SmartShots: An Optimization Approach for Generating Videos with Data Visualizations Embedded

TAN TANG, JUNXIU TANG, JIEWEN LAI, LU YING, and YINGCAI WU, Zhejiang University
LINGYUN YU, Xi'an Jiaotong-Liverpool University
PEIRAN REN, Alibaba Group

Videos are well-received methods for storytellers to communicate various narratives. To further engage viewers, we introduce a novel visual medium where data visualizations are embedded into videos to present data insights. However, creating such data-driven videos requires professional video editing skills, data visualization knowledge, and even design talents. To ease the difficulty, we propose an optimization method and develop SmartShots, which facilitates the automatic integration of in-video visualizations. For its development, we first collaborated with experts from different backgrounds, including information visualization, design, and video production. Our discussions led to a design space that summarizes crucial design considerations along three dimensions: visualization, embedded layout, and rhythm. Based on that, we formulated an optimization problem that aims to address two challenges: (1) embedding visualizations while considering both contextual relevance and aesthetic principles and (2) generating videos by assembling multi-media materials. We show how SmartShots solves this optimization problem and demonstrate its usage in three cases. Finally, we report the results of semi-structured interviews with experts and amateur users on the usability of SmartShots.

CCS Concepts: • **Human-centered computing** → *Visualization systems and tools*; • **Information systems** → **Multimedia information systems;**

Additional Key Words and Phrases: Visualization, data-driven videos, optimization

ACM Reference format:
Tan Tang, Junxiu Tang, Jiewen Lai, Lu Ying, Yingcai Wu, Lingyun Yu, and Peiran Ren. 2022. SmartShots: An Optimization Approach for Generating Videos with Data Visualizations Embedded. *ACM Trans. Interact. Intell. Syst.* 12, 1, Article 4 (February 2022), 21 pages.
https://doi.org/10.1145/3484506

The reviewing of this article was managed by associate editor Chang, Remco.
The work was supported by NSFC (61761136020), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1609217), and Zhejiang Provincial Natural Science Foundation (LR18F020001). This work was also supported by Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies and partially funded by Microsoft Research Asia. L. Yu was supported by XJTLU Research Development Funding RDF-19-02-11.
Authors' addresses: T. Tang, J. Tang, J. Lai, L. Ying, and Y. Wu (corresponding author), Zhejiang University; emails: {tantan, tangjunxiu, laijiewen, yingluu, ycwu}@zju.edu.cn; L. Yu, Xi'ian Jiaotong-Liverpool University; email: Lingyun.Yu@xjtlu.edu.cn; P. Ren, Alibaba Group; email: renpeiran@gmail.com.

2160-6455/2022/02-ART4 $15.00
https://doi.org/10.1145/3484506

1 INTRODUCTION

Videos have been adopted in various disciplines, such as journalism [41], education [14], and advertising [1], to communicate stories. As an effective means for telling visual stories [29, 31], data visualization [38] has been embedded into videos to enhance narrations. Real-world examples of this novel visual medium can be seen in a video from AMD (Figure 1(a)), which highlights the power-saving feature of its newest GPU using a skew bar chart, and a video from Ekko Hub (Figure 1(b)), which uses a horizontal bar chart to highlight the benefits of its WIFI technology. The major advantage of the data-driven content is the support by convincing and objective facts that can increase the persuasiveness and expressiveness of the videos. However, creating **videos with data visualizations embedded (VDE)** is not an easy task.

Many factors can influence the effectiveness of VDE. First, the interplay between data visualizations and video content is very important in producing a complete and consistent story. However, such a relationship cannot be easily balanced because various factors in the design space need to be considered, such as *where* in the video should the data visualization be embedded for both the visualization and video content to be well noticed and *how long* the data visualization should be played to keep viewers oriented. Second, the type of data visualization and the associated animations directly affect the effectiveness of reading and understanding the video content. Third, the aesthetic quality of videos and the synthesis of music are quite important for real-world videos that target attracting ordinary viewers. Thus, producing such an information-rich video may require considerable efforts and experiences from designers.

To produce general data-driven videos [7], previous studies [7, 9] have developed a template-based tool that generates videos by composing a set of data motion graphics. However, this tool cannot be used for producing VDE because of two reasons. First, it mainly focuses on animated data visualizations and ignores the video content [22]. To consider the interplay between the inserted entities and videos, many approaches have been applied in the field of in-video advertising [37, 49, 59]. However, these methods mainly focus on image or text rather than data visualizations and ignore the aesthetic principles [10, 33] when identifying the insertion points. Second, this tool does not support the synthesis of music that is vital for real-world videos to inspire and attract viewers. Some music-driven approaches [53, 54] have been proposed to automatically compose video and music content, but these approaches require a complete video in advance instead of assembling multi-media materials into a new one.

These limitations motivate us to develop a new tool that facilitates the automatic generation of VDE. However, the development of the tool is hindered by the following challenges.

One challenge is that it is difficult to abstract a sufficient and essential design space among all **design considerations (DCs)**. As discussed earlier, many factors should be considered in the design of VDE. To seamlessly integrate data visualizations in a video context, these visualizations must be initially associated with their respective referents (e.g., physical entities in videos), then the embedded layouts should be optimized based on certain aesthetic goals. For example, the visualizations should be embedded in the same region of videos to preserve the viewers' mental map and to prevent an occlusion of referents, which can negatively influence readability. Furthermore, designers may not be able to easily select a suitable visualization type for the data in a specific video scene. Thus, a comprehensive design space that considers both contextual relevance and aesthetic principles must be developed to produce appealing layouts.

Another challenge is that it is difficult but necessary to propose a computational model corresponding to the design space and enable automatic video generation. The production of digital videos involves processing and assembling multi-media materials, including data visualizations, video clips, images, and music, which requires considerable effort [7] and design expertise. First,

Fig. 1. Videos with data visualizations embedded. (a) The bar chart[1] is used to highlight the power-saving feature. (b) The horitontal bar chart[2] compares the different wireless techniques. White boxes are manually labeled to distinguish the embedded visualizations.

manually editing videos requires professional skills. Second, assembling data visualizations, images, and video clips while considering music rhythm is also a tedious task. Thus, an intelligent approach to process multi-media materials, especially one that can automatically embed data visualizations into videos, must be developed.

To address these challenges, we must fully understand the design space of VDE and consider the different design aspects of the video generation. Therefore, we collaborate closely with a group of experts to determine the crucial DCs for VDE. We target video makers who want to integrate data visualizations into their videos, including both designers who are not experts in visualization designs and amateur users who are novice in video editing skills. Based on these considerations, we develop a novel optimization model that targets (1) *embedded visualization*: we extract five high-level aesthetic principles from the DCs and then define five energy functions to determine the suitable visualization type and optimize the embedded layouts accordingly; (2) *video generation*: we initially translate the input data tables, video clips, and images into shots that refer to video pieces and then formulate a dynamic programming problem that matches the shot transitions with the music rhythm to automate the composition of source materials. Furthermore, we develop SmartShots, a video generation system that integrates a set of post-editing interactions to facilitate the easy creation of VDE. We present three use cases to demonstrate the usage and extensibility of our approach and conduct semi-structured interviews to validate the usability of SmartShots. Our preliminary work [48] designed interface components for the interactive editing of VDE. This article provides a comprehensive final review of the entire SmartShots project with the following new contributions:

- We characterize a design space and propose a set of DCs that can guide the design and creation of VDE.
- We formulate an energy-based optimization problem according to the DCs to automate the video generation process. In particular, we consider the aesthetic principles to optimize the in-video layouts for data visualizations.
- We collect user feedback to evaluate SmartShots and present three use cases to demonstrate the usage and effectiveness of this tool.

2 RELATED WORK

This section presents the relevant studies on data videos and discusses the state-of-the-art methods that address the challenges in embedded visualizations, label placement, layout optimization, and video generation.

[1] https://www.youtube.com/watch?v=dTa8GVO3GFI.
[2] https://www.youtube.com/watch?v=raok7kqii_w.

2.1 Data Video

Data video is the video with data-driven arguments or visualizations [7–9], which is one of the seven genres of narrative visualizations [45, 46, 64]. Amini et al. [7] studied data videos in the wild and summarized the authoring processes adapted by designers. Based on that, they developed *DataClips*, an authoring tool that can help users create data motion graphics and aggregate separate units into a complete video [9]. MTurk [6] studies evaluated the effect of data animation and pictographic representations on the viewer engagement of data videos, which reported a set of design suggestions [8]. Although these studies have provided a deep understanding of data videos, they mainly focus on data-driven motion graphics and ignore the combination between data visualizations and video content. Recent studies [21, 34] on data videos mainly focus on specific data formats or tasks, which cannot be easily extended for the overall pipeline of video generation. For example, Lu et al. [34] proposed a novel approach to track the temporal changes of time series in data videos, and Ge et al. [21] focused on developing high-level grammars for data-driven chart animations. To understand how to augment videos with data visualizations, Tang et al. [47] conducted group studies to collect design guidelines from professional designers. However, it is still unknown to how to develop an authoring tool that can create VDE. To bridge the gap, we propose a novel video generation system for producing VDE that extends the form of data videos from motion graphics to videos with real scenes and embedded data visualizations.

