Journal of Visual Languages and Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

A visualisation technique for large temporal social network datasets in Hyperbolic space ☆

Q1 Uraz Cengiz Turker, Selim Balcisoy

Q3 *Sabanci University, Orhanli, University Street No: 27, 34956 Tuzla, Istanbul, Turkey*

ARTICLE INFO

Article history:
Received 14 December 2011
Received in revised form
8 October 2013
Accepted 27 October 2013

Keywords:
Social networks
Hyperbolic layout
Visualisation
Temporal data

ABSTRACT

Visualisations of temporal social network datasets have the potential to be complex and require a lot of cognitive input. In this paper, we present a novel visualisation approach that depicts both relational and statistical information of evolving social structures. The underlying framework is implemented by the usage of *Hyperbolic Geometry* to support focus context rendering. The proposed method guarantees representing prominent social actors through scaling their representations, preserves user's mental map, and provides the user to reduce visual clutter by means of filtering.

1. Introduction

In recent years visualisation of large temporal social network datasets has become one of the hot topics in information visualisation domain. Existing techniques rely on relational information to compute layouts as node-link diagrams and omit network metrics [1]. Therefore, as experienced in [2,3] analysts employ at least two software suits such as Net-Draw [4] and ORA [5] in parallel to understand the structure and the communication channels between actors of evolving social network datasets. Without loss of generality, we can summarize the requirements for visualising temporal social network datasets as follows: The underlying method must use an appropriate visual language to combine information space and the visualisation space. The technique must reflect the evolution of the network through minimizing the topological differences between adjacent images [6]. Finally the method must highlight the important aspects of the data with interactive, topology preserving clutter reduction techniques. In this paper, we address these requirements and propose a *Hyperbolic Temporal Layout Method* (HTLM for short) which uses nodes as visual metaphor for social actors, and links for social relations. HTLM draws the layout in the Hyperbolic space and project this Hyperbolic space in an Euclidean sphere. As illustrated in Fig. 1, it derives prominences of nodes from user selected network metrics, then uses these values to position prominent nodes close to the centre of the Hyperbolic space and others on circular orbits. Therefore, users can recognize prominences of nodes by comparing their distances to the centre. Our layout method represents evolution of a social network by performing a few topological modifications on the layout to enhance dynamic stability. The major contributions of this paper are as follows:

- We propose a Hyperbolic temporal layout method that represents the evolution of relations amongst network actors and structural patterns of a social network.
- We propose encoding user defined network metrics into a temporal layout to improve the information density of the generated visuals.
- We introduce a visual clutter control mechanism for HTLM which can significantly reduce overall visual clutter without modifying the topology of the layout.

We evaluate the proposed layout method on several synthetic and well known real-life temporal social network datasets through using multiple network metrics.

☆ This paper has been recommended for acceptance by Shi Kho Chang.

E-mail addresses: urazc@sabanciuniv.edu (U. Cengiz Turker), balcisoy@sabanciuniv.edu (S. Balcisoy).

1045-926X/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jvlc.2013.10.008

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

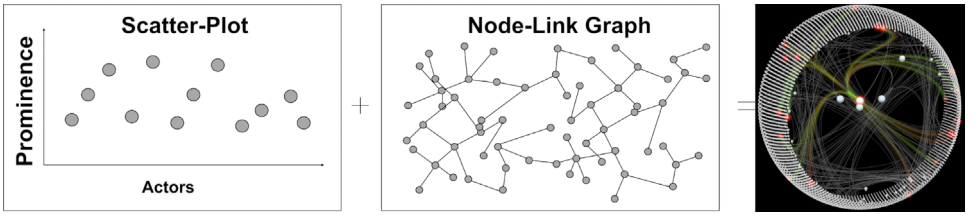

Fig. 1. We use actor prominence values according to a given network metric to calculate layout hence provide both relational and structural information to the user.

We measure the dynamic stability of the HTLM with two fundamental network metrics: *Average Distance* and *Orthogonal Ordering*. We conducted a usability study to evaluate the effectiveness of the proposed method. This paper is organized as follows: the following section reviews existing work that relates to our method. In Section 2, we present the analogy that we have drawn to control nodes' sizes. In Section 3, we discuss the details of our method. Performed case study with synthetic and real-life datasets is explained in Section 4. We present quantitative test results in Section 5. In Section 6, we conclude with the discussion of the layout and present future work.

2. Related work

In this section we briefly summarize the existing literature, since HTLM spans several aspects of visualisation spectra, we split this section into appropriate subsections.

2.1. Social network analysis

Social network analysis is the study of social actors and their relations through formal methods to reveal statistical properties [7]. In [8] Henry et al. divide computer based social network analysis spectra into two subcategories as *menu based systems* and *exploration systems*. Menu based systems such as Pajek [9] and Ucinet [10] provide advanced functionalities to domain-experts to evaluate social network datasets. Exploration systems generate visualisations of network datasets and provide interaction mechanisms such as zooming and filtering to convey details to the user. In [3] Henry et al. described how synthesizing a node-link diagram with matrix representation enhances user performances in terms of readability. In [11] Lee et al. propose an iterative graph representation technique with no global view of the dataset. Exploration of network data through coordinated views is presented in [12]. The semantic substrates [13] utilize visual attributes of data items to represent statistical properties of dataset. Moreover, it can visualise temporal social network datasets through static images with limited scalability.

2.2. Temporal graph drawing

Temporal graph drawing techniques utilize animations and use variations of force directed methods to represent temporal social network datasets as node-link graphs [14]. In [15] a modified GRIP algorithm is used to produce animations of evolving graphs. Kumar and Garland [16] proposed a hierarchical force-directed layout technique that sorts nodes according to weights. This method stratifies nodes into levels, therefore a user can easily get disoriented while exploring relations between the nodes. Besides, it is reported that this method does not scale well for large datasets [1]. In [1] Frishman and Tal proposed a promising solution to represent large temporal datasets. It uses a modified force-directed algorithm that runs on the GPU by passing data values to textures to calculate the layout. This technique advances the layout calculation performance of a force-directed method and its scalability. PieSpy can visualise small temporal datasets based on Fruchterman–Reingold method, where the initial topology of the layout is obtained randomly [17]. Condor [18] uses an algorithm called *sliding frame* to represent the evolution of the dynamics of a social network through time. SoNIA [19] is a graph animation tool that generates videos of animated graphs based on different layout techniques with little interactivity. A similar approach has been presented in [20]. This method uses a mixture of force directed placement and circular layout placement on the 2D surface.

As these methods follow a force-directed placement algorithm, they produce visuals based on relational information amongst social actors. In comparison, our method encodes the network metrics into layout, so users can perform not only relational but also statistical analysis on a given temporal network.

2.3. Quantitative visualisation techniques

As 2D scatter-plots cover main properties of other approaches such as 2D bar-charts, and 2D pie charts, throughout this paper we use *scatter-plot* as a general term to refer 2D quantitative visualisation techniques. Scatter-plots are one of the oldest visualisation techniques to express statistical properties of a dataset. Generally, scatter plots use two axes of Cartesian coordinates to form spatial mappings of the data where each axis corresponds to one data attribute. The main idea behind scatter-plots is to represent dependency between different data attributes [21]. The first scatter-plot representation of a temporal social network is given by Moreno's drawings [22]. The graphic provides user to observe change of roles of social actors as time progresses. While, this representation method is widely applied and well suited to many real-world networks, it is inapplicable when both the time span and the number of network items get larger. Gapminder's Trendanalyzer [23] and similar approaches [24] address this issue through the usage of animations. These techniques do not use an axis to represent time, they however

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

supply adjacent images, which visualise the state of the data, one after another. Therefore, users can observe correlations between several data attributes as time advances. As both Gapminder's Trendanalyzer and other animated scatter-plots do not represent relational information to the user, analysts cannot trace intercommunication patterns. Our method, however, provides animated representation of both relational and statistical information for a social network.

2.4. Focus-context techniques

Focus context techniques represent large datasets by manipulating the quantitative attributes of visual variables. Heat model [25] and Fish-eye [26] are two focus-context methods that alter the parameters of visualisation to establish focus-context images. Hyperbolic techniques are also used to generate focus-context views of datasets. In [27] Munzner demonstrated that large datasets can be visualised by positioning tree nodes on hemispheres according to their spanning tree locations using the Klein model. In [28] Walter proposed a Hyperbolic multi-dimensional scaling technique to represent large clustered items in Hyperbolic space, which uses Poincare's disk model. In [28] Ritter introduces the usage of Hyperbolic self-organizing maps to represent feature maps of high dimensional datasets in Hyperbolic space. Kobourov and Wampler proposed a non-Euclidean spring embedders in [2]. This technique provides focus context images for small non-temporal graphs. Although these techniques provide a high quality images they are not designed for evolving social network datasets. Our methodology, which combines theories of focus-context drawing with techniques of temporal data visualisation, is a promising candidate to visualise temporal social network datasets.

3. Layout generation

In this section we give a detailed description for the proposed visualisation approach. As illustrated in Fig. 2, our layout method first derives actor's prominences through user-selected network metric. Second it calculates nodes' sizes and finally it uses these values to compute the layout in 3D Hyperbolic space. A sample output of our method is represented in Fig. 3.

3.1. Deriving actor prominence

In the literature, two design principles are reported for computing a layout for a temporal social network: using either instantaneous information or historical information [6,29]. Instantaneous information reflects the state of a social network for a particular time frame however as the state of the social network might change rapidly, this schema has the potential of producing layouts which require extra smoothing algorithms. On the other hand, using historical information produces more stable visualisations, however this method prevents the user from realizing the instantaneous events. HTLM uses historical information to derive prominences of social actors.

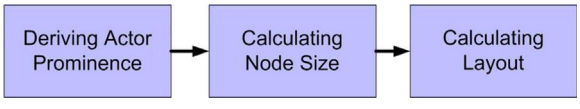

Fig. 2. Flow chart of proposed algorithm.

We now present some definitions used throughout this section. See Table 7 for the descriptions of symbols used in this subsection. We assume that the underlying temporal social network dataset contains a finite set of social actors that interact with each other for a period of time, called the *time span S*. We also assume that the underlying social network dataset possesses information such as names, telephone-numbers or e-mail addresses that allow us to explicitly differentiate the social actors. From now on we refer to such differentiating information as an α-*attribute*. That is, we assume that by inspecting the value of α-attribute one can uniquely identify the actors in the social network.

