

# Timages: Enhancing Time Graphs with Iconographic Information

Stefan Jänicke

Image and Signal Processing Group, Institute for Computer Science, Leipzig University, Leipzig, Germany  
E-mail: stjienicke@vizcovery.de

**Abstract**—Various time-based visualization techniques have been designed to support the temporal analysis of data collections. While quantities play a secondary role in traditional timelines that reserve space for each individual data item to be observed, time graphs rather display quantitative information and they provide interaction means to filter for a subset of the data. Timages is a hybrid approach that enhances quantitative time graphs with qualitative information in an infographic-style. By (1) scaling thumbnails of data items dependent on relevance to the observed topic and by (2) time-dependent positioning these thumbnails inside a temporally aligned area with a novel space-filling strategy, the most relevant items in the entire data collection as well as predominant data items of certain time ranges are instantly seizable without the need to interact with the time graph.

**Index Terms**—timeline visualization, space-filling technique, digital humanities

## I. INTRODUCTION

Timelines display lists of events in chronological order [26], thus, they attach greater importance to the individual event or data item than to the aggregate. However, for large data collections the amount of information to be displayed often exceeds the available screen space, so that the user is forced to scroll for exploration purposes. Also, quantitative information is often only rudimentarily visualized. An example is given by Zhao et al. [32], who combine a timeline with a histogram for quantitative analyses.

The means of choice to support the visual analysis of large timestamped data collections are time graphs [4] as they intuitively illustrate the temporal distribution of the data while giving information about the highs and lows of the observed topic. Typically, the time graph is part of a system that implements Shneiderman’s Visual Information-Seeking Mantra “Overview, Zoom and Filter, Details on demand” [28], i.e., users are enabled to drill-down for a specific time span in order to retrieve associated data items. Few approaches directly visualize more detailed information on the topical construction of a time graph. For example, Shi et al. [27] fill the streams of stacked time graphs with tags indicating predominant tags for certain time periods. Still, interaction means are necessary to observe individual data items as tags represent multiple ones.

The idea of Timages is to combine both concepts of charting temporal data. While providing quantitative information in the

form of a time graph, information on individual data items is attached, thus, visible at first glance without the need to drill down or interact with the time graph. The proposed design is inspired by Adams’ *Synchronological Chart* [3]; a snippet is shown in Figure 1. Adams juxtaposed biblical and world historical events from 4000BC to the 19th century, and he iconographically exposed important events or influential individuals. The size of an image seems to represent the importance of an event, e.g., significant events such as the crucifixion of Christ reserve a large amount of space. In Adams’ spirit, Timages automatically fills a given polygonal area with thumbnails representing individual data items while considering the relevance of a data item to the observed topic—the relevance value is used for thumbnail scaling. This paper outlines a space filling strategy for arranging thumbnails of the most relevant data items temporally in a given area and it outlines the value of Timages to generate visually appealing infographic-like time graphs on the basis of large data collections.

## II. RELATED WORK

Temporal information is inherent in many data collections, so that numerous visualizations have been developed in order to support the analysis of trends. Different static time charting methods to illustrate quantitative and qualitative temporal information exist [13], and an overview of sophisticated interactive time-based visualization techniques is given by Aigner et al. [4].

Time graph visualizations can be embedded in a linked views system to interact with temporal metadata [29]. The purposes of such environments range from the dynamic exploration of spatio-temporal data [5], [11] to determining meaningful events in text collections [30]. LeadLine [12] and CloudLines [23] use small multiples of time graphs to support the comparative analysis of different topics in news and social media data. Topical information can be provided in the form of tags such as in Parallel Tag Clouds [8], in which each column lists tags of a certain time slice, or SparkClouds [24], in which each tag is annotated with a time graph in order to illustrate the tag’s usage over time.