2.2 Embedded Visualization

Embedded data visualization has become an emerging research topic in the fields of information visualization [58]. To make embedded data visualization more applicable, some AR (VR)-based toolkits [15] and techniques [26, 36] have been developed. Chen et al. [15] developed a mobile-based authoring tool, namely *MARVisT*, to enable non-experts to easily bind data to virtual glyphs and real objects. McNamara et al. [36] developed a novel technique for placing textual labels in a virtual environment where the eye-tracking data is used to improve the identification of objects of interests. Hegde et al. [26] focused on embedding textual labels in a physical environment and proposed a novel technique for an occluded-free layout. However, all these applications mainly focus on embedding visualizations into a 3D space that is established on AR (VR)-based technologies [52]. Unlike these approaches, the goal of our study is to integrate data visualizations into a 2D video context. Recently, researchers [17, 62] also proposed a few tools to support the design of infographics that embed visualizations into 2D images. Coelho and Mueller [17] developed a system to create such kinds of infographics where visualizations are placed on some areas in images and fitted in the shapes of the embedded areas. Zhang et al. [62] proposed DataQuilt to provide designers an iterative workflow to design pictorial visualizations with images like paintings and photos. However, the existing works target at creating static image-style infographics, which is different from embedding visualizations in dynamic video shots. For example, the embedded area and visualizations must be chosen in advance, which could be a tedious task for designing data videos. Instead, our approach considers the relationships between video context and visualizations, which enable us to automatically determine the embedded areas and visualization types.

2.3 Label Placement

The problem of embedding visualizations into 2D video context is similar to the label placement issue in videos [27, 30, 37, 49]. Mei et al. [37, 49] developed VideoSense that determines the insertion points in the video timeline according to the contextual relevance between inserted advertisements and video content. Hu et al. [27] proposed an approach to present subtitles of conversations near to the characters in videos. Kurzhals et al. [30] introduced novel methods to dynamically place

subtitles or captions in videos according to the viewer's real-time gaze. However, these approaches are not sufficient for embedding visualizations into videos. First, the previous studies either focus on the specific scenarios like dialog scenes [27] or merely considered text labels (e.g., subtitles [27] or captions [30]). Due to the different scenarios and inserted objects, the DCs of the existing works are not enough for embedding data visualizations into videos because the visualizations are far more informative than text labels. For example, a data visualization not only includes textual labels but also provides graphical elements that should be colored according to video scenes. To integrate such informative visualizations, a more comprehensive design space must be considered, which is still unknown in the existing studies. Second, the goal of label placement is to insert information without disturbing original videos, whereas the goal of VDE is to balance visualizations and video content because both are important components of the final videos. On the one hand, it is necessary to place visualizations in a proper area without hindering the presentation of other visual referents in videos. On the other hand, it may be desirable to highlight data visualizations because they can reveal data patterns in a more expressive way. Thus, it is inevitable to achieve a trade-off among different design factors, which makes the design of VDE much more complicated than existing label placement issues.

2.4 Layout Optimization

The layout optimization problem has been a hot research issue in the multimedia community for a long time. Yadati et al. [59] proposed a novel approach that is established on consumer psychology rules and considers affective factors, such as valence and arousal, to insert advertisements in videos. However, these methods do not always take the aesthetic aspect of embedded layouts into account, which can be inspired by the studies of the single-page layout optimizations [39, 40, 60]. Optimizing a single-page layout is a non-trivial issue that needs to be solved by considering various aesthetic design principles, including visual balance [33], alignment [10], and style harmony [61]. Based on these aesthetic principles, Purvis et al. [42] proposed a genetic optimization method for the layout of personalized documents, whereas O'Donovan et al. [39] introduced a pixel-based approach that automates single-page graphic designs and developed *DesignScape* [40] to provide layout suggestions to novice designers. However, their approaches are time consuming and do not take proximity and temporal coherence into consideration, thereby making them unsuitable for video generation. Yang et al. [60] developed a template-based approach for layering the text labels on top of a background image. Yet the magazine-style layouts cannot be directly applied in the designs of embedded visualizations since visualizations are much more informative than text information. To facilitate an efficient generation of VDE, we employ an object-based optimization model that produces visually appealing and aesthetically pleasing layouts for embedded visualizations in a video context. We consider a more comprehensive set of design principles, including balance, alignment, proximity, readability, and temporal coherence. Moreover, our model also considers the contextual relevance between in-video data visualizations and the respective visual referents to optimize the embedded layouts.

2.5 Video Generation

Creating videos manually is tedious and requires significant human effort. Researchers have proposed various (semi-)automatic approaches [16, 53, 54] to facilitate the efficient generation of digital videos. Ngo et al. [16] employed a Markov chain model to balance the content and perceptual quality of video shots. They then aggregated shots with high attention values into a new video. We follow the same idea, but we first translate the source materials into shots before aggregating shots into a complete video. To facilitate music composition [11], various approaches [53, 54] have been developed to match shots to the music rhythm. Wang et al. [54] introduced a novel music-centric

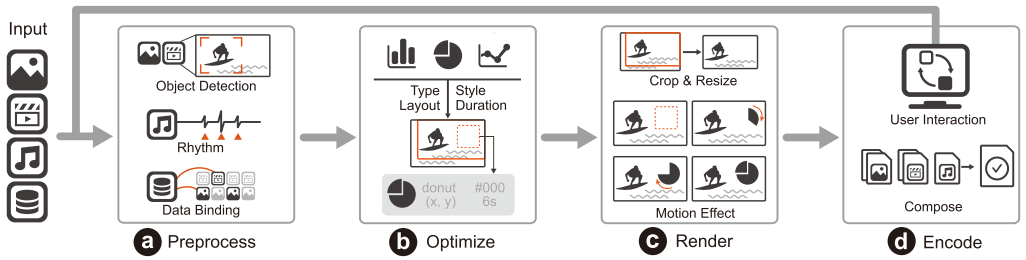

Fig. 2. System pipeline: preprocess input materials (a), optimize embedded layouts and shot durations (b), render visualizations and motion effects (c), and encode videos through composing shots and music (d). User commands are incorporated to facilitate the iterative editing of output videos.

approach that aligns shots according to the semantic boundaries of the music clips. A highly advanced model [53] was further developed to select a subset of shots that best match music beats in terms of time differences. However, these music-centric models may abandon some shots during the optimization process. To fully use shots, we propose a video-centric model that optimizes the match between the shots and the music rhythm.

3 APPROACH OVERVIEW AND DESIGN CONSIDERATIONS

In this section, we introduce the overall video generation pipeline of SmartShots (Figure 2) and then summarize the crucial DCs of VDE. Our approach is designed to integrate non-verbal video clips, images, a melody, and a text file that describes tabular data in the JSON format to create VDE. To fully explore the design space and collect the DCs of VDE, we collaborated with a group of five experts from various backgrounds, including information visualization (two), design (two), and video production (one). We selected these experts based on the following considerations. The two visualization experts were senior researchers who had decades of experience in visual analytics and design [38]. They could provide the necessary knowledge to answer the questions regarding visualization design, such as *how to craft effective data visualizations in a video context?* The two designers had more than 10 years of experience in graphic/animation design. Finally, the video expert was a research director in a large corporation, and he was leading a video generation project. They provided useful answers to some of our questions, like *how to improve the aesthetic quality of VDE?* and *how to make VDE appeal to ordinary viewers?* Our discussions with these experts were roughly divided into two phases. In the initial meetings, we explored the overall design space of producing VDE and collected professional videos that included at least one embedded data visualization to identify design issues. Based on our findings in this phase, we proposed and iteratively refined an initial design space to associate the design issues until all these issues were covered. We then manually crafted VDE and performed a literature review to answer these design-related questions. Based on our video-making experiences and expert reviews, we identified the following design space and DCs.

3.1 Dimension I: Visualization

Given that "a picture is worth a thousand words," data visualization shows great potential in enhancing the video content [38]. This dimension answers the visualization-associated issues.

