VideoForest: Interactive Visual Summarization of Video Streams Based on Danmu Data

Zhida Sun*[1], Mingfei Sun[1], Nan Cao[2], and Xiaojuan Ma[1]

[1]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
[2]College of Design and Innovation, TongJi University

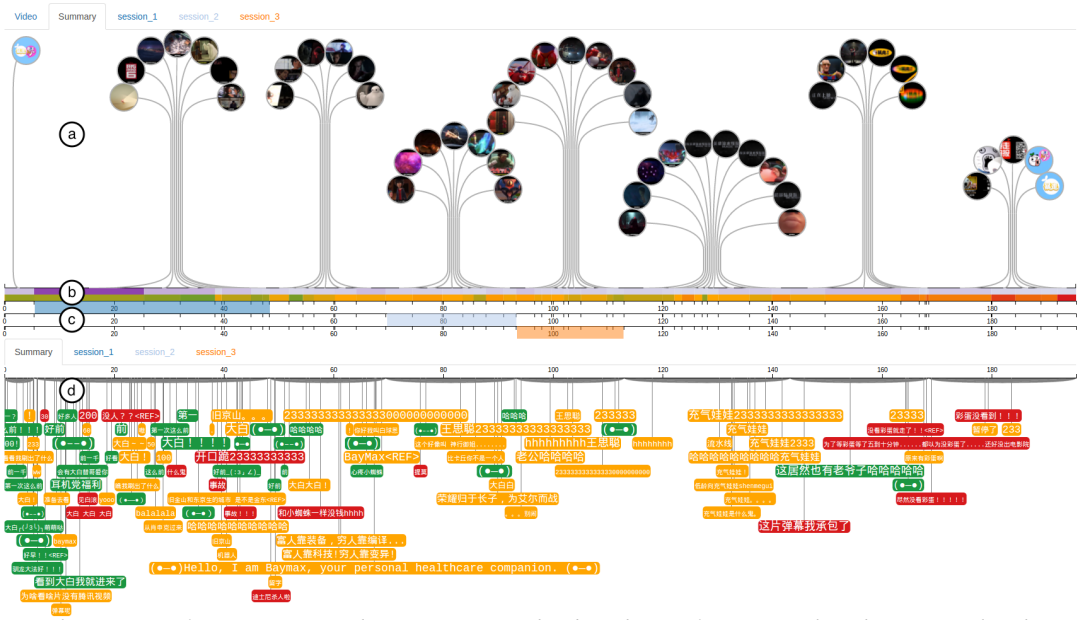


Figure 1: *A visual summary of a user-generated movie review video based on a forest-metaphor design. In this design, session-level key-frames of the video are clustered and visualized in scene trees (a) planted in a video timeline ground (b,c) with roots (d) showing the corresponding danmu posts. The video timeline ground consists of (b) a heatmap view that presents the contextual information extracted from danmu posts such as sentiments and volume and, (c) an interactive timeline view that enables users to specify the video sessions of interest.*

Abstract

Emerging online video-sharing websites such as YouTube allow users to access a huge number of videos and generate feedback via reviews and/or live comments (a.k.a. danmu), making it possible to summarize videos based on media and user responses collectively. A video summary produced by existing techniques may not fully capture an audience's perception of and reaction to the source video, and thus may be less reflective. In this paper, we introduce VideoForest, a visualization system designed to convert an input video augmented with danmu commentary data into a tree-like visual summary of content highlights under user supervision. The proposed visualization design employs a forest metaphor. The overall summary of different video sessions is illustrated as scene trees on top of the session timeline ground, with the roots depicting the corresponding danmu messages. VideoForest can also generate a detailed synopsis of user-selected video segment(s) as a compact storyline in the form of circle packing. We evaluate our system via case studies with real video data based on experts' feedback. The results suggest the power and potential of the system.

Keywords: Multimedia (Image/Video/Music) Visualization

Concepts: •**Computing methodologies** → **Image manipulation;** *Computational photography;*

1 Introduction

With the rapid growth of online video-sharing websites such as YouTube, users now have easy access to a large collection of videos generated by content providers or individuals, and they can also share their feedback of the clips. Besides leaving reviews below the video, some video sharing sites in China provide a feature called danmu, which allows viewers to post "live" comments asynchronously while watching a video. These messages get superimposed onto the video content, aligning with the video timeline regardless of their actual submission time to create a group viewing experience. The danmu data instantaneously reveals viewers' thoughts and feelings that usually coincide with a specific video scene. Content and service providers, as well as analysts can leverage such information to better summarize the content, characteristics, and potential values of the online videos.

*e-mail:zhida.sun@connect.ust.hk

SA '16 Symposium on Visualization, December 05-08, 2016, Macao
ISBN: 978-1-4503-4547-7/16/12
DOI: http://dx.doi.org/10.1145/3002151.3002159

Existing video summarization techniques have been developed mainly in the multimedia domain [Rajendra and Keshaveni 2014]. Many of these are automatic algorithms that aim to identify the key parts of a clip by analyzing its video, audio, and text (i.e., captions) features. These methods, although efficient, usually depend highly on the video's innate structure [Chu et al. 2015] and overlook viewers' perceptions and feelings. As a result, the storylines extracted from source videos may not be meaningful, interpretable, or exciting to audiences. Such techniques still rely heavily on professional judgment to manually identify any informative and evocative video scenes. In this paper, we introduce a semi-automatic approach to visually summarize a video stream based on danmu data.