As the time span refers to a duration, its unit can be year, day, minute, second, millisecond, etc. Moreover, the time span can be split into atomic, measurable frames. The duration of a frame is given by the *time granularity* (Gr), which is the level of detail in which the time is considered. The unit of a time granularity is similar to that of time span i.e. year, day, minute, second, millisecond, etc. For instance "Apr-10-1958, Apr-11-1958" are frames with day granularity but "2013, 2014" are frames with year granularity.

Before deriving actor prominences, we first decide the time granularity for the given temporal social network. There is no restriction on the time granularity and it is decided by a social network expert. After the granularity is set, we, by the usage of the social network dataset, generate a sequence of frames which defines a sequence of *time graphs* $\mathcal{G} = (G_0, G_1, \ldots, G_T)$. Here the t-th graph (or t-th frame) covers the state of the social network within interval $[t{*}Gr, t{*}Gr + Gr]$ where $0 \le t \le T$. In HTLM the time granularity should be uniform, that is the duration of a frame covered by any pair of time graphs should be the same. Moreover, the time granularity cannot be zero.

For a given index t, the time graph G_t contains a finite set of nodes $N_t = \{n_0, n_1, \ldots, n_{|N_t|}\}$, and a finite set of edges $E_t = \{e_0, e_1, \ldots, e_{|E_t|}\}$. Each node n is associated with a unique non-negative integer k called as the *key*. We form a relation between a social actor and a node by mapping the α-*attribute* value to a single key through the usage of the invertible mapping function $\pi : \alpha \to \mathbb{Z}_{\ge 0}$. A key k for node n is assigned to the node while the actor prominence is being computed.

A user-selected network metric, called the *Actor Metric* (AM for short), is retrieved by function f which is defined as $f : N \times G_t \to \mathbb{R}_{\ge 0}$ i.e. it returns the value of user-selected network metric for node n at t-th frame. We use the *Temporal Actor Metric* (TAM for short) value, which gives the sum of the actor metric values over time, to derive actor prominences. Intuitively the temporal actor metric value is given as follows:

$$\mathcal{F}(n,t) = \sum_{0 \le j \le t} f(n, G_j) \quad (1)$$

Finally, we define the *Temporal Importance Metric* (TIM for short) $\mathcal{P}(G_t)$, which is the weight of the social network at

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

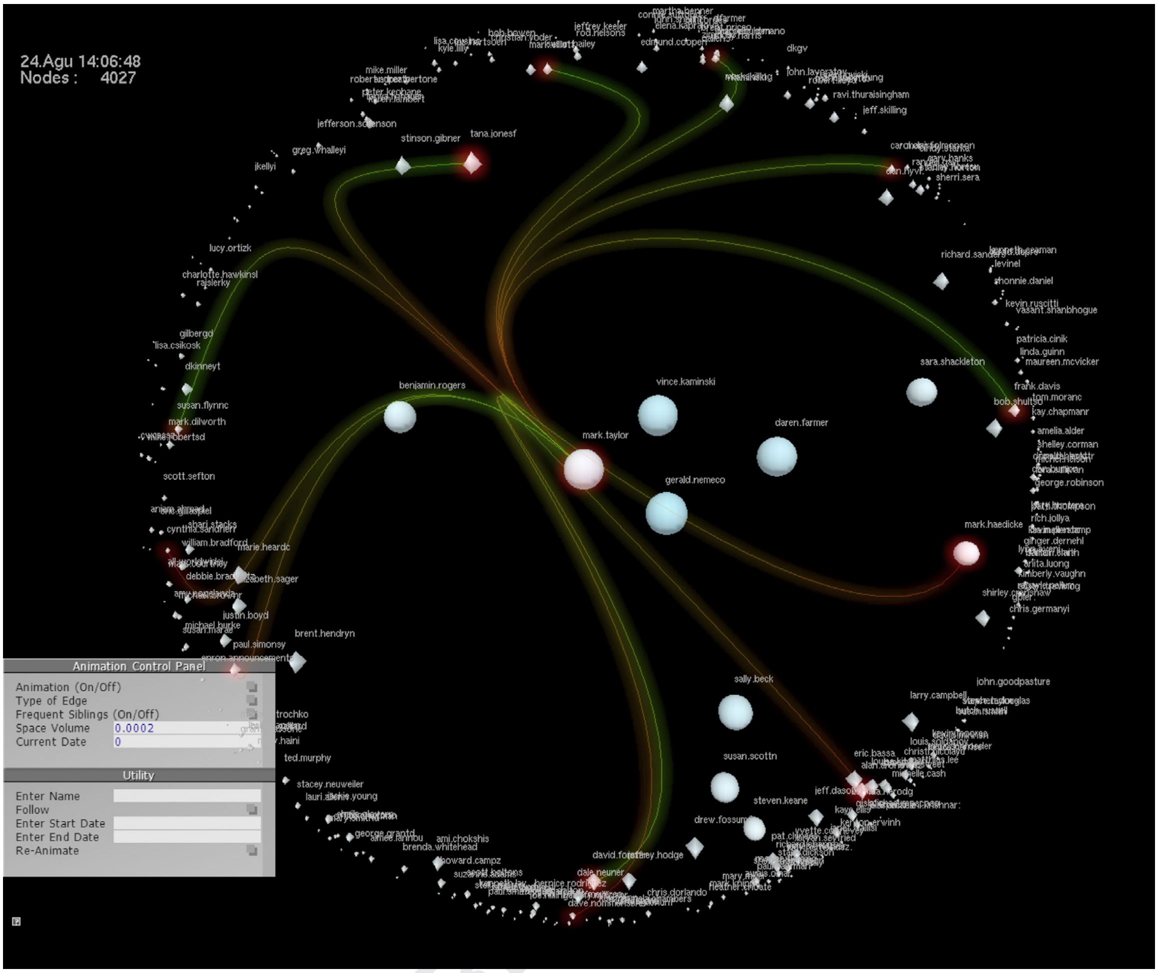

Fig. 3. Visual representation of a temporal dataset with 4027 nodes and 12 edges. Central nodes represent important actors given prominences derived from the closeness centrality. Glow effect is used to highlight selected actor and its relations. The user interface allows filtering the low prominence nodes (left bottom).

Table 1
The time graph G_i.

E_i		N_i
Source	Destination	
v_1	v_2	$v_1, k=96$
v_1	v_3	v_2, k=97
v_2	v_3	v_3, k=98

Table 2
The time graph G_{i+1}.

E_{i+1}		N_{i+1}
Source	Destination	
v_1	u_3	$v_1, k=96$
u_2	u_3	$u_2, k=\infty$
u_2	v_1	$u_3, k=\infty$

t-th frame, as follows:

$$\mathcal{P}(G_t) = \sum_{n \in N_t} \mathcal{F}(n, t) \tag{2}$$

In order to compute Eqs. (1) and (2), we merge each time graph with its adjacent time graph as follows.

Let G_i, G_{i+1} are adjacent time graphs as depicted in Tables 1 and 2. We first compute the AM value of each node in set N_{i+1}, and we assign a unique key to a node if its key value is not defined (nodes u_2, u_3 retrieve keys 99, 100 respectively). Then we merge nodes of these graphs. For each node $u \in N_{i+1}$ we first check if there exists a node $v \in N_i$ such that $k_u = k_v$ (note that node v_1 in Table 2 holds this requirement). If it holds then we compute the *TAM* value of the new node n by adding the *TAM* value of node v to the *AM* value of node u. Otherwise for node n *TAM* value equals to the *AM* value. Afterwards we add the computed *TAM* value to the *TIM* value of the current graph. Finally we copy the existing edge information (E_i) to the new edge set (E_{i+1}) (Table 3) such that in the end the following

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

holds: Let (n_a, n_b) denotes the edge between a pair of nodes n_a and n_b.

1. $\forall v \in N_i$, $\exists u \in N_{i+1}$ such that $k_v = k_u$.
2. $\forall (v_a, v_b) \in E_i$, $\exists (u_a, u_b) \in E_{i+1}$ such that $\{k_{v_a} = k_{u_a}\} \wedge \{k_{v_b} = k_{u_b}\}$.

The overall process is summarized in Algorithm 1.

Algorithm 1. Iterative accumulation algorithm.

Data: A set of time graphs ($\mathcal{G}$)
Result: Accumulate network metric to adjacent graphs.
begin
- $T \leftarrow |\mathcal{G}|$, $i = 0$;
- **for** *each node* $u \in N_0$ **do**
  - Compute $f(u, 0)$;
  - $k_u =$ retrieve_key(u);
  - $\mathcal{F}(u, 0) = f(u, 0)$;
  - $\mathcal{P}(G_0) = \mathcal{P}(G_0) + \mathcal{F}(u, 0)$
- **while** $i < T$ **do**
  - **for** *each node* $u \in N_{i+1}$ **do**
    - Compute $f(u, i+1)$;
    - **if** $k_u =$ *not_defined* **then**
      - $k_u =$ retrieve_key(u);
  - **for** *each node* $v \in N_i$ **do**
    - **if** $u \in N_{i+1}$ *such that* $k_v = k_u$ **then**
      - $\mathcal{F}(u, i+1) = f(u, i+1) + \mathcal{F}(v, i)$;
      - $\mathcal{P}(G_{i+1}) = \mathcal{P}(G_{i+1}) + \mathcal{F}(u, i+1)$;
  - Copy_(E_i, E_{i+1});
  - $i = i+1$; Copy E_i to E_{i+1};
- **end**

Table 3
The time graph G_{i+1} after merge operation.

E_{i+1}		N_{i+1}
Source	Destination	
v_1	v_2	v_1, k=96
v_1	v_3	v_2, k=97
v_1	u_3	v_3, k=98
v_2	v_3	u_2, k=99
u_2	u_3	u_3, k=100
u_2	v_1	

3.2. Calculating node size

Let V_{n_t} denotes the size of node n at t-th frame. We compute the node size V_{n_t} by taking the ratio of temporal actor metric value to the TIM value and multiply it with the user defined volume $\mathcal{V}$ as follows:

$$V_{n_t} = \frac{\mathcal{F}(n, t)}{\mathcal{P}(G_t)} * \mathcal{V} \tag{3}$$

Multiplying the fraction of temporal actor metric to the TIM value with volume $\mathcal{V}$ provides the user to define a proportion. So, when we define the value of volume $\mathcal{V}$ as factors of the display area, node sizes will be computed relative to this space and thus important nodes always gain larger display areas. The value of the volume $\mathcal{V}$ can be changed by the user through the GUI of HTLM. We demonstrate the idea in Fig. 4.

