Illustrating Changes in Time-Series Data With Data Video

Junhua Lu
State Key Lab of CAD&CG, Zhejiang University

Jie Wang
State Key Lab of CAD&CG, Zhejiang University

Hui Ye
State Key Lab of CAD&CG, Zhejiang University

Yuhui Gu
State Key Lab of CAD&CG, Zhejiang University

Zhiyu Ding
Cloud BU, Huawei Technologies Co Ltd

Mingliang Xu
School of Information Engineering,
Zhengzhou University

Wei Chen
State Key Lab of CAD&CG, Zhejiang University

***Abstract*—Understanding the changes of time-series is a common task in many application domains. Converting time-series data into videos helps an audience with little or no background knowledge gain insights and deep impressions. It essentially integrates data visualizations and animations to present the evolution of data expressively. However, it remains challenging to create this kind of data video. First, it is difficult to efficiently detect important changes and include them in the video sequence. Existing methods require much manual effort to explore the data and find changes. Second, how these changes are emphasized in the videos is also worth studying. A video without emphasis will hinder an audience from noticing those important changes. This article presents an approach that extracts and visualizes important changes of a time-series. Users can explore and modify these changes, and apply visual effects on them. Case studies and user feedback demonstrate the effectiveness and usability of our approach.**

Digital Object Identifier 10.1109/MCG.2020.2968249
Date of publication 21 January 2020; date of current version 28 February 2020.

Showing change over time is prevalent in data storytelling and data journalism.* Data videos, as a way to show the change in the data, are widely used for visualization in public spaces. They can be found on TV news, news application, and short video applications on mobile devices, and public spaces like large screens at libraries or museums. Typically, the general audience (an audience that does not have much background knowledge of the data presented in the video) watches these videos to learn some insights from the data. Thus, it is critical to communicate and inform them with facts, rather than display the time-series data linearly without any emphasis. Rosling's famous talk** "200 countries, 200 years, 4 minutes" narrated the changes of 200 countries' income and expected lifespan across 200 years with an animated bubble chart. He used attention cues (e.g., highlighting, zooming, textual, and graphical annotations, varying the displaying speed) to highlight important historical events or trends that he discovered, and thus to make the video informative and engaging.

While data videos for storytelling appeal to a wide range of audiences, it is challenging to create a data video of time-series which presents desired patterns. The author first creates a raw video that displays the time-series data chronologically. Then, the author needs to either explore the raw data or watch the raw video to identify those timestamps of interest when desired trends are shown. Based on how the data behaves on these timestamps, the author may apply different attention cues to convey the information behind them in a more engaging way. The author has to review the video and probably makes some modification before finally exporting it as a video.

Most existing solutions suppose that the author already decides what insights to be conveyed in videos. In the case of time-series data, the timestamps to be highlighted are selected before creating the video. However, this is not always the case. As one data storytelling practitioner mentioned,[2] "with numbers, it is good to explore them and turn them on their heads and try them in different ways and see where the story is." To our best knowledge, existing methods either require much effort or are restricted by templates while highlighting these insights.

To address these problems, we propose an approach which focuses on visually enhancing important changes in time-series data with data video. The approach assists data video creation by better support of data exploration, identification for important timestamps, and providing flexible operation with visual effects to highlight them in the video sequences. Authors can review and refine the sequence. Thus, the sequence can be enhanced by incorporating user experiences into the video generation process so as to better convey important changes in the data.

We design **data2video**, an authoring tool that supports interactive detection and depiction of changes in time-series. Case studies are conducted to evaluate the usefulness of our system.

To summarize, the contributions are twofold.

- A prototype system that supports preference configuration, interactive preview, and sequence editing.
- Curated visualization and interaction to support data exploration and visual enhancing.

The rest of this article is structured as follows. The "Related Work" section reviews related work. The "Problem Schema and Approach Overview" section defines the problem and gives an overview of our approach. The "Change Point Detection and Segmentation of Time-Series" section introduces the algorithm for detecting trends in the time-series data. The "Visual Interface" section describes the prototype implementation of our approach. Case studies and user feedback are elaborated in the "Evaluations" section. The "Discussion" section discusses the lessons learned, limitation, scalability, and generalizability of our approach. The "Conclusion and Future Work" section concludes this article.

RELATED WORK

Narrative Visualization

Segel and Heer[3] introduced the term "narrative visualization" and organized the design space as three divisions, namely, narrative genre, visual narrative tactics, and narrative structure tactics.

*Martha Kang, exploring the seven different types of data stories, https://mediashift.org/2015/06/exploring-the-7-different-types-of-data-stories/

**https://www.youtube.com/watch?v=jbkSRLYSojo

Many narrative visualization works center on developing new techniques or tackling problems that belong to the latter two divisions. Stolper *et al.*[4] analyzed narrative visualization stories to obtain detailed techniques for guiding designers to tell better stories. Amini *et al.*[5] examined the narrative structures of 50 data videos through the lens of cinematography. Hullman and Diakopoulos[6] investigated the rhetorical techniques used in narrative visualization. Beyond these divisions, Lee *et al.*[7] proposed a visual data storytelling process (VDSP) which summarizes how visual storytellers transform raw data into visually shared stories. VDSP includes three components: exploring data, making a story, and telling a story. Our approach focuses on better support the first two components.

Data Video and Animated Data Visualizations

Among the seven genres of narrative visualization defined in the work by Segel and Heer,[3] data-driven film/video/animation has become increasingly popular with the development of multimedia techniques. Plenty of examples can be found in online media.

Besides Rosling's numerous famous talks on world socio-economic data using GapMinder, there are other applications of animated scatterplot in data analysis.[8,9] Wang *et al.*[10] applied time remapping and foreshadowing techniques while generating the animated visualization to convey data patterns to the general audience. Sigovan *et al.*[11] used animation to illustrate dynamic communication patterns and helped users to analyze large datasets in parallel application execution. In the area of scientific visualization, animation is used to summarize events in time-varying data[12] or summarize users' navigation through the data[13] or parameter space.[14]

There has been an increasing demand for data video authoring tools with the popularity of data videos. Existing data video authoring tools range from commercial authoring tool applications (e.g., Animaker[†]) to authoring tools developed in the visualization community.[15] However, these tools assume that the authors have already explored the data and are aware of what to present in the video. Also, the entire creating process using these tools are completely carried out manually by authors. Our approach can ease the burden of authors by serializing the exploration process.