However, synopsizing a video stream based on danmu data is non-trivial. First, extracting useful information from massive danmu data is most often difficult. Danmu data is usually noisy, consisting of a wide variety of information from viewers' own opinions; questions regarding things in the video; or to spam messages. The language used in danmu messages is rather informative, with a lot of Internet slangs and emoticons. Therefore, pure natural language processing techniques may not be able to effectively differentiate among the different types of danmu posts and extract the most relevant information for the synopsis. Second, combining and balancing heterogeneous information, i.e., visual data from a video and textual data from danmu comments, to guide the selection of essential video scenes and frames can be complicated. Third, specifically for the poster-style summary which is composed of static frames drawn from the video stream, organizing the images in a meaningful way to provide an overview at a glance is a challenging task. Most existing methods either align the images in the form of a grid [Lu and Grauman 2013] or pack them into a treemap [Tan et al. 2012], in which the contextual information showing users' interests is largely missing.

To address the above challenges, we introduce VideoForest, a visualization system designed to convert an input video stream augmented with danmu data into a visual summary of its content with highlights along the storyline under users' supervision. The generation of video synopsis follows a semi-automatic procedure. First, an algorithm produces a visual timeline view of an initial forest-style summary of the entire input clip by clustering related video sessions based on their graphical and danmu features. The corresponding danmu posts are illustrated as the root system of the scene trees. Users can further explore specific video segments through this timeline view, and locate parts of interest guided by contextual information derived from the danmu data such as the topic, volume, and sentiment of user-generated comments within a given time frame. Once a focal section of the video is selected, VideoForest creates a detailed video summarization based on a weighted, time-oriented circle image packing algorithm that captures both the storyline and climaxes at a finer granularity. In particular, this paper makes the following contributions:

- **Visualization Design and Algorithm.** We introduce two novel visualization designs: (1) a compound timeline view that consists of a scene forest on the top, tag roots at the bottom, and session timelines in the middle as the ground, providing rich contextual information to help users skim and navigate through a video stream to find interesting stories and scenes; and (2) a visual summarization of key video frames based on a novel circle packing algorithm following a storyline. Such a layout preserves the temporal relationship between frames and ensures efficient use of the screen space.
- **System.** To the best of our knowledge, VideoForest is the first visual analysis and authoring system designed for summarizing videos based on danmu data in an interactive manner.
- **Experiments.** We apply the proposed technique to a real world dataset and showcase several interesting summarizations created by expert users via our system.

2 Related Work

2.1 Visualizing Time-Oriented Data

Representing time-oriented data has long been one of the focal areas of visualization research. Among a wide variety of visualization techniques, this work is closer to those designed for showing multivariate time series data and streaming data given the characteristics of video and danmu commentary.

Visualizing multivariate time series data. Past research efforts in this area have mostly been devoted to visualizing, analyzing, and comparing data consisting of a set of numeric time-varying variables. Many visualization techniques have been developed to illustrate the changes in numerical multivariate data such as Silhouette Graph [Harris and Schreiner 1997], Horizon Graph [Heer et al. 2009], Layered area graphs [Byron and Wattenberg 2008]. MultiComb and TimeWheel [Tominski et al. 2004] applied radial arrangement instead of the conventional linear layout. Interactions have been widely used to help with time-oriented data exploration, such as multi-focal used in TrendDisplay [Brodbeck and Girardin 2003] and Chronolens [Zhao et al. 2011] and linked brushing used in LiveRAC [McLachlan et al. 2008] and TimeSearcher [Buono et al. 2007]. These techniques are helpful for revealing numerical features or measures extracted from a video stream, however, they fail to provide an interpretable summarization of a video due to the loss of connection to its content.

Visualizing streaming data. Another category of visualization techniques shows the evolution of high-level constructs in assorted streaming data, such as texts [Alsakran et al. 2011], events [Cao et al. 2016], and more generally, objects [Huron et al. 2013]. The rapid growth of variety and complexity of streaming data has attracted more attention and led to more systematic studies in this area. For example, Cottam et al. [Cottam et al. 2012] presented a comprehensive taxonomy of streaming data visualization. It revealed the relationship between changes in data and the interpretability of their visual presentations. Tanahashi et al. [Tanahashi et al. 2015] developed a framework for creating the visualization of a storyline extracted from streaming data. Although prior work has provided useful design principles and inspirations for our system, none has been developed specifically for visually summarizing a video stream and its corresponding danmu commentary.

2.2 Visualizing Video Streams

Video stream summarization has been extensively studied in the field of multimedia, with many techniques developed over decades [Rajendra and Keshaveni 2014]. These techniques all try to achieve one goal, i.e., producing a condensed version of the raw video so that users can quickly grasp its content, but from different perspectives. Existing research has proposed to exploit internal data, external data, or a hybrid of the two as the basis of video synopsis. Internal summarization techniques (e.g., [Chu et al. 2015]) use information extracted from the video itself, such as video/audio/linguistic features to identify important scenes. These methods depend highly on innate video structures and overlook viewers' perception and reaction. In contrast, external approaches rely on user-based information such as audience's preferences and interests [Kannan et al. 2013] to prioritize different content in a video stream. Manual labeling of video contents generated by authors of the videos or outsourced workers is required, which is expensive and less scalable. In this paper, we leverage danmu comments, a different source of user-generated data that is abundant and diverse, covering a larger scope of topics.

The video summaries are most commonly represented in a matrix form, in which selected images are packed line by line in a tem-

poral order [Lu and Grauman 2013]. Some advanced techniques can further illustrate the relationship among images extracted from the video. For example, Chen et al. [Chen et al. 2012] introduced a storyline visualization that uses arrows to suggest the flow of the narrative through the images arranged in a treemap [Shneiderman and Wattenberg 2001]. ImageHive [Tan et al. 2012] organized images inside a Voronoi diagram optimized to preserve the relationships among images and reduce visual overlaps. However, compared to our proposed technique, all these visualizations display images without showing much of the contexts such as references to the inner structure of the video and viewers' responses. In addition, none of them support interactions, and thus are incapable of guiding users to navigate in the video stream. Visualizations have also been introduced to support video navigation. For example, SceneSkim [Pavel et al. 2015] enables efficient caption-based video frame searching. VideoLens [Matejka et al. 2014] is designed to help with rapid exploration of large video collections. In comparison, VideoForest supports video exploration at a finer granularity.