Eq. (3) implies that independent from node importance, if the temporal actor value of a node decreases or remains static as time progresses, the size of the node may change.

We exemplify this scenario as follows: consider an online customer care service department with ten number of employees. We can decide if a given employee is active or passive according to response criteria i.e. *how frequent does the employee response to the incoming e-mails*. That is, the actor metric $f(n, G_t)$ returns the number of replied e-mails divided by the number of received e-mails for a given employee n at t-th frame. For instance, if $f(n, G_t) = 0.5$ then the actor n replies one e-mail after receiving two e-mails.

We present the actor metric values and the temporal actor metric values of employees in the customer service department in Tables 4 and 5 respectively. Note that nodes corresponding to most active employees at frame t_1 are n_1 and n_5. Check that as time passes one employee becomes more active (n_5) and another employee (n_1) reduces response frequency and becomes a less active.

From Table 4, we see that other employees start to increase the response frequency at least by two fold. Thus due to Eqs. (2) and (3), the size of the node n_1 reduces rapidly (approximately 50%). However as the rate of increment of the eight

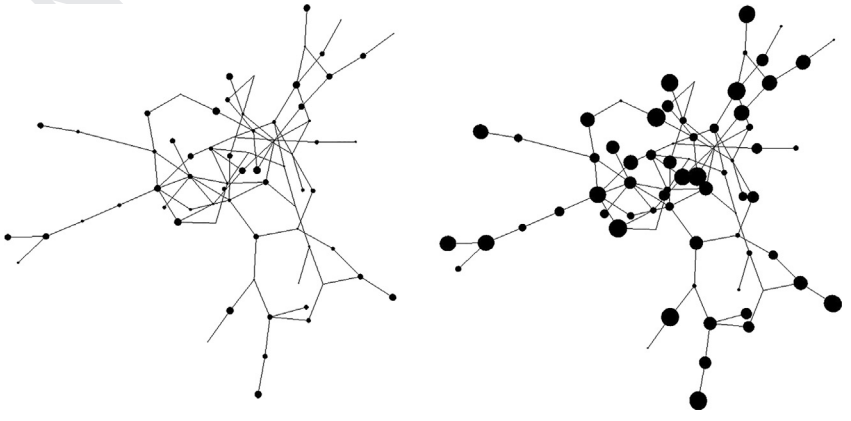

Fig. 4. On the left image ($\mathcal{V}_L$), nodes' sizes are set by their normalized temporal importance values. On the right image ($\mathcal{V}_R$), nodes' sizes are calculated by their normalized temporal importance values proportional to the display size $\mathcal{V}_R = \mathcal{V}_L * 4$. For this example the display area is given as 10 (i.e. $\mathcal{V}_L = 10$) grids. So $\mathcal{V}_R = 40$ grids.

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

Table 4
AM values of the Costumer Care Service Department at frames t_1, t_2, t_3.

Node n	Corresponding node AM value at t-th frame		
	$f(n,t_1)$	$f(n,t_2)$	$f(n,t_3)$
n_0	0.84	0.6	0.45
n_1	0.2	0.44	0.47
n_2	0.29	0.73	0.35
n_3	0.03	0.34	0.64
n_4	0.49	0.68	0.78
n_5	0.19	0.81	0.85
n_6	0.08	0.45	0.39
n_7	0.2	0.78	0
n_8	0	0.13	0
n_9	0	0.34	0

Table 5
TAM values of the Costumer Care Service Department at frames t_1, t_2, t_3.

Node n	Corresponding node TAM value at t-th frame		
	$\mathcal{F}(n,t_1)$	$\mathcal{F}(n,t_2)$	$\mathcal{F}(n,t_3)$
n_0	0.84	1.44	1.89
n_1	0.2	0.64	1.11
n_2	0.29	1.02	1.37
n_3	0.03	0.37	1.01
n_4	0.49	1.17	1.95
n_5	0.19	1	1.85
n_6	0.08	0.53	0.92
n_7	0.2	0.98	0.98
n_8	0	0.13	0.13
n_9	0	0.34	0.34

Table 6
Node sizes of corresponding actors given in the Costumer Care Service Department at frames t_1, t_2, t_3. TIM values are given as $\mathcal{P}(G_1)=2.32$, $\mathcal{P}(G_2)=7.62$, $\mathcal{P}(G_3)=11.55$.

Node n	Corresponding node size at frame t		
	$V_1(n)$	$V_2(n)$	$V_3(n)$
n_0	0.36	0.18	0.17
n_1	0.09	0.08	0.10
n_2	0.13	0.13	0.12
n_3	0.01	0.05	0.09
n_4	0.21	0.15	0.17
n_5	0.08	0.13	0.16
n_6	0.03	0.07	0.08
n_7	0.09	0.12	0.09
n_8	0	0.02	<0.1
n_9	0	0.04	0.03

employees is larger than that of actor n_5, the size of node n_5 reduces slowly (see Table 6).

Clearly, in the visual presentation, the sizes of nodes will change during each time slice (see sizes of nodes given in Table 6). Recently, Ghani et al. reported that, although the size alternation is an important way of conveying information of evolving graphs, it carries a huge risk of disturbing the dynamic stability [30]. In Section 5, we show that, in practice, the node size alternation, caused by Eqs. (1)–(3), has a limited negative effect on the dynamic stability. Moreover, considering the tests and discussions presented in [30], we claim that by this method, we can convey information about how importance of a node evolves.

3.3. Calculating layout

The proposed method computes the layout according to the laws that govern the Hyperbolic Geometry. HTLM computes the Hyperbolic Geometry by using the Klein model, which does not distort primitive shapes such as lines, cubes, and spheres but distorts the angles. Moreover, the Klein model provides interaction mechanisms such as panning and zooming to acquire high- and low-level information while working on a large dataset [31]. Furthermore, as angles are distorted in the Klein model, all the Hyperbolic space curvatures are visible when the space has three dimensions. Before rendering the Hyperbolic Geometry, we project the Hyperbolic space on an Euclidean sphere. Following this, we use a circular layout topology as the base line to adopt the space curvature to represent the entire network.

HTLM adopts the *Historiograph Placement Strategy* [32]. Recall that node ids are assigned when the corresponding social actor starts participating in the social structure. We claim that positioning nodes in ascending order on circular orbits provides user to infer when or how long a social actor participates in the social network. This also provides us to reveal social actors that started to participate at consecutive time slices. Furthermore as the nodes index values are unique, for each time graph, we can calculate a unique position to a node on a circular orbit once. An illustration of this schema is given in Fig. 5.

In order to place A nodes on circular orbits, we first calculate circumferences and the number of the circular orbits. Although there are no optimal values, we propose distributing nodes to $\sqrt{A}$ number of circular orbits each having $\sqrt{A}$ number of nodes to provide a uniform distribution of nodes amongst circular orbits.

We have to select the circumference of an orbit in such a way that each node located on this orbit is close to the surface of the Euclidean Sphere, therefore the radius r must be large enough to prevent collisions. As the Hyperbolic circumference formula, $c=2\pi*\sinh(r)$, exhibits the significant property that circumference grows exponentially with the radius r. Supplying factors of $\sqrt{A}$ as a radius provides enough space area to place nodes on the circular orbits in practice. While calculating the positions, we first need to assign a reference point and a distance d between a circular orbit.

This reference point is the centre of the Hyperbolic space and the distance of the initial circular orbit d is given by the Hyperbolic distance function $d=d_{hyp}(\overline{a},\overline{b})$. We define $d_{hyp}(\overline{a},\overline{b})$ as follows:

$$2*\cosh^{-1}\sqrt{\frac{\langle\overline{a},\overline{b}\rangle_h^2}{\langle\overline{a},\overline{a}\rangle_h*\langle\overline{b},\overline{b}\rangle_h}} \quad (4)$$

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

Table 7
Nomenclature table for symbols used in this section.

Symbol	Description	Range/cardinality
$\mathcal{G}$	A sequence of time graphs	$[0 \ldots T]$
G_t	t-th time graph	$\in \mathcal{G}$
N_t	Set of nodes of time graph G_t	$[0 \ldots \lvert N_t \rvert]$
E_t	Set of edges of time graph G_t	$[0 \ldots \lvert E_t \rvert]$
n_i	Node i	$\in N_t$
$f(n,t)$	Actor metric value of node n in t-th time graph	$\in \mathbb{R}_{\geq 0}$
$\mathcal{F}(n,t)$	Temporal actor metric value of node n accumulated between 0-th and t-th time graphs	$\in \mathbb{R}_{\geq 0}$
$\mathcal{P}(G_t)$	The temporal importance value of time graph G_t	$\in \mathbb{R}_{\geq 0}$
V_{n_t}	Size of the node n at t-th time graph	$\in \mathbb{R}_{\geq 0}$
$\mathcal{V}$	User (defined and controlled) volume	$\in \mathbb{R}_{\geq 0}$
α-Attribute	A set of attributes	Alphanumeric
k_n	The key of node n	$\in \mathbb{Z}_{\geq 0}$
π	An invertible mapping function	$\{\in \mathbb{Z}_{\geq 0}\} \leftrightarrow \{\in \alpha\}$
S	The time span of the underlying social network	s, min, …, year
Gr	The time granularity used in the HTLM method	s, min, …, year