PROBLEM SCHEMA AND APPROACH OVERVIEW

In this section, we formalize the problem and describe user requirements. The pipeline of our proposed approach is also illustrated.

Problem Definition and Tasks

Two-dimensional (2-D) time-series data are ubiquitous. This type of data can be viewed as two tables, each for one *dimension* (e.g., income and life expectancy). Each row of the table represents time-series of the one *item* (e.g., one country), each column represents one timestamp. The value in the cell represents the value of a corresponding item at that timestamp. Each value in the cell is numerical.

Changes from the recorded time-series data are extracted since they may contain meaningful information about the different states and transitions of the items for analysis purposes.[16]

The target users of our method are data storytellers. Thus, we had discussions with two data storytellers about their requirements while telling the stories with time-series data and identified the following tasks.

- *T1: Obtain changes in the time-series data*. Various changes of time-series exist in the dataset; thus, how to find and organize them in a sequence is an important problem.
- *T2: Review detected changes*. Changes should be reviewed for their interestingness or clarity. Authors may remove some changes and keep the most important changes to make a story informative, or simply modify the timespan of changes to make them more reasonable.
- *T3: Enhance changes and generate the video*. Users can apply different strategies to highlight those changes in the video sequence. Some Refinements can also be made. Finally data videos are generated based on the sequences.

Approach Overview

Our approach aims at enhancing changes in a 2-D time-series data through generating a data

[†] https://www.animaker.com/

video. This is accomplished by integrating the capabilities of both data mining and visualization techniques within the following pipeline (see Figure 1).

Step 1: Important changes extraction in the time-series data: A set of important changes are extracted from the 2-D time-series data. The authors can make some configurations based on their preference before extraction.

Step 2: Exploring changes in the data using the data timeline: The extracted changes are stored and visualized on the data timeline. Authors can explore them with the time indicator. The changes can be modified, removed, or added. With this timeline, authors can also explore the entire time-series to find some facts that interest them but not extracted automatically.

Step 3: Previewing and modifying enhancements using the editing timeline: Enhancements (effects like slowing down of the displaying speed or zoom into a specific area of visualization) for those changes are generated based on heuristic rules. After previewing these enhancements, authors can change the effects.

Step 4: Refining and generating data videos: Authors can repeat Step 2 and Step 3 to refine the video sequence. Data videos are then generated based on the sequence.

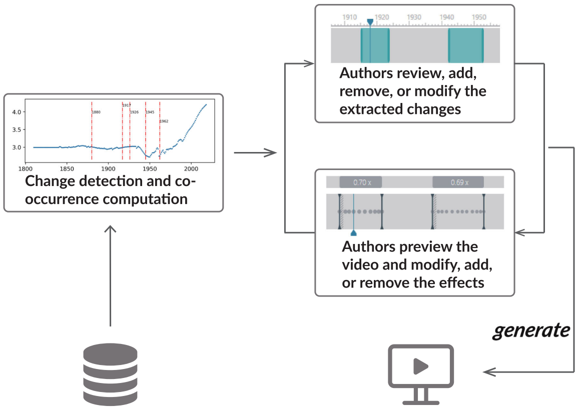
Change detection and co-occurrence computation
4.0
3.5
3.0
1800
1850
1900
1950
2000
1910
1920
1930
1940
1950
Authors review, add, remove, or modify the extracted changes
0.70 x
0.69 x
Authors preview the video and modify, add, or remove the effects
generate

Figure 1. Pipeline of our approach. First, important changes are extracted from the time-series data. They are visualized on the data timeline, and authors can view and explore them. Correspondingly, an editing timeline is provided to show how the visual enhancements are applied to the changes on the data timeline. Authors can make modifications to both timelines. Finally, the video will be generated on the editing timeline.

CHANGE POINT DETECTION AND SEGMENTATION OF TIME-SERIES

While showing the change over time with visualization is common in data journalism,[17] it is vague what does "change" exactly mean and what types of change are important, and others are trivial. This section defines what is change from the perspective of time-series data mining. We then identify two types of important changing segments based on this definition. Algorithms for detecting and aggregating these changes are also provided.

Detection of Change Points and Segmentation of Time-Series

Consider a time-series $y = \{y_1, \ldots, y_T\}$ that takes value in $\mathbb{R}^2$ and has T timestamps (i.e., 2-D time-series data). The time-series y is assumed to be piecewise stationary, which means that some characteristics of y change abruptly at some timestamps $t_1^* < t_2^* < \ldots < t_K^*$. The process of identifying these timestamps is called point detection. It results in a succession of nonoverlapping *segments*. We denote the segments from timestamp a to b as $y_{a..b}$.

We choose the change detection algorithm based on two considerations. First, the time complexity should be low. The system can response quickly after authors make some initial configurations. Second, the parameters and corresponding results should be easy to understand. The authors (data journalism or data enthusiasts) are familiar with the data they collect, but their familiarity with mining algorithms is not guaranteed. Thus, the tunable parameters (authors tune with interactions to meet their requirements) should be as few as possible, and authors can easily understand their effect on the results. Based on these criteria, we choose the window-sliding (denoted as Win) method,[16] which is a fast approximate approach for detecting change points. Its complexity is linear in the number of timestamps. The method computes the *discrepancy* between two adjacent windows that slide along the time-series y. The discrepancy between subsets of time-series in two windows is computed by

$$\begin{aligned} d(y_{a..t}, y_{t..b}) = c(y_{a..b}) - c(y_{a..t}) \\ - c(y_{t..b}) (1 \le a < t < b \le T) \end{aligned} \tag{1}$$