3 Visualization Design

In this section, we introduce the detailed visualization designs of VideoForest, in particular the timeline view and session summary view that aims to depict the video content at different granularity. More specifically, the timeline view describes the temporal and semantic relationships between video content and danmu commentary on two levels. At the macro level, it provides an overview of all the video sessions and representatives of their associated danmu comments, while at the micro level it serves as a lens into user-selected video segments, obtaining further details from video frames and individual comments. The session summary view consequently packs these micro-level details into a glanceable form. In the rest of this section, we first discuss the design rationales behind the VideoForest system. Then, we present the detailed visual metaphors and encoding schema used in our forest-like design, including the scene tree, the session ground timeline, and the commentary root system. Lastly, we describe the design of corresponding interactions.

3.1 Design Rationales

The visual design of VideoForest aims to help users gain the gist and flow of a video clip and explore the evocative parts of it based on previous viewers' instantaneous feedback. We compile the following design principles to fulfill the above objectives.

R1. Narrative summarization of heterogeneous data. A visual summary of the video should preserve important contents based on viewers' interests. It should contain sufficient information from both the video itself and the audience to support further pattern and climax analysis. In other words, this requires the visualization to process danmu posts (texts) and video frames (images) collectively.

R2. Visual storyline unfolding. An effective video summary should reveal the main plot of the video so that users can grasp the storyline at a glance. This requires not only extracting proper representative frames from the video but also capturing the innate relationship among these frames, which can be a challenging task.

R3. Hot spots identification. The visualization should highlight viewers' points of interest and enable a quick climax detection. This requires the understanding of individual danmu post and leveraging the meta-information derived from aggregated danmu data.

R4. Easy exploration of video streams. The system should enable flexible video exploration guidance by the aforementioned contextual information derived from the danmu data.

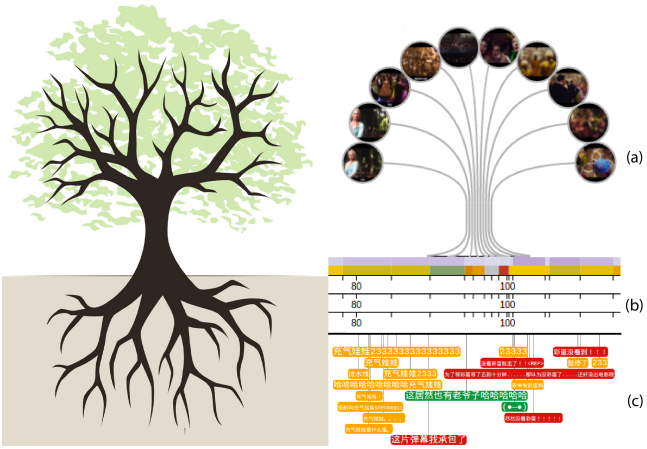


Figure 2: *Visualization design of a scene cluster in the primary summary based on a tree metaphor: (a) the scene tree, (b) video timeline ground, and (c) commentary root system.*

R5. Visual clutter reduction. Visual clutter should be reduced to maintain a neat and clear view of the summary. This ensures the legibility of the video storyline and improves the system's usability.

3.2 Visual Metaphor and Encoding

It has been shown that a well-defined visual metaphor is of great help in representing complex heterogeneous data [Cao et al. 2012]. A video usually contains multiple plot branches, and each scene may be developed around some major branch. To resonate with this notion, the design of the primary view in VideoForest follows a tree morphology which encodes different components of a video summary via different parts of the tree (**R1**) as shown in Figure 2. Here, the branches growing out of each tree are an analog of the successive video sessions that are semantically and emotionally related, since the clustering of session key frames takes both the image content and the contextual danmu sentiment into consideration (**R3**). The tree is planted in the ground to show the video timeline. Its roots reaching into the ground are an analog of the danmu commentary stimulated by the video content, and are visualized as temporal based tag clouds. The spread of the trees of varying heights is determined by the volume of danmu data (**R3**) which indicates the development of the story in the video (**R2**, Figure 1). When users select a video segment, they zoom into the tips of the corresponding session branches in the primary view, and a separate summary of the selected segment is displayed as a twig pruned from the tree (**R2**).

As shown in Figure 1, the whole visualization employs a consistent encoding scheme. Colors are used to encode sentiments, with green meaning positive and red meaning negative. The size of the danmu tags and image circles indicate their importance. Note that the importance of a danmu tag is measured by the frequency of its occurrence in a period of time, whereas the importance of an image is determined by the value of its information calculated over a set of features.

3.3 Interactions

The VideoForest system supports a set of rich interactions to facilitate an effective exploration of a video stream (**R4**). (1) View switching. A user can switch to a raw video view (Figure 3(a1)), a temporal summary view of the whole video stream (Figure 3(a2)), or a view showing summaries of selected session(s) (Figure 3(a3)) by clicking the corresponding tabs. (2) New session generation.