Stacked Graphs [6] are often used to illustrate topical changes in a dataset. The original idea introduced as the ThemeRiver by Havre et al. [14] aimed at visualizing thematic variations over time within large document collections. Shi et al. [27] enhance such graphs by placing predominant tags

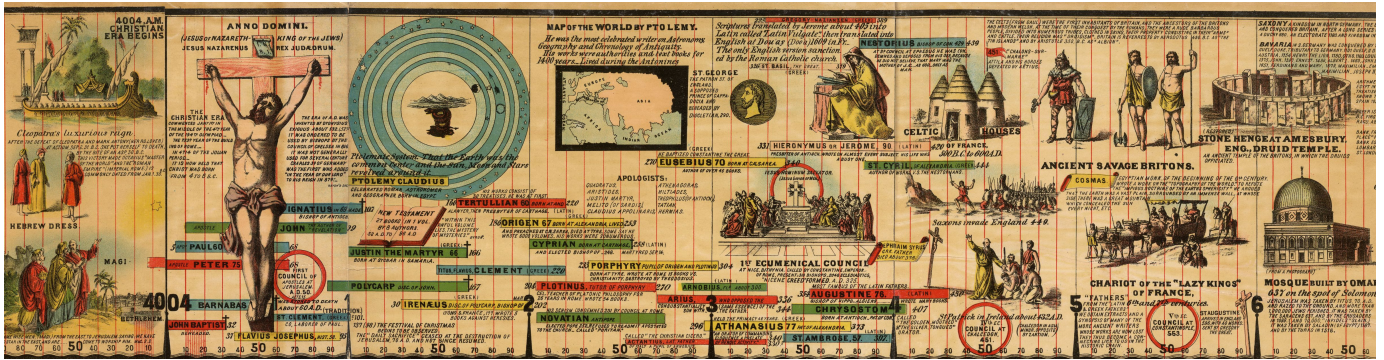


Fig. 1: Snippet of Adams’ *Synchronological Chart* focusing on the biblical section. According to the image sizes, the crucifixion of Christ was the most relevant event between 90BC to 670AC.

at certain positions in streams. Although their approach is similar to ours as they fill “trend segments” (time ranges of a stream with many data items) with (rectangular) tags. A bag-of-words approach is used to define the tag set for a trend segment, and tags are arranged in a greedy manner within the trend segment’s area—irrespective of the tags’ dedicated vertical positions that could be derived from the temporal information. Furthermore, a tag might represent more than one data item, so that detailed information on individual data items belonging to specific time eras are only shown when interacting with the time graph (cf. [31]). A similar example is given by RoseRiver [10], a time-based visualization that supports the visual analysis of evolving topics over time. Major topics are represented as color-coded streams over time that merge and split. Few words indicate contextual information of topics, and a comprehensive tag cloud is shown on demand. By contrast, TextFlow [9] provides access to individual data items that constitute a time graph with a timeline listing a subset of data items related to a selected topic.

Cultural heritage collections use time graphs to display inherent temporal information [21], [22]. For example, Hinrichs et al. [15] use a time-based bar chart for the exploration of commodity trading, and Cho et al. [7] use a stacked time graph for analyzing events in Roman history. A rather abstract timeline visualization is EMDialog [16]. The timeline is a tree cut section—each ring represents a decade—that visualizes statements from and about Emily Carr’s life and work. MusikerProfiling [20] takes uncertain datings into account and overlaps the life spans of musicians on a vertical timeline to support comparative analyses. TimeSlice [32] is a traditional timeline visualization that displays a representative for each data item, and multiple layers support a comparative analysis of data facets.

While traditional timeline approaches are inappropriate for visualizing large data collections as each data item has an own visual representative, none of the above mentioned techniques enhances a quantitative temporal representation with details on individual data items that constitute a time graph. Huynh et al. [18] arrange images on a timeline in columns, and representative images are scaled to facilitate semantic zooming.