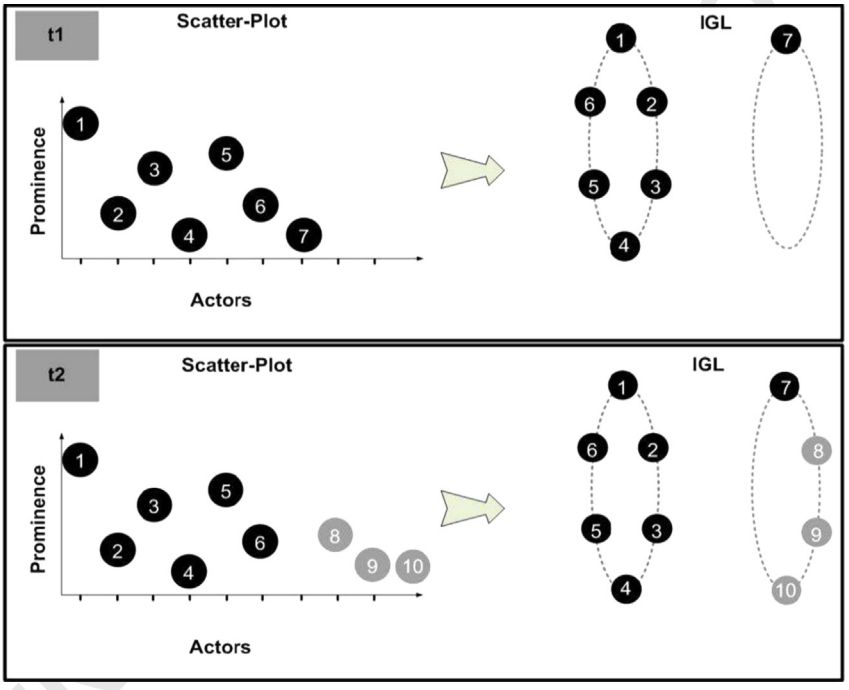

Fig. 5. Scatter plots represent participating actors in the network at time t_1 and t_2. Grey nodes start participating at time t_2. HTLM places these new nodes close to each other and HTLM keeps actor 7 visible until its temporal importance value fades out. Note that node 7 does not participate at time t_2.

here $\langle \cdot, \cdot \rangle_h$ is called as a *Minkowski Inner Product* [31]. The Minkowski Inner Product is defined as follows. Let

$$I^{3,1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{5}$$

Then $\langle \overline{a}, \overline{b} \rangle_h = \overline{a}^T I^{3,1} \overline{b}$. The parameters $\overline{a}$, $\overline{b} \in \mathbb{R}^4$ are 4×1 matrices and intuitively $\overline{a}^T$ is the transpose matrix of $\overline{a}$. As we want to apply Historiographic placement, for d_{init}, we set the values of matrices $\overline{a}$, $\overline{b}$ in such a way that earliest social actors are represented at the surface of the Euclidean Sphere. Therefore we have that $\overline{a} = \langle 1-\varepsilon, 0, 0, 1 \rangle$ and $\overline{b} = \langle 0, 0, 0, 1 \rangle$, where ε is a small real number larger than 0. We set the first parameter of matrix $\overline{a}$ as $1-\varepsilon$ because we are going to project the Hyperbolic space onto an Euclidean sphere of radius 1. We have to consider subtracting ε from 1 because the surface of the Euclidean Sphere will correspond to "infinity" in Hyperbolic Space. At this point we must emphasize that these values are implementation specific, one can try to represent Hyperbolic space on an Euclidean Sphere of different radius values. The important thing is if one perform transformations properly, the value of the radius would not affect the visual representation of the Hyperbolic space [31].

Moreover we have to assign an incremental distance between adjacent orbits. Although there is no restriction,

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

practical evaluations show that the following formula produces satisfying results: $d_{inc} = d_{hyp}(\langle 1/\sqrt{A}, 0, 0, 1\rangle, \langle 0, 0, 0, 1\rangle)$. Having d_{inc}, the distance between adjacent orbits satisfies the requirements when we have $\sqrt{A}$ orbits and we lay one orbit on the origin of the Hyperbolic space.

After having r, $\sqrt{A}$, d, and d_{inc} we use Hyperbolic Spherical angles to place nodes. We direct the reader to Ref. [27] for derivation of these angles. But we have to consider that as we are placing nodes on circular orbits, the distance between an adjacent node on the circle must be computed. Recall that we assign the circumference of a circular orbit as multiples of $\sqrt{A}$ therefore, we use d_{inc} as a distance between any adjacent node on the orbit. Again this value is implementation specific.

The placement method is done as follows: we select the node that has the lowest index value of the current list. Then we place this node on the current circular orbit (first orbit is d away from the centre of the Hyperbolic space) then we select next node and place it d_{inc} away from the previous node at the same circular orbit. When $\sqrt{A}$ nodes are placed on the current circular orbit, we start placing remaining nodes on next circular orbit, which is d_{inc} away from the previous circular orbit as demonstrated in Fig. 6. This routine finishes when all nodes are placed on circular orbits.

Although Historiograph placement strategy provides visual landmarks to specify the duration for which the social actor participates in the social network, it does not provide enough information to the viewer to discriminate nodes according to their prominences. To resolve this, we use the size and the distance between the centre of the Hyperbolic space and the position of a node on the circular orbit as a visual landmark.

This is done by taking the centre of the Hyperbolic space as the location for the *ego* of the network, a node that has the maximum TIM value, and comparing the TIM values between the ego and all other nodes one by one through $(\mathcal{F}(n,t)/\mathcal{F}(ego,t))$. Based on the ratio, we apply Hyperbolic translation formulas given in [31] to place nodes between positions at circular orbits and the centre of the Hyperbolic space. For completeness we give the transition function formula. Let us assume that we translate a node from point defined by matrix $\overline{a}$ to point defined by matrix $\overline{b}$. Then the translation is defined by the reflection $r^{\overline{p}}_{hyp}$ at point $\overline{a}$. The reflection operation is done as follows, we consider an object O in Hyperbolic space and point defined by matrix $\overline{p}$. In order to find the reflection of object O we must multiply its coordinates with the $r^{\overline{p}}_{hyp}$ which is given by

$$r^{\overline{p}}_{hyp} = \frac{I - 2pp^T I^{3,1}}{\langle \overline{p}, \overline{p}\rangle_h} \tag{6}$$

where matrix I defines 4×4 identity matrix. To translate one object from a point defined by matrix $\overline{a}$ to point defined by a matrix $\overline{b}$, we use the translation function $T^{a,b}_{hyp}$ defined as follows:

$$T^{\overline{a},\overline{b}}_{hyp} = r^{m}_{hyp}, r^{\overline{a}}_{hyp} \tag{7}$$

here r^{m}_{hyp} defines the reflection of the middle point between points $\overline{a}$, $\overline{b}$. In Hyperbolic space m is given by

$$m = \overline{a}\sqrt{\langle \overline{b}, \overline{b}\rangle_h \langle \overline{a}, \overline{b}\rangle_h} + \overline{b}\sqrt{\langle \overline{a}, \overline{a}\rangle_h \langle \overline{a}, \overline{b}\rangle_h} \tag{8}$$

Finally, in order to translate object we have to set matrices $\overline{a}$ and $\overline{b}$. Let $\varepsilon = 1 - \varepsilon$ and $\omega = \mathcal{F}(n,t)/\mathcal{F}(ego,t)$ then the first parameter $\overline{a}$ of $T^{a,b}_{hyp}$ is given by the current position of the node, we assign matrix $\overline{b}$ as $b = \langle \varepsilon - \omega, \varepsilon - \omega, \varepsilon - \omega, 1\rangle$.

Since these translations are directly proportional to the ratio between a node and the *ego* of the network, the node is positioned either close to the centre of Hyperbolic space or close to its position on the corresponding circular orbit. Therefore when the viewer starts to visualise the evolution of the network, the viewer observes that nodes fluctuate between their original positions and the centre of the Hyperbolic space. When the social network under consideration has more than one ego nodes then they form a group near the centre of the Hyperbolic space.

To exemplify the general schema let us assume that we have a social structure, which has several actors of low importance values and two actors, actor 2 and 11, having higher importance values respectively. As actor 2 has the highest importance value, and it is the ideal node, and our method places actor 2 at the centre of the Hyperbolic space. Then according to the ratios, our method translates all actors to their circular orbits and translates actor 11 to a position that is close to the centre of the Hyperbolic space as illustrated in Fig. 7.

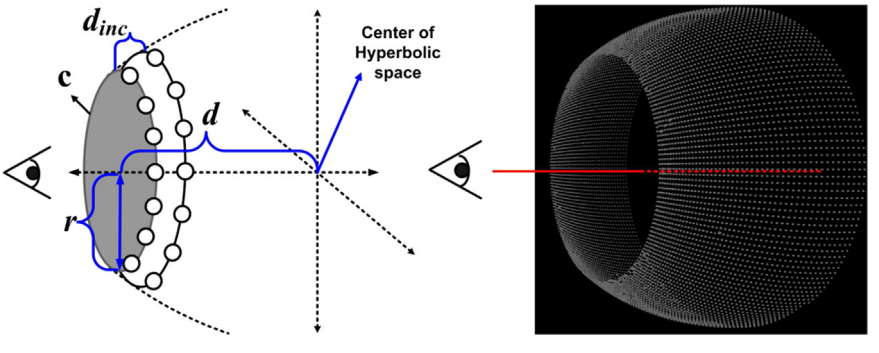

Fig. 6. Topology of the HTLM. Nodes are positioned on circular orbits according to their initial time values. Distance d, circumference c and radius r are calculated to compose required orbits (left). A corresponding representation is depicted on the right image.

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

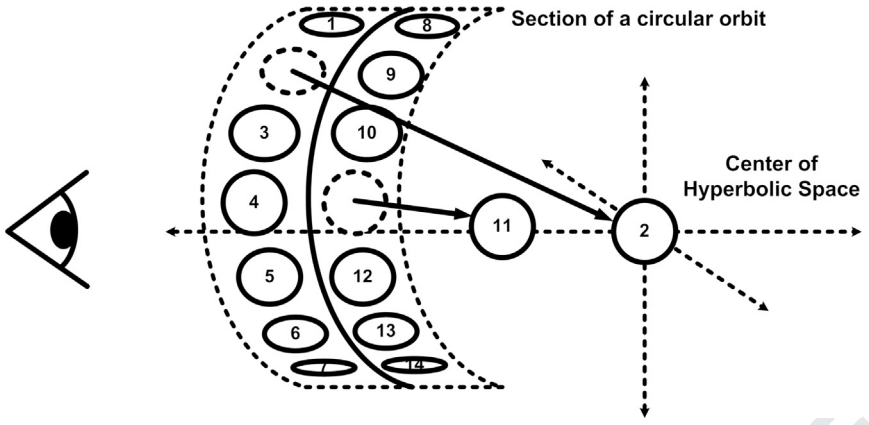

Fig. 7. A social structure with two prominent actors 2 and 11. As actor 2 is the most prominent social actor, it is placed at the centre. Actor 11 is translated close to the centre of the Hyperbolic space due to its high prominence value. Arrows show actors' positions on circular orbits.

