8 (Aug. 2019), 823–838. https://doi.org/10.1109/TSE.2018.2802916

[11] Miri Barak, Tamar Ashkar, and Yehudit J. Dori. 2011. Learning science via animated movies: Its effect on students' thinking and motivation. *Computers & Education* 56, 3 (April 2011), 839–846. https://doi.org/10.1016/j.compedu.2010.10.025

[12] Spencer Barnes. 2017. Studies in the Efficacy of Motion Graphics: The impact of narrative structure on exposition. *Digital Journalism* 5, 10 (Nov. 2017), 1260–1280. https://doi.org/10.1080/21670811.2017.1279020

[13] Biteable. 2019. Biteable. https://biteable.com/

[14] Matthew Brehmer, Bongshin Lee, Benjamin Bach, Nathalie Henry Riche, and Tamara Munzner. 2017. Timelines Revisited: A Design Space and Considerations for Expressive Storytelling. *IEEE Transactions on Visualization and Computer Graphics* 23, 9 (Sept. 2017), 2151–2164. https://doi.org/10.1109/TVCG.2016.2614803

[15] Eric Brill and Raymond J Mooney. 1997. An Overview of Empirical Natural Language Processing. (Oct. 1997), 12.

[16] Yan Chen, Sang Won Lee, Yin Xie, YiWei Yang, Walter S. Lasecki, and Steve Oney. 2017. Codeon: On-Demand Software Development Assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 6220–6231. https://doi.org/10.1145/3025453.3025972

[17] 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*. ACM, Virtual Event USA, 279–292. https://doi.org/10.1145/3379337.3415814

[18] Parmit K. Chilana, Amy J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: selection-based crowdsourced contextual help for web applications. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. ACM Press, Austin, Texas, USA, 1549. https://doi.org/10.1145/2207676.2208620

[19] Mark Christiansen. 2012. *Adobe After Effects CS3 Professional Studio Techniques*. https://www.adobepress.com/store/adobe-after-effects-cs3-professional-studio-techniques-9780321850553

[20] Devan Ciccarelli. 2020. Google Slides vs PowerPoint vs Keynote: Everything You Need to Know. https://www.goskills.com/Microsoft-Office/Resources/Google-Slides-vs-PowerPoint

[21] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2015. Supporting Novice to Expert Transitions in User Interfaces. *Comput. Surveys* 47, 2 (Jan. 2015), 1–36. https://doi.org/10.1145/2659796

[22] Justin Cone. 2015. 6 challenges facing motion design education. http://motionographer.com/2015/04/01/7-problems-with-motion-design-education/

[23] Juliet Corbin and Anselm Strauss. 2007. Basics of qualitative research: techniques and procedures for developing grounded theory.

[24] Carman Coto. 2018. What Are Explainer Videos, and Why Are They So Popular. https://topexplainers.com/how-effective-are-explainer-videos-statistics-and-how-to-step-by-step/

[25] James Davis, Maneesh Agrawala, Erika Chuang, and Zoran Popovi. 2003. A Sketching Interface for Articulated Figure Animation. (2003), 10.

[26] Richard C. Davis, Brien Colwell, and James A. Landay. 2008. K-sketch: a 'kinetic' sketch pad for novice animators. In *Proceeding of the twenty-sixth annual CHI*

conference on Human factors in computing systems - CHI '08. ACM Press, Florence, Italy, 413. https://doi.org/10.1145/1357054.1357122

[27] Richard C Davis and James A Landay. 2004. Informal Animation Sketching: Requirements and Design. *AAAI Technical Report* (2004), 7.

[28] Marco de Sá, David A. Shamma, and Elizabeth F. Churchill. 2014. Live mobile collaboration for video production: design, guidelines, and requirements. *Personal and Ubiquitous Computing* 18, 3 (March 2014), 693–707. https://doi.org/10.1007/s00779-013-0700-0

[29] Doodly. 2021. Doodly. https://www.doodly.com/

[30] Figma. 2021. Figma. https://www.figma.com/

[31] Firebase. 2021. Firebase. https://firebase.google.com/

[32] Sarah Flanagan. 2015. How does storytelling within higher education contribute to the learning experience of early years students? *Journal of Practice Teaching and Learning*, (2015), 23.