Thus, the size of an image does not represent its individual but the importance of the cluster it belongs to. In addition, the arrangement of images leads to whitespaces, and the timeline does not reflect time-specific quantities. Itoh et al. [19] stack image thumbnails extracted from blogs and TV on a 3D timeline. While time-dependent quantities are conveyed, the importance of an image is not outlined as all images have the same size. Compared to these methods, Timages operates from a different direction. First, a timeline is computed that reflects the temporal distribution of a given dataset. Second, the resultant time graph in the form of a polygonal area is filled with thumbnails of the most relevant data items of specific time periods; relevance of a data item is mapped to image size. In contrast to the two methods listed above, Timages communicates the temporal distribution accordingly *and* it keeps the sizes of images coherent to the given relevance values. As the priority of the presented approach is filling the given space, a drawback is that not all data items can be drawn, in the worst case, some of the most relevant ones. But given the overview purpose of Timages, and the possible use of different timeline scalings to increase the number of thumbnails to be displayed (see Figures 2(a) and 2(b)), this drawback can be acceptable for many tasks and scenarios.

### III. TIMAGES DESIGN

Timages enables filling a polygonal region that represents a horizontal time graph with thumbnails of data items. In the following, let  $D = \{d_1, \dots, d_n\}$  denote a data set with  $n$  data items. The following constraints are given:

- The polygonal region is temporally aligned. The thumbnail of a data item  $d_i$  shall be placed as close as possible to the x-position that corresponds to  $d_i$ ’s time stamp  $t_i$ .
- In order to visualize data items having a high relevance to the observed topic more prominently, thumbnails are scaled dependent on relevance values. The thumbnail of  $d_i$  is scaled according to  $d_i$ ’s relevance  $R_i$ , which is a positive floating-point number in the range  $(0, 1]$ .
- The aspect ratios of thumbnails vary. Thus, using a regular grid in order to place the images aiming to minimize white spaces is not possible. The rectangular

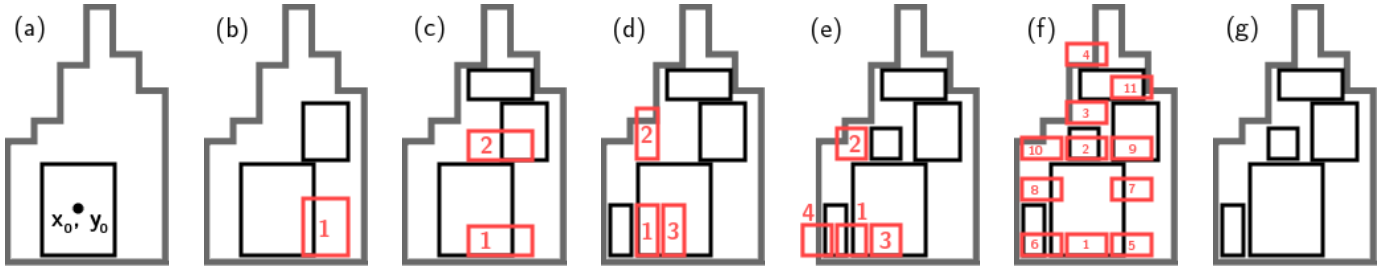


Fig. 2: Filling a *bars* area. Each subfigure illustrates the shift operations performed to place a single thumbnail.

thumbnail size of data item  $d_i$  is defined by width  $w_i$  and height  $h_i$ .

- The thumbnails of the most relevant data items shall be placed first, thus,  $D$  is sorted by decreasing relevance values.

**Thumbnail Scaling:** A user-defined maximum image height  $h_{max}$  is used to scale the thumbnail of the most relevant data item  $d_1$ . Scaled width  $h_1^s$  and scaled height  $w_1^s$  of  $d_1$  are defined by

$$h_1^s = h_{max} \quad \text{and} \quad w_1^s = \frac{w_1}{h_1} \cdot h_{max},$$

and  $A_1 = h_1^s \cdot w_1^s$  denotes the area covered by the thumbnail of  $d_1$ . The area  $A_i$  of the thumbnail of an arbitrary data item  $d_i$  is linearly interpolated according to the respective relevance values. The resultant scaled width  $w_i^s$  and scaled height  $h_i^s$  are defined by

$$w_i^s = \sqrt{R_i \cdot A_1 \cdot \frac{w_i}{h_i}} \quad \text{and} \quad h_i^s = \frac{h_i}{w_i} \cdot \sqrt{R_i \cdot A_1 \cdot \frac{w_i}{h_i}}.$$

A linear interpolation is sufficient. If a logarithmic or square root scaling is desired, relevance values can be defined accordingly. If the resultant area is smaller than a user-defined minimum area  $A_{min}$ , corresponding thumbnails are scaled to fit  $A_{min}$ .

