Micallef, Luana and Rodgers, Peter (2009) ***Extended Abstract: Force-Directed Layout for Euler