Q1. How many visualizations are suitable in a shot? Putting data visualizations into videos can spatially relate data with the physical relevant entities [58], thereby presenting data insights more effectively. However, such enhancement may also increase cognitive load since viewers must pay attention to both the data visualizations and the referents simultaneously. Given that animated data

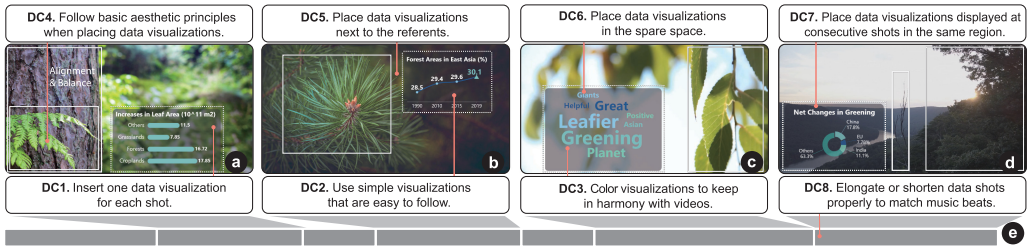

Fig. 3. A news video about environment protection is generated by SmartShots: net changes in leaf area (a), increases of forest area (b), comments on social media (c), and distribution of greening changes (d). The DCs guiding the video generation are highlighted. White dashed and solid boxes are labeled to identify visualizations and their referents. The timeline (e) presents shots' duration, and the white gaps between shots are the detected music beats aligned with shot transitions.

visualizations may catch viewers' attention [9], they may also distract from the original video content. Thus, a trade-off exists between enriching videos and limiting the number of visualizations.

DC1. Insert one data visualization for each shot. According to our observations from the professional data videos and the experiences of the manually crafted VDE, a data visualization is informative enough to be presented in one shot. The experts suggested "it may exceed viewers' cognitive ability if they watch multiple visualizations concurrently." To well balance embedded visualizations and video content, we follow this empirical rule that only inserts one data visualization for each shot (Figure 3(a)).

Q2. What kind of visualizations are applicable for videos? Although various visualizations have been developed to deal with different types of data and tasks [38], how to choose a proper one remains an unsolved challenge [56]. Complicated data visualizations may be able to handle complex analytical tasks but can also lead to heavy perceptual burden, especially played in a video context.

DC2. Use simple visualizations that are easy to follow. The experts indicated that "videos should avoid unfamiliar and complex visual designs to appeal to ordinary viewers." Moreover, previous studies also introduced well-established guidelines [25] for animated visualizations and summarized commonly used visualizations for data videos [9]. Inspired by these works, we employ the common visualizations, namely *line chart*, *bar chart*, *pie/donut chart*, and *pictographs*. To further increase the expressiveness of the library, we develop variants of the basic visualizations by changing their visual glyphs or orientations and finally obtain 15 visualizations (see Figure 6(f) later in this article).

Q3. How does the style of data visualizations interact with the video content? Color design plays an important role in creating data visualizations where the *effectiveness* principle [38] is the primary consideration [55]. However, the color design for in-video data visualizations must consider the influences of the periphery objects. Different color schemes are preferred on a "case-by-case" basis. For example, harmonic colors can be used to avoid the intrusiveness of graphic designs, whereas contrasting colors may be favorable for highlighting embedded entities.

DC3. Color visualizations to keep in harmony with videos. Our goal is to ensure that viewers could pay equal attention to both visualizations and video content instead of focusing on one side. Since some intelligent methods [60] are developed to balance embedded visualizations and video content, we adopt them to obtain harmonic colors for in-video visualizations (see Figure 3(c)).

3.2 Dimension II: Embedded Layout

Properly positioning data visualizations in a shot results in an *embedded* layout. In contrast to optimizing single-page layouts [39, 42], the positions and sizes of the visual referents in videos are fixed. This dimension discusses the design issues of an embedded layout.

Q4. How to guarantee the aesthetic quality of embedded layouts? Artists have proposed various high-level aesthetic principles for graphic design [61, 63]. However, the design of embedded layouts is different because the position of visual referents are fixed. Thus, it remains unclear for embedded layouts what kind of aesthetic principles should be followed.

DC4. Follow basic aesthetic principles when placing data visualizations. Pursuing the high aesthetic quality of videos is critical since the general goal of videos is to attract viewers. The experts commented that "the crucial criterias for designing embedded layouts were aligning the visual objects and balancing their visual weights." Therefore, we employ the two layout principles (see Figure 3(a)), namely balance [33] and alignment [10], to optimize embedded layouts.

Q5. How to preserve the semantic links between visualizations and referents? The semantic relations between the in-video visualizations and the respective referents are often conceptual and usually defined by users [58]. Some visual linking methods can reveal such semantic relationships intuitively. For example, users can use lines or shapes to connect or group visualizations and referents.

DC5. Place data visualizations next to the referents. Although the explicit linking methods can reveal semantic relationships visually, they also introduce additional visual labels that may hinder the presentation of both the data visualizations and the video content. The experts indicated that "an alternative way was spatially placing visualizations and referents to reveal the hidden relationship." According to gestalt principles [50], we adopt *proximity* [32], which indicates the underlying relationships by putting visualizations close to the referents (see Figure 3(b)).

Q6. How to choose the spatial anchor point of visualizations? Random placement of embedded visualizations may result in an illegible layout, which may break viewers' mental map and decrease engaging feelings. Determining the proper positions in videos for data visualizations poses a great challenge since many design factors should be considered.

DC6. Place data visualizations in the spare space. White space is a fundamental element in graphic design that is associated with the *readability* principle [39]. To maintain the readability of embedded layouts, we place data visualizations in a space that is not occupied by the salient visual objects of videos (see Figure 3(c)). Otherwise, in-video data visualizations may hinder the presentation of the original video content and prevent the dissemination of important information.

DC7. Place data visualizations displayed at consecutive shots in the same region. Generating embedded layouts in different shots without considering the stability of videos may lead to poor-quality results. For example, data visualizations embedded in consecutive shots may appear in opposing positions in videos, thereby breaking the mental map of viewers. The experts suggested that "video captions should always appear at the bottom, which could inspire the design of embedded layouts." We place the consecutively displayed data visualizations in the same region to maintain the coherence of videos (see Figure 3(d)).

3.3 Dimension III: Rhythm

When assembling data shots into a complete video, identifying the correct timing of shot transitions presents a challenge because (1) the durations of data shots build on images are unknown and (2) the extent to which the shots can be shortened or elongated is unclear. Our dimension of rhythm intends to address these issues and determine the time points of shot transitions precisely.

Q7. How to determine the shot durations according to the music rhythm? One method to determine shot durations is evenly dividing the timeline according to the number of shots, which may produce extreme short shots when dealing with a large number of video clips. It is difficult to keep viewers oriented and ensure they understand the visualization within a limited time if the shot is too short.

DC8. Elongate or shorten data shots properly to match music beats. There are two situations to consider. First, for data shots built on images, we employ animations to simulate a sense

of motion. The experts commented that "each shot should have sufficient time to be played." According to our video-making experiences, we set the default duration of those shots to 4seconds. Second, for data shots built on video clips, we keep the original duration before assembling them into the final video. Real-world videos are often accompanied by music to promote user enjoyment and satisfaction. To inspire more engaging feelings for the viewers, we assemble data shots according to the music rhythm. To ensure that the shot transitions are in line with the music beats, we slightly elongate or shorten both types of data shots whenever needed (see Figure 3(e)).

4 VIDEO GENERATION APPROACH

In this section, we formulate the design of VDE as an optimization problem and introduce in-video visualization designs. To find satisfactory visualizations, generate appealing layouts, and determine suitable durations for data shots, we propose a set of energy functions in accordance with the preceding DCs. We also present an efficient solver and employ a dynamic programming algorithm to address the optimization problem.

4.1 Problem Definition

To produce VDE via optimization, we define the problem as follows: given m data shots $S(D) = \{S_i(D_i)\}_i^m$, optimize (V, X, T) by

$$\min_{V,X,T} E(V, X, T) = E_{music}(T) + \sum_{i=1}^{m} E_{shot}(V_i, X_i), \tag{1}$$

where $E(\cdot)$ represents an energy function and $D = \{D_i\}_1^m$ represents the input data. According to DC1, each shot should only contain one data visualization. For simplicity, we allow the data D_i to be *null*. $V = \{V_i\}_1^m$ and $X = \{X_i\}_1^m$ denote the visualization and layout designs for the shot S_i, and $T = \{t_i\}_1^m$ denotes the durations of S.