**Thumbnail Placing:** Dependent on the user-defined, rectangular space, Timages computes different types of polygonal areas  $P$  (see Figure 3) where thumbnails will be placed:

- *bars*: A horizontal bar chart having the time axis at the bottom and representing the temporal distribution of data items is computed. The union of all bars defines the polygonal area  $P$ .
- *stream*: As above, but in order to center the thumbnails of the most relevant data items the horizontal time axis is oriented in the vertical center of the given screen space.
- *rectangle*: This variant uses the entire rectangular space for thumbnail placement, and it can be accompanied with a horizontal time graph illustrating the temporal data distribution. The time axis is oriented in the vertical center.
- *unbounded rectangle*: As above, the time axis is oriented in the vertical center, but in order to enable placing all images, this variant does not use a boundary for the  $y$  dimension. The composition of all images visually composes a quantitative time graph.

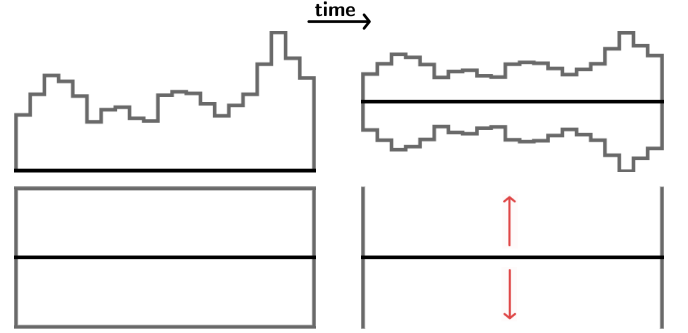


Fig. 3: Area types *bars*, *stream*, *rectangle* and *unbounded rectangle*

Timages implements a strategy to fill the given polygonal area  $P$  with thumbnails of  $D = \{d_1, \dots, d_n\}$  dependent on the orientation of the time axis. The procedure for filling a *bars* area is depicted in Algorithm 1 and illustrated with an example in Figure 2. First, the algorithm tries to position the thumbnail's bounding box  $b_i$  centered at a vertical axis  $x(t_i)$  that stands for the respective time stamp  $t_i$  of data item  $d_i$  (see Figure 2(a)). When occlusions with previously positioned thumbnails occur,  $y_i$  is increased accordingly as shown in Figures 2(b) and (c). Finally, if  $b_i$  intersects the boundary of  $P$  (see Figure 2(d)),  $b_i$ 's  $x$ -value is alternately increased and decreased by a user-defined number of pixels  $x_s$ , and  $b_i$  is again vertically shifted (see Figure 2(e)). If no valid position is found in  $N$   $x$ -shifting iterations,  $d_i$ 's thumbnail is discarded (see Figures 2(f) and 2(g)).

When the time axis is vertically centered (for *stream* and *rectangle* areas), the filling strategy operates similarly. The only difference is that also vertical shifting operations alternate with increasing and decreasing  $y$ -values to find thumbnail locations for given  $x$ -values that are close to  $y = 0$ . For *unbounded rectangle* areas, only vertical shiftings are performed.

The presented solution for packing thumbnails is insofar unique as rectangle packing algorithms (e.g., [17], [25]) typically do not constrain rectangle placement with pre-defined vertical positions. Nevertheless, the presented method is only a heuristic solving a complex problem. However, as Timages is implemented as a JavaScript plugin, this straightforward shifting-based space-filling approach serves to quickly deliver an infographic-like time graph for Web-based applications.



Fig. 4: Timages *unbounded rectangle* graph of Wikipedia inventor portraits.