[33] Dustin Freeman, Stephanie Santosa, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. 2014. LACES: live authoring through compositing and editing of streaming video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1207–1216. https://doi.org/10.1145/2556288.2557304

[34] Lu Geng. 2016. Study of the Motion Graphic Design at the Digital Age. In *Proceedings of the 2016 International Conference on Arts, Design and Contemporary Education*. Atlantis Press, Moscow, Russia. https://doi.org/10.2991/icadce-16.2016.183

[35] Andreas Girgensohn, Sara Bly, Frank Shipman, John Boreczky, and Lynn Wilcox. 2001. Home Video Editing Made Easy — Balancing Automation and User Control. *INTERACT* (2001), 8.

[36] Brandon Gorthy. 2013. Coffee Cup Proper Disposal PSA. https://vimeo.com/62218268

[37] Chirag Goyal. 2021. Part 11: Step by Step Guide to Master NLP – Syntactic Analysis. https://www.analyticsvidhya.com/blog/2021/06/part-11-step-by-step-guide-to-master-nlp-syntactic-analysis/

[38] Masaki Hayashi, Seiki Inoue, Mamoru Douke, Narichika Hamaguchi, Hiroyuki Kaneko, Steven Bachelder, and Masayuki Nakajima. 2014. T2V new Technology of Converting Text to CG Animation. 2, 1 (2014), 8.

[39] Xian-Sheng Hua, Zengzhi Wang, and Shipeng Li. 2005. LazyCut content-aware template-based video authoring. In *Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05*. ACM Press, Hilton, Singapore, 792. https://doi.org/10.1145/1101149.1101318

[40] Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. ACM Press, Santa Clara, California, USA, 384–396. https://doi.org/10.1145/2858036.2858266

[41] Jessica Hullman, Steven Drucker, Nathalie Henry Riche, Bongshin Lee, Danyel Fisher, and Eytan Adar. 2013. A Deeper Understanding of Sequence in Narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2406–2415. https://doi.org/10.1109/TVCG.2013.119

[42] Adobe Illustrator. 2021. Adobe Illustrator. https://www.adobe.com/ca/products/illustrator.html

[43] iMovie iMovie. 2021. iMovie. https://www.apple.com/ca/imovie/

[44] Amir Jahanlou, William Odom, and Parmit Chilana. 2021. Challenges in Getting Started in Motion Graphic Design: Perspectives from Casual and Professional Motion Designers. *Graphics Interface 2021* (2021), 11.

[45] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. ACM Press, Honolulu, Hawaii, USA, 395–405. https://doi.org/10.1145/2642918.2647375

[46] Rubaiat Habib Kazi, Tovi Grossman, Cory Mogk, Ryan Schmidt, and George Fitzmaurice. 2016. ChronoFab: Fabricating Motion. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. ACM Press, Santa Clara, California, USA, 908–918. https://doi.org/10.1145/2858036.2858138

[47] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. Motion Amplifiers: Sketching Dynamic Illustrations Using the Principles of 2D Animation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 4599–4609. https://doi.org/10.1145/2858036.2858386

[48] Kimia Kiani, Parmit K. Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. 2020. "I Would Just Ask Someone": Learning Feature-Rich Design Software in the Modern Workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, Dunedin, New Zealand, 1–10. https://doi.org/10.1109/VL/HCC50065.2020.9127288

[49] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond "One-Size-Fits-All": Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, Glasgow, Scotland Uk, 1–14. https://doi.org/10.1145/3290605.3300570

[50] Juho Kim. 2015. Learnersourcing: Improving Learning with Collective Learner Activity. *Doctoral dissertation, Massachusetts Institute of Technology* (2015), 213.

[51] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, Toronto, Ontario, Canada, 4017–4026. https://doi.org/10.1145/2556288.2556986

[52] Nam Wook Kim, Hanspeter Pfister, Nathalie Henry Riche, Benjamin Bach, Guanpeng Xu, Matthew Brehmer, Ken Hinckley, Michel Pahud, Haijun Xia, and Michael J. McGuffin. 2019. DataToon Drawing Dynamic Network Comics With Pen + Touch Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, Glasgow, Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300335

[53] Jon S. Krasner. 2008. *Motion graphic design: applied history and aesthetics* (2nd ed. ed.). Focal Press, Amsterdam ; Boston.

