Social Network Visualization Oriented Multi-level Layout Method

Xiaolin Du¹, Yunming Ye¹, Yueping Li², and Xiaomeng Wang¹

¹ Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School, Harbin Institute of Technology, China
² ShenZhen Polytechnic, Shenzhen, China
duxiaolinhitsz@gmail.com, yeyunming@hit.edu.cn
leeyueping@gmail.com, lanmengmen@126.com

Abstract. We describe a fast multi-level layout for visualizing social networks, which can visualize social networks high quality and rapidly. There are two innovations in our fast multi-level layout. Firstly, we proposed a new graph multi-layered compression method based on random walk. The multi-layered compression process groups vertices to form “planet” systems and then abstract these “planet” systems as new vertices to define a new graph and is repeated until the graph size falls below some threshold. And we also proposed a new single level force-directed layout based on sampling. The multi-level layout process can be accelerated based on these two innovations. Finally, we have evaluated our layout on several well-known data sets. The experimental results show that our layout outperforms the state-of-the-art method.

Keywords: graph compression, random walk, multi-level layout, sampling

1 Introduction

Social networks appear everywhere in our modern lives, such as twitter, micro-blog, MSN, Facebook, co-citation relation, credit network, etc. The modern science of networks has brought significant advances in our understanding of complex systems [1]. In research, visualization techniques are always employed to illustrate social networks to users and assist social networks analysis. Social networks are usually represented by different types of graphs. Vertices represent entities, and edges represent interactions between pairs of entities. Graph visualization helps users to gain insight into social networks by turning the network elements and their internal relationships into graphs. There have been many graph layout algorithms designed for graph visualization. Each layout algorithm has its own characteristic and pertinence to different types of graph and different applications.

A graph \( G = (V_G, E_G) \) is an abstract structure that is used to model a relation \( E_G \) over a set \( V_G \) of entities. Graph drawing is a conventional tool for the visualization of relational information, and its usefulness depends on its readability, that is, the capability of conveying the meaning of the diagram quickly and clearly. In recent
years, many algorithms for drawing graphs automatically were proposed (the state of
the art is surveyed comprehensively in [2], [3]).

The rest of paper is organized as follows: Section 2 reviews several areas of related
work; Section 3 introduces the detailed description of our fast multi-level graph
layout algorithm; Section 4 will evaluate the proposed layout through some
comparable study; finally, the paper concludes in Section 5 with a review and
discussion of future work.

2 Related Works

2.1 Graph Layouts

Graph visualization helps users to gain insight into data by turning the data elements
and their internal relationships into graphs [5]. Graph layout problems are a particular
class of combinatorial optimization problems whose goal is to find a linear layout of
an input graph in such a way that a certain objective function is optimized [6]. Given
a general graph consisting of vertices and edges, graph layout is a problem of drawing
the graph. Vertices are assigned coordinates, and if two vertices share an edge it is
drawn between them as curves. The popular graph layouts include Node-link layout
[7], [8], [9], [10], Space filling layout [11], [12], Matrix Layout [13], [14], [15] and so
on. Node-link layout is one of the most used graph layouts, which uses links between
vertices to indicate the relationships of vertices. As one of the well-known Node-link
layouts for drawing general graphs, spring layout is proposed by Eades [16] in 1984.
Since then, his method is revisited and improved [17], [18], [19], [20], [21] in
different ways. There are mainly two kinds of space filling layout: space division
layout and space nested layout. In space division layouts, the parent-child relationship
is indicated by attaching child vertices to the parent vertices. Since the parent-child
and sibling relationships are both expressed by adjacency. Space nested layouts, such
as Treemaps [22], draw the hierarchical structure in the nested way. They place child
vertices within their parent vertices. Matrix Layout is an alternative approach to graph
visualization which is using matrix-based representations. Graphs can be presented by
their connectivity matrixes. Each row and each column corresponds to a vertex. The
glyph at the interaction encodes the edge from corresponding vertex.

2.2 Multi-level Layouts

Multilevel layouts are largely used in graph visualization as multilevel graph drawing
methods can accelerate run time and also improve the visual quality of graph drawing
algorithms. Chris Walshaw [23] presents a multilevel optimization of the
Fruchterman’s and Reingold’s spring embedder algorithm. The GRIP algorithm [24]
coarsens a graph by applying a filtration to the vertices. This filtration is based on
shortest path distance. Fast Multiple Multilevel Method (FMM) [25] is also a force-
directed layout algorithm. \( FM^3 \) [25] is proved subquadratic (more precisely in \( O(N \log N + E) \) in time, contrary to previous algorithms. Work in [26] is based on the detection of topological structures in graphs. This algorithm encodes each topological structure by a meta-node to construct a hierarchical graph.