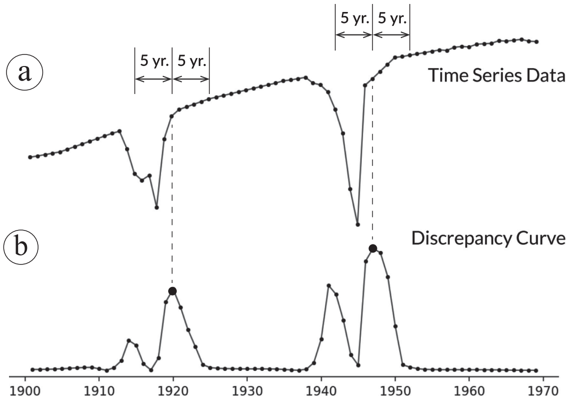

Figure 2. Illustration of the Win method. (a) Life expectancy in Germany between 1900 and 2000. Win computes the discrepancy and finds two peaks in (b) as two change points. The discrepancy is computed using Euclidean distance.

here $c(\cdot)$ is a cost function which measures goodness-of-fit of the subset of time-series to a specific model. We adopt $c(y_{a..b}) = \sum_{t=a}^{b} \|y_t - \bar{y}_{a..b}\|$, where $\bar{y}_{a..b}$ is the empirical mean of $y_{a..b}$. Here, we adopt Euclidean distance as the distance metric. If the two windows cover dissimilar segments, the discrepancy reaches large values. That means, for each timestamp t, Win measures the discrepancy between the immediate past (data in the left window) and the immediate future (data in the right window). After the complete discrepancy curve is computed, Win finds timestamps of change points through peak search. Higher peaks indicate more significant changes in the data. Figure 2 illustrates such a process in 1-D data, using the data of life expectancy in Germany between 1900 and 2000. The change points segment one item (the corresponding time-series) into different segments. We denote the detected results of item I as $C_I = \{c_{I1}, c_{I2}, \ldots, c_{IK+1}\}$. Note that, the algorithm has two parameters to specify. The first one is K, which denotes the number of points at which the time-series changes most drastically. The second one is the width of window (i.e., $b - a$), which specifies the preferred range during which changes are captured. That means, if the value of window width is small, the algorithm tends to emphasize changes happens in a short timespan, and vice versa.

As authors do not know the number of change points in the data *a priori*, our approach allows authors to choose the number of most significant change points (N) with simple interactions. We use a python library called *ruptures*[‡] to achieve this algorithm.

Identification of Important Changes

Given the change points and segments of different items, we define the following two types of important change segments.

- A: Segments with salient peaks. Peaks (or valleys) are included in the segments, which indicates drastic *increase/decrease* in the data.
- B: Segments that have significantly different slopes with adjacent segments. Such slope changes indicate changes in the *trend*.

Since the data is 2-D, both types of important changes may happen on one dimension or both dimensions. We take the union of results from each dimension.

For type A, we detect salient peaks for each item (each 2-D time-series) with three steps. First, we pick up all the peaks (the value at timestamp t is larger than values of $t - 1$ and $t + 1$). We keep the difference of peaks and their neighbors. Then, we rank the differences in ascending order. Finally, we apply change detection to find peaks with salient difference with neighbors. These peaks are regarded as salient peaks, and the corresponding segments are marked as type A.

For type B, we first fit each segment with linear regression and obtain the slope. The difference of slopes between adjacent segments are then computed. Given the difference of segments from all items in the dataset, we apply three sigma rule to find those segments with great changes in trend.

After these steps, we obtain the important segments of type A and B for each item.

Discovery of Co-Occurrence of Important Changes

The important segments of individual item is computed with the above methods. However, sometimes, many items may show similar patterns simultaneously. For example, World War I

[‡] http://ctruong.perso.math.cnrs.fr/ruptures-docs/build/html/index.html

leads to a huge drop in life expectancy, as demonstrated in Hans Rosling's video. We regard these co-occurrence of important changes across different items as another type of pattern.

To identify these occurrences, we use a simple count-based method. That is, given $C_I,\ I \in \{1, 2, \ldots, \#items\}$ of all items, we count how many times one timestamp appears in $C_I,\ I \in \{1, 2, \ldots, \#items\}$ for each timestamp ($t \in \{1, 2, \ldots, T\}$). We then select those timestamps with top-K highest frequencies, where K is the user specified number of most significant change points.

The points are further processed as follows. For each point, we extend it to adjacent timestamps: three previous timestamps and three following timestamps. We will merge the segments if several points of top-K are located quite closely. Users can modify the start and end of segments interactively in the prototype system described in the next section.

Comparison With Alternative Change Detection Methods

As discussed in the "Detection of Change Points and Segmentation of Time-Series" section, we have two criteria to choose the desired detection algorithm. As discussed in the work by Truong *et al.*,[16] there are two classes of change detection method: optimal detection and approximate ones. The time complexity of optimal ones are very high since they require exhaustive enumeration of segments. The approximate methods include window sliding, binary segmentation, and bottom-up segmentation. The binary segmentation method has no parameters to tune and uses global information to binary segment. Its accuracy is not guaranteed. The bottom-up segmentation is not robust to specific conditions and no theoretical convergence study is available. Taking all these factors into consideration, we select Win as the background algorithm.

VISUAL INTERFACE

The user interface consists of two major components, namely configuration panel and exploration and editing panel. The configuration panel provides a set of parameters for authors to choose or adjust. Authors can tune them based on own knowledge or preference. The exploration and editing panel can be subdivided into video preview (visualization), data timeline, and editing timeline. Widgets located on the right of interface enable authors to apply effects. By default, the system shows a raw video in video preview, with all the items selected and two random selected dimensions. Authors are allowed to change the selection of items and dimensions. They can explore the time-series data before configuration of algorithm-related parameters, and video length.

Configuration Panel

As depicted in Figure 3(a), authors can specify the X and Y dimensions and their corresponding scale (linear or log scale). By default, all the data items are selected. If items are selected, those items will be highlighted in the video preview. The highlighting is realized through diminishing other points' transparency. Authors can also specify the time range of data.