The optimization problem Equation (1) is decomposed into two subproblems, namely shots-music composition E_{music} and shot-layout optimization E_{shot}, which can be solved independently. For the data shot S_i, the shot-layout problem is further decomposed into a set of single-frame layout problems:

$$E_{shot}(V_i, X_i) = \sum_{k=1}^{K_i} E_{frame}^{(k)}(V_i, X_i; \theta_i, X_{i-1}), \tag{2}$$

where K_i refers to the total number of frames in the shot S_i, $E_{frame}^{(k)}$ denotes the energy function defined in the kth frame, θ_i denotes the model parameters $\theta_i = [\bar{\omega}, \omega_b, \omega_a, \omega_p, \omega_r, \omega_c]$ (Equation 11), and X_{i-1} represents the layout design of the previous data shot. The single-frame layout problem will be further decomposed into five optimization components, which are detailed in Section 4.3.

4.2 Visualization Design

Choosing suitable visualizations for the input data in the given video context remains a big challenge. On the one hand, the various data types and the aims of presenting the data need to be considered. On the other hand, the chosen visualization should also fit the video context without affecting the aesthetic quality. To reduce the difficulty, we introduce the visualization design V in Equation (2). Our main idea is to first collect a set of visualization templates and then select the optimal one for the input data via optimization. To optimize the visualization design V, we extract the bounding box $[w_v, h_v]$ and visual center $[cx_v, cy_v]$ for each visualization v (Figure 4). In accordance with DC2, the visualization v is defined as one of the 15 visualization types shown later in Figure 6(f). To maintain the simplicity of the mathematical notations, we denote w_v, h_v, cx_v, cy_v

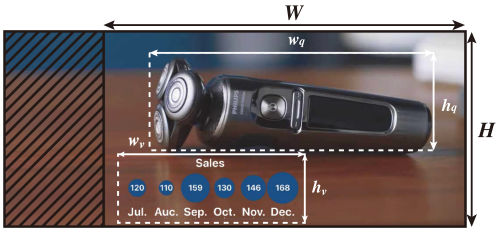

Fig. 4. Semantic analysis for the video context. Only one visual object is presented for illustration. The black rectangle refers to the viewport $[W, H]$. The bounding boxes $[w_q, h_q]$ and $[w_v, h_v]$ of visual object q and visualization v are labeled. The shadow region is abandoned after the resize-and-cut stage.

as hidden variable $\mathbf{h}_v$ because these variables are associated and determined by the visualization type v. We then define the visualization design as $V = [v, \mathbf{h}_v] = [v, w_v, h_v, cx_v, cy_v]$. We also adopt *growing* motion [9] to animate in-video data visualizations and highlight the increasing trends in the data. According to DC3, harmonic colors should be applied for in-video data visualizations to avoid intrusiveness. Inspired by previous studies [18, 51], we acquire several color templates from ColorBrewer [24] and then extract the main color of a data shot. Afterward, we pick the harmonic colors [51] from the templates according to the extracted one.

4.3 Embedded Layout Optimization

To optimize the visualization design V and obtain the embedded layout X, we define a set of energy functions according to high-level aesthetic principles. To incorporate these energy terms, we further extend Equation (2) as

$$E_{shot}(V_i, X_i) = \sum_{k=1}^{K_i} E_{frame}^{(k)}(v_i, p_i, \tilde{p}_i; \theta_i, \mathbf{h}_i, X_{i-1}), \tag{3}$$

where the layout design is defined as $X_i = [p_i, \tilde{p}_i]$ with $p_i = [x_i, y_i]$ denoting the position of an embedded data visualization and $\tilde{p}_i = [\tilde{x}_i, \tilde{y}_i]$ denoting the position of video *viewport*. The input shots may be in different sizes. Thus, we employ a resize-and-cut scheme [60, 63] to normalize these shots. As shown in Figure 4, the frames of these shots are initially resized to match the shortest edge of the viewport while preserving their aspect ratio to avoid distortion. The viewport then moves along the longest edge and cuts the frames to match the output video size specified by users denoted as $[W, H]$.

4.3.1 Energy Function. In accordance with DC4 through DC7, we extract five high-level aesthetic principles, namely *balance, alignment, proximity, readability*, and *coherence*, and then propose five energy terms. Afterward, we define the single-frame layout optimization problem through a linear combination of these energy terms.

Contextual relevance. Two challenges may be encountered when embedding a visualization into a video frame. First, we need to understand the contents of the frame. Second, we should determine to which visual object in a frame the visualization is referring. We employ an advanced computer vision technique [28] to extract object-level semantic information, including the bounding box and semantic labels of the visual objects (see Figure 4). Following the same notations presented in Equation (3), we denote the bounding box and visual center of the visual object q by $[w_q, h_q]$ and $[cx_q, cy_q]$, respectively. We assume that each data shot should only have one referent. To determine the referent, we follow the majority principle that the visual object with the most

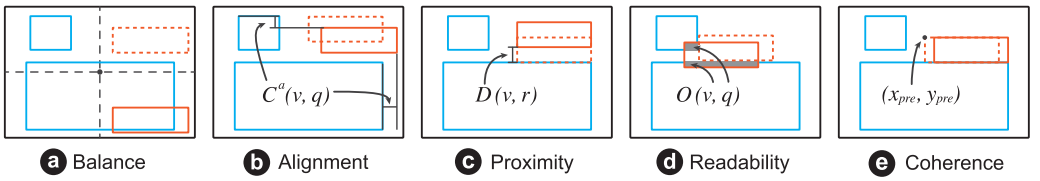

Fig. 5. Algorithm illustration. The blue rectangles represent visual objects/referents in a frame, the solid red rectangle represents a visualization to be inserted, and the dashed red rectangle indicates the position of the visualization after optimization. The mathematical notations used by different aesthetic energy terms are highlighted. (a) Balance: the symmetry center is highlighted at the geometric center of the viewport. (b) Alignment: the misalignment of the right and top edges is penalized. (c) Proximity: the distance between the visualization and referent is shortened. (d) Readability: the overlapping region colored by shadows is minimized. (e) Coherence: the visualization moves to the insertion position in the last shot.

repetitive semantic labels among all shots should be the referent. For shots without the referent, we choose the largest visual object as the alternative.

Balance. To satisfy DC4, we first describe an energy item for the *balance* principle [33], which refers to the spatial equilibrium of visual objects in a single-frame presentation. Symmetry balances, including *left-right* and *top-bottom* symmetries, are often used to evaluate graphic designs [61]. We measure the balance quality of the embedded layouts from both the horizontal and vertical directions. To simplify the computation, we obtain the visual center by calculating the geometry center that $cx = w/2, cy = h/2$. In case users want to highlight visualizations or the other visual objects, we provide a parameter ω to enable the direct control of the visual weights for the visualizations. If ω increases, then the visualizations would be strengthened and occupy a larger space in the central region of the viewport. Suppose there are Q objects in a frame; we define *balance* by

$$\begin{aligned} E_{balance} = &\left[\omega(x_i + cx_{v_i}) + (1-\omega)\frac{\sum_q (x_q + cx_q)}{Q} - \left(\tilde{x}_i + \frac{W}{2}\right)\right]^2 \\ &+ \left[\omega(y_i + cy_{v_i}) + (1-\omega)\frac{\sum_q (x_q + cy_q)}{Q} - \left(\tilde{y}_i + \frac{H}{2}\right)\right]^2. \end{aligned} \tag{4}$$

Alignment. To satisfy the *alignment* principle [10] of DC4, we define an energy term that considers six alignment types, namely *left*, *right*, *top*, *bottom*, and *vertical-* and *horizontal-center* [39]. Only one alignment type, denoted by a, can exist between a visualization v_i and a visual object q. To align visualizations and visual objects as much as possible, we define an energy term to penalize the misalignments:

$$E_{alignment} = \frac{1}{Q}\sum_q \operatorname*{argmin}_a C^a(v_i, q), \tag{5}$$

where $C^a(v_i, q)$ denotes the misalignment distance defined by

$$\begin{cases} (x_i - x_q)^2, & if\ a = left \\ (x_i + w_{v_i} - x_q - w_q)^2, & if\ a = right \\ (y_i - y_q)^2, & if\ a = top \\ (y_i + h_{v_i} - y_q - h_q)^2, & if\ a = bottom \\ (cx_{v_i} - cx_q)^2, & if\ a = vertical\ center \\ (cy_{v_i} - cy_q)^2, & if\ a = horizontal\ center. \end{cases} \tag{6}$$

Proximity. On the basis of DC5, we introduce an energy term to ensure the *proximity* between visualizations and the referents. We obtain the horizontal (HD) and vertical (VD) relative distances of the visualization v_i and its referent r by

$$\begin{aligned} HD(v_i, r) &= \min\{x_i + w_{v_i}, x_r + w_r\} - \max\{x_{v_i}, x_r\} \\ VD(v_i, r) &= \min\{y_i + h_{v_i}, y_r + h_r\} - \max\{y_{v_i}, y_r\}. \end{aligned} \tag{7}$$