3.4. Controlling visual clutter

In Section 3.3 we state that the size of a node n is related to user defined volume $\mathcal{V}$. We claim that any adjustment on the space volume will affect nodes' sizes without modifying the topology of the layout as diagrammed in Fig. 9. This adjustment provides users to filter the network nodes, where the filtering is based on comparison between the temporal importance values. Therefore obscuring is applied on to less important nodes.

To demonstrate the effect of filtering, we applied the *Braths Metric* [33], which is given by the ratio of the number of occluded data items to the number of available data items on four distinct visualisations taken from Enron Corpus [34] and 20 Newsgroup [35] dataset visualisations. The results are given in Table 8. Four visualisations of the Enron dataset are represented in Fig. 8.

Table 8
Clutter reduction based on *Brath's Metrics* (occluded nodes/available nodes) under V (Space Volume).

Dataset	$V/8$	$V/6$	$V/4$	V
Enron	0/418	5/741	409/2045	3403/17,843
20 News	0/179	0/856	48/1210	410/3418

3.5. Discussion

We can summarize the advantages of the proposed layout method as follows:

1. HTLM represents edges between social actors. Besides the analytical properties of social actors are visually encoded by its position and size; therefore both network metrics and communication structures are supplied to the viewer.
2. As we limit the displacement of a node, a node can only move either to the centre of the Hyperbolic space or back to its position on the circular orbit. Therefore, we decrease the visual complexity caused by the extensive motions and alleviate the amount of topological differences between two consecutive images. Moreover we avoid implementing extra smoothing algorithms to enhance dynamic stability.
3. Using circular orientation in Hyperbolic space provides us to generate readable overviews of large temporal social network datasets, minimize disorientation issues, and express the properties of the social structure efficiently.
4. Users have the ability to control the visibility of data items so that they can improve their information gathering performances.

4. Case studies and performance

In this section, we present initial visuals computed by the proposed method. We first present results of Synthetic Datasets which posses different kinds of social structures. Then we present results of real life datasets to evaluate the performance of the proposed method.

4.1. Synthetic datasets

We generated datasets that practice special types of relations such as star and many to some. Each one of the generated datasets has temporal dimensions that contain 100 nodes with 30,000 edges.

Star: This is a relational pattern, in which all members in a social group have only one direct link to a single unique actor. For a star type network the central actor is represented with a large node due to its high temporal importance value and is located at the centre of the Hyperbolic space.

Many to some relations: Many to some relations occur when all members in a social group have direct links with a small number of actors. That is to say there are more than one *ego* in the entire social network. Visuals of many to some network reveal that the method emphasizes the central group by positioning them at the central area of the Hyperbolic space. Other nodes are positioned around this group. Our results are given in Fig. 10.

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

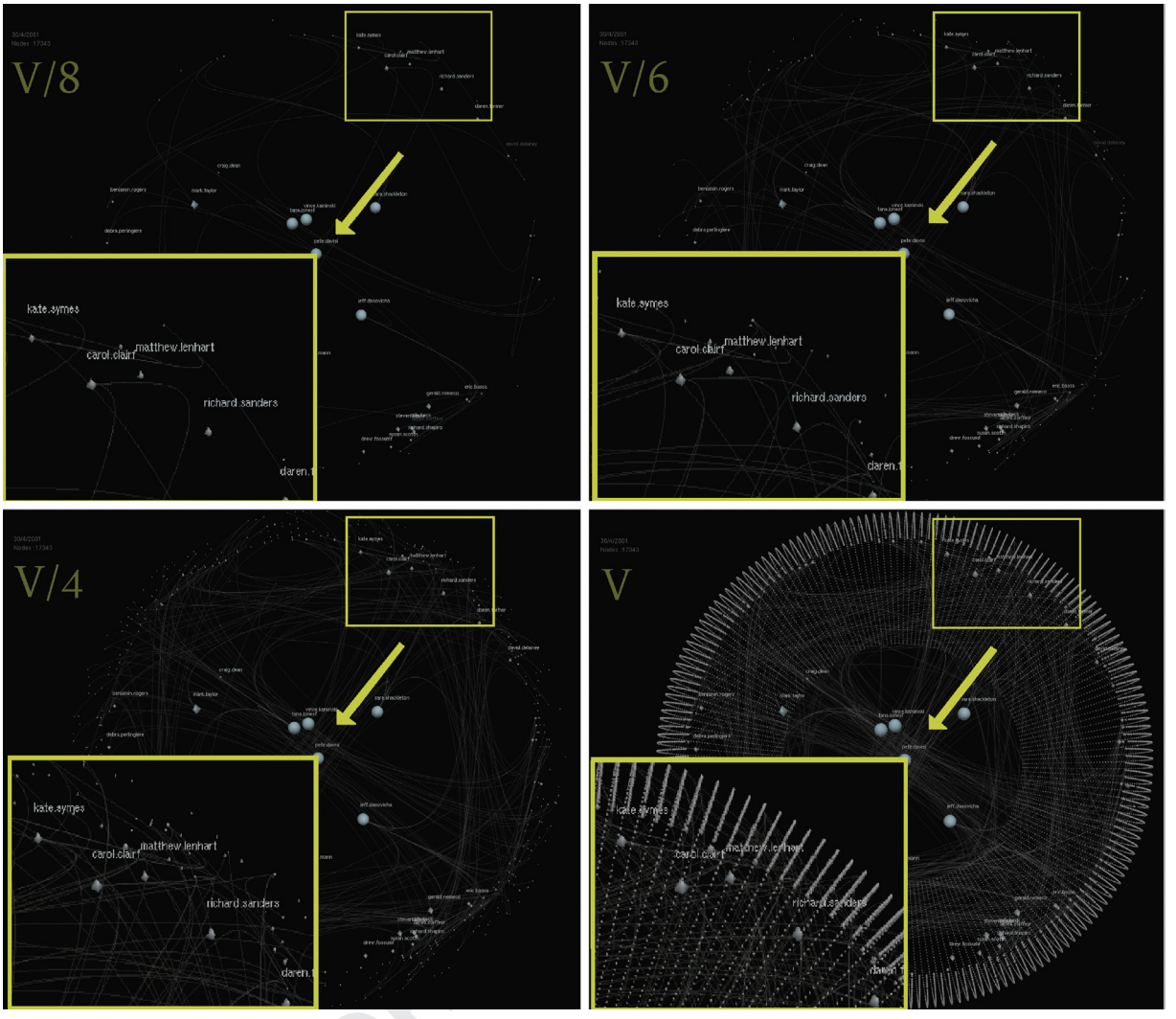

Fig. 8. Demonstration of visual clutter control. The user increases the space volume from *V*/8 to *V* to observe 17,843 nodes in a temporal dataset. The camera remains static during the demonstration.

4.2. Real-life datasets

The primary objective of using real life dataset as case studies is to see the performance of HTLM and to judge if HTLM produce reliable, readable visuals. In order to achieve these goals we collect and preprocess datasets that are used as benchmark datasets in information visualisation spectra. We try to use different types of datasets (large, small), to show how we can use HTLM under such cases. Before going further we want to introduce the social metrics that we have employed during the case studies.

In–Out Degree Centrality: Let n be a social actor and n_{in} be the set of social actors who form a direct relation with actor n then *In-Degree Centrality* D_{in_n} defines the cardinality of the set n_{in}. Likewise let n_{out} be the set of social actors who have been accessed by the social actor n, then *Out-Degree Centrality* D_{out_n} defines the cardinality of the set n_{out}. The degree centrality of an actor is the sum of D_{in_n}, D_{out_n}.

The Closeness Centrality: The Closeness Centrality of a social actor n is given by inverse of the sum of all shortest paths between n and the rest of the social actors. Therefore the higher the closeness centrality, the more the central social actor n.

4.2.1. McFarland classroom dataset

We visualise the McFarland's classroom 33 [36] dataset as the first case study for our analysis. McFarland's classroom is a small temporal social network dataset that belongs to a class and has been used as a benchmark dataset in numerous works [36]. This dataset is formed through direct conversations between individuals in the class. The $f(n)$ is derived using the degree centrality metric.

Comments: From Fig. 11 we see that at the beginning of the visualisation node 14 communicates with all other nodes, and becomes the most important node (T1). Later, node 7 starts to communicate with other nodes and approaches to the centre, and as time passes node 7 becomes the most important node (T2).

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

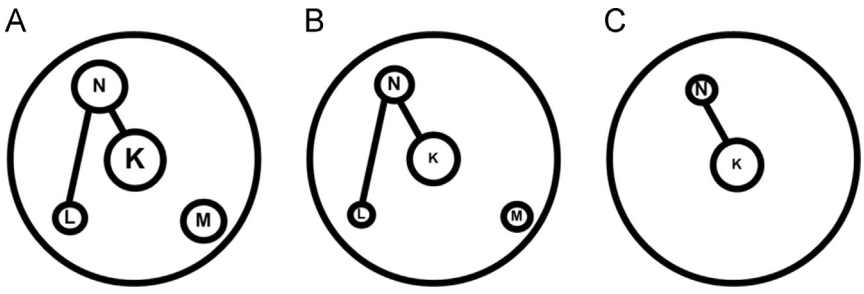

Fig. 9. The space volume is step by step decreased by user at time B and time C. Thus nodes get smaller and less important nodes fade out.