---

**Algorithm 1** FillBarsArea(D,P)

---

```

for all  $d_i \in D$  do
 $x_i \leftarrow x(t_i), y_i \leftarrow \frac{h_i^s}{2}, b_i \leftarrow \emptyset, n \leftarrow 0$ 
while  $n < N \wedge b_i \equiv \emptyset$  do
 $b_i^r \leftarrow (\{x_i - \frac{w_i^s}{2} + n \cdot x_s, y_i - \frac{h_i^s}{2}\}, \{x_i + \frac{w_i^s}{2} + n \cdot x_s, y_i + \frac{h_i^s}{2}\})$ 
 $b_i^l \leftarrow (\{x_i - \frac{w_i^s}{2} - n \cdot x_s, y_i - \frac{h_i^s}{2}\}, \{x_i + \frac{w_i^s}{2} - n \cdot x_s, y_i + \frac{h_i^s}{2}\})$ 
if  $b_i^r \cap P \equiv b_i^r \wedge b_i^r$  does not intersect any  $b_j$  ( $j < i$ ) then
 $b_i \leftarrow b_i^r$ 
else if  $b_i^l \cap P \equiv b_i^l \wedge b_i^l$  does not intersect any  $b_k$  ( $k < i$ ) then
 $b_i \leftarrow b_i^l$ 
else
 $y_r \leftarrow \infty, y_l \leftarrow \infty$ 
if  $b_i^r \cap P \equiv b_i^r \wedge b_i^r$  intersects  $b_j$  then
 $y_r \leftarrow y_j + \frac{h_j^s}{2} + 1$ 
end if
if  $b_i^l \cap P \equiv b_i^l \wedge b_i^l$  intersects  $b_k$  then
 $y_l \leftarrow y_k + \frac{h_k^s}{2} + 1$ 
end if
 $y_i \leftarrow \min(y_r, y_l) + \frac{h_i^s}{2}$ 
if  $y_i \equiv \infty$  then
 $n \leftarrow n + 1$ 
end if
end if
end while
end for

```

---

#### IV. USE CASES

Three scenarios illustrate the capability of the approach to generate infographic-like time graph visualizations. Except in the first scenario, the height of the given rectangular space was 500 pixels, and  $h_{max} = 150$  was used as maximum thumbnail height. Dependent on the number of data items  $n$ , we observe

the following parameters to evaluate the quality of resulting Timages graphs:

- $p_n$  is the percentage of placed vs. discarded thumbnails
- $p_{100}$  denotes the number of placed images out of the 100 most relevant data items
- $f$  is the first top-ranked data item for which the thumbnail could not be printed in the graph

**Inventor Portraits.** In the German Wikipedia, 1,861 individuals are listed in the category *inventor*.<sup>1</sup> By crawling the corresponding web pages, 703 (public domain) portraits could be retrieved. Inventors without portrait were regarded too less relevant. To determine the relevance of an inventor with portrait, a simple heuristic was chosen: the amount of text written about the inventor in the Wikipedia article. This strategy delivers Friedrich Wilhelm Bessel, a pioneering German astronomer, mathematician, physicist and geodesist, as most important inventor. Other known inventors like Gottfried Wilhelm Leibnitz, Thomas Alva Edison, Wernher von Braun or Carl Friedrich Gauß are also ranked in the Top 10. Konrad Adenauer is ranked second though he might gained his popularity by being the first Chancellor of the Federal Republic of Germany instead of inventing the “Kriegsbrot” and “Friedenswurst” during World War I. The resultant time graph is shown in Figure 4. To get a readable layout, a smaller value of 120 pixels for  $h_{max}$  was used, and all inventors who lived before 1500 were disregarded. As all thumbnails

<sup>1</sup><https://de.wikipedia.org/wiki/Kategorie:Erfinder>



Fig. 5: Timages *rectangle* graph showing portraits of musicians of the MusiX database.

for all remaining 703 inventors are placed ( $p_n = 100\%$ ,  $p_{100} = 100\%$ ,  $f = \emptyset$ ), the composition of images reflects the history of inventorship with the highest peak before World War I. The following steady decrease of the number of inventors lasts until the end of World War II.