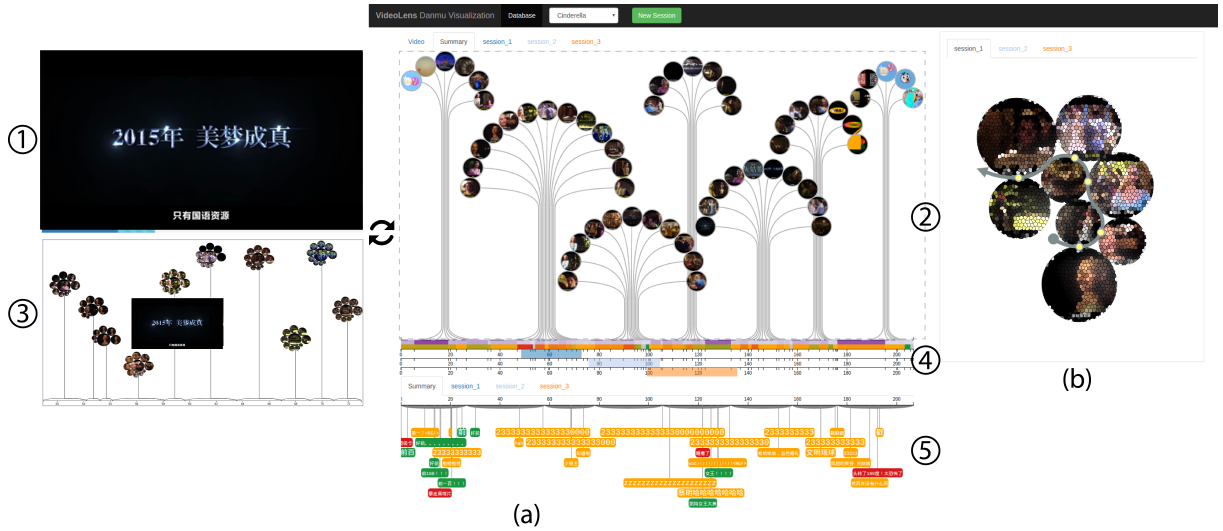


Figure 3: *The VideoForest system's user interface consists of (a) the primary timeline summary view and (b) the session summary view.*

Users can create a new session by clicking the "create session" button on the toolbar (Figure 3(a4)), and then brush on to select a video segment or multiple related segments in the new session. (3) Smart summarization. VideoForest enable users to select the danmu data of interest from the commentary root system to help refine the summary results shown in the session summary view. (4) Scene Preview. An animation related to a session shown in the scene tree or in the detailed session summary view will be shown in a tooltip when a user hovers over the representative image. (5) Highlighting. Users can also hover over a danmu tag to highlight it. When the mouse is hovering over a circular leaf on the scene tree, both the session image and its corresponding time window on the video timeline become highlighted.

4 System Overview

The VideoForest system aims to create an interactive video summarization based on the video content and the danmu commentary data. The system comprises three modules, i.e., preprocessing, analysis, and visualization . In particular, the *preprocessing module* extracts a sequence of key information frames (a.k.a. I-frames) and predicted motion frames (a.k.a. P-frames) from the video stream. It then computes the visual features of each image in the sequence, and re-assembles shots between two subsequent I-frames into animated previews. This module also performs preliminary text processing on the comments, including sentiment classification, similarity detection, topic assignment, and symbol categorization. The results are converted into textual features that characterize each post.

The *analysis module* takes the image and textual features as input and analyses the video contents and audience reactions on three levels: (1) meta-frame, i.e., each I-/P-frame and the danmu messages posted within the time interval around it (2) video coding session, i.e., the video segment and corresponding danmu data between two adjacent I-frames; and (3) scene cluster, i.e., successive sessions and related danmu data grouped by scenes.

The *visualization module* comprises two major video summarization views (Figure 3): (1) the primary timeline view and (2) the focal session summary view. The primary timeline view contains multiple components, including two multi-tab views coordinated via a video timeline view in the middle (Figure 3(a4)), with the top view hosting the raw video (Figure 3(a1)), a temporal summary of the whole video stream (Figure 3(a2)), and summaries of selected session(s) (Figure 3(a3)) on each of its tab, and the bottom view displaying a synoptic danmu cloud and collections of danmu posts within the selected session(s) on separate tabs (Figure 3(a5)). The whole design of the primary timeline view follows a forest metaphor which is introduced in Section 5. The session summary view illustrates a storyline derived from the selected session(s) in a circle packing-based layout.

5 Data Preprossessing and Analysis

We obtained the danmu data and video clips from the *bilibili*[1], a major Chinese danmu sharing website. In this section, we describe the functions in the data preprocessing and analysis modules.

Feature Extraction. We extract features from both video streams and danmu posts for conducting a clustering based video summarization analysis. In particular, we extract visual features from the key frames obtained from the raw video stream. First, we leverage the innate structures of an MPEG video and extract two types of frames, i.e. I-Frames and P-Frames, as keyframes based on *FFmpeg*[2]. In particular, I-Frames are fully-specified, uncompressed images in a video stream which usually indicate the switching of shots in the video. P-frames, in comparison, preserve only the data that are different from the preceding I-frame and capture the major changes in a shot. We extract features from each I/P-frames based on Histogram of Oriented Gradients (HOG) [Dalal and Triggs 2005] which was originally introduced for detecting objects in images and used for capturing the changing objects in the case of video processing. For each frame, its HOG vector is formed by concatenating the components of normalized cell histograms over all block regions. The Principal Component Analysis (PCA) is also used to balance the dimensions between HOG features and post features. Employing the HOG feature enables us to filter out small local differences and place emphasis on detecting major scene changes.

