Drawings of Compound Graph Using Free-form Curves

Jie GAO Kazuo MISUE Jiro TANAKA

Department of Computer Science, University of Tsukuba

1. Introduction

Compound graph is a kind of graph representation that can represent both hierarchy and adjacency relationships. It integrates inclusion graph and adjacency graph within adjacency edge and hierarchy edge, which are drawn as straight line segment and inclusion area, respectively. Cluster is an important concept in compound graph and is defined as the vertex that contains another one or more vertices.

Drawings of compound graph are discussed in some works [1, 2], in which we found representation for cluster is regular figure such as rectangle and circle.

Area efficiency is the main focus of this paper. It is concerned in drawing area consumed by a certain scale of graph. We think the area efficiency of previous methods is not well regarded and it is important to improve it. To address the problem of low area efficiency, we propose to use free-curve instead as representation of cluster, so that we can gather the vertices and clusters closer, meanwhile the cluster itself consume less space.

2. Compound Graph

A compound graph G is defined as the combination of inclusion graph $G_c = (V, F)$ and adjacency graph $G_a = (V, A)$, where V is a set of vertices, F is a set of inclusion edges, and A is a set of adjacency edges. The details of its definition are as follows.

$$G = (V, F, A)$$
$$V = \{v_1, v_2, v_3, \cdots, v_n\}$$
$$F \subseteq \{(m, n) \mid m, n \in V, m \neq n\}$$
$$A \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$$

We denote a vertex containing one or more other vertices is a **cluster** and a contained vertex is a **child**, and a vertex without children is a **leaf**.

The inclusion edges are generally represented as inclusion area containing vertices, while the adjacency edges are represented as straight line segment between vertices. Inclusion graph is directed graph and adjacency graph is undirected one.

3. Drawing Conventions

A drawing convention is a protocol for the placement of vertices and the drawing of edges. Effectively it consists of constraints that must be satisfied by a drawing [2]. In our method we created conventions as follows:

C1: A leaf is drawn as a circle.
C2: A cluster is drawn as a free curve surrounding all its vertices.
C3: the child should be placed inside of the cluster it belongs to.
C4: An adjacency edge is represented by a straight line.
C5: The child shared by one or more clusters should be placed inside of their intersection.

4. Drawing rules

Since a graph is an extremely simple abstract representation, despite minor differences, we can investigate several elementary and common criteria for "good" drawings, reflecting the characteristics of graphs. In our method, we followed the rules below.

R1: The area efficiency should be as high as possible.
R2: The unnecessary overlap of vertices should be avoided.
R3: The curve should not cling to the vertices beside it.

5. Drawing method

5.1 Outline

We implemented spring embedder model to place vertices and edges in the drawing area. Spring embedder model is designed to lay out simple undirected graph. So we convert compound graph to simple undirected graph by merging the inclusion edge and adjacency into an edge set. Thus we enabled the vertices and the edges of compound graph to be forced to move automatically until reaching a stable state.

Using the coordinates of the vertices, we calculate the convex hull surrounding all the vertices in each cluster, and draw cardinal curve through it. We will elaborate the flow in detail in the following subsections.

5.2 Converting

In order to adapt compound graph to the spring embedder model, we merge the two sets of edges into one, without considering of their direction. As a result, compound graph $G = (V, F, A)$ is converted into undirected simple graph G' through the processing below.

$$G' = (V, E)$$
$$E = A \bigcup \{\{u, v\} \mid (u, v) \in F\}$$

The graph G' is an undirected graph that can be drawn as a force-directed layout.

5.3 Extended spring model

In spring embedder model, vertices are considered as mutually repulsive charges and edges as springs that attract or repulsive connected vertices. The essential to use the model is giving a proper strength of repulsive force, and a proper nature length of the spring. We leverage the spring embedder model by assigning different nature length to the edges according to their type. The classification of edges is based on the type and relationship of the two vertices connected in the original compound graph. (See Table 1 and Table 2)

Table 1. Type and length of adjacency edge

	cluster	leaf	
cluster	l_1	l_2	
leaf	l_2	sister	otherwise
		l_3	l_4

Table 2. Type and length of hierarchy edge

	leaf	cluster
Cluster	l_5	l_6

The value of each kind of edge is determined empirically. We tried to find the optimal combination of the lengths for all the kinds of edges. Primarily, "optimal" combination is the one that fulfill the predefined rules as much as possible. Here we will also show a rough expression to describe these lengths

$$(l_1, l_2, l_6) > (l_4, l_5) > (l_3)$$

5.4 Rendering

Firstly, we group a set of points for each cluster by catching the coordinates of its children. Then we use Graham Scan to compute the convex hull of the point set. By connecting the vertices in order we get a minimal convex polygon, which containing all the vertices. At next step we compute the centroid of the polygon, and extend the line between centroid and its vertices respectively by a certain length. As a result, another set of points are generated. By connecting these points in order we get a new expanded polygon. The new polygon is similar to former one and keeps proper space to the vertices. Finally, we draw cardinal spline with all vertices in the expanded polygon.

Figure 1 is an example of drawing of compound graph. The free curves we draw consume less area than the regular figures such as circle and rectangle, as they look like elastics band that stretched open to encompass all their children, by which area efficiency is improved.

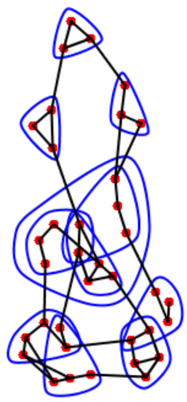

Figure 1. Drawing example

6. Conclusion

We presented an area efficient drawing for compound graph. The principle of the method is using closed free curve to represent cluster. Additionally, we adopt and extend the spring embedder model to place the graph on plane.

7. Reference

[1] H. Omote and K. Sugiyama. Development of drawing method for kj-diagram. In *The 4th Comference on the Support Systems for Knowledge Creation, 2007*, pages 76–83, 2007.
[2] K. Sugiyama. *Graph Drawing and Applications for Software and Knowledge Engineers*. World Scientific Publishing Co.Pte.Ltd, 2004.