To penalize the relative distance between a visualization and its referent, we first determine their relative direction M as reflected in their alignment type (Figure 5(c)). M is equal to 1 if the alignment type is *left*, *right*, or *vertical center* and is equal to 0 otherwise. We then define an energy term to maintain proximity as follows:

$$E_{proximity} = [M \times VD(v_i, r) + (1 - M) \times HD(v_i, r)]^2. \tag{8}$$

Readability. In accordance with DC6, we describe an energy term to minimize the overlaps between visualizations and visual objects and to fully utilize the spare space. Equation (8) also penalizes the potential overlap between a visualization and its referent. Therefore, we exclude the referent to reduce abundant computations. For a visualization v_i and a visual object q, we define an energy term to penalize the overlaps as

$$E_{readability} = \frac{1}{Q} \sum_{q \neq r} \max\{0, HD(v_i, q)\} \times \max\{0, VD(v_i, q)\}. \tag{9}$$

Coherence. According to DC7, we optimize the embedded layouts in the consecutively displayed shots where the visualizations should be located close to the same insertion point to maintain viewers' mental map. We define an energy term as

$$E_{coherence} = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2. \tag{10}$$

To optimize the embedded layout in a single frame, we define the energy function using a weighting scheme:

$$\begin{aligned} E^{(k)}_{frame}(v_i, X_i; \theta_i, \mathbf{h}_i, X_{i-1}) &= \omega_b E_{balance} + \omega_a E_{alighment} \\ &+ \omega_p E_{proximity} + \omega_r E_{readability} + \omega_c E_{coherence}. \end{aligned} \tag{11}$$

4.3.2 Optimization Solver. The intra-shot optimization problem (Equation (2)) is a mix-type problem that cannot be simply solved by continuous solvers. We employ a two-stage strategy in which we first develop and establish a common convex optimization solver on a gradient descending algorithm for a given visualization type. We then traverse all visualization types to determine the optimum one with the lowest energy value and obtain the corresponding embedded layout. However, directly searching such a huge solving space of Equation (2) is time consuming. Given that this equation is further decomposed into a set of single-frame optimization problems (Equation (11)), we employ a sampling scheme to extract the key frames and improve optimization efficiency. To optimize the energy function defined in Equation (11), the gradient descending solver heavily relies on the initial value and easily converges to a local optimum. We employ a random seeding method that randomly picks several initial positions from the spare space to perform an optimization. The model parameter $\theta_i = [\bar{\omega}, \omega_b, \omega_a, \omega_p, \omega_r, \omega_c]$ is specified by users to obtain satisfactory layouts. In our experiments, we observe users prefer occlusion-free and aligned layouts and tend to balance visualizations and the video content. Therefore, the default parameter is set to $\theta_i = [0.5, 1, 3, 1, 5, 1]$.

4.4 Shots Composition

After obtaining optimum visualizations and optimizing embedded layouts, we further compose data shots and music. Let $T = [t_1, \ldots, t_m]$ denote the shot durations and $B_l = [b_0, b_1, \ldots, b_l]$ denote the music beats. We derive the transitions of shots $\bar{T}_m = [\bar{t}_0, \bar{t}_1, \ldots, \bar{t}_m]$ through $\bar{t}_i = \bar{t}_{i-1} + t_i$. In accordance with DC8, the shot transitions should match the beats of the input music. Thus, we define the energy function of Equation (12) as

$$\min E_{music}(T) = -M(\bar{t}_m, b_l), \tag{12}$$

where $M(\cdot)$ is the maximum number of matched data shots.

To avoid the compression loss of data shots, the following border conditions must be satisfied: $\bar{t}_0 = b_0 = 0$ and $\bar{t}_m < b_l$. Given that the transitions $\{\bar{t}_i\}_1^m$ may not strictly match the beats $\{b_j\}_1^l$ due to time differences, we slightly elongate or shorten the data shots that built on images to perfectly match beats (DC8). We define the matching function of $\bar{t}_i$ and b_j as

$$match(\bar{t}_i, b_j) = \begin{cases} 1, \ if \ |\bar{t}_i - b_j| < \alpha * t_i \\ 0, \ otherwise, \end{cases} \tag{13}$$

where α is a threshold and set to 0.1 in our case. There exists a possibility that $\bar{t}_i$ matches two adjacent beats b_j and b_{j+1} according to Equation (13). Thus, we propose a heuristic that the closest beat should be selected.

For any m and l, the shot transitions $\bar{T}_m$ and the music beats B_l represent two sequences. The problem of finding the maximum matching beats and shots is identical to obtaining the LCS (longest common subsequence) [19] of B_l and $\bar{T}_m$. Thus, the inter-shots composition problem is converted into a typical LCS problem that is defined as

$$M(\bar{t}_m, b_l) = \begin{cases} \max\{\, M(\bar{t}_{m-1}, b_{l-1}) + match(\bar{t}_m, b_l),\ M(\bar{t}_m, b_{l-1}),\ M(\bar{t}_{m-1}, b_l)\,\}, & \text{if } m, l > 0 \\ 1, & \text{if } m, l = 0. \end{cases} \tag{14}$$

We solve the recursive function via dynamic programming. If $[\bar{t}_{i_0}, \bar{t}_{i_1}, \ldots, \bar{t}_{i_k}]$ is the obtained matching sequence that corresponds to music beats $[b_{j_0}, b_{j_1}, \ldots, b_{j_k}]$, then we update shot durations by $t_{i_n} = b_{j_n} - b_{j_{n-1}}$, $n = 1, \ldots, k$. However, some shots cannot be updated ($k < m$) because of insufficient music beats or a transition-beat mismatch. In this case, we adopt a weighting allocation scheme to update the mismatched shots. For example, two adjacent shots can be updated by $t'_{i_k-1} = \frac{t_{i_k-1}}{t_{i_k-1}+t_{i_k}}(b_{j_k} - b_{j_{k-1}})$ and $t'_{i_k} = \frac{t_{i_k}}{t_{i_k-1}+t_{i_k}}(b_{j_k} - b_{j_{k-1}})$ when they do not match the music beats.

5 SMARTSHOTS

This section presents SmartShots, an automatic system that facilitates the easy creation of VDE. SmartShots consists of three components, namely *material*, *timeline*, and *preview*, to create a minimalist interface (Figure 6). Users first upload data tables and other multi-media materials (Figure 6(h)) using the *material* panel (Figure 6(a)). To construct stories, users can then *drag and drop* any entity from *material* to *timeline*. To distinguish the chosen entities, shots are surrounded by colorful labels and placed in the middle of the *timeline* (Figure 6(b)). SmartShots also enables users to adjust the aesthetic parameters (Figure 6(g)), and preview and export videos (Figure 6(c) and (e)).

We implement SmartShots using a client-server architecture that comprises a backend that runs the optimization model to generate videos and a web interface that allows users to edit the output videos. The web interface is built in the *TypeScript* [12] and *React* framework [4], which supports the uploading of multi-media materials and the editing of videos. The backend is implemented using Python, which integrates *librosa* [35] to analyze the music rhythm, *TensorFlow* [5] to facilitate

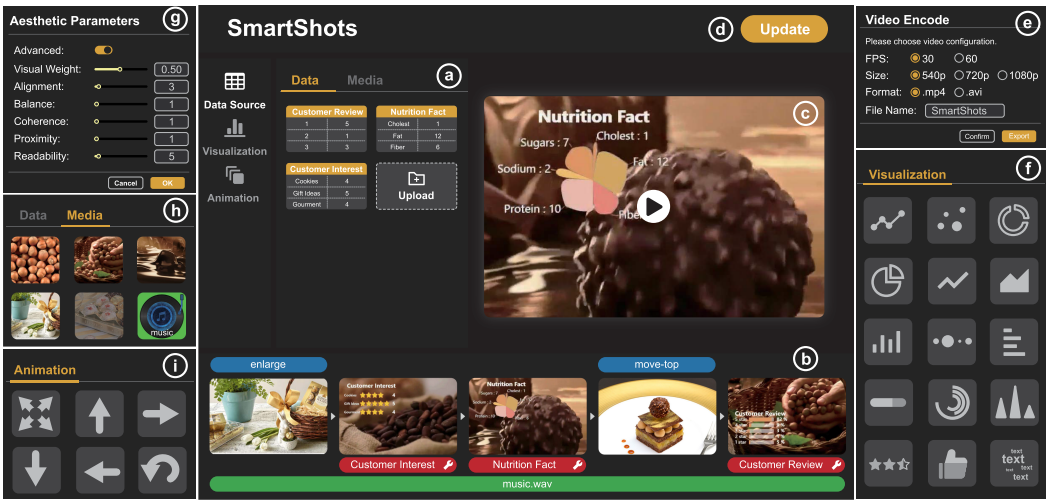

Fig. 6. SmartShots: material with data tables (a), timeline for constructing stories (b), video preview (c), video encode & export (d, e), data visualization selection (f), adjusting of aesthetic parameters (g), shot selection (h), and motion effect selection (i).

object detection [28] with a pre-trained model [43], and *OpenCV* [3] to process images and videos. We also develop a canvas renderer built in Chrome and two well-established JS libraries (*D3* [13] and *G2* [2]) to render animated visualizations.