The raw danmu data are crawled and stored in an XML file which contains eight attributes, i.e., each comment's physical and video time-stamp, position on the screen, font size and color, content, visual effects, and poster ID. For each danmu post, we extract the linguistic features categorized as follows: (1) Similarity, i.e., the number of danmu comments above a certain cosine similarity (0.4 and 0.8 respectively) within a given post. (2) Sentiment. (3) The

[1]http://www.bilibili.com/
[2]http://ffmpeg.org

topic of the danmu text, including actor, background music, review, character voice, danmu, graphics, producer, scene, title, movie, and socializing, etc. (4) Communication function of danmu text, including emphasis, thought, observation, feeling, and need. (5) Use of symbols, including word, punctuation, number, letter, Emoji, and pointer. (6) Language pattern, including normal, repeat, rephrase, and reference (replying to a previous danmu). This process yields a total of 32 textual features for each danmu message. To find a balance, we apply principle component analysis to reduce dimensions of HOG features so that the latter analysis can be performed fairly.

Meta-frame Construction and Analysis. With the two types of features above, we conduct further analysis at different levels. The lowest level is called meta-frame, i.e., assigning the danmu messages that occur within the time interval between its preceding and succeeding frames, if any, to each I/P-frame. A meta-frame inherits the visual descriptor of its image component, and aggregates over the textual features of all the associated danmu comments to derive a set of collective textual attributes. Together these form the feature vector of the meta-frame. The visual attributes of each meta-frame describe its content, while the textual part reflects the context derived from the audiences' reactions.

Session Analysis and Preview Assembly. We define the video segment between two adjacent I-frames as a video coding session. In other words, a session consists of an I-frame that indicates the start of a new shot followed by a sequence of P-frames that signal the motion and any changes to the elements in the shot [Richardson 2011]. For each session, we compute the total volume of damnu and the aggregated sentiment within its time window and generate its animated GIF preview using FFmpeg command. We appoint the I-frame (or the first succeeding P-frame if the I-frame is too dark) of every session as its representative frame. Similar to meta-frame, we compose a hybrid feature vector for each session. It consists of the color palette of the entire session generated by the FFmpeg command as well as the synoptic linguistic attributes drawn out of all related danmu posts in its time range. We also add the session start time to the session descriptor.

Scene Cluster Construction and Analysis. From the video compression point of view, a session roughly corresponds to a shot in the video stream. It often takes multiple successive shots, likely from different angles, locations, or views, to make a scene. Therefore, we group shots into scene clusters based on their temporal adjacency, the similarity in color tone, and congruence of audience's feedback in terms of both content and emotion. For each cluster, we again calculate the density of danmu commentary appearing in its time window as an indicator of the evocativeness of the scene.

6 VideoForest Interface

In this section, we describe the technical details for implementing the design of the aforementioned two views: the primary timeline view and the session summary view (i.e., story twig).

6.1 Primary Timeline View

This view consists of three parts, i.e., *scene forest*, *video timeline ground*, and *commentary root system*. To produce the *scene forest* and the *commentary root system*, both the session-level frames extracted from the video and the danmu posts are first clustered into groups More specifically, the frames are clustered based on their temporal adjacency, graphical features, and textual features extracted from danmu data (Section 4.4). Each session cluster is visualized as a scene tree growing on top of the *video timeline ground*. The danmu comments are clustered based on their content similarity. Posts with the same messages are merged with the font size denoting the frequency of occurrence.

We produce a scene tree for each session cluster based on a series of calculations. The representative images of sessions in the cluster are displayed as circular leaves and packed in a radial layout to form an umbrella-shaped canopy anchored at the top of the scene tree. Each tree is planted into the timeline ground at the middle time point of its associated video session sequence, with its height directly proportional to the number of danmu messages posted within its corresponding time window. The heights of the trees are further adjusted to remove any overlaps based on the force-directed algorithm.

In *commentary root system*, the aggregated danmu posts are laid out as a temporal tag cloud in a direction opposite to the scene forest by following a similar procedure. Each tag is placed at the video time point as appointed by its poster, and the height is adjusted to avoid overlap. Note that the bounding box of a danmu tag is a rectangle instead of a half circle as that of the canopy of a *scene tree*.

6.2 Session Summary View

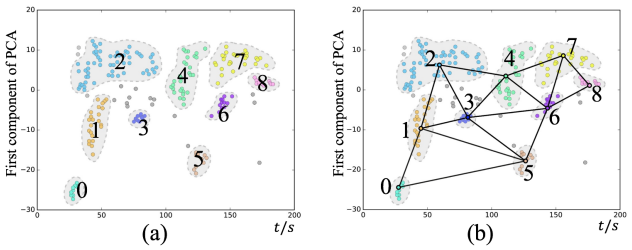


Figure 4: *(a) Mega-frame aggregation with each cluster labeled sequentially; the outliers are excluded. (b) The construction of triangular complex. In this complex, nodes* 3 *and* 6 *have 5 degrees, which are higher than the others. Hence clusters 3 and 6 will be the anchors and the others are the non-anchors.*

The mega-frame feature vectors constructed in the analysis module are projected onto $\mathbb{R}$ by PCA with their timestamps being mapped onto the x-axis (Figure 4). In other words, mega-frames are transformed into two-dimensional data points called chron-units in $\mathbb{R}^2$(the first dimension of each data point is the first component of its PCA while the second dimension is its timestamp). This process stresses the visual and textual relationships among the meta-frames without tampering with their chronological arrangement. We cluster the normalized chron-units with each resulting cluster indexed by its chronological order in the sequence and weighted by its size (Figure 4(a)). Then, we choose the image of the meta-frame closest to the center of a cluster as its representative to offer a glimpse of the shot in the associated time range.

Further, we want to compose a meaningful timeline that threads all the representative shots to create an overview of the video plot in the specified time windows. Rather than simply placing these shots in $\mathbb{R}^2$, we propose a two-phase holistic layout method that is reminiscent of circle packing [Collins and Stephenson 2003] to display the video summarization results of user-selected video segments.