Besides the above data properties, two more algorithm-related parameters should be specified since they influence the generation of video [see Figure 3(b)]. They are (i) the number of segments which specifies the number of most significant change points, and (ii) window width specifies the range to detect, as discussed in the last paragraph of the "Detection of Change Points and Segmentation of Time-Series" section. By default, the former value is set as 5, and the latter one is 10^{odm-1}, where odm denotes *the order of magnitudes* of the number of timestamps. After that, the video length is also decided by authors [see Figure 3(c)]. The system will scale the length video sequence to the desired video length.

Exploration and Editing Panel

The exploration and editing panel allows authors to dive into data and create their own video sequences. *Video preview* [see Figure 3(d)] shows the animated scatterplot visualization. *Timeline exploration and editing* [see Figure 3(e1, e2)] consists of several timelines that depicts the data and video sequence from different perspectives. As discussed in the work by Riche *et al.*,[18] there are three notions of time in data-driven storytelling, namely, authoring-time, presentation-time, and data-time. Similarly, we

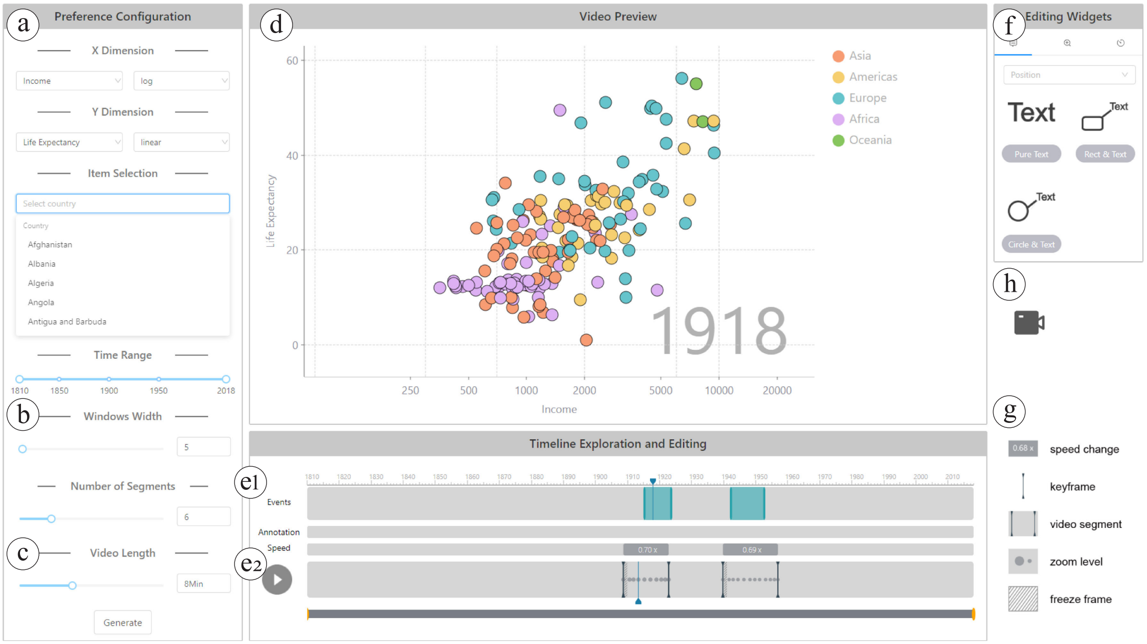

Figure 3. Data2video prototype system used to explore the Gapminder dataset.[1] The user interface mainly consists of (a)–(c) configurations for generating the data video, (d) video preview, (e1)–(e2) timeline exploration and editing, (f) widgets for enhancing segments, (g) legends, and (h) exporting button. Two important change segments are visualized on both data timeline and editing timeline. Authors can review the data timeline and edit on the editing timeline to better narrate changes in the time-series data.

distinguish between *editing timeline* [see Figure 3 (e2)] and *data timeline* [see Figure 3(e1)]. The data timeline mainly visualizes the important change segments or author specified interesting segments (strips colored with cyan) in the time-series. The editing timeline (includes annotation and speed tracks) is responsible for editing effects on video sequences. To keep these two timelines connected in the dimension of time, we synchronize the two blue colored current time indicators (CTIs) on both timelines. Even if the editing timeline's length changes due to speed change, these two CTIs always point to the same timestamp in the time-series data. By dragging and moving CTIs, authors can easily preview the video.

After the data processing (**T1**) described in the "Change Point Detection and Segmentation of Time-Series" section, the results are visualized in the timelines. Authors can review the algorithm results and explore the data (**T2**) using the data timeline. The data timeline has two different modes. If authors only view the time-series of several data items, *individual mode* will display an individual timeline of each item. If the authors care about the co-occurrence of changes across several countries, the data timeline will become a wider timeline (i.e., *aggregation mode*). Under both modes, the cyan strips indicates important segments detected by backend algorithms. If the authors discover interested facts that not detected automatically during their exploration process, they can manually add interesting segments by moving the CTI to a position and double clicking the handle to drag and move. A video segment that covers the same time range will show on the editing timeline for further editing.

Editing Widgets

After authors finish reviewing and exploration of data, they can further highlight those segments of interest on the editing timeline (**T3**). Authors can add, remove, or modify video segments. The prototype system provides three major types of effects, namely annotation, speed change, and zooming. Two tracks that placed above the editing timeline display the applied annotations and speed changes. Zooming level is visualized on the editing timeline if the zoom level is not equal to 1.

Annotation is an essential part of data-driven storytelling. Most existing annotation technique for data stories are designed for static visualizations. It may be time consuming for authors to add data-related annotations even with animated scatterplot. We distinguish between two types of annotation in data video: *fixed-position annotation* and *semantic annotation*. Geometric annotations are fixed in position across timestamps, which is quite common in video editing software. Semantic annotations are annotations that bound to data items. So this type of annotations may move with data items. Currently, our system supports both type of annotation, as shown in the Figure 3(f). For semantic annotations, authors need to specify which data item to be bound with.