**Musician Portraits.** The MusiX, a successor of the Bavarian Musicians Encyclopedia Online (German: “Bayerisches Musiker-Lexikon Online (BMLO)”) [1], is a musicological database that contains biographical information about around 26,500 musicians with various professions, e.g., composers, instrument makers, musicologists, etc. The popularity of a musician is the total number of references about the musician from online and print media. Whereas Wolfgang Amadeus Mozart is the most popular musician with more than 150,000 references, for many musicians only a single reference, the BMLO, exists. The popularity values—after square root scaling and normalization to  $(0, 1]$ —are used as relevance values for thumbnail scaling. The database contains portraits for only 1,774 musicians, for all other musicians a default thumbnail is used. The Timages visualization for musicians who lived between the 15th and the 19th century is shown in Figure 5. The *rectangle* time graph reflects the major focus of musicology in researching the *heroes* of the 18th and 19th century. Earlier, composers are rarely present as rather portraits of librettists, courtly patrons and fine artist are prominent. While only a small amount of 1,932 portraits can be shown ( $p_n = 7,3\%$ ), thumbnails of most of the top-ranked musicians could be placed ( $p_{100} = 94\%$ ). The first discarded musician with the highest relevance is Friedrich Wilhelm Joseph von Schelling at rank  $f = 65$ .

**Movie Posters.** The Internet Movie Database (IMDb) [2] contains information about more than 180,000 movies, for most of them poster thumbnails are available. For arranging most relevant movie posters on the time graph, the relevance  $R_i$  of a movie  $d_i$  is defined according to IMDb’s weighted rating (also used to define the IMDb Top250) as

$$R_i = \frac{v_i}{v_i + 25,000} \cdot r_i + \frac{25,000}{25,000 + v_i} \cdot 7$$

with the number of votes  $v_i$  for the movie, and the average rating  $r_i$ . The resultant values range from 1 to 10, and in order to make minimal differences more salient, a logarithmic scaling is applied before normalizing to  $(0, 1]$ . Figure 6 illustrates different scenarios. Timages graphs in the *bar* variant arranging the posters of 10,491 movies having more than 5,000 votes ranked by decreasing weighted rating are outlined with a linear (see Figure 6a) and a logarithmic scaling (see Figure 6b). While using the latter variant yields whitespaces in the first 50 years, the total of movies represented increases from  $p_n^{lin} = 5,7\%$  to  $p_n^{log} = 15,8\%$ , respectively, from  $p_{100}^{lin} = 44\%$  to  $p_{100}^{log} = 88\%$ . Using the linear scaling, the second most relevant movie ( $f^{lin} = 2$ ), *The Godfather* (1972), cannot be displayed due to the low amount of movies in the corresponding time range. But using a logarithmic scaling provides more space for such time ranges ( $f^{log} = 13$ ). Figure 6c illustrates the same set of movies ranked by increasing weighted rating, thus, the worse a movie the bigger its thumbnail. The *rectangle* Timages graph has the following features:  $p_n = 11,6\%$ ,  $p_{100} = 45\%$  and  $f = 10$ . The graph illustrates that the total number of low-ranked movies steadily increases. For example, the movie *Manos: The Hands of Fate* with  $r_i = 1.9$  from 1966 sticks out. Figure 6d shows a Timages *stream* graph with a linear scaling. All 1,870 Western movies with more than 100 votes are considered and ranked by decreasing weighted rating. Most of the movie posters are displayed ( $p_n = 76,8\%$ ), especially, all except one movie out of the Top 100 ( $p_{100} = 99\%$ ,  $f = 11$ ). The graph illustrates different golden ages of the Western genre. In the 1950s, many classical Western movies were produced in the United States—popular movies like *High Noon* (1952) and *The Searchers* (1956) stick out. In the 1960s, movies establishing the Spaghetti Western like *The Good, the Bad and the Ugly* or *Once Upon a Time in the West* are prominent, and the fall of the whole Western genre by the end of the 1970s is visible as only small thumbnails occur. A slight revival of the genre after 2000 is seizable featuring movies like *Django Unchained* (2012) and *The Revenant* (2015).



