HYBRID FORCE-DIRECTED AND SPACE-FILLING ALGORITHM FOR EULER DIAGRAM DRAWING

Maki Higashihara *Takayuki Itoh*
Ochanomizu University

ABSTRACT

Euler diagram drawing is an important problem because we may often have extended tree structures that children nodes are connected to multiple parent nodes. We expect space-filling tree data visualization techniques can be effectively applied to Euler diagram drawing. This paper presents an Euler diagram drawing technique applying a hybrid force-directed and space-filling node layout algorithm. The paper also introduces a subjective evaluation of the presented visualization style with recent Euler diagram drawing techniques.

1. INTRODUCTION

There have been many tree visualization techniques, which are divided into node-link and space-filling techniques; commonly-used tree datasets have been effectively visualized by applying the existing techniques. Meanwhile, we may often have extended tree structures that children nodes are connected to multiple parent nodes, which can be drawn as Euler diagrams. There have been several node-link techniques which effectively represent such data structures; however, we may often prefer space-filling techniques to visualize tree structures because it is easier to display all the lowest layer of nodes in a single display space. We are therefore focusing on drawing Euler diagrams by a space-filling tree visualization technique.

This paper presents an Euler diagram drawing technique as an extension of a space-filling tree visualization technique [1,2]. The paper first defines the Euler diagram as an extension of tree structure, and presents techniques to place nodes of the tree structure onto the display space, and draw the belonging information of the nodes. The paper also compares its visual style with recent drawing techniques [3,4,5].

2. RELATED WORK

2.1 Euler Diagram Drawing

Automatic Euler diagram drawing is a recent active topic, and survey papers in this field [6] has been also recently published. Several existing techniques [7] realizes reliable algorithms to draw theoretically complete Euler diagrams. On the other hand, it often happens that sets in real world applications are so complicated that they are often theoretically impossible to be drawn as complete Euler diagrams.

Several Euler-diagram-like representations for set visualization have been also recently presented and applied to real world datasets. Riche et al. [3] presented a novel technique "Untangling Euler Diagrams", which applies a force-directed algorithm as our technique does. The technique displays the categories as comprehensive shapes; however, it may require very large display space and computation time for large datasets. Collins et al. [4] presented "Bubble Sets" which draws arbitrary shaped categories on the top of various visualization spaces. Its visualization results are very flexible and good-looking; however, it may be sometimes difficult to complete in a reasonable computation time. Alper et al. [5] presented "Line Sets" which connects nodes which share the same categories by natural curved lines. They demonstrate Line Sets could generate better results rather than Bubble Sets.

2.2 Space-filling Tree Visualization

Treemaps [8] is the most famous space-filling approach which packs rectangular regions onto a display space. The space-filling tree visualization technique [1] applied in this study is also a space-filling technique, however, it is different from Treemaps since it represents leaf-nodes as square icons and branch-nodes as rectangular regions enclosing the icons. The technique applies a fast rectangle packing algorithm to obtain adequate looking of the hierarchy layout.

This hierarchy representation has been extended to the clustered graphs visualization technique [2]. The technique supposes that one or more items are assigned to leaf-nodes of the graph, while the items are represented as colors of the leaf-nodes. The technique firstly generates clusters of leaf-nodes based on their connectivity and item commonality. It then generates a simpler graph where its nodes correspond to clusters of leaf-nodes of the input graph. It then applies two-pass algorithm to calculate positions of leaf-nodes. The technique firstly applies a force-directed graph layout algorithm to briefly calculate positions of clusters so that tightly-connected or commonly-itemed clusters are placed closer. The technique then adjusts the layout by the rectangle packing algorithm [1] so that the clusters of the leaf-nodes never overlap each other.

Buchin [9] presented a variation of Treemaps algorithm which preserves the constraints of adjacency among the leaf nodes. It is possible to apply this variation of Treemaps to our problem; however, the space-filling technique we apply in this work [1,2] has advantages in aspect ratios of rectangles and visual similarity among similar datasets comparing with Treemaps.

Simonetto et al. [10] and Santamaria et al. [11] respectively presented techniques which calculates positions of nodes by connecting parent nodes by virtual links as our technique does. However, it does not take the maximization of display space utility into account.

3. TECHNICAL EXTENSION FOR EULER DIAGRAM DRAWING

3.1 Problem Statement

Space-filling hierarchical data visualization techniques [1,8] generally do not suppose that a leaf-node is connected to multiple parent nodes. This is a common limitation of space-filling visualization techniques including variety of Treemaps techniques.