Speed Change is a powerful way to convey emotions and help authors/audience better digest the information in the data. Dragging the frame tick on the editing timeline can speed up or slow down the speed. Freeze frame functionality (third editing widget) is also provided if authors want to stop at specific frame and elaborate the story at this timestamp.

Zooming also helps to view more detailed information in the video. Authors can zoom and pan with the second editing widget.

Rule-Based Generation of Effects

To reduce authors' time on editing, we propose simple empirical rules to automatically generate effects for enhancing important segments. These rules are extracted based on our observation of data videos and experience. We watched and investigated the animated scatterplot based data videos collected from Gapminder and news on public media such as China Central Television. We estimate the zoom scale and speed variation with video editing tools. To allow the authors better refine the effects based on their preference, these generated effects can also be modified (e.g., changing the range) or removed.

Zooming rule: If the data points during important or interesting segments are located in a small area (less than 45%) inside the video display, the system will add zooming in effect during these segments. However, it is quite confusing if the video zooms in and data points move simultaneously. Thus, we add still frames before zooming.

Slowing down rule: We expected that those detected or user-interested segments require more attention. To avoid abrupt fall of video speed, which may also confuse the authors/audience, the speed gradually slows down with interpolation. For items in individual mode, the system we will merge these segments based on the speed of respective segments. The merged results enable authors to further inspect the similarity and difference between individual items.

Design Rationale

The above design follows several design rationales. First, we let authors to choose their desired dimensions, items, number of segment, and video length in the configuration panel. Authors have various preference about the data subset and data patterns. With this design, they can customize what they intend to show and the facts to convey.

Second, the timelines and tracks visually summarize the data, the extracted changes, and the video sequence. Timelines are common in daily life and are intuitive to understand. To enable detailed exploration of data and avoid inconsistency between timelines due to speed change, we add CTIs to help coordinate the data and editing timelines. Thus, the timeline not only allows authors to review the automatically detected changes, but also allows them to manually explore the data in detail and find if there are other interesting facts in the time-series. The individual mode and aggregation mode also support customization of authors' interested data. Therefore, the approach provides the authors with flexibility to create data videos.

Last, we separate data, annotations, speed changes, and video sequence as different timelines (tracks). This design release mental burden of authors and avoid visual clutter.

EVALUATIONS

We describe two use cases with our prototype system. The dataset we used is from Gapminder, which includes socioeconomic status of 203 countries around the world from 1810 to 2018. After removing those countries with missing data, there are 184 countries left in the dataset. We encode the continent information

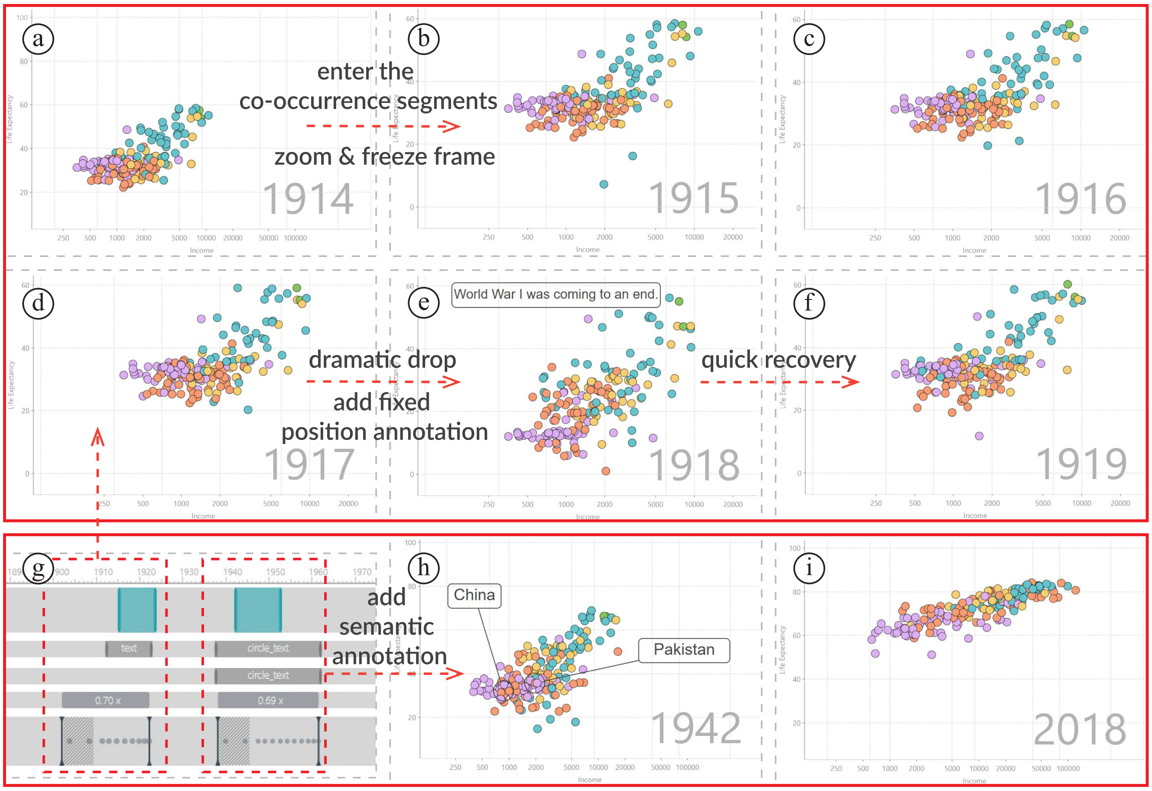

Figure 4. Key frames from the important co-occurrence segments of the first case. This case uses aggregation mode and shows changes that occur across items (i.e., changes occurred across those countries). The video freezes and zooms in while the timestamp enters the co-occurrence segments at 1914 (a) and (b). The segment is automatically detected and emphasized. The points of the scatterplot drop dramatically from 1917 to 1918 and then quickly recover at 1919 (c)–(f). Alice adds a fixed annotation to explain relevant historical events (e). Semantic annotations are added around 1942 (h) to show the evolution of countries. Timelines (g) are also shown.

with different colors. This encoding is widely used while visualizing this dataset and does not affect result of the backend algorithms.