## V. DISCUSSION

Though Timages delivers aesthetically appealing, infographic-like time-based representations, the used heuristic for the complex space-filling task comprises some disadvantages.

Scaling image sizes and computing polygonal areas are not interlinked, and both steps depend on user-configurable parameters. Thus, in the *bars* and *stream* variant it might happen that thumbnails of most relevant data items cannot be placed as the available space for a certain time range is too small and the data item of the thumbnail to be placed is too relevant. The *rectangle* area type ensures having enough space for that purpose, but to also increase the coverable screen space for the *bars* and *stream* area types, one can switch from linear to logarithmic scaling of the y-axis. However, only the *unbounded rectangle* variant—applicable to small datasets—ensures that all images, especially the most relevant ones, are placed. Another workaround for the problem would be if images of highly relevant data items are shrunk if required, but this would harm the comparability of data items.

As the space-filling strategy performs horizontal shifts to the left and to the right, occasionally, the temporal order of adjacent thumbnails may be incorrect. To keep such cases rare, only slight horizontal movements are allowed; the maximum horizontal movement of a thumbnail of an arbitrary data item  $d_i$  ranges from  $-\max(w_i^s, w_y)$  to  $+\max(w_i^s, w_y)$ , with  $w_y$  being the width of a time slice of the polygonal area  $P$ .

The complexity of the space-filling strategy depends on the available screen size assigned to  $P$ , maximum image height  $h_{max}$  and the diversity of relevance values. The larger the difference between minimum and maximum relevance, the smaller thumbnails with low relevance values will be scaled. This leads to an increase of vertical shift operations. When trying to place a thumbnail of data item  $d_i$ , the maximum number of possible overlaps is  $i - 1$ , but considering that only a subset of already placed thumbnails is in the reachable time range of  $d_i$ , the number of overlap tests is smaller. With  $k$  being the average number of overlap tests when placing an arbitrary thumbnail, the upper bound complexity of the space-filling approach can be assessed by  $O(k \cdot n)$ .

## VI. SUMMARY

The prior goal of Timages is giving a quantitative overview of a time-stamped data collection while also providing samples of relevant data items that construct the time graph. Like tag clouds, Timages cannot show all data items iconographically, but showing the most relevant ones is targeted. One of the major challenges were varying aspect ratios of images to be placed. Having equally sized images, a space-filling approach operating on a regular grid would be possible.

Although this paper focuses on the space-filling strategy aiming to generate visually appealing, infographic-like time graphs, Timages can be used in an interactive Web-based environment, i.e. details about individual data items are shown

in popups. Timages is implemented as a user-configurable JavaScript library based on D3.js.<sup>2</sup>