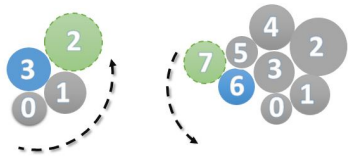
Circle image packing. We represent each of the output clusters from the previous step as a circle, with the radius logarithmically proportional to the cluster weight to avoid exceedingly large radii. The circles are indexed according to the temporal order of the corresponding clusters (Figure 4(a)). In our circle packing algorithm, we divide all the circles into two categories, i.e., anchors and non-anchors whose positions are determined by the former. We adopt the Delaunay triangulation to construct a complex with each node denoting a cluster(Figure 4(b)). Namely, each circle in our final packing corresponds to a node in the complex. Different from conventional circle packing algorithms (e.g., [Collins and Stephenson

Algorithm 1: Storyline

Data: Graph (V, E) with node weight w_i, start vertex v_s, end vertex v_e
Result: The path from v_s to v_e: $[v_s, v_1, \ldots, v_e]$
begin
  # Initialize
  for $v_i \in V$ **do**
    $dist[v_i] \leftarrow Infinity$; $direc[v_i] \leftarrow 0$; $pre[v_i] \leftarrow v_i$;
  # Calculate distance for each vertex
  $changed \leftarrow True$
  while $changed$ **do**
    $changed \leftarrow Flase$;
    for $v_i \in V$ **do**
      for $v_j \in V$ *and* $(v_i, v_j) \in E$ **do**
        if $v_i, v_j, pre[v_i]$ *are from same circle* **then**
          $sameCircle \leftarrow -1$;
        else
          $sameCircle \leftarrow +1$;
        if $v_i \rightarrow v_j$ *is clockwise* **then**
          $currDirec \leftarrow +1$;
        else
          $currDirec \leftarrow -1$;
        if $dist[v_j] > dist[v_i] + w_i$ *and* $sameCircle * currDirec * direc[v_i] \leq 0$ **then**
          $dist[v_j] \leftarrow dist[v_i] + w_i$; $changed \leftarrow True$;
          $direc[v_j] \leftarrow currDirec$; $pre[v_j] \leftarrow v_i$;
  # Reverse traverse from v_e to find the path
  $path \leftarrow \emptyset$; $currV \leftarrow v_e$;
  while $currV \neq v_s$ **do**
    Insert $currV$ into the front of $path$;
    $currV \leftarrow pre[currV]$;
  Insert v_s into the front of $path$;
  Return $path$;

2003]), which directly convert the complex edges to the pattern of tangencies, we use the complex to determine the anchors in our circle image packing. In this way, we maintain the chronological relationship and gain the flexibility to determine individual circle size. On top of that, we minimize any potential waste of space.

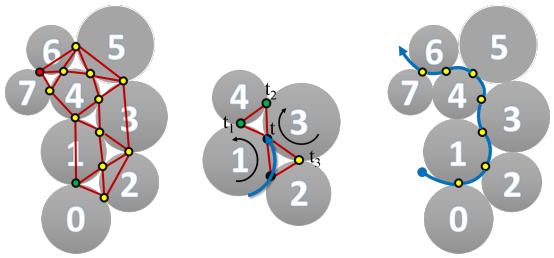
In particular, we choose circles of which the corresponding nodes have a higher degree to become the anchors. In a complex, the internal nodes tend to have higher degrees than the external ones. Therefore, in our circle image packing, we place the non-anchor circles around the anchors. We position the first anchor at the origin. Then, all the circles before the next anchor are sequentially positioned in the following manner (see Figure 5(a)). They surround the current anchor clockwise, only intersecting with it at a common external tangent. The pattern of tangencies also applies to consecutive non-anchors. Once a new anchor is activated, it will become the center around which the successive non-anchors go in the same manner as mentioned before (see Figure 5(b)). We further remove the overlaps (**R5**) among circles to produce the final result.



(a) (b)

Figure 5: *(a) Circles are placed by the anchor, circle* 3*; (b) a new anchor, circle* 6*, is activated.*

Storyline threading. We draw a smooth and spatially compact narrative thread to connect circle-circle intersecting points in a chronological order. This forms a video summary within the user-selected time windows $\mathbb{R}^2$(Figure 5). Users can follow the storyline to view the representative shots in associated circles and browse the related danmu cloud in the pop-up tooltips at the threaded tangent points.



(a) (b) (c)

Figure 6: *(a) The corresponding node-link graph (green node: v_s, red node: v_e); (b) circular motion principle. The blue path just crosses node t anticlockwise. The next feasible nodes would be t_1, t_2 and t_3. However, the motion of direction $t \rightarrow t_3$ compromises the path's smoothness. Hence, only t_1 and t_2 would be the choices. More specifically, t_i is feasible if and only if t_i and t are on the same circle and $t \rightarrow t_i$ agrees with the previous circular motion or t_i and t are not on the same circle and $t \rightarrow t_i$ disagrees with the previous circular motion; and (c) the circle image packing with a storyline.*

To derive such a storyline, we construct a weighted node-link graph to illustrate the tangent relationships among circles. More specifically, all the common external tangent points are mapped to the graph nodes. We refer to the common tangent point that connects to the first and second circles in the sequence as the start point (v_s), and the one between the two last circles as the end point (v_e). We assign a weight w_i to each node v_i, which is determined by the time differences between two clusters represented by the circles intersecting at the corresponding common tangent point. For each pair of adjacent tangent points on the same circle, we insert an edge between their corresponding nodes (see Figure 6). The distance between a node v_i to the origin v_s is defined as: $dist(v_i) = min(\sum_{k \neq i} w_k)$, i.e., the sum of weight for all v_k except v_i in the path $(v_s \rightarrow \ldots \rightarrow v_k \rightarrow \ldots \rightarrow v_i)$.