Figure 1(Left) shows an illustration of an extended tree structure we suppose in this paper. The leaf-node "2" is connected to parent nodes "A" and "B" in this example. We must duplicate the leaf-node "2" under the two parent nodes, if we need to represent this data structure by the existing space-filling visualization techniques. However, this visualization may prevent understanding of the input data structure because it is not easy to visually recognize this duplication. Therefore, we suppose an additional parent node "AB" to maintain the multiple connections to parent nodes from "2", as shown in Figure 1(Center). The presented technique also adds "categories" to denote the parent nodes of original data structure. Green and red rectangles in Figure 1(Right) are categories which denote the parent nodes "A" and "B" in Figure 1(Left).

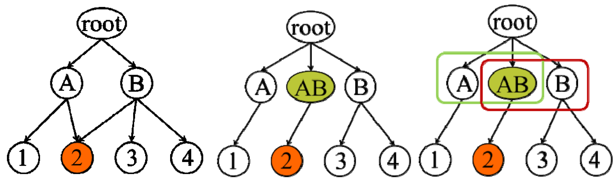

Figure 1. (Left) Extended tree structure the presented technique supposes. (Center) Additional parent nodes to maintain the connection between leaf and parent nodes. (Right) Categories which denote parent nodes of original data structure.

3.2 Node Layout

This technique visualizes Euler diagrams by drawing categories over the visualization result by a space-filling visualization technique [1]. Figure 2(Left) is an illustration of category drawing, where green and red rectangles correspond to categories in Figure 1(Right).

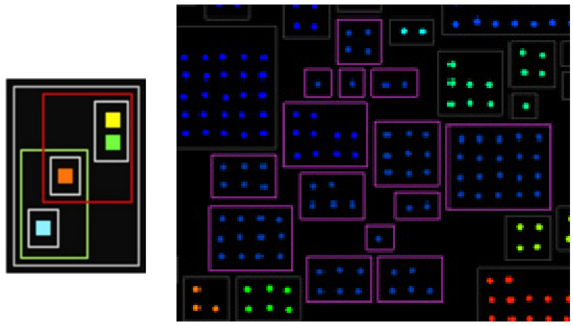

Figure 2. (Left) Additional drawing of categories. (Right) Layout of parent nodes. Rectangles which share same categories are closely placed.

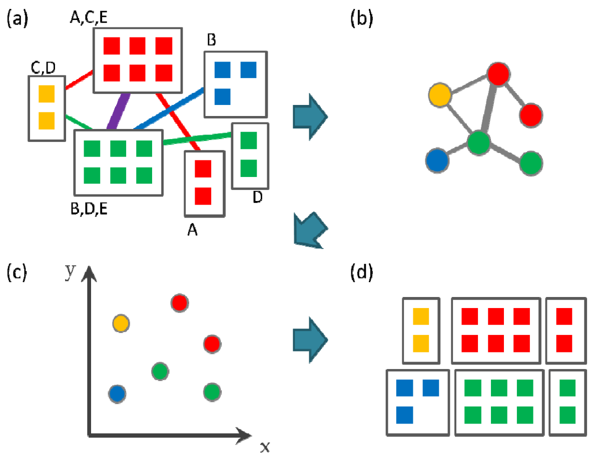

Figure 3. Hybrid force-directed and space-filling node layout. (a) Virtual links among parent nodes. (b) Force-directed cluster layout. (c) Ideal positions of clusters as the result of (b). (d) Space-filling layout of rectangular cluster regions.

Node layout is a very important problem to draw visually preferable Euler diagrams. We would like to place rectangles corresponding to parent nodes which share same categories closer in the display space. However, the space-filling visualization technique does not directly aim to place such rectangles closer. To solve the problem, the presented technique applies the graph layout technique [2] for the node layout of Euler diagrams. Figure 3 illustrates the processing flow of hybrid force-directed and space-filling algorithm for Euler diagram drawing. Our implementation generates virtual links among parent nodes which share same categories, and constructs a graph of clusters of leaf-nodes. It then applies a force-directed graph layout algorithm so that the linked parent nodes are closely placed in the display space. Finally, it applies the rectangle packing algorithm so that the visualization results never overlap the nodes and rectangles while maximizing the display space utility.

Figure 2(Right) shows an example of node layout by our implementation. Rectangular borders drawn in pink denote parent nodes sharing the same category. This result demonstrates our implementation successfully places such parent nodes closer in the display space.

3.3 Category Drawing

We firstly implemented category drawing functions as the following simple two types: convex hulls and connectors. Figure 4(Left) is an illustration of categories drawn as convex hulls. We think this representation is quite intuitive; however, it may unexpectedly enclose rectangles which do not share the same categories. Figure 4(Right) is an illustration of connectors linking rectangles which share the same categories. It never enclose unnecessary rectangles; however, we think it is sometimes bothering to follow the category-based connectivity of parent nodes. These illustrations show that the both implementation have drawbacks.