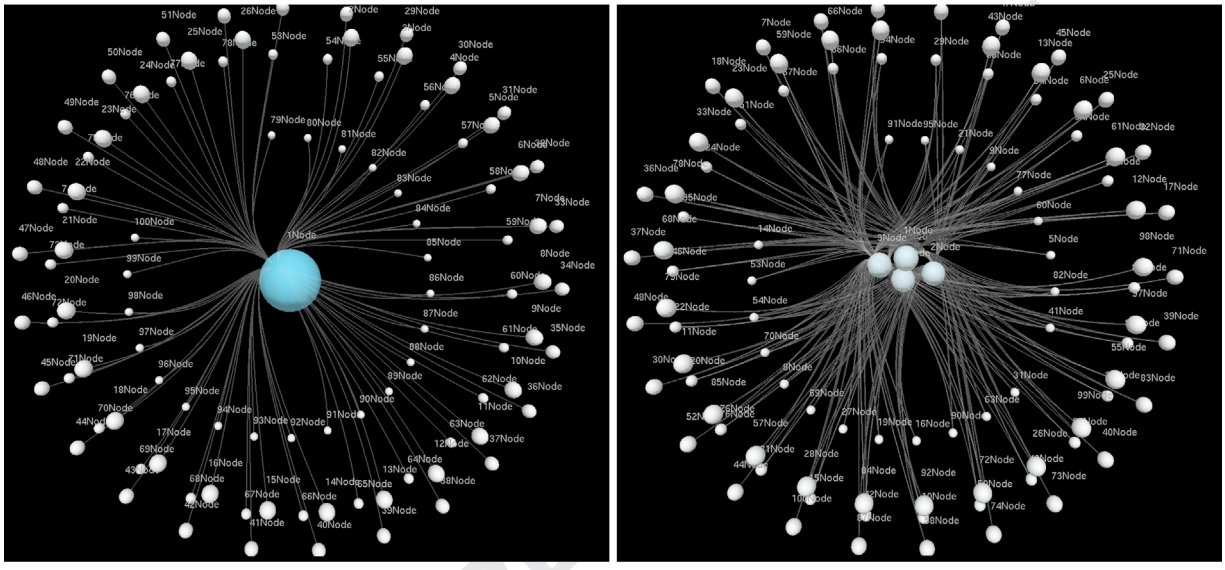

Fig. 10. Synthetic data visualizations. Star type network (left). Many to some relation (right).

Combining the visualisation with the related discussion in [36], one can realise that node 7 is the teacher and node 14 is the guest speaker. In addition, similar to the discussions, we observe that during the class the node 7 issues links to some nodes when these nodes start to communicate with each other, which infers that the teacher tries to collect attentions of students when they become too social or the teacher explains some topic which the group may discuss about. Moreover, we can also observe that two students (node 2 and 19) do not participate in the lecture and students 4 and 12 are the most active students. Finally, as can be seen in Fig. 11 the hierarchy and the centrality of the social structure remain stable throughout the lecture.

4.2.2. Enron corpus

The Enron e-mail dataset, due to the Enron Corporation, is one of the most popular datasets known in social network analysis spectra, the importance of the Enron email dataset is two-fold. (1) It is a dataset belonging to a real-life large social structure. (2) As the organization collapsed after a crisis, social scientists can develop new techniques focused on guessing the wellness of the social structure. Therefore we believe that, in our case, Enron Dataset provides us to observe the weaknesses and strengths of the proposed method. The original data contains 619,449 emails belonging to 158 Enron employees. For detailed investigations of this dataset we refer to [34].

Data Pre-Processing: We obtained 323,078 e-mails belonging to 19,067 different e-mail addresses (social actors) accumulated in four years. We selected a time period between 30 August 2000 and 30 October 2000 to represent the evolution of the Enron structure as the firm approaches to the bankruptcy. The time step is one day and the *degree* and *closeness centrality* analyses of Enron network between 18 and 19 October 2000 in Fig. 12.

Comments: Based on the degree centrality; between 18 and 19 October 2000 the number of prominent social actors remains stable. However according to the closeness centrality analysis the number of prominent actors increases drastically, which reveals a formation of a large number of new communication channels between 18 and 19 October 2000. So we can say that the number of communication channels between social actors increases. Synthesizing this observation we suggest that actors, who

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

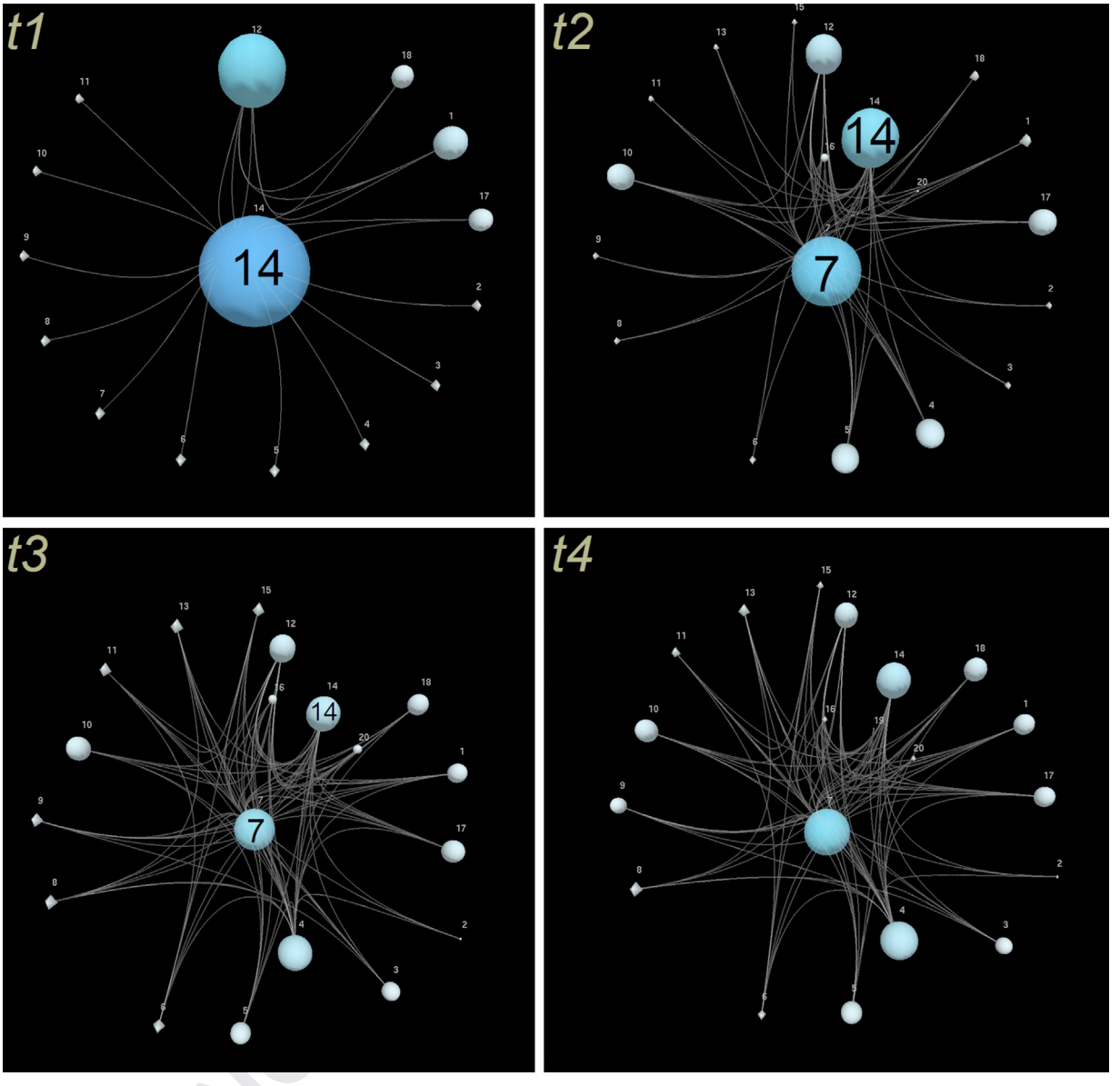

Fig. 11. McFarland classroom no. 33. At time step t1 (image 11.1) image guest speaker (14) is at the centre, as she meets students. Later the lecturer (7) talks (time t2, image 11.2) to students and eventually positioned at the centre and holds his position throughout the course. Student 4 becomes more social and moves towards the centre (time t3, image 11.3). Last figure represents important actors of McFarland's class (time t4, image 11.4).

have small degree central values, start communicating with many actors in the network at 19 October 2000. This result indicates that the hierarchical structure of the Enron firm starts collapsing after Enron employees are aware of the crisis at 19 October 2000.

4.3. Performance

We tested our framework on a single core of Intel Dual Xeon 3.4 GHz with 4 GB of memory. Although we do not optimize our implementation, the runtime performance of the method is satisfying (Table 9). However, depending on the network metric, f the required time to compute layouts can change significantly.

5. Experimental results

In this section we first measure the dynamic stability of our layout. Later we perform a usability study to represent the effectiveness of the proposed method. In addition to McFarland's Class number 33 dataset and Enron Corpus we also use New Comb's fraternity dataset [37], 20 Newsgroup datasets as well. As we wanted to decrease the risk of not being able to detect weaknesses of the HTLM, this variation is inevitable.