Case 1: Demonstrate the Co-occurrence of Changes in Global Life Expectancy and Income Data

Alice wants to learn about some big events that happened across many countries in history. She chooses the dimensions of life expectancy and income. In the preference configuration panel, she sets *X* as income and *Y* as life expectancy. Other parameters are left as default (number of segments is six and video length is 1 min).

After clicking the generate button, she obtains two cyan strips [see Figure 4(g)] on the data timeline (aggregation mode) which indicate two important change segmentations (big events) are found. This amount is smaller than the default number of segments, which means segments are merged as described in the “Discovery of Co-occurrence of Important Changes” section. Correspondingly, two video segments with autogenerated effects are also shown on the editing timeline.

Alice first reviews the data timeline and finds that these two segments are around the period of World War I and World War II. She then clicks the display button of the editing timeline to view the video. At the beginning (1810), a few European countries slightly move upward due to the industrial revolution. The majority of countries move in a quite small range. After the twentieth century, the range of movement becomes relatively larger. When it comes to 1914 and 1915 [see Figure 4(a) and (b)], the video zooms in and freezes for 2 s. At this time, the video begins enhancing the first detected important change segment. On 1918 [see Figure 4(d) and (e)], most points drop dramatically in the dimension of life expectancy. This was due to World War I and the Spanish flu epidemic. The world recovered from the disaster after 1919 [see Figure 4(f)], when the points again move upwards. Alice adds fix-position annotations during this period.

The video zooms out and resumes normal speed after 1924. Note that we add three timestamps before and after important co-occurrence change points for easy exploration and better speed change effect. That is the reason for ending at 1924.

When the time comes to 1942, the Axis powers were defeated, and other countries started to develop. During this period, Asian and Latin American countries that were located at the left bottom corner of the scatterplot also began developing their economies. Alice adds semantic annotations for countries like China and Pakistan [see Figure 4(h)] to show their great effort and development. Both life expectancy and income increased. This video segment ends in 1953.

After that, most countries developed steadily until now [see Figure 4(i)]. No significant fluctuations occurred across a majority of countries.

This case shows our system's ability to automatically extract and enhance important changes across countries. With several operations, users can obtain an expressive data video.

Case 2: Inspect the Commonality and Difference Between Two Countries' Income and Birth Rate Data

In this case, Bob selects China and Japan for comparison. Other countries are diminished with higher transparency. The selected dimensions are children per woman and income. The time range is set to 1900–2018, the number of segments is 15, and the video length is of 3 min. Data timelines are generated for each country [see Figure 5(g)]. Bob finds that China has two important segments while Japan has three. The editing timeline shows one merged video segment with effects.

Bob moves the CTI of the data timeline from the beginning. From 1900, most countries have a high children per woman value. Western countries move very slowly toward the right bottom of the scatterplot (higher income, lower children per woman). When CTI points to 1933, two countries enter the first important segment [see Figure 5(a)–(d)]. Japan's income increased sharply while children per woman dropped dramatically to 2.1 in 1957 [see Figure 5(a) and (b)]. However, at this time in China, both income and children per woman drop to the lowest level [see Figure 5(b) and (c)] of this segment. The reason behind this is the Great Chinese Famine. After 1962, the Chinese economy recovered and birth rate also increased. Bob realizes that, although both countries' children per woman values droped rapidly, the causes can be quite different.

Bob keeps dragging the CTI to view how China and Japan changed after this important segment. He finds that China's babies per women value was gradually decreasing. He double clicks the handle of CTI to select 1971–1992 as an interesting segment. He switches to the editing timeline and slows down this segment's speed. During this period, China adopted the One Child Policy, and this resulted in the drop of babies per women value from 6 to 2 [see Figure 5(e) and (f)]. Income increased quite slowly. Japan's babies per women further drops with slow speed and income also increased slowly.

The second case demonstrates the system's ability to incorporate users' knowledge while exploring the data and further editing the data video. Besides automatically detected segments, authors can easily add segments they are interested in and apply customized visual effects.

User Feedback

The prototype was assessed by four experts from a data journalism company in China. They all have experience in creating data videos or other forms of data stories. After a brief introduction of the system, they used the system to make videos using Gapminder dataset and stock dataset. The stock dataset is collected by their company, which includes the up/down value *versus* net inflow of different industries in China. They provided us preliminary feedback about the tool.

Overall, they were satisfied with our approach. They all commented that the interface is neat and well organized. They appreciated the idea of automatically extracting the changes and applying visual effects first and then letting authors review and refine. One claimed, "The backend algorithm