## REFERENCES

- [1] Bayerisches Musiker-Lexikon Online, 2018. ed. Josef Focht. <http://www.bmlo.lmu.de/> (Accessed 2018-03-02).
- [2] Internet Movie Database (IMDb), 2018. <http://www.imdb.com/> (Accessed 2018-03-02).
- [3] S. C. Adams. Chronological Chart of Ancient, Modern and Biblical History, 1871. [https://en.wikipedia.org/wiki/Adams\\_Synchronological\\_Chart\\_or\\_Map\\_of\\_History](https://en.wikipedia.org/wiki/Adams_Synchronological_Chart_or_Map_of_History) (Retrieved 2018-02-28).
- [4] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer Publishing Company, Incorporated, 1st edition, 2011.
- [5] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
- [6] L. Byron and M. Wattenberg. Stacked Graphs – Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, Nov 2008.
- [7] I. Cho, W. Dou, D. X. Wang, E. Sauda, and W. Ribarsky. VAIroma: A Visual Analytics System for Making Sense of Places, Times, and Events in Roman History. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):210–219, 2016.
- [8] C. Collins, F. Viegas, and M. Wattenberg. Parallel Tag Clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91–98, Oct 2009.
- [9] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. TextFlow: Towards Better Understanding of Evolving Topics in Text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, Dec 2011.
- [10] W. Cui, S. Liu, Z. Wu, and H. Wei. How Hierarchical Topics Evolve in Large Text Corpora. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2281–2290, Dec 2014.
- [11] M. Dörk, S. Carpendale, C. Collins, and C. Williamson. VisGets: Coordinated Visualizations for Web-based Information Exploration and Discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1205–1212, Nov 2008.
- [12] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. Zhou. LeadLine: Interactive visual analysis of text data through event identification and exploration. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 93–102, Oct 2012.
- [13] R. L. Harris. *Information Graphics: A Comprehensive Illustrated Reference*. Oxford University Press, Inc., New York, NY, USA, 1999.
- [14] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, Jan. 2002.
- [15] U. Hinrichs, B. Alex, J. Clifford, A. Watson, A. Quigley, E. Klein, and C. M. Coates. Trading Consequences: A Case Study of Combining Text Mining and Visualization to Facilitate Document Exploration. *Digital Scholarship in the Humanities*, 2015.
- [16] U. Hinrichs, H. Schmidt, and S. Carpendale. EMDialog: Bringing Information Visualization into the Museum. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1181–1188, Nov 2008.
- [17] E. Huang and R. E. Korf. Optimal Rectangle Packing: An Absolute Placement Approach. *J. Artif. Int. Res.*, 46(1):47–87, Jan. 2013.
- [18] D. Huynh, S. M. Drucker, P. Baudisch, and C. Wong. Time Quilt: Scaling up Zoomable Photo Browsers for Large, Unstructured Photo Collections. In *Conference on Human Factors in Computing Systems (CHI'05)*, pages 1937–1940. ACM New York, NY, USA, 2005.
- [19] M. Itoh, M. Toyoda, C. Z. Zhu, S. Satoh, and M. Kitsuregawa. Image Flows Visualization for Inter-media Comparison. In *2014 IEEE Pacific Visualization Symposium*, pages 129–136, March 2014.
- [20] S. Jänicke, J. Focht, and G. Scheuermann. Interactive Visual Profiling of Musicians. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):200–209, Jan 2016.

<sup>2</sup><http://timages.vizcovery.de>

- [21] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARS*. The Eurographics Association, 2015.
- [22] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. Visual Text Analysis in Digital Humanities. *Computer Graphics Forum*, 2016.
- [23] M. Krstajic, E. Bertini, and D. Keim. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432–2439, Dec 2011.
- [24] B. Lee, N. Riche, A. Karlson, and S. Carpendale. SparkClouds: Visualizing Trends in Tag Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, Nov 2010.
- [25] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. Rectangle-packing-based Module Placement. In *Proceedings of the 1995 IEEE/ACM International Conference on Computer-aided Design, IC-CAD '95*, pages 472–479, Washington, DC, USA, 1995. IEEE Computer Society.
- [26] D. Rosenberg and A. Grafton. *Cartographies of Time: A History of the Timeline*. Princeton Architectural Press, 2013.
- [27] L. Shi, F. Wei, S. Liu, L. Tan, X. Lian, and M. Zhou. Understanding text corpora with multiple facets. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 99–106, Oct 2010.
- [28] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.
- [29] F. B. Viegas, M. Wattenberg, F. Van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.
- [30] C. Weaver. Multidimensional Visual Analysis Using Cross-Filtered Views. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, pages 163–170, Oct 2008.
- [31] P. Xu, Y. Wu, E. Wei, T.-Q. Peng, S. Liu, J. Zhu, and H. Qu. Visual Analysis of Topic Competition on Social Media. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2012–2021, Dec 2013.
- [32] J. Zhao, S. M. Drucker, D. Fisher, and D. Brinkman. TimeSlice: Interactive Faceted Browsing of Timeline Data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12*, pages 433–436, New York, NY, USA, 2012. ACM.