We then convert the storyline construction on the circle image packing to a minimum-weight path finding the problem on the node-link graph. This can be efficiently solved by dynamic programming (as describe in Algorithm 1). We also introduce the circular motion principle for each node (see Figure 6(c) for more detail) to preserve the smoothness of the path. The circular motion for each node is flagged as follows: $+1$ for clockwise, -1 for counterclockwise, and 0 for neither. We also define the preceding node for each node v_i as the one which precedes it in the path $(v_s \rightarrow \ldots \rightarrow v_i)$.

7 Evaluation

Since VideoForest is the first of its kind to leverage danmu commentary for video summarization, there is a lack of baseline systems for a comparative study. Therefore, we evaluate our system via a case study and an informal focus group session with thirteen experts. This section describes the evaluation methods and results which showcase the power of our system.

7.1 Case Study

This case study focuses on understanding the video content and style. Here, we choose a Chinese movie "Silent Separation" as an example, given that people often compare it with another most viewed but disliked movie "Tiny Times" which we have watched before. Both movies are poorly rated by the viewers, and thus we are interested to know the similarity between them. We imported an audience's cut version of the "Silent Separation" movie into the VideoForest system. The video is 205 seconds long and comes

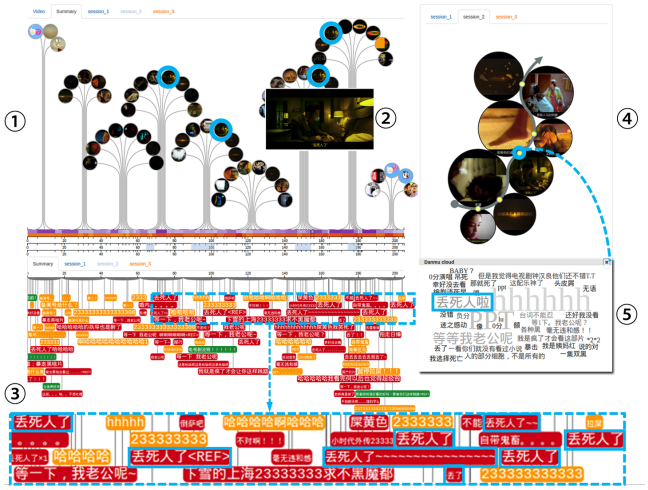


Figure 7: *Summary of a review video about the movie "Silent Separation".*

with 2052 danmu comments. We obtained 61 I-frames and 1804 P-frames from the video stream, and grouped the 61 sessions into nine scenes.

We immediately noticed that the director likes to create close-up shots by skimming through the scene forest Figure. 7(1). We recognized several familiar faces and realized that the movie employed almost the same cast as "Tiny Times". In addition, we found that many of the shots were in sepia tones such as those in old photos, which is consistent with the critic that the movie has a strong preference for earthy-yellow filters. From the danmu cloud (Figure. 7(3)), we noticed that a comment "What a disgrace" appears repeatedly throughout the video. It seems that many viewers posted this message. We could guess that people were expressing their dislike about the movie, but puzzled by their repetitive use of the phrase. When turned back to the scene trees and went over each branch more closely, we realized that one key frame also occurred multiple times in summary. After looking at the preview shown on the tool tip (Figure. 7(2)), we found that it was a shot from the movie in which the leading actress said "What a disgrace!". This finding connected the different scenes into a coherent story and the commentary started to make sense. We selected the video segments before all the instances of the "disgrace" shot. The resulting session summary (Figure. 7(4, 5)) basically revealed the major criticism of "Silent Separation": PowerPoint(PPT)-style narrative, nonsense conversations, and a boring cast, according to user reviews on *douban*[3], a dedicated movie review website in China.

7.2 User Observation

To evaluate the design of our system and its potential usage, we organized a demonstration with thirteen expert guests from related areas. Four of them were from one of the leading animation companies in China, including a chief executive, a project manager, a marketing manager, and a technical director. The other participants were university faculty and students of different backgrounds: machine learning and data mining (one faculty and three graduate students), data visualization (one faculty, experienced in video visualization), computer vision (one faculty, specializing in video processing), and new media art (two graduate students). We started with an introduction of the basic concepts related to danmu and an overview of the VideoForest system. Then we showcased the different views, interactions and functionalities of our system using a user-generated review video of the animation movie, Big Hero 6 (2014). After the demonstration, we conducted a semi-structured interview with the participants followed by an open discussion.

We asked the participants about their perceptions of the VideoForest system during the interview. Overall, they found the VideoForest system easy to follow and the logic across different views very clear. They also liked the visual metaphor design, as one of the arts student said, "It is novel and fun." The data mining participants showed a specific interest in generating scene bundles based on heterogeneous data, i.e., combining pictorial and textual information. Participants from the animation company found the system particularly useful for several reasons. First, the variation in the size and height of the scene trees allow them to capture the pace and climax of the video. "We can easily identify the most eye-catching and evocative parts of the video," according to the project manager of the animation company. Second, VideoForest can help with sentiment analysis, discovering characters or plots that viewers liked or disliked based on their reactions. Third, from a producer's perspective, it is critical to study in detail the audiences' collective opinions and feelings triggered by specific content of interests. They want to know "if the audience actually recognize and understand the theme, clues, punchlines and twists deliberately embedded in the video."

In the open discussion, we invited the participants to envision the potential applications of our system. The visualization, arts and design participants proposed that VideoForest could be used for bookmarking and navigation within a video. As one of the university students commented, "It would be easier to locate and refer to a specific scene during conversations with friends, family, colleagues, etc." The animation company suggested that the VideoForest system would be beneficial for their marketing campaigns. They can better decide on the design of consumer products related to their animations and guests to their offline promotion events. Analysis via VideoForest can also guide in the customization of trailers and follow-up advertisements according to the consumers. The marketing manager added, "We can even tailor the next episode or next production to the taste of our target audiences."

