ABSTRACT

Storyline visualizations help users understand and analyze the interactions between entities in a story and explore how entity relationships evolve over time. Existing storyline techniques support limited or no user interaction due to the high cost of layout. Typical design considerations for storyline layout include minimizing line crossing and line wiggling, which are NP-hard or NP-complete problems. Interactive generation of layouts, such as in response to dynamic querying, is a substantial performance challenge. Motivated by application to visualization of classical Latin texts, we present work in progress on a new approach that uses force directed layout to dynamically position storyline elements in an agile, legible fashion in real time. We outline how this approach can support appealing layout even in response to diverse user interactions including rapid panning and zooming and dynamic filtering of storyline elements.

Keywords: Storyline visualization, force directed layout, dynamic queries, textual criticism, critical edition, Latin texts.

1 INTRODUCTION

Munroe popularized the storyline visualization technique in hand drawn form in his XKCD comic about movie narratives. There is growing interest in automating the visual layout of data using storylines [4, 6, 3, 5]. The methods described by Tanahashi, et al. [6, 5] produce storylines for hundreds of entities and event times but take minutes to lay out, making them too slow for many user interactions including dynamic queries. Storyflow [3] generates layouts more quickly, but provides a limited set of interactions, such as bundling and straightening, for fine-tuning generated layouts.

Like other literatures, Latin texts have three major categories of readings, called editions: the manuscripts, early printed editions, and modern editions. Prior to 14th century, Latin texts were mostly preserved by scribes, who copied the original text to a manuscript by hand, introducing textual variations and transcription errors in the process. The original text—which is often lost—can have multiple manuscripts that vary from each other substantially.

At the end of 14th century, publishers started publishing printed editions of the texts. In many cases, publishers chose a manuscript at random to publish as a printed edition, and then discarded the manuscript itself, considering it redundant to the published edition. The printing process itself also introduced errors. Manuscripts and printed editions are called witnesses. Textual differences between witnesses are called variant readings or simply variants.

An editor’s ultimate goal is to produce a critical edition that presents a conjectured original text reconstructed using the editor’s choices of variants. The text in a critical edition is accompanied by a critical apparatus, which lists the variants that the editor considered along with the sigla (abbreviated names) of the witnesses that supply those variants. The apparatus appears as a sequence of footnotes at the bottom of each page, in a compactly encoded format that requires substantial training to read. The information in an apparatus is generally inaccessible to novice readers of Latin texts.

The Digital Latin Library (DLL) project aims to take a step forward in publishing digital critical editions of Latin texts, presenting text and apparatus in ways that are useful to scholars but also accessible to students and other novice readers. With our storyline approach, we aim to create an alternative representation of the critical apparatus that will help DLL users read and understand variant readings as conjectured by an editor. The dynamic query capabilities of our storyline layout will also help scholars reason about manuscripts families (stemma) based on relationships between variants. They can filter witnesses based on the stemma, analyze how witnesses vary and co-occur throughout the text, and select variants to explore the editor’s word choices in a critical edition.

2 LAYOUT ALGORITHMS

Figure 1 shows a storyline visualization of Giarratano’s critical edition of the classical Latin poem Calpurnius Siculus. We implemented a prototype of our layout algorithm and visualization by adapting the existing general-purpose graph view in Improvise [7] to the special case graph topology of storylines. The bottom of the layout contains words for which there are some variant readings in the apparatus (see figure 1A). The words are equally spaced from left to right in the same order as the original text, forming vertical slots in text order in a manner similar to narrative time in existing storyline visualizations. Each line (figure 1B) represents a witness that has variant readings for these words. Each variant of a word is aligned vertically with the word, and horizontally with the contributing witness as shown in figure 1C. The green cylindrical glyphs group the witnesses specifying same variants (figure 1H).