Therefore, we implemented a hybrid drawing method which conquers the drawbacks. This implementation generates convex hulls which enclose groups of adjacent rectangles which share the same categories, and then connects the convex hulls by polylines which ward the remaining rectangles. Figure 5 shows an example of the drawing result. We are currently improving the implementation so that multiple categories are simultaneously and clearly visualized.

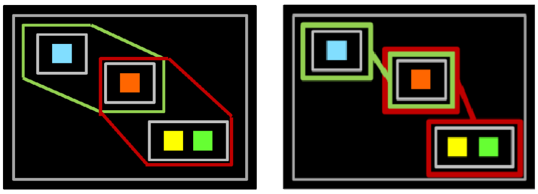

Figure 4. (Left) Categories as convex hulls. (Right) Categories as connectors.

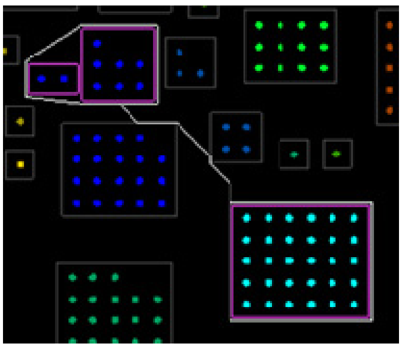

Figure 5. Category drawing as combination of convex hulls and connectors.

4. COMPARISON

4.1 Subjective Evaluation

We conducted a user evaluation to subjectively compare our drawing style with existing techniques, including Untangling Euler Diagram [3], BubbleSets [4], and LineSets [5]. Since it was difficult to implement all the existing techniques, we created illustrations of visualization results of the same datasets mimicking the results of our and existing techniques, as shown in Figure 6. The four pictures shown to the participants were as follows:

[Presented-1] Representing categories by connectors, similar to LineSet. See Figure 6 (Upper-left).
[Presented-2] Representing categories by convex hulls and connectors. See Figure 6 (Upper-right).
[Existing-1] Untangling Euler Diagram. See Figure 6 (Lower-left).
[Existing-2] Bubble Sets. See Figure 6 (Lower-right).

We shows them to 16 subjects and asked their subjective evaluations regarding the comprehensibility and discovery of the following five points:

- **Q1:** Overall category structures
- **Q2:** Particular pairs of categories
- **Q3:** Contents of a particular category
- **Q4:** Category belonging information of particular parent nodes
- **Q5:** Interested parent nodes

We asked them to answer their evaluations as five grade scores and comments.

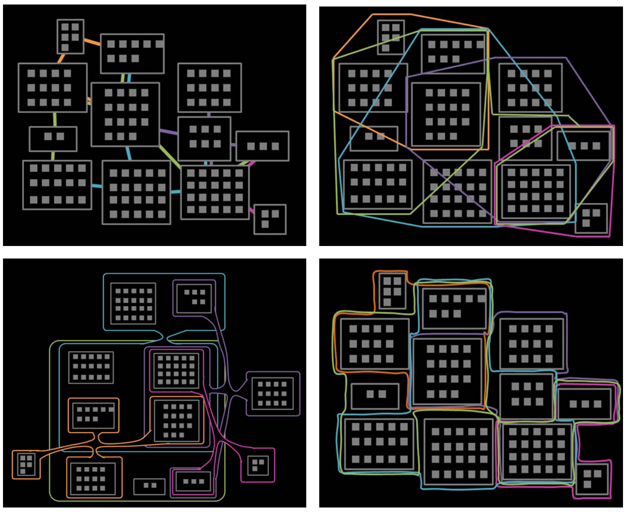

Figure 6. Illustrations shown to the participants.

Table 1. Average scores given by participants.

	Q1	Q2	Q3	Q4	Q5
Presented-1	2.69	3.07	2.56	3.81	2.50
Presented-2	3.13	4.50	2.69	2.53	3.13
Existing-1	2.06	2.19	1.31	1.69	1.38
Existing-2	4.50	4.44	4.69	3.75	4.56

Table 1 shows the average of five grade scores for each of the questions. This result shows Existing-1 got totally better evaluation. On the other hand, Presented-1 got the higher score on Q4, and Presented-2 got the higher score on Q2.

Following are the comments from the participants.