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

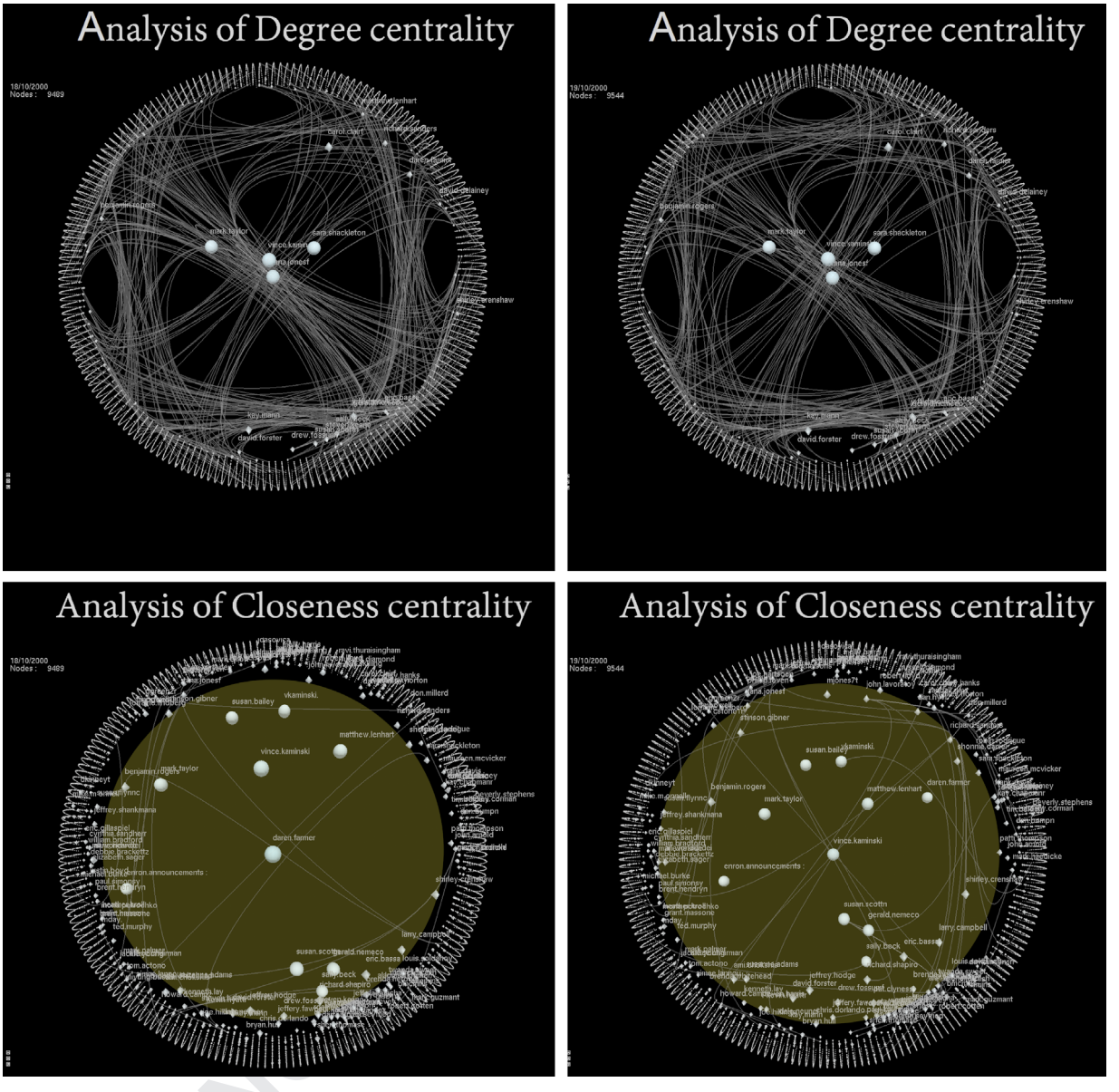

Fig. 12. The degree and closeness centrality analysis of Enron between 18 and 19 October 2000. Between 18 and 19 October the degree centrality analysis does not reveal any abnormal behaviour as the number of prominent actors remains the same. However according to the closeness centrality analysis; the number of prominent actors increases drastically as highlighted in the bottom two images, we infer that Enron employees form a large amount of new communication channels in one day.

Table 9
Time comparison table—the values are averages for the degree centrality. TS (s) is the average time in seconds to compute the layout for a single time slice. Runtime in frame per second is RT (fps).

Dataset	TS (s)	Slices	Node	Edge	RT (fps)
Enron	0.54	1300	19,067	323,078	45
20 News	0.11	90	5417	44,797	50

McFarland's Class number 33 and New Comb's Fraternity datasets represent small networks, where relations are made by direct conversations of social actors. About 20 Newsgroup dataset contains 44,567 e-mails of 4096 group members accumulated in 3 months.

5.1. Dynamic stability

While rendering visualisations of a temporal data in a sequential order, it is essential to minimize topological differences between adjacent images. Rapid change of the underlying layout can cause disorientation and can disturb user's perception [38]. As reported by Bridgeman several metrics can be used to measure the stability of temporal, animated layouts. These metrics are mostly based on comparing the relative positions of identical elements that

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

are represented in adjacent visuals. The interpretation of these metrics is the same: if the difference between subsequent positions of identical elements is low, the underlying visualisation is a stable. To show that our method generates stable visualisations, we apply *Average Distance* and *Orthogonal Ordering* tests [38]. These tests compare the positions and the relative orientations of identical items in adjacent images where the current image is represented by S and the next image is represented by $\overline{S}$. *Average Distance*: this metric calculates the average distance between locations of each node that are represented in adjacent images by the following formula:

$$dist_{S,\overline{S}} = \frac{1}{|S|} \sum_{s_i \in S, \overline{s}_i \in \overline{S}} d(s_i, \overline{s}_i) \tag{9}$$

Because we distribute the dataset in a Hyperbolic space, and then we project the layout into an Euclidean sphere, we use the Euclidean distance function $d(s_i, \overline{s}_i)$. In our calculations the distance function d is normalized by the maximum distance value, which is one (1) for the Hyperbolic manifold. *Orthogonal ordering*: The orthogonal ordering measures the stability of a layout through comparing the orientation of an item that are represented in adjacent images with the following formula:

$$order_{S,\overline{S}} = \frac{1}{W|S|} \sum_{s_i \in S,\ \overline{s}_i \in \overline{S}} \min \int_{\theta}^{\overline{\theta}} weight(\theta)\, d\theta \tag{10}$$

The $weight_\theta$ is the cost function $weight_\theta : \{1^\circ, \ldots, 360^\circ \rightarrow 1, \ldots, 360\}$, and W is the maximum possible angle (360°) between two positions of a node.

We applied these measurements on a reference tool, Sonia, by using Newcomb's fraternity and McFarland's 33 Classroom datasets to illustrate the stability of our layout. We failed to run Enron Corpus and 20 Newsgroup datasets on Sonia, since Sonia does not support such large datasets. Moreover, we do not know any visualisation technique that supports large temporal social network datasets; therefore we use small datasets for comparison. As recommended in [19] we run Sonia's Peer Influence *PI* layout, where each node is positioned according to the averages of its direct relations (peers) and the weight of their edges (influences), to minimize redundant motions of nodes. Moreover, we used a circular layout as the base line for PI.

According to the average distance and the orthogonal ordering metrics, our layout yields better performances on the stability as presented in Table 10. These results suggest that HTLM produces animations for Classroom 33 dataset that are smoother by a factor of 5 than the Sonias P.I. algorithm. However when we consider Newcomb Fraternity Dataset, we see that there is almost no difference between HTLM and P.I. algorithms. These results are expected since Classroom 33 dataset covers a social network in which the rate of fluctuations of the communication channels between actors is high. Therefore this case enforces the P.I. algorithm to rapidly change positions of nodes.

To support our observations, we perform a non-parametric *Kruskal–Vallis Significance* test on HTLM and PI results. For each metric, we construct two vectors such that each vector holds the results retrieved from HTLM or PI. Then we run Kruskal–Vallis difference test on these vectors. Initially the test assumes that two vectors have identical distributions (null Hypothesis), if the *p−value* is near zero, this casts doubt on the null hypothesis and suggests that at least one sample median is significantly different from the other. Table 11 suggests that for each metric distributions are statistically significant. This result supports the results given in Table 10.

Table 10
Comparison on dynamic stability—values are means for HTLM and peer influence methods (lower values are better).

Dataset	Average distance		Orthogonal ordering	
	HTLM	P.I.	HTLM	P.I.
NewComb	0.3	0.32	0.45	0.48
$Classroom_{33}$	0.014	0.275	0.13	0.388

5.2. Usability study

The usability study was designed to test our hypothesis that HTLM conveys both relational and structural information. The subjects were asked to perform three basic analysis tasks. It is important to highlight that we could not replicate these tasks on any other well known and accessible visualisation tool due to the large number of nodes (over 5000) involved.

The experiments are conducted with 10 participants, all computer science graduate students. All participants had either normal or corrected-to-normal vision. No participant was colour-blind. The only requirement was fundamental computer skills.

5.2.1. Experiment design

The tasks were designed to represent potential queries that a user looks for while working on the Enron Corpus (Table 12). The objective of the first task (T1) is to find a clique of size six amongst 19,000 nodes. Note that this task is known to be *NP-Complete* problem [39] can be performed by a force-directed method when the size of the dataset is small (nodes ≤ 200) [25]. The objective of the second task (T2) is to find the number of actors whose out-degree centralities fluctuate rapidly; as a result, their prominences change.

To our knowledge, in information visualisation domain, there is no visualisation tool that can convey this information for a large temporal social network dataset without requiring a domain expert to analyse the dataset or using extra visual attributes such as labels and side bars.

The usability study was carried out as follows: the participant was seated in front of the screen with

Table 11
Comparison on dynamic stability—*p*-values for *Kruskal–Vallis Significance Test* for the results of HTLM and PI (no significant difference if $p > 0.1$).

Dataset	Average distance	Orthogonal ordering
NewComb	0.03	0.04
$Classroom_{33}$	0.0014	0.0077

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

Table 12
The tasks for usability study.

Task ID	Task definition
Task 0 (warm up)	*Count the actors that have frequent prominence fluctuations in between* 21 *February* 2000 *and* 1 *November* 2000
Task 1	*Identify a clique of size 6 at* 10 *April* 2001
Task 2	*Count the actors whose out-degree centralities fluctuate rapidly in between* 10 *October* 2000 *and* 20 *October* 2000

a keyboard and a mouse. A controller has a mirrored screen, a mouse and a keyboard. After each task, the controller would power down the participant's screen, set up the next task, and check with the participant to see if he/she was ready for the next task and then power up the participant's display.

The experiment starts with a warm-up session during which with the controller briefly describes the objective of the experiment and the nature of the tasks. After the controller's introduction, the warm-up session ends when the participant finalizes the Task 0.

Evaluation tasks start only after the end of the warm up session and training task. The participants are not made aware of the distinction between training and evaluation tasks because during the initial tests it has been observed that revealing this loosens the participants' attention and prevents effective acquisition of information about the experiment. To conceal the training tasks as evaluation tasks, following the warm-up session the users are given three cards each of which displays the task number on the front area and the task description in the back.

The time that passes from the moment the participants are allowed to see and interact with the tool to the moment they provide the answer is recorded. The tasks are query tasks that designed such that when animations begin, the participant may or may not find the correct answer. We measure the time that it takes to complete the task and the accuracy.

5.2.2. Results and discussion

We examined the Task 1 and Task 2 separately. In the first task (T1) we only consider the task completion time due to the nature of this task. Participants mostly spend less than 2 min to reveal a clique of size six amongst 19,000 nodes and 237,000 edges.

The participants accomplished the task (T2) of finding social actors whose out-degree centralities fluctuate rapidly in between 10 October 2000 and 20 October 2000 with 77% accuracy. Considering the fact that participants have no experience in the field, and they have finished the task in less than 1 min, these results are promising.

We can derive two important results from the conducted usability study: first we can conclude that an inexperienced user can quickly perform high-level analysis regarding the structural properties of actors in a large temporal social network dataset accurately. Second, considering the accuracy of the tests we can say that HTLM conveys both relational and structural information for a given large temporal social network dataset.