[54] Andreas Krämer and Sandra Böhrs. 2016. How Do Consumers Evaluate Explainer Videos? An Empirical Study on the Effectiveness and Efficiency of Different Explainer Video Formats. *Journal of Education and Learning* 6, 1 (Dec. 2016), 254. https://doi.org/10.5539/jel.v6n1p254

[55] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community enhanced tutorials: improving tutorials with multiple demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, Paris, France, 1779. https://doi.org/10.1145/2470654.2466235

[56] Mackenzie Leake, Hijung Valentina Shin, Joy O. Kim, and Maneesh Agrawala. 2020. Generating Audio-Visual Slideshows from Text Articles Using Word Concreteness. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–11. https://doi.org/10.1145/3313831.3376519

[57] Bongshin Lee, Nathalie Henry Riche, Petra Isenberg, and Sheelagh Carpendale. 2015. More Than Telling a Story: Transforming Data into Visually Shared Stories. *IEEE Computer Graphics and Applications* 35, 5 (Sept. 2015), 84–90. https://doi.org/10.1109/MCG.2015.99

[58] Ghang Lee, Charles M. Eastman, Tarang Taunk, and Chun-Heng Ho. 2010. Usability principles and best practices for the user interface design of complex 3D architectural design and engineering tools. *International Journal of Human-Computer Studies* 68, 1-2 (Jan. 2010), 90–104. https://doi.org/10.1016/j.ijhcs.2009.10.001

[59] Germán Leiva and Michel Beaudouin-Lafon. 2018. Montage: A Video Prototyping System to Reduce Re-Shooting and Increase Re-Usability. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press, Berlin, Germany, 675–682. https://doi.org/10.1145/3242587.3242613

[60] Wei Li, Tovi Grossman, and George Fitzmaurice. 2012. GamiCAD: a gamified tutorial system for first time autocad users. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. ACM Press, Cambridge, Massachusetts, USA, 103. https://doi.org/10.1145/2380116.2380131

[61] Min Lu, Chufeng Wang, Joel Lanir, Nanxuan Zhao, Hanspeter Pfister, Daniel Cohen-Or, and Hui Huang. 2020. Exploring Visual Information Flows in Infographics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376263

[62] Magicons. 2021. Magicons. https://graphicburger.com/magicons-2867-flat-icons/

[63] Autodesk Maya. 2021. Autodesk Maya. https://www.autodesk.ca/en/products/maya/overview

[64] Britta Meixner, Katarzyna Matusik, Christoph Grill, and Harald Kosch. 2014. Towards an easy to use authoring tool for interactive non-linear video. *Multimedia Tools and Applications* 70, 2 (May 2014), 1251–1276. https://doi.org/10.1007/s11042-012-1218-6

[65] Justin Mifsud. 2018. The Difference (And Relationship) Between Usability And Learnability. https://usabilitygeek.com/the-difference-and-relationship-between-usability-and-learnability/

[66] Tomer Moscovich and John F Hughes. 2001. Animation Sketching: An Approach to Accessible Animation. *Unpublished Master's Thesis, CS Department, Brown University* (2001), 5.

[67] Tsai-Yun Mou. 2015. Creative Story Design Method In Animation Production Pipeline. *Proceedings of The Third International Conference on Design Creativity, Indian Institute of Science, Bangalore* (2015), 8.

[68] Santiago Negrete-Yankelevich and Nora Morales-Zaragoza. 2013. e-Motion: a system for the development of creative animatics. *Proceedings of the Fourth International Conference on Computational Creativity (p. 184)* (2013), 5.

[69] Naoko Nitta and Noboru Babaguchi. 2011. Example-based video remixing for home videos. In *2011 IEEE International Conference on Multimedia and Expo*. IEEE, Barcelona, Spain, 1–6. https://doi.org/10.1109/ICME.2011.6012080

[70] Susanna Nordmark and Marcelo Milrad. 2012. Mobile Digital Storytelling for Promoting Creative Collaborative Learning. In *2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education*. IEEE, Takamatsu, Kagawa, Japan, 9–16. https://doi.org/10.1109/WMUTE.2012.10

[71] Lucia Pechersky. 2018. How Long Does It Takes To Produce A Customized Explainer Video? https://www.yumyumvideos.com/how-long-does-it-takes-to-produce-a-customized-explainer-video-wp/

[72] Adobe Photoshop. 2021. Adobe Photoshop. https://www.adobe.com/ca/products/photoshop.html

[73] Chris Potter. 2013. Video production is a collaborative effort. Whether you are a client, a producer, or an editor, providing feedback to others is part of the game. https://screenlight.tv/blog/2013/04/11/tips-for-providing-feedback-on-video-production

[74] Point Power. 2021. Powerpoint. https://www.microsoft.com/en-us/microsoft-365/powerpoint

[75] Adobe Premiere. 2021. Adobe Premiere. https://www.adobe.com/ca/products/premiere.html

[76] J Quevedo-Fernández. 2013. idAnimate A General-Purpose Animation Sketching Tool for Multi-Touch Devices. (2013), 10.