Presented-1
[Pros] Easy to follow all the parent nodes of a particular category.
[Cons] Difficult to understand structures of multiple categories simultaneously.
Presented-2
[Pros] Easy to understand structures of multiple categories simultaneously.
[Cons] Sometimes confusing if convex hull lines of categories go across the rectangular regions of parent nodes. Also, sometime confusing to visually distinguish edges of convex hulls and connectors.
Existing-1
[Pros] Less overlap among lines of categories. More intuitive.
[Cons] Will be too complicated if there are many categories.
Existing-2
[Pros] Categories do not overlap each other.
[Cons] Many lines representing categories concentrate at narrow regions.

4.2 Discussion

We recognized that our category drawing has two problems from the above comments. On the other hand, Existing-1 received a positive comment "less overlap among lines of categories.", and Existing-2 received another positive comment "Categories do not overlap each other." We think the solutions for the problems of the presented technique are related to the above positive comments for the existing techniques. Figure 7 illustrates the solution for the problems of the presented technique learned from the advantages of the existing techniques. We expect the comprehensibility of the categories will be improved by drawing a group of adjacent parent nodes as concave polygons, not convex hulls, and adjusting the drawing styles of connectors to visually distinguish with the concave polygons.

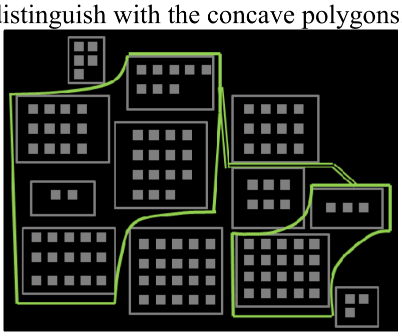

Figure 7. Illustration of the improved category drawing.

The user evaluation introduced in this paper just subjectively scored the visual comprehensibility of the presented and existing techniques. As future work, we would like to implement all the presented and existing techniques, and justify the comprehensibility by measuring the understanding of users. Also, we would like to apply larger datasets and compare the results of presented and existing techniques. Space-filling tree data visualization techniques has advantages on quick and reasonable visualization of large datasets. Recent Euler diagram drawing techniques [3,4,5] has not been applied to large datasets containing hundreds or thousands of nodes. It may be somewhat difficult to apply such scale of datasets to the existing techniques. We would like to evaluate the advantages of our technique on visualization of large scale Euler diagrams.

5. CONCLUSION

This paper presented an Euler diagram drawing technique applying a hybrid force-directed and space-filling node layout technique. The paper defined the input structure as an extended tree structure which contain "category" as parents of branch nodes. The technique draws categories as convex hulls surrounding branch nodes and connectors between the convex hulls. The paper also introduced a subjective evaluation with illustration of the visual styles of presented and existing techniques. We found several problems and solutions from the evaluation result.

We would like to complete the implementation of presented and existing technique as future work, and evaluate these techniques with larger datasets. Also, we would like to implement the improved category drawing shown in Figure 7, and compare again between the visualization results of presented and existing techniques.

REFERENCES

[1] T. Itoh, Y. Yamaguchi, Y. Ikehata, Y. Kajinaga, "Hierarchical Data Visualization Using a Fast Rectangle Packing Algorithm", *IEEE Transactions on Visualization and Computer Graphics*, 10(3), 302-313, 2004.

[2] T. Itoh, C. Muelder, K.-L. Ma, J. Sese, "A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs", *IEEE Pacific Visualization Symposium*, 121-128, 2009.

[3] N. H. Riche, T. Dwyer, "Untangling Euler Diagrams", *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1090-1099, 2010.

[4] C. Collins, G. Penn, S. Carpendale., "Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations", *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1009-1016, 2009.

[5] B. Alper, N. H. Riche, G. Ramos, M. Czerwinski, "Design Study of Line Sets, A Novel Set

Visualization Technique", *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2259-2267, 2011.

[6] P. Rodgers, "A survey of Euler diagrams", *Journal of Visual Languages & Computing*, 25(3), June 134-155, 2014.

[7] G. Stapleton, P. Rodgers, J. Howse, Z. Leishi, "Inductively Generating Euler Diagrams", *IEEE Transactions on Visualization and Computer Graphics*, 17(1), 88-100, 2011.

[8] B. Johnson, B. Shneiderman, "Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space", *IEEE Visualization*, 275-282, 1991.

[9] K. Buchin, D. Eppstein, M. Löffler, M. Nöllenburg, R. I. Silveira. "Adjacency-Preserving Spatial Treemaps", *12th International Conference on Algorithms and Data Structures*, 159-170, 2011.

[10] P. Shimonetto, D. Auber, "A Heuristic for the Construction of Intersection Graphs", *13th International Conference on Information Visualization,* 637-678, 2009.

[11] R. Santamaria, R. Theron, "Visualization of Interesting Groups Based on HyperGraphs", *IEICE Transactions on Information and Systems*, E93-D(7), 1957-1964, 2010.