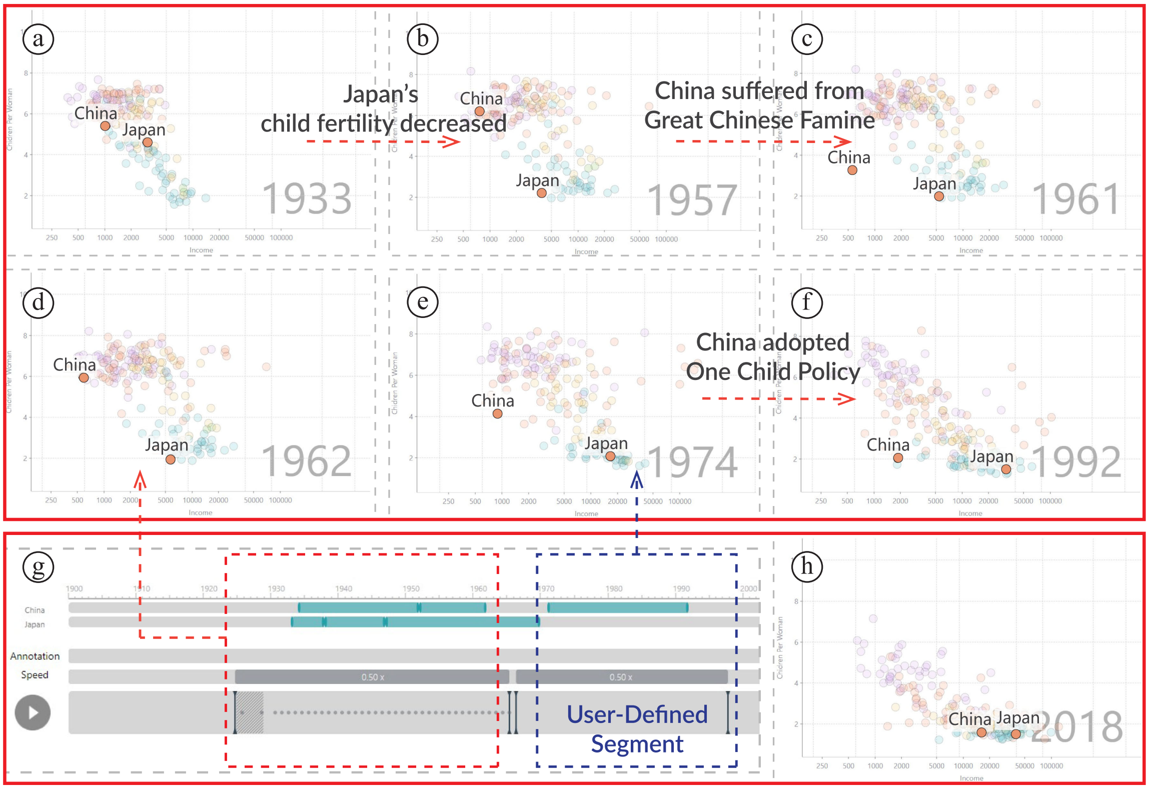

Figure 5. Key frames from the individual segments and user added segment of the second case. This case uses individual mode and shows changes of different items (i.e., countries). Here, Bob uses this mode as a way to compare between two countries along time. Different historical events, which are detected from the data automatically, influence the development of income and birth rate of these two countries (a)–(d). Bob adds a segment he is interested in (e) and (f) and slows down the speed. (g) shows the corresponding timelines and segments.

that computes the segments and applies visual effects can increase the efficiency of creating a data video. A combination of the automatic algorithm and manual effort enables this approach to be applied in different scenarios, like the stock data." Another expert liked the design of semantic annotation, which can reduce much effort of editing and helps highlight those important items (i.e., points in the scatterplot).

They also pointed out some limitations. Because the visual effects are generated based on empirical rules, the animation result can be unsatisfactory sometimes. One possible solution is to use advanced deep learning methods to summarize a better visual effect recommendation policy from data videos. When testing the system on stock data, one commented, "There are too many changes in the data if the window size is set as 2. Can you provide a more intelligent solution that ranks them or filter out some of them?". One also suggested that we provide the functionality of adding background music and voice-overs.

A more comprehensive user study is demanded to evaluate and improve our approach. We plan to interview a broader range of possible users (like data enthusiasts, not very familiar with visualization or video editing) and conduct more real cases for this purpose.

DISCUSSION

Lessons Learned: When reflecting on the design of our approach, we think that providing the flexibility for authors to explore data and authoring video is critical. Therefore, our method is not designed as a fully automatic one. We provide the functionality of exploring the data before configuration and reviewing, exploring, and refining after initial automatic generated results are generated. The widgets help authors make the video more expressive.

We also consider that the intuitiveness of visual encodings is quite important. Thus, we employ timeline visualization to show all the data, extracted facts, and the video sequence. Timeline displays chronological information and distribution of events, and is easy to understand.

Implication and Limitation of Using Automated Algorithm: The basic idea behind our design is to accelerate the video creation process by reducing the effort on data exploration and insight extraction. To release the burden of data exploration and insight extraction, we integrate a data mining algorithm in our system. With only two algorithm-related parameters to specify, the system will provide some initial results of important changes among the data. As we design algorithms to further compute important changes in the time-series data, our approach supports both global patterns (e.g., changes of many countries during World War I) and local patterns (e.g., changes of specific countries). With modification functionality of timelines and editing widgets, more customized animated scatterplot data videos can be created accurately.

However, our prototype implementation lacks the diversity of fact types of time-series data. For example, our system does not support abnormal subsequence detection, which can be widely applied in different domains. Despite manual effort can fix these problems to some extent, this increases authors' burden. One possible solution is to implement more algorithms to extract facts in the data, or allow plugins of these algorithms as an input. Since the data type is restricted to 2-D time-series data, it is feasible to add these algorithms as long as they share the same input and ouput formats.

Generalizability: Our approach can be applied to various 2-D time-series datasets where the attributes are numerical. For example, given a time-series dataset which contains the number of tweets and the number of retweets of hashtags that related to Brexit, our system can demonstrate interesting changes in the hashtag evolution. Combined with news or authors' background knowledge, authors can use this system to tell a good story about Brexit.

Our approach adopts intuitive design, and thus data journalists and data enthusiasts can use it to create data videos, as long as they have some knowledge of their data. Suppose that a data enthusiast librarian wants to make a video to illustrate the visitors and borrowers of a public library and show it on the large screen of the library. With our approach, he or she can make the video with less effort because the approach lighten the load of data analysis and video editing.

While our approach focuses on 2-D datasets and visualization of animated scatterplot, it can be adapted to 1-D time-series data and animated line charts. Given items with 1-D time-series, we visualize these items' evolution together in one line chart. The background algorithm can detect same types of important change segments without modification. Effects like zooming and changing speed are orthogonal to the visualization type so they can be directly applied on line charts. Annotations of fixed positions also applies. Since line charts can be viewed as a line connecting between points at different timestamps, semantic annotations can be bound to each item and move as the points move forward.

Scalability: We consider both the scalability of the algorithm and the visualization. For the scalability of the algorithm, for each item, the complexity of change point detection is linear in the number of timestamps. Similarly, the computation of important changes is also linear in the number of timestamps. The count-based method is linear in the results of previous step. So the complexity is $O(m * n)$, where m is the number of timestamps, and n is the number of countries.

