Laplacian based Dynamic Graph Drawing

Limei Che*
Baidu, Inc

Xiaoru Yuan†
Peking University

Lun Zhang‡
Baidu, Inc

ABSTRACT

In this paper, we propose a new strategy for drawing dynamic graphs. We developed an algorithm based on Laplacian constrained distance embedding to maintain the topological information of the previous graph layout. It's a online algorithm which can preserve user's mental map very well by keeping the dynamic graph layout consistent stable. Finally, our experimental results suggest this new approach to be very effective.

Index Terms: dynamic graph, graph layout algorithm, Laplican matrix, force directed layout, stress model

1 INTRODUCTION

In many applications, graphs are widely used to represent social connections, physical networks, or other relationships. Most of these graphs are changing over time, for example in a social connection, some new people add in, some old ones leave. To reflect the evolution of the organization or system represented by the graph, we can keep a track of snapshots once the graph changed. This kind of sequence graph layout is called dynamic graph layout problem.

Traditional Graph drawing algorithms are well studied since early year, layout methods, such as force-directed or stress model layout algorithms are all concentrate on static graph. They produce a totally new layout each time we run the algorithm. Since the dynamic graph sequence is a evolution of a graph, there must be some parts not changed, which we hope their layout are also not changed. Therefore in dynamic graph drawing, the challenge is that we have to layout a current graph aesthetically good while preserving the "mental map" of previous time step.

Mental map was first proposed as a concept in [4], it means a abstract structural information a user forms by looking at the layout of a graph. The mental map facilitates memory of a graph or comparison between it and other graph layouts. Preserving the mental map helps users to quickly recognize the new graph, which is important in dynamic graph drawing.

2 RELATED WORKS

There are algorithms solving the off line dynamic graph drawing, where the whole graph sequence is known in advance. Early work DynaDag [9] generated a heuristic method for incremental layout of directed acyclic graphs drawn as hierarchies. [3] proposed a general context preserving algorithm, in which for each graph, it's layout should have very limited difference between its previous and next time step graph. [6] developed a stratification hierarchical layout algorithm, which not only speed up the general force directed algorithm but also accommodates time-varying graphs.

There are also few works developed online dynamic graph drawing algorithms, where the whole graph sequence is not known ahead. [2] introduced random field models for graph layout which based on Bayesian decision theory.

*e-mail: chelimei@baidu.com
†e-mail:xiaoru.yuan@pku.edu.cn
‡e-mail:zhanglun@baidu.com

There are some existing algorithms solving the mental map problem, such as [8] maintaining the orthogonal ordering, or adding constrains to the graph [1] [5].

3 LAPLACIAN BASED DYNAMIC GRAPH DRAWING ALGORITHM

In this section we describe our algorithm in detail. Given a sequence of online graphs $G_0, ..., G_n$, where $G_0 = (V_0, E_0), ..., G_n = (V_n, E_n)$, the goal of our algorithm is to produce a sequence of layouts $L_0, ..., L_n$, where L_i is a straight line drawing of G_i. We first describe how to compute the online dynamic layout $L_i, i \geq 1$, when given L_{i-1} and G_i. Then, we discuss the way to compute initial layout L_0.

3.1 Laplacian based dynamic graph layout algorithm

In the dynamic graph sequence, for each graph $G_i = (V_i, E_i)$, each node has a unique id 'UId'. Through this unique id, we can recognize the same nodes in different time step. The following are the basic steps of our algorithm:

1. Calculate initial node positions from previous graph layout L_{i-1}.

2. Run force-directed layout algorithm.

3. Run Laplacian Constrained Distance Embedding algorithm to refine the mental map.

Step 1: For each node v in G_i, which is existed in G_{i-1}, copy the coordinates from L_{i-1} to this node; for those new added nodes, calculate their coordinates according to their neighbors as follows: if node v has more than one positioned neighbors, then this node is assigned the barycenter of all its positioned neighbours; if node v has only one positioned neighbour p, select a random position on the circle which take p as center of this circle, and the average edge length 'avg$_e dge_l ength' of \mathrm{L}_{i-1}$ as radius; if node v has no positioned neighbour, then it can be placed randomly.

Step 2: Run force-directed layout algorithm starting from the positions assigned in step 1. Here we use the simulated annealing force-directed layout algorithm [**?**] to adjust the layout L_i. For different nodes we set different cooling parameters. In order to keep the existed nodes as stable as possible, we have to frozen these nodes which copy coordinates from L_{i-1}, only small movements allowed. For those nodes that are totally random positioned, we should let them move far away enough to find their best positions, so the cooling parameter should be high. In our implementation, scores of 0.25, 0.5, 0.75 and 1 are assigned to nodes positioned copied from L_{i-1}, at the barycenter of two or more neighbors, according to one neighbor, and totally random ones, respectively.

Step 3: Although we have preserved user mental map by setting a small parameter of cooling step, but it's still far from stable. The nodes both existed in graph G_{i-1} and G_i move a lot by the forces from new added and disappeared nodes and edges. Inspired by the Laplacian constrained graph layout algorithm [**?**], we can take the subgraph consist of nodes that are copied from L_{i-1} as a user input subgraph, then run the Laplacian constrained distance embedding algorithm.

Table 1: Vis Paper Acceptance Rate

Year	Submitted	Accepted	Accepted (%)
1994	91	41	45.1
1995	102	41	40.2
1996	101	43	42.6
1997	117	44	37.6
1998	118	50	42.4
1999	129	47	36.4
2000	151	52	34.4
2001	152	51	33.6
2002	172	58	33.7
2003	192	63	32.8
2004	167	46	27.6
2005	268	88	32.8
2006	228	63	27.6

3.2 Initial layout of dynamic graph

4 Experiment

5 Conclusion

6 Exposition

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

$$\sum_{j=1}^{z} j = \frac{z(z+1)}{2} \tag{1}$$

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem

rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Figure 1: Sample illustration.

6.1 Mezcal Head

Duis autem [?] vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat[1].

6.1.1 Ejector Seat Reservation

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat [?]. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

[1] Footnotes appear at the bottom of the column

Rejected Ejector Seat Reservation Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie

7 Conclusion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Acknowledgements

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

References

[1] K. Bőhringer and F. Paulisch. Using constraints to achieve stability in automatic graph layout algorithms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 43–51, 1990.

[2] C. Brandes and D. Wagner. A bayesian paradigm for dynamic graph layout. In *The 5th International Symposium on Graph Drawing*, GD '97, pages 236–247, 1997.

[3] S. Diehl and C. Görg. Graphs, they are changing. In *the 10th International Symposium on Graph Drawing*, GD '02, pages 23–30, 2002.

[4] P. Eades, W. Lai, K. Misue, and S. K. Preserving the mental map of a diagram. In *Proceedings of Compugraphics*, volume 91, pages 24–33, 1991.

[5] W. He and K. Marriott. Constrained graph layout. *Constraints*, 3:289–314, 1998.

[6] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12:805–812, 2006.

[7] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.

[8] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Jurnal of Visual Languages and Computing*, 6:183–210, 1995.

[9] S. C. North. Incremental layout in dynadag. In *In Proceedings of the 4th Symposium on Graph Drawing (GD)*, pages 409–418, 1996.