We observed that two participants struggled with different parts of the HTLM (i.e. the positions of GUI buttons, the shapes of the nodes, colour scheme of HTLM, etc.). Therefore we are not planning to address these issues. On the other hand, we observe that in task two (T2) 33% of the participants fail to grasp same actors (four actors) due to the visual clutter. They agree that even if they use the visual clutter reduction mechanism, they had difficulties on detecting those actors. Hopefully, when they retake the task two they were able to give correct answer. However visual clutter gets unavoidable as the number of nodes and edges increases. Therefore we conclude that 33% failure is normal and indicates that in some cases HTLM fails to prevent the visual clutter.

Another issue was the time granularity. About 40% of the participants were not be able to decide which time granularity is the best for viewing such dataset. Recall that Enron Corpus is an e-mail dataset and e-mails are sent in any time of the day. Since the time granularity affects the density of information in the visualisation, we take this issue seriously. One idea is to reduce time granularity. At first, this idea seems to solve the problem, however it is impractical to do. To see this consider that when we set the time granularity to be one day, we have to produce 968 time graphs for Enron dataset. If we want to reduce time granularity to 1 h we will eventually have more than 20,000 time graphs. Considering that each graph holds approximately 10,000 nodes on the average, we realised that in most of the cases we cannot reduce the time granularity as we wish.

To sum up, usability studies indicate that HTLM provides users to analyse large temporal social network datasets. Moreover these studies reveal that visual clutter and scalability are weaknesses of the HTLM. In future these problems can be solved by distributing the work load amongst Central Processing and Graphical Processing Units.

6. Conclusion

We have presented a new Hyperbolic temporal layout method for large sparse temporal social network datasets. Our novel method calculates temporal node link layouts using network metrics.

We have provided a detailed analysis to demonstrate the efficiency of the proposed method: first, we have presented that our layout algorithm is able to represent basic social structures. Second, we have applied the HTLM method on Enron e-mail dataset and have performed structural analysis as well as identified important actors of the Enron firm. Our structural analysis showed that while the number of prominent actors decreases, its hierarchical structure was collapsing as the firm approaches to the crisis. Third, we have demonstrated that the proposed dynamic visualisation method enhances dynamic stability. Fourth, we have conducted a usability study and showed that novice users can perform high-level visual analysis with little effort. The proposed method draws actors with high temporal importance near to the centre of Hyperbolic space. As a result visualisations

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008

may get cluttered if the data has more than ten cliques. Our method does not use relational information to layout a network, hence representing edges without any modification results in over-crowded images. In this paper, we developed a simple topology-based-edge-clustering method to tackle this challenge. Other edge-clustering algorithms such as edge bundling [40] and geometry-based-edge-clustering methods [41] can be easily applied to enhance visual quality, clarity as well as information retrieval performance. Conducted usability study reveals some weaknesses of the proposed method, as a future work, we will generate a series of synthetic datasets (possibly each dataset is designed for a special task) and perform another usability study with a large number of participants to reveal the strengths and the weaknesses of the proposed approach.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org.10.1016/j.jvlc.2013.10.008.

References

[1] Yaniv Frishman, Ayellet Tal, Online dynamic graph drawing, IEEE Trans. Vis. Comput. Graph. 14 (July (4)) (2008) 727–740.
[2] Stephen G. Kobourov, Kevin Wampler, Non-Euclidean spring embedders, IEEE Trans. Vis. Comput. Graph. 11 (November (6)) (2005) 757–767.
[3] N. Henry, J.-D. Fekete, M.J. McGuffin, Nodetrix: a hybrid visualization of social networks, IEEE Trans. Vis. Comput. Graph. 13 (November–December (6)) (2007) 1302–1309.
[4] S.P. Borgatti, Net Draw: Network Visualization Softwares, Analytic Technologies, Harvard, 2002.
[5] K.M. Karley, J. Reminga. ORA: Organization Risk Analyzer.
[6] Wolfgang Aigner, Silvia Miksch, Wolfgang Mller, Heidrun Schumann, Christian Tominski, Visualizing time oriented data a systematic view, Comput. Graph. 31 (2007) 401–409.
[7] Katherine Faust, Stanley Wasserman, in: Social Network Analysis: Methods and Applications, Cambridge University Press, 1994, p. 857.
[8] Nathalie Henry, Jean-Daniel Fekete, Matlink: enhanced matrix visualization for analyzing social networks, in: Proceedings of the 11th IFIP TC 13 International Conference on Human–Computer Interaction, Volume Part I, INTERACT'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 288–302.
[9] V. Batagelj, W. de Nooy, A. Mrvar, Exploratory Social Network Analysis with Pajek, Structural Analysis in the Social Sciences, Cambridge University Press, 2005.
[10] S. Borgatti, in: Ucinet User's Guide, Analytic Technologies, 1999.
[11] B. Lee, C.S. Parr, C. Plaisant, B.B. Bederson, V.D. Veksler, W.D. Gray, C. Kotfila, Treeplus: interactive exploration of networks with enhanced tree layouts, IEEE Trans. Vis. Comput. Graph. 12 (November–December (6)) (2006) 1414–1426.
[12] Orland Hoeber, Xue Dong Yang, Exploring web search results using coordinated views, in: Proceedings of the 4th International Conference on Coordinated & Multiple Views in Exploratory Visualization, CMV '06, IEEE Computer Society, Washington, DC, USA, 2006, pp. 3–13.
[13] Tom A.B. Snijders, Gerhard G. Van De Bunt, Christian E.G. Steglich, Introduction to Stochastic Actor-Based Models for Network Dynamics, Social Networks, 2009.
[14] Visualization of social media data: mapping changing social networks (Masters Theses), Enschede, the Netherlands.
[15] Stephen G. Kobourov, Pawel Gajer, Grip: graph drawing with intelligent placement, J. Graph Algorithms Appl. 6 (3) (2002) 203–224.
[16] G. Kumar, M. Garland, Visual exploration of complex time-varying graphs, IEEE Trans. Vis. Comput. Graph. 12 (September–October (5)) (2006) 805–812.
[17] PieSpy—Inferring and Visualizing Social Network on IRC, ⟨http://www.jibble.org/piespy/⟩.
[18] Peter A. Gloor, Rob Laubacher, Yan Zhao, Scott B.C. Dynes, Temporal visualization and analysis of social networks, in: NAACSOS Conference, Pittsburgh, PA, North American Association for Computational Social and Organizational Science, June 27–29, 2004.
[19] Dynamic network visualization: methods for meaning with longitudinal network movies, Am. J. Sociol. (AJS) (2004).
[20] Lei Shi, Chen Wang, Zhen Wen, Dynamic network visualization in 1.5d, in: Pacific Visualization Symposium (PacificVis), 2011 IEEE, March 2011, pp. 179–186.
[21] Paul Craig, Kennedy Jessie, Andrew Cumming, Animated interval scatter-plot views for the exploratory analysis of large-scale microarray time-course data, Inf. Vis. 4 (September (3)) (2005) 149–163.
[22] Lois Moreno, Who Shall Survive, Beacon, 1953.
[23] Gapminder, ⟨http://www.gapminder.org⟩, retrieved 3.7.2008.
[24] Tatiana Tekusova, Jorn Kohlhammer, Applying animation to the visual analysis of financial time-dependent data, in: Proceedings of the 11th International Conference Information Visualization, IV '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 101–108.
[25] N. Osawa, A multiple-focus graph browsing technique using heat models and force-directed layout, in: Proceedings of the 5th International Conference on Information Visualisation, 2001, pp. 277–283.
[26] Emden R. Gansner, Yehuda Koren, Stephen C. North, Topological fisheye views for visualizing large graphs, IEEE Trans. Vis. Comput. Graph. 11 (July (4)) (2005) 457–468.
[27] T. Munzner, H3: laying out large directed graphs in 3d hyperbolic space, in: Proceedings of the IEEE Symposium on Information Visualization, October 1997, pp. 2–10.
[28] Helge Ritter, Self-organizing maps on non-Euclidean spaces, in: Kohonen Maps, Elsevier, 1999, pp. 97–108.
[29] Peter A. Gloor, Capturing team dynamics through temporal social surfaces, in: Proceedings of the 9th International Conference on Information Visualization, 2005, pp. 6–8.
[30] S. Ghani, N. Elmqvist, J.S. Yi, Perception of animated node-link diagrams for dynamic graphs, Comput. Graph. Forum 31 (3 (Pt 3)) (2012) 1205–1214.
[31] Mark Phillips, Charlie Gunn, Visualizing hyperbolic space: unusual uses of 4×4 matrices, in: Proceedings of the 1992 Symposium on Interactive 3D Graphics, I3D '92, ACM, New York, NY, USA, 1992, pp. 209–214.
[32] E. Garfield, Historiographic Mapping of Knowledge Domains Literature, vol. 30, 2004, pp. 119–145.
[33] R. Brath, Metrics for effective information visualization, in: Proceedings of the IEEE Symposium on Information Visualization, October 1997, pp. 108–111.
[34] Enron e-Mail Dataset. ⟨http://www-2.cs.cmu.edu/enron/enron_mail_030204.tar.gz⟩, retrieved 5.2.2007.
[35] 20 Newsgroups Dataset. ⟨http://www.qwone.com/~jason/20Newsgroups/⟩, retrieved 5.8.2008, accessed in 7/11/2012.
[36] McFarland Data Classroom 33. ⟨http://www.stanford.edu/group/sonia/examples/McFarlaClass.html⟩, retrieved 5.8.2008.
[37] New Fraternity Dataset, ⟨http://www.casos.cs.cmu.edu/computational_tools/datasets/external/newfrat/index2.html⟩, retrieved 5.8.2008, accessed in 7/11/2012.
[38] Stina Bridgeman, Roberto Tamassia, Difference metrics for interactive orthogonal graph drawing algorithms, J. Graph Algorithms Appl. (1998) 57–71.
[39] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Company, New York, 1979.
[40] Danny Holten, Hierarchical edge bundles: visualization of adjacency relations in hierarchical data, IEEE Trans. Vis. Comput. Graph. 12 (September (5)) (2006) 741–748.
[41] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, Xiaoming Li, Geometry-based edge clustering for graph visualization, IEEE Trans. Vis. Comput. Graph. 14 (November–December (6)) (2008) 1277–1284.

Please cite this article as: U. Cengiz Turker, S. Balcisoy, A visualisation technique for large temporal social network datasets in Hyperbolic space, Journal of Visual Languages and Computing (2013), http://dx.doi.org/10.1016/j.jvlc.2013.10.008