The lines in our storyline layout exist throughout the entire text, as opposed to characters in movie narrative charts, where the characters appear and disappear over time, causing the lines to be discontinuous throughout the visualization. This calls for obvious design considerations for spacing and alignments of the lines in our storyline layout. The grouping of witnesses in terms of variants (e.g., figure 1H) often varies substantially across consecutive vertical slots (i.e., the words in reading order). In contrast, characters in a movie tend to interact with each other over long periods of time, causing lines to stay together across multiple slots. Consequently, the differences between our application case and the general case is that we need to group entities (lines) within individual slots but also accommodate frequent changes from slot to slot. This motivated us to develop a new storyline layout technique using a specialized force directed layout (FDL) algorithm. The following section elaborates the force model we have used to incorporate the functional and design considerations specific to our application.

For our layout, we use a modified version of the Fruchterman-Reingold model. Nodes attract and repulse each other in the usual way. Iterative movement minimizes the cumulative force on each
node, approaching an equilibrium state [2]. In our force model, nodes represent variants of a word at a particular point in the text. Edges are connections between variants for a given witness, and “packs” are sets of witnesses that use the same variant for a word.

Application of storylines to critical editions can be treated as a special case of graph layout. The variant nodes for each word are initially placed in a slot, with slots laid out from left to right for each word in text order. All forces and node movements are vertical only. The forces included in our current force model are:

- **Node-node forces**: Nodes in a slot attract and repulse each other using a vertical inverse-squared force with equilibrium distance. This groups and separates variants in each word slot.
- **Node-edge forces**: Nodes connected across slots (along a storyline) attract each other vertically, thus pulling the nodes to align horizontally. This force helps to reduce line wiggling.
- **Node-pack attraction**: Nodes that specify the same variant are initially positioned in a pack within each word’s slot. Nodes within each variant pack attract each other. This force helps to cluster occurrences of each variant for a given word and reduce overlap between packs in a slot.
- **Pack-pack repulsion**: Within each slot, packs themselves repulse each other, using an inverse-square force on their centroids. This helps to separate variants within each word’s slot.

We apply these forces iteratively and continuously, including during navigation and dynamic query interactions. In each iteration, we combine the forces to achieve a good balance. This results in convergence to reasonable layouts regardless of prior layout state. We hypothesize that these characteristics of our model will allow for graceful response to a wide variety of interaction types, speeds, and patterns. This interactive behaviour will in turn greatly facilitate the flexible design of coordinated multiple view visualizations that include storyline views.

## 3 Customizing Forces for Specialized Layouts

We are currently experimenting with combinations of the above forces. Figure 1 shows progress for a portion of a critical edition. Moving forward, we plan to experiment with additional forces, like wall forces and symmetry forces, that we expect to improve layout aesthetics. Minimizing the cumulative force on each node can result in indefinite spreading out of nodes [1]. In our case, top and bottom wall forces can be applied to keep all nodes within a defined vertical space. Symmetry forces can be applied to each word’s variant groups to more evenly space them within their packs. We anticipate that this will further improve horizontal alignment (and hence also readability and aesthetics) of witness lines individually and collectively, while maintaining consistent layout during dynamic filtering.

Once we decide on which forces to include in our force model, we plan to fine-tune the combination of forces. We have been experimenting with how strong these forces should be relative to one another. We are exploring random placement and hill climbing algorithms as ways to tune the parameters. We are also planning an evaluation to assess usability of promising combinations.

We anticipate that the dynamic interactive querying capability of our storyline visualization will help users explore and analyze various relational structures in a given dataset. The high dynamics of interactive querying requires layout techniques that will dynamically generate readable layouts in real time. Current state-of-the-art storyline techniques do not produce layouts in real time, making them generally unsuitable for use with many common kinds of interactive querying including dynamic filtering. We aim to develop a storyline generation technique that will dynamically generate decent layout in real time, allowing users to interactively explore complex queries with the support of an appealing layout.

## Acknowledgements

This work was supported as a part of the Digital Latin Library project by a grant from the Andrew W. Mellon Foundation.

## References