2.3 Our Proposed Method

The proposed fast multi-level layout combines force-directed layout method, graph partition method and graph compression method. More specifically, the fast multi-level layout proposed a graph multi-layered compression method based on random walk and the layout process groups vertices to form “planet” systems and then abstract these “planet” systems as new vertices to define a new graph and is repeated until the graph size falls below some threshold. In each level graph, a new force-directed method based on sampling is employed to assign vertex coordinates. The characteristic of our proposed multi-level layout is visualizing social networks high quality and quickly.

3 Fast Multi-level Layout

3.1 Multi-level Compression

The multi-layered compression in \( FM^3 \) [25] is a very complex process. There, solar systems are created, which consist of vertices at a distance of two edges or less from the center of the solar system. It needs more spaces to store the paths between compression units and more complexity to calculate the initial positions of vertices. Walshaw [23] algorithm gives a more global quality to the force-directed placement through compressing vertex with one of its neighbors, but it may lead too many levels, which would complicate the algorithm. Thus, in this paper our compress strategy mixes the advantages of these two classical algorithms. We next detailed introduce the compression process of FML algorithm.

In FML algorithm, we create a “Planet System” on graphs. Some vertices are chosen as “planet” vertices, and then the neighbors of planet vertices are the matching “moon” vertices of the “planet”. A planet and its matching moons construct a planet system. Fig. 1 shows an original graph with 17 vertices and 16 edges. Our compress strategy is that we compute the weights of all vertices by Random Walk algorithm, and then compress the vertices with less weight and their uncompressed moons in the graph.

A random walk of length \( k \) on a graph \( G \) is a stochastic process with random variables \( W_0, W_1, W_2, \ldots, W_k \) such that \( W_0 = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) \) and \( W_{i+1} \) is a vertex chosen
uniformly at random from the neighbors of \( w_i \). Let \( B \) be the column-normalized adjacency matrix of the graph \( G \).

\[
W_{k+1} = BW_k
\]  

(1)

By random walk on graph, we get the weights of all vertices. Fig. 1 shows an example graph \( G_E \) with 17 vertices and 16 edges and Table 1 shows the vertex weights after running random walk on example graph \( G_E \).

![Example graph](image1.png)

**Fig. 1.** Example graph.

According to the weights in Table 1, we first compress the vertex with minimum weight and its “moons”. Then, if there still are vertices uncompressed, we continue choose the rest minimum weighted vertex to compress until all vertices are compressed. Fig. 2 demonstrates the compress process. Fig. 2 (a) is the compress process and Fig. 2 (b) presents the compressed graph.

![Compress process](image2.png)

**Fig. 2.** Compress process.

**Table 1.** Weights of vertices after running random walk.

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Weight</th>
<th>Vertex</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.315522</td>
<td>17</td>
<td>0.493981</td>
</tr>
<tr>
<td>15</td>
<td>0.315522</td>
<td>13</td>
<td>0.563252</td>
</tr>
<tr>
<td>11</td>
<td>0.315522</td>
<td>3</td>
<td>0.563252</td>
</tr>
<tr>
<td>12</td>
<td>0.315522</td>
<td>10</td>
<td>0.563252</td>
</tr>
<tr>
<td>1</td>
<td>0.315522</td>
<td>5</td>
<td>0.563252</td>
</tr>
<tr>
<td>2</td>
<td>0.315522</td>
<td>8</td>
<td>0.737913</td>
</tr>
<tr>
<td>6</td>
<td>0.315522</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0.315522</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0.493981</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
4 Conclusions

In this paper, we firstly proposed a new graph multi-layered compression method based on random walk, and then we presented a new fast multi-level layout based on the compressed graph structure. We detailed described the fast multi-level layout and performed experiments on several datasets. We compared our layout with the FM* layout [25] and Walshaw [23] layout from three aspects. The comparable results show our fast layout can visualize the social networks high quality and rapidly.

Reference

5. Cui W, Qu H. A survey on graph visualization [J]. Hong Kong University of Science and Technology, 2007.