[77] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative Adversarial Text to Image Synthesis. (Oct. 2016), 10.

[78] Widi Sarinastiti, Dwi Susanto, and Rodhiyah Mutammimah. 2016. Skill level animation technique on dental care motion graphic for children. In *2016 International Electronics Symposium (IES)*. IEEE, Denpasar, Indonesia, 389–394. https://doi.org/10.1109/ELECSYM.2016.7861037

[79] Klaus Schoeffmann, Marco A. Hudelist, and Jochen Huber. 2015. Video Interaction Tools: A Survey of Recent Work. *Comput. Surveys* 48, 1 (Sept. 2015), 1–34. https://doi.org/10.1145/2808796

[80] Yang Shi, Xingyu Lan, Jingwen Li, Zhaorui Li, and Nan Cao. 2021. Communicating with Motion: A Design Space for Animated Visual Narratives in Data Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–13. https://doi.org/10.1145/3411764.3445337

[81] Google Slides. 2021. Google Slides. https://www.google.ca/slides/about/

[82] Kaoru Sumi and Mizue Nagata. 2006. Animated storytelling system via text. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology - ACE '06*. ACM Press, Hollywood, California, 55. https://doi.org/10.1145/1178823.1178889

[83] Naimdjon Takhirov and Fabien Duchateau. 2011. A cloud-based and social authoring tool for video. In *Proceedings of the 11th ACM symposium on Document engineering - DocEng '11*. ACM Press, Mountain View, California, USA, 65. https://doi.org/10.1145/2034691.2034705

[84] TedEd TedEd. 2018. Build a lesson around any TED-Ed Animation, TED Talk or YouTube video. https://ed.ted.com/lessons/making-a-ted-ed-lesson-concept-and-design

[85] J. Thompson, Z. Liu, W. Li, and J. Stasko. 2020. Understanding the Design Space and Authoring Paradigms for Animated Data Graphics. *Computer Graphics Forum* 39, 3 (June 2020), 207–218. https://doi.org/10.1111/cgf.13974

[86] Matthew Thorne, David Burke, and Michiel van de Panne. 2004. Motion Doodles: An Interface for Sketching Character Motion. (2004), 8.

[87] TikTok TikTok. 2021. TikTok. https://www.tiktok.com/

[88] Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. QuickCut: An Interactive Tool for Editing Narrated Video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 497–507. https://doi.org/10.1145/2984511.2984569

[89] Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. 2017. Social CheatSheet: An Interactive Community-Curated Information Overlay for Web Applications. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (Dec. 2017), 1–19. https://doi.org/10.1145/3134737

[90] Vimeo. 2019. Vimeo. https://www.vimeo.com/

[91] Daniel Wilches, Pablo Figueroa, Agustin Conde, and Faramarz Samavati. 2012. Sketch-Based Interface for animation for non-experts. In *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)*. IEEE, Medellin, Colombia, 1–8. https://doi.org/10.1109/CLEI.2012.6427124

[92] Nora S Willett, Hijung Valentina Shin, Zeyu Jin, Wilmot Li, and Adam Finkelstein. 2020. Pose2Pose: pose selection and transfer for 2D character animation. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. ACM, Cagliari Italy, 88–99. https://doi.org/10.1145/3377325.3377505

[93] Adobe XD. 2021. Adobe XD. https://www.adobe.com/ca/products/xd.html

[94] Haijun Xia. 2020. Crosspower: Bridging Graphics and Linguistics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 722–734. https://doi.org/10.1145/3379337.3415845

[95] 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*. ACM, Virtual Event USA, 735–746. https://doi.org/10.1145/3379337.3415882

[96] Taner Çifçi. 2016. Effects of Infographics on Students Achievement and Attitude towards Geography Lessons. *Journal of Education and Learning* 5, 1 (Jan. 2016), 154. https://doi.org/10.5539/jel.v5n1p154