8 Conclusion and Future Work

In this paper, we have introduced VideoForest, the first system designed to interactively convert an input video stream augmented with danmu commentary data into a forest-like visual summary of its content and compile highlights along a storyline under users' supervision. The proposed visualization design employs a forest metaphor, in which an overall summary of different sessions in a video is illustrated as scene trees growing out of a video timeline ground, with the roots depicting the corresponding danmu messages. VideoForest can also generate a detailed synopsis of the user-selected video segment(s) and present it as a compact storyline in the form of circle packing. We evaluate the proposed system via case studies with real video data according to thirteen domain experts' feedback. The results demonstrate the power and potential of our system. Note that scalability to accommodate longer video streams is the most commonly mentioned issue among the experts. We believe it can be addressed using hierarchical aggregation of the video streams, as part of our future work. In addition, we will employ more advanced video analysis techniques to refine the summary quality and deploy our system on video sharing websites to serve ordinary users.

Acknowledgements

The work is supported by National Natural Science Foundation of China (No. 61602306).

[3] https://movie.douban.com/

References

ALSAKRAN, J., CHEN, Y., ZHAO, Y., YANG, J., AND LUO, D. 2011. Streamit: Dynamic visualization and interactive exploration of text streams. In *2011 IEEE Pacific Visualization Symposium*, 131–138.

BRODBECK, D., AND GIRARDIN, L. 2003. Interactive poster: Trend analysis in large timeseries of high-throughput screening data using a distortion-oriented lens with semantic zooming. In *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis03)*, Citeseer, 74–75.

BUONO, P., PLAISANT, C., SIMEONE, A., ARIS, A., SHMUELI, G., AND JANK, W. 2007. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Information Visualization, 2007. IV '07. 11th International Conference*, 191–196.

BYRON, L., AND WATTENBERG, M. 2008. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov.), 1245–1252.

CAO, N., LIN, Y. R., SUN, X., LAZER, D., LIU, S., AND QU, H. 2012. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec), 2649–2658.

CAO, N., LIN, Y. R., DU, F., AND WANG, D. 2016. Episogram: Visual summarization of egocentric social interactions. *IEEE Computer Graphics and Applications 36*, 5 (Sept), 72–81.

CHEN, T., LU, A., AND HU, S.-M. 2012. Visual storylines: Semantic visualization of movie sequence. *Computers & Graphics 36*, 4, 241 – 249. Applications of Geometry Processing.

CHU, W. S., SONG, Y., AND JAIMES, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3584–3592.

COLLINS, C. R., AND STEPHENSON, K. 2003. A circle packing algorithm. *Computational Geometry 25*, 3, 233 – 256.

COTTAM, J. A., LUMSDAINE, A., AND WEAVER, C. 2012. Watch this: A taxonomy for dynamic data visualization. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, 193–202.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, IEEE Computer Society, Washington, DC, USA, CVPR '05, 886–893.

HARRIS, R. L., AND SCHREINER, D. E. 1997. Information graphics: A comprehensive illustrated reference. *Technical Communication 44*, 2, 174.

HEER, J., KONG, N., AND AGRAWALA, M. 2009. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '09, 1303–1312.

HURON, S., VUILLEMOT, R., AND FEKETE, J. D. 2013. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec), 2446–2455.

KANNAN, R., GHINEA, G., SWAMINATHAN, S., AND KANNAIYAN, S. 2013. Improving video summarization based on user preferences. In *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on*, 1–4.

LU, Z., AND GRAUMAN, K. 2013. Story-driven summarization for egocentric video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2714–2721.

MATEJKA, J., GROSSMAN, T., AND FITZMAURICE, G. 2014. Video lens: Rapid playback and exploration of large video collections and associated metadata. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, NY, USA, UIST '14, 541–550.

MCLACHLAN, P., MUNZNER, T., KOUTSOFIOS, E., AND NORTH, S. 2008. Liverac: Interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '08, 1483–1492.

PAVEL, A., GOLDMAN, D. B., HARTMANN, B., AND AGRAWALA, M. 2015. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ACM, New York, NY, USA, UIST '15, 181–190.

RAJENDRA, S. P., AND KESHAVENI, N. 2014. A survey of automatic video summarization techniques. *International Journal of Electronics, Electrical and Computational System 2*, 1.

RICHARDSON, I. E. 2011. *The H. 264 advanced video compression standard*. John Wiley & Sons.

SHNEIDERMAN, B., AND WATTENBERG, M. 2001. Ordered treemap layouts. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, IEEE Computer Society, Washington, DC, USA, INFOVIS '01, 73–.

TAN, L., SONG, Y., LIU, S., AND XIE, L. 2012. Imagehive: Interactive content-aware image summarization. *IEEE Comput. Graph. Appl. 32*, 1 (Jan.), 46–55.

TANAHASHI, Y., HSUEH, C. H., AND MA, K. L. 2015. An efficient framework for generating storyline visualizations from streaming data. *IEEE Transactions on Visualization and Computer Graphics 21*, 6 (June), 730–742.

TOMINSKI, C., ABELLO, J., AND SCHUMANN, H. 2004. Axes-based visualizations with radial layouts. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, ACM, New York, NY, USA, SAC '04, 1242–1247.

WIKIPEDIA, 2016. k-means clustering — Wikipedia, the free encyclopedia. [Online; accessed 28-March-2016].

WIKIPEDIA, 2016. Mean shift — Wikipedia, the free encyclopedia. [Online; accessed 28-March-2016].

ZHAO, J., CHEVALIER, F., PIETRIGA, E., AND BALAKRISHNAN, R. 2011. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec.), 2422–2431.