Scatterplot supports visualization of *hundreds* of items discussed by Munzner.[19] Too many items lead to severe visual clutter. The clutter can become even worse if the points are animated. Although those video effects may reduce some clutter, too many points are still a burden to the authors and possible audience.

CONCLUSION AND FUTURE WORK

This study describes a novel and interactive approach that transforms 2-D time-series datasets into videos that can capture and highlight important changes in the data. To achieve this goal, we use several algorithms to detect and summarize changes in the dataset. We also implement a prototype system that shows great potential for producing expressive data videos.

In the future, we hope to apply this approach to other types of visualization, such as line charts. We also hope to enhance the video effect generation and recommendation functionality. In this case, we plan to use state-of-the-art machine learning methods to learn recommendation rules from a large set of data storytelling videos.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61772456, 61761136020, U1736109).

REFERENCES

1. "Gapminder dataset," 2019. [Online]. Available: https://www.gapminder.org/data/
2. W. Weber, M. Engebretsen, and H. Kennedy, "Data stories: Rethinking journalistic storytelling in the context of data journalism," *Studies Commun. Sci.*, vol. 2018, no. 1, pp. 191–206, 2018.
3. E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1139–1148, Nov./Dec. 2010.
4. C. D. Stolper, B. Lee, N. H. Riche, and J. Stasko, "Emerging and recurring data-driven storytelling techniques: Analysis of a curated collection of recent stories," Microsoft Research, Redmond, WA, USA, Tech. Rep. MSR-TR-2016-14, 2016.
5. F. Amini, N. H. Riche, B. Lee, C. Hurter, and P. Irani, "Understanding data videos: Looking at narrative visualization through the cinematography lens," in *Proc. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 1459–1468.
6. J. Hullman and N. Diakopoulos, "Visualization rhetoric: Framing effects in narrative visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2231–2240, Dec. 2011.
7. B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale, "More than telling a story: Transforming data into visually shared stories," *IEEE Comput. Graph. Appl.*, vol. 35, no. 5, pp. 84–90, Sep./Oct. 2015.
8. S. Sirisack and A. Grimvall, "Visual detection of change points and trends using animated bubble charts, " in *Environmental Monitoring*. London, U.K.: IntechOpen, 2011.
9. A. Rind *et al.*, "Visually exploring multivariate trends in patient cohorts using animated scatter plots," in *Proc. Int. Conf. Ergonom. Health Aspects Work Comput.*, 2011, pp. 139–148.
10. Y. Wang, Z. Chen, Q. Li, X. Ma, Q. Luo, and H. Qu, "Animated narrative visualization for video clickstream data," in *Proc. SIGGRAPH Asia Symp. Visualization*, 2016, pp. 11:1–11:8.
11. C. Sigovan, C. W. Muelder, and K.-L. Ma, "Visualizing large-scale parallel communication traces using a particle animation technique," in *Proc. Eurographics Conf. Visualization*, 2013, pp. 141–150.
12. L. Yu, A. Lu, W. Ribarsky, and W. Chen, "Automatic animation for time-varying data visualization," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2271–2280, 2010.
13. I. Liao, W.-H. Hsu, and K.-L. Ma, "Storytelling via navigation: A novel approach to animation for scientific visualization," in *Proc. Int. Symp. Smart Graph.*, 2014, pp. 1–14.
14. S. Bruckner and T. Möller, "Result-driven exploration of simulation parameter spaces for visual effects design," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1468–1476, Nov./Dec. 2010.
15. F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani, "Authoring data-driven videos with DataClips," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 501–510, Jan. 2017.
16. C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," 2018, *arXiv:1801.00718*.
17. J. Gray, L. Chambers, and L. Bounegru, *The Data Journalism Handbook: How Journalists Can Use Data to Improve the News*. Newton, MA, USA: O'Reilly Media, Inc., 2012.
18. N. H. Riche, C. Hurter, N. Diakopoulos, and S. Carpendale, *Data-Driven Storytelling*. Boca Raton, FL, USA: CRC Press, 2018.
19. T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014.

Junhua Lu is currently working toward the Ph.D. degree with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. His research interests include information visualization, narrative visualization, and visual analytics. He received the B.S. degree in information and computing science from Zhejiang University. Contact him at akiori@zju.edu.cn.

Jie Wang is currently working toward the master's degree with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. His research interests include information visualization and visual analytics. He received the B.S. degree in computer science from the Zhejiang University of Technology, Hangzhou, China. Contact him at mcfatewang@zju.edu.cn.

Hui Ye is currently working toward the master's degree with Visual Analysis Group, State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. Her research interests include information visualization and visual analytics. She received the B.S. degree in software engineering from the Central South University, Changsha, China. Contact her at yehui_leaf@zju.edu.cn.

Yuhui Gu is currently working toward the master's degree with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. His research interests include information visualization and visual analytics. He received the B.S. degree in computer science from Hangzhou Dianzi University, Hangzhou, China. Contact him at esora@zju.edu.cn.

Zhiyu Ding is currently Director of Visualization, Manager of WEB Competence Center with Cloud BU, Huawei Technologies Company, Ltd., Shenzhen, China. His research interests include visualization and visual computing. He received the Ph.D. degree in computer science from the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. Contact him at dingzhiyu@huawei.com.

Mingliang Xu is currently a Professor with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. His research interests include computer graphics, multimedia, and artificial intelligence. He received the Ph.D. degree in computer science and technology from the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China, and the B.S. and M.S. degrees from the Computer Science Department, Zhengzhou University, respectively. Contact him at iexumingliang@zzu.edu.cn.

Wei Chen is currently a Professor with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. His current research interests include visualization, visual analytics, and biomedical image computing. He received the Ph.D. degree in mathematics from Zhejiang University, Hangzhou, China. He is the corresponding author of this article. Contact him at chenvis@zju.edu.cn. For more information, please refer to http://www.cad.zju.edu.cn/home/chenwei/.