6 USE CASES

In this section, we craft three use cases, including a news video, an introductory video, and an advertisement, to demonstrate the usage and generalizability of SmartShots.

Environment news. By using SmartShots, we craft a video (see Figure 3) about an environment-related news item that reports the progress that China and India have achieved in land greening [23]. We collect pictures and short videos about forests and trees to illustrate the beauty of a green environment. Crucial data about how China and India became leafier and people's opinions on this change are exhibited. Such data include *net changes in leaf area from 2000 to 2017*, *increases of forest areas in east Asia*, *the distribution of greening changes*, and *comments on social media*, according to the reports [15, 20]. This information can increase the video's persuasiveness and attract viewers' attention. Thus, we decide to integrate the information in the video. In total, the source materials include three video clips, four images, one JSON file that stores the data mentioned previously, and one music melody without lyrics.

To make such a news video with supporting data information, we first upload all source materials to SmartShots. Then, we arrange the order of data shots for the story: the net changes in leaf area is put at the beginning to inspire viewers' interests and show that these areas are greening; then the increases of forest area and people's social media comments are presented in the next shots; finally, the distribution of greening changes are shown to highlight the great achievements made by the two Asian countries.

After the arrangement of the order of images, videos, and data, SmartShots automatically generates a video. It chooses a *bar chart* to show net changes in leaf area and places it at the bottom right of the shot to *balance the layout* (DC4). The forest area is shown by a *line chart* to ensure that viewers can easily notice its rapid expansion. Notice that SmartShots *places the line chart next to*

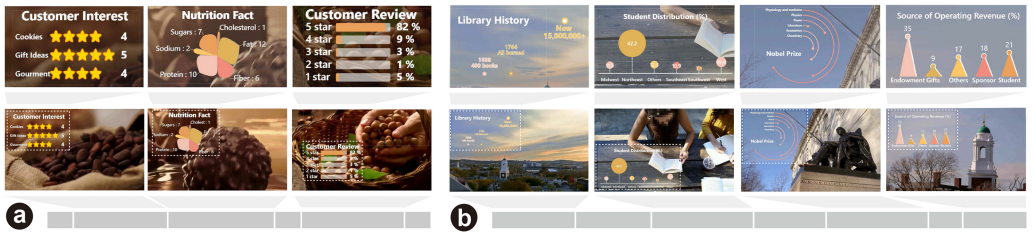

Fig. 7. Use cases. (a) A chocolate advertisement is created to demonstrate the diverse visualizations that SmartShots can support. (b) A university introductory video is crafted to indicate that SmartShots can be applied to various use scenarios.

the referent (DC5)—the leaf. In the next shots, SmartShots presents a *word cloud* of people's comments and uses a *donut chart* to show the distribution of greening changes, which is *aligned with the referent at the bottom edge* (DC4). Moreover, these two consecutive visualizations are *embedded in the same area* (DC7)—the bottom left of the shots. The four visualizations are *embedded in the spare area* (DC6) to ensure that the referent is visible.

Finally, we fine-tune the aesthetic parameters of the embedded layouts and add motion effects to animate the images.

Advertisement. We create a chocolate advertisement video (Figure 7(a)) to show that SmartShots can support the creation of advertising videos. The goal of advertisements is to promote products. Thus, we collect some relevant information about chocolates, such as the taste and nutrition facts, and customers' reviews. To integrate the information in the video, SmartShots embeds *a single visualization in each shot* (DC1) and supports *common and intuitive visualizations* (DC2). Specifically, it uses a *star-glyph bar chart*, a *filling bar chart*, and a *flower-based donut chart* to present data insights. SmartShots also automatically extracts the main color of the video and generates *visualizations with harmonic colors* (DC3).

University introduction. We create an introductory video of a famous university to show that SmartShots can be applied in diverse use scenarios and supports various visualizations. This video is used to attract more registrations. Thus, we want to greatly enhance such an introductory video through the history of the university, the distribution of the students, the scientific achievements, and school revenues. These source materials are uploaded to SmartShots. Different from the previous cases, we manually select the visualizations for the data information of the university. We choose the following: a *dot chart* to highlight significant historical events for the school library, a *bubble bar chart* to indicate the student diversity, a *radial chart* to reveal the number of the Nobel winners from the university, and a *triangle plot* to reveal the school financial status. We arrange the order of shots with which the information appears together. After that, SmartShots automatically aggregates these shots and optimizes the embedded positions for the chosen visualizations. Additionally, as shown in Figure 7(b), the transitions of shots *match the music rhythm* perfectly (DC8).

7 EVALUATION

In this section, we first evaluated the efficiency of the proposed approach. Then, we validated the effectiveness and usability of SmartShots through semi-structured interviews with both experts and amateur users who are novice video makers in video editing skills or tools.

7.1 Quantitative Analysis

To better understand the time efficiency of SmartShots, we conducted quantitative experiments on a desktop computer that is equipped with a 3.4-GHz CPU and runs Windows 10. We measured

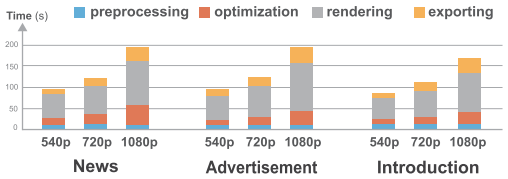

Fig. 8. Time costs of the creation processes for the three use cases, namely news, advertisement, and introductory video.

the time performance of SmartShots when generating the three use cases in terms of different resolution. We recorded the time cost of each stage (see Figure 2). We repeated the time recording of SmartShots three times and calculated the average time cost to avoid the potential influence of CPU scheduling. As shown in Figure 8, the bottleneck of SmartShots is the rendering stage. The optimization stage spends 6.2 seconds per data shot on average, which indicates that the optimization method is efficient (see orange bars in Figure 8). Although the average time cost (2.2 minutes) of the video generation seems to be a little long for SmartShots, the experiments were conducted on a personal computer and the implementation of SmartShots is not fully optimized for time performance.

7.2 User Study

To evaluate the usage of SmartShots, we conducted user studies and collected qualitative feedback.

7.2.1 Participants. We invited six participants, including three designers (D1 through D3) and three amateur users (U1 through U3). Their average age was 25 years (range = 20 − 33 years). All the designers had extensive experience in UI/UX design and were familiar with professional video tools. None of them were involved in our previous studies. All the amateur users reported having no prior experience in authoring videos or visualizations.

7.2.2 Procedure. First, we showed participants the process of creating a use case (see Figure 3) (10 minutes). Second, participants were asked to freely explore SmartShots until they were familiar with the interactions (5 minutes). Third, they were asked to make a new VDE using given materials (20 minutes). The materials included three images, three video clips, one non-lyrical music, and one JSON file that records tabular datasets. Participants needed to upload the materials to SmartShots and then edit their VDEs. They could select shots and bind data into some chosen shots. They could also make some adjustments, including the story timeline, motion effects, data visualization types, and aesthetic parameters according to their preference. They were allowed to update the authoring results iteratively. In the end, we discussed with each participant about their user experiences and the usability of SmartShots (25 minutes). The user studies and interviews were conducted independently with each participant.

7.2.3 Results. The outcome of user interviews can be summarized as follows.

- *Authoring process*: We observed the authoring process of the participants. They first uploaded all the source material to SmartShots and spent 10 minutes on average to construct the data stories using the *timeline* panel. They added/removed shots and changed the order of shots to create appealing stories. They also bound data tables to different shots to enhance the narrative. After that, they invoked the optimization module to generate VDEs. Due to the benefits of the automatic workflow (see Figure 2), the participants did not spend time dealing with low-level design tasks, such as cropping images or videos to normalize their sizes. After

obtaining the output videos, the participants would try different optimization parameters to adjust the embedding layouts. For example, U2 preferred to assign the visualizations in the same place, so he turned up the *coherence* parameter, which generated a similar visualizations layout in each shot. But in most cases, they used the default parameters because the output matched their design knowledge (D1, D3) or seemed in a reasonable form (D2, U1, U3). Some participants would also change the shots for better storytelling and refine the VDEs iteratively. According to the feedback from participants, all of them were satisfied with the final VDEs created by SmartShots.

- *User experiences*: In general, all of the participants liked the minimalist interface and easy-to-use interactions of SmartShots. They had similar comments that it was very easy to generate VDE by using SmartShots. As D1 and D2 commented, "SmartShots makes it quite easy to integrate data charts into videos," and all designers agree that SmartShots could save their efforts in designing videos. For example, Montage is a typical method in film-making that combines a series of video shots into a sequence, which can be efficiently completed using SmartShots. U3 complimented this: "I love the idea that users only need to bind data for shots and a video will be created in a few minutes." Moreover, U2 was impressed by the automatic composition of music and commented that "the beats of video transitions make the whole video more affective." **D3** also complemented, "I love the ideas of mixing music and data visualizations which produces an engaging feeling." These comments support the effectiveness of the music-video composition method.
- *User creativity*: We are also interested in whether the optimization module could inspire designers' talents and facilitate video designs. Altogether, the designers of the participants had shown great interest in the optimization model. They agreed that SmartShots can be used as a fast prototyping tool to achieve some design ideas without too much effort. For example, both D1 and D2 provided a potential usage that designers could first try different design alternatives using SmartShots and then choose the most promising design for further development.
- *Improvement*: The participants also suggested several improvements for SmartShots. First, the video *preview* panel should enable users to navigate or fast-forward the output videos. Second, the *timeline* panel only supports shot-level manipulations while ignoring inter-shots operations. It could be better for users to add transitions between two consecutive shots. Third, the data visualizations can be animated according to music rhythm. We plan to implement these features as future works to further improve SmartShots.

7.3 Expert Interview

To evaluate the effectiveness of SmartShots, we conducted semi-structured interviews with two experts. The first expert was a university teacher (ET) who taught students to make digital videos with professional tools. He can evaluate the output videos of SmartShots from the design aspect and compare SmartShots with the other existing tools. The second expert (ED) was the research director, who also participated in the discussions of the DCs. He was the only expert involved in the design space analysis of VDEs and served as one of the co-authors due to his contributions. He can evaluate the system development of SmartShots. The interviews include a 20-minute demonstration of SmartShots and use cases and a 40-minute discussion.

ET mainly focused on the design of the use cases and agreed with the optimization goals that pursue aesthetic layouts for embedded visualizations. He appreciated the *coherence* goal of our model, "viewers are accustomed to obtaining information according to their past experiences. Thus, it is necessary to present visualizations in the same region of videos." As for the comparison with professional tools, he commented, "SmartShots is definitely more efficient and intelligent." To discuss

this, he demonstrated to us a general design procedure for manually creating such a data video. Designers first analyze data using analytic tools (e.g., Excel) and select a suitable visualization for the data. Then, they search for an appropriate position, craft motions with professional tools, and fine-tune iteratively. Thus, it usually takes hours to generate a data video. However, that can be shortened greatly using SmartShots—it only needs a few minutes to generate such a video. Nevertheless, ET thought that "professionals may still prefer the advanced tools since these tools support more powerful functions, such as editing frames."

ED commented on the system development of SmartShots. He was impressed by the optimization module that produces appealing embedded layouts and commented that "the layouts seem to be all right since they avoid the occlusion with products skillfully." In general, he confirmed the usefulness and the intuitiveness of the interface but also pointed out that "it is tedious to bind data manually if there are many shots and datasets." To address the scalability issue, we plan to employ a heuristic method to support automatic data binding. The idea is to construct data shots according to the semantic connections. For example, we can extract the semantic labels from the video content and then calculate the similarity between the semantic labels and the textual information of data tables, such as table title or data attributes. The shots can be associated with data tables according to their similarity scores.

8 DISCUSSION

In this section, we thoroughly discuss the implications and limitations of this study, along with the potential usage of SmartShots.

Implications. The implications of this work are twofold. First, to the best of our knowledge, we are the first to focus on the design of VDE. To fully explore the design space, we systematically study design issues related to this new form of data videos [9, 45] with a group of experts. We further propose a set of DCs to answer these issues from three dimensions, namely *visualization*, *embedded layout*, and *rhythm*. The DCs can be applied as fundamental guidelines to inspire designers in creating effective VDE. Second, VDE effectively reflects the concept of *visualization for mass* with the explosive development of video marketing. However, creating such information-rich videos manually is difficult because users need to select suitable data visualizations, fit them in the video context (e.g., arrange positions), and finally generate a new video that integrates data visualizations, images, video clips, and music. To ease the difficulty, we develop SmartShots, which can make this new form of data videos more accessible and serve numerous amateur users and audiences.

Potential usage. With the explosive development of the video market, it is difficult to make a video stand out from the crowd. Yet the data-driven videos provide a promising paradigm that could inspire video creators to improve their works. Expect for generating VDEs, SmartShots is potential to create other types of videos. In the expert interview, the experts provided some promising uses of SmartShots, which are summarized as follows:

- *Social videos*: Video blogs, or vlogs [57], which integrate text, images, and short videos to present life moments, are well received on social media recently. SmartShots integrates an optimization model to assemble multi-media materials, which can help generate data-rich vlogs (e.g., exercise videos with calorie consumption).
- *Live videos*: Enriching live videos, such as online TED Talks, had a great potential to make SmartShots more accessible. A similar case is the Gapminder video [44]. In addition, SmartShots could be further improved to enhance the increasingly popular live videos hosted by social media influencers to promote products where numerous audiences would be served. Although we currently focus on offline video generation, we would like to explore how to extend SmartShots to embed visualizations in a live video stream.

Limitations. We summarize three limitations of SmartShots. First, the optimization results may be heavily affected by inaccurate semantic information. For example, mislabeled bounding boxes may result in undesired layouts that cover salient objects in videos. Thus, we plan to employ a more advanced deep learning model to improve the accuracy of the semantic analysis. Second, the optimization model mainly focuses on the contextual relevance and aesthetic principles when determining the visualization type and the insertion point of data. However, the complexity of data representation and the underlying insights are also vital design factors when choosing a suitable visualization [38], which is difficult to formulate due to the richness of data. Thus, SmartShots enables users to manually set data visualizations when the users are not satisfied with the automatically generated ones. To further improve SmartShots, we also plan to take the data complexity and richness into account when optimizing visualizations. Third, SmartShots is not highly optimized for generating long videos with dozens of shots. We plan to improve the time performance of SmartShots via GPUs. Fourth, we now mainly consider to embed a single visualization into a video scene. However, there may be some design situations where users want to play multiple visualizations in a single view. Thus, we plan to extend the optimization model to incorporate multiple embedded visualizations.

9 CONCLUSION

In this work, we thoroughly explore the design space and summarize primary DCs (DC1 through DC8) to design VDE. Based on that, we formulate the design of VDE as an optimization problem and propose a set of energy functions. We implement the optimization algorithm and develop SmartShots that facilitates the automatic generation of VDE. We demonstrate the usage and usability of SmartShots through use cases and user evaluations. Our work takes the initial step toward a deep understanding of how to embed data visualizations in videos, which is full of opportunities for further studies.

REFERENCES

[1] E-tailing Group. 2013. *How Consumers Shop with Video: Based on a 4Q 2012 Research Study of 1000 Consumers.* Technical Report.

[2] GitHub. 2019. G2. Retrieved March 31, 2019 from https://github.com/antvis/g2.

[3] OpenCV. 2019. OpenCV. Retrieved March 31, 2019 from https://opencv.org/.

[4] React. 2019. React. Retrieved March 31, 2019 from https://reactjs.org/.

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*. 265–283.

[6] Amazon. 2005. Amazon Mechanical Turk. Retrieved March 31, 2019 from https://www.mturk.com/.

[7] 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 ACM Conference on Human Factors in Computing Systems*. 1459–1468.

[8] 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 Working Conference on Advanced Visual Interfaces*. 1–9.

[9] 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 (2017), 501–510.

[10] Helen Y. Balinsky, Anthony J. Wiley, and Matthew C. Roberts. 2009. Aesthetic measure of alignment and regularity. In *Proceedings of ACM Symposium on Document Engineering*. 56–65.

[11] Ronald C. Barker and Chester L. Schuler. 1985. Video composition method and apparatus. US Patent 4,538,188.

[12] Gavin Bierman, Martín Abadi, and Mads Torgersen. 2014. Understanding typescript. In *Proceedings of the European Conference on Object-Oriented Programming*. 257–281.

[13] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309.

[14] Jere Brophy. 2003. *Using Video in Teacher Education*. Emerald Group Publishing Limited.

[15] Zhutian Chen, Yijia Su, Yifang Wang, Qianwen Wang, Huamin Qu, and Yingcai Wu. 2020. MARVisT: Authoring glyph-based visualization in mobile augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 26, 8 (2020), 2645–2658.
[16] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. 2003. Automatic video summarization by graph modeling. In *Proceedings of IEEE Conference on Computer Vision*. 104–109.
[17] D. Coelho and K. Mueller. 2020. Infomages: Embedding data into thematic images. *Computer Graphics Forum* 39, 3 (2020), 593–606.
[18] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. 2006. Color harmonization. *ACM Transactions on Graphics* 25, 3 (2006), 624–630.
[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (3rd ed.). MIT Press, Cambridge, MA.
[20] Food and Agriculture Organization of the United Nations. 2018. The State of World's Forests. Retrieved March 31, 2019 from http://www.fao.org/state-of-forests/en.
[21] T. Ge, Y. Zhao, B. Lee, D. Ren, B. Chen, and Y. Wang. 2020. Canis: A high-level language for data-driven chart animations. *Computer Graphics Forum* 39, 3 (2020), 607–617.
[22] Jinlian Guo, Tao Mei, Falin Liu, and Xian-Sheng Hua. 2009. AdOn: An intelligent overlay video advertising system. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*. 628–629.
[23] Roger Harrabin. 2019. China and India Help Make Planet Leafier. Retrieved March 31, 2019 from https://www.bbc.com/news/science-environment-47210849.
[24] Mark Harrower and Cynthia A. Brewer. 2003. ColorBrewer.org: An online tool for selecting colour schemes for maps. Cartographic Journal 40, 1 (2003), 27–37.
[25] Jeffrey Heer and George G. Robertson. 2007. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1240–1247.
[26] Srinidhi Hegde, Jitender Maurya, Aniruddha Kalkar, and Ramya Hebbalaguppe. 2020. SmartOverlays: A visual saliency driven label placement for intelligent human-computer interfaces. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 1121–1130.
[27] Yongtao Hu, Jan Kautz, Yizhou Yu, and Wenping Wang. 2015. Speaker-following video subtitles. *ACM Transactions on Multimedia Computing, Communications, and Applications* 11, 2 (2015), 1–17.
[28] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, et al. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7310–7311.
[29] Robert Kosara and Jock Mackinlay. 2013. Storytelling: The next step for visualization. *IEEE Computer* 46, 5 (2013), 44–50.
[30] Kuno Kurzhals, Fabian Göbel, Katrin Angerbauer, Michael Sedlmair, and Martin Raubal. 2020. A view on the viewer: Gaze-adaptive captions for videos. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 1–12.
[31] 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 (2015), 84–90.
[32] William Lidwell, Kritina Holden, and Jill Butler. 2010. Proximity. In *Universal Principles of Design*. Rockport Publishers, Beverly, MA, 196–197.
[33] Simon Lok, Steven Feiner, and Gary Ngai. 2004. Evaluation of visual balance for automated layout. In *Proceedings of the ACM Conference on Intelligent User Interfaces*. 101–108.
[34] Junhua Lu, Jie Wang, Hui Ye, Yuhui Gu, Zhiyu Ding, Mingliang Xu, and Wei Chen. 2020. Illustrating changes in time-series data with data video. *IEEE Computer Graphics and Applications* 40, 2 (2020), 18–31.
[35] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in Python. In *Proceedings of the Python in Science Conference*. 18–25.
[36] Ann McNamara, Katherine Boyd, Joanne George, Weston Jones, Somyung Oh, and Annie Suther. 2019. Information placement in virtual reality. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces*. IEEE, Los Alamitos, CA, 1765–1769.
[37] Tao Mei, Xian-Sheng Hua, Linjun Yang, and Shipeng Li. 2007. VideoSense: Towards effective online video advertising. In *Proceedings of the ACM International Conference on Multimedia*. 1075–1084.
[38] Tamara Munzner. 2014. *Visualization Analysis and Design*. CRC Press, Boca Raton, FL.
[39] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2014. Learning layouts for single-page graphic designs. *IEEE Transactions on Visualization and Computer Graphics* 20, 8 (2014), 1200–1213.
[40] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with interactive layout suggestions. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 1221–1224.
[41] John Pavlik. 2000. The impact of technology on journalism. *Journalism Studies* 1, 2 (2000), 229–237.

[42] Lisa Purvis, Steven Harrington, Barry O'Sullivan, and Eugene C. Freuder. 2003. Creating personalized documents: An optimization approach. In *Proceedings of the ACM Symposium on Document Engineering*. 68–77.
[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2017), 1137–1149.
[44] Hans Rosling. 2009. Gapminder. *Gapminder Foundation*. Retrieved March 31, 2019 from http://www.gapminder.org.
[45] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 1139–1148.
[46] Tan Tang, Sadia Rubab, Jiewen Lai, Weiwei Cui, Lingyun Yu, and Yingcai Wu. 2019. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 769–778.
[47] Tan Tang, Junxiu Tang, Jiayi Hong, Lingyun Yu, Peiran Ren, and Yingcai Wu. 2020. Design guidelines for augmenting short-form videos using animated data visualizations. *Journal of Visualization* 23 (2020), 707–720.
[48] Tan Tang, Junxiu Tang, Jiewen Lai, Lu Ying, Peiran Ren, Lingyun Yu, and Yingcai Wu. 2020. SmartShots: Enabling automatic generation of videos with data visualizations embedded. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4509–4511.
[49] Tao Mei, Xian-Sheng Hua, and Shipeng Li. 2009. VideoSense: A contextual in-video advertising system. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 12 (2009), 1866–1879.
[50] Dejan Todorovic. 2008. Gestalt principles. *Scholarpedia* 3, 12 (2008), 5345.
[51] Masataka Tokumaru, Noriaki Muranaka, and Shigeru Imanishi. 2002. Color design support system considering color harmony. In *Proceedings of the IEEE Conference on Fuzzy Systems*. 378–383.
[52] D. Van Krevelen and R. Poelman. 2010. A survey of augmented reality: Technologies, applications, and limitations. *International Journal of Virtual Reality* 9, 2 (2010), 1.
[53] Jinjun Wang, Engsiong Chng, and Changsheng Xu. 2006. Fully and semi-automatic music sports video composition. In *Proceedings of the IEEE Conference on Multimedia and Expo*. 1897–1900.
[54] Jinjun Wang, Changsheng Xu, Engsiong Chng, Lingyu Duan, Kongwah Wan, and Qi Tian. 2005. Automatic generation of personalized music sports video. In *Proceedings of the ACM Conference on Multimedia*. 735–744.
[55] Yunhai Wang, Xin Chen, Tong Ge, Chen Bao, Michael Sedlmair, Chi-Wing Fu, Oliver Deussen, and Baoquan Chen. 2019. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 820–829.
[56] Yunhai Wang, Fubo Han, Lifeng Zhu, Oliver Deussen, and Baoquan Chen. 2018. Line graph or scatter plot? Automatic selection of methods for visualizing trends in time series. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (2018), 1141–1154.
[57] Wikipedia. 2019. Vlog. Retrieved March 31, 2019 from https://en.wikipedia.org/wiki/Vlog.
[58] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. 2017. Embedded data representations. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 461–470.
[59] Karthik Yadati, Harish Katti, and Mohan Kankanhalli. 2014. CAVVA: Computational affective video-in-video advertising. *IEEE Transactions on Multimedia* 16, 1 (2014), 15–23.
[60] Xuyong Yang, Tao Mei, Ying-Qing Xu, Yong Rui, and Shipeng Li. 2016. Automatic generation of visual-textual presentation layout. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 2 (2016), 1–22.
[61] Jiajing Zhang, Jinhui Yu, Kang Zhang, Xianjun Sam Zheng, and Junsong Zhang. 2017. Computational aesthetic evaluation of logos. *ACM Transactions on Applied Perception* 14, 3 (2017), 1–21.
[62] Jiayi Eris Zhang, Nicole Sultanum, Anastasia Bezerianos, and Fanny Chevalier. 2020. DataQuilt: Extracting visual elements from images to craft pictorial visualizations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 1–13.
[63] Yunke Zhang, Kangkang Hu, Peiran Ren, Changyuan Yang, Weiwei Xu, and Xian-Sheng Hua. 2017. Layout style modeling for automating banner design. In *Proceedings of the ACM Conference on Multimedia Thematic Workshops*. 451–459.
[64] Ying Zhao, Haojin Jiang, Qi'an Chen, Yaqi Qin, Yitao Wu, Shixia Liu, Zhiguang Zhou, Jiazhi Xia, and Fangfang Zhou. 2021. Preserving minority structures in graph sampling. *IEEE Transactions on Visualization and Computer Graphics* 27 (2021), 1698–1708.

Received August 2020; revised May 2021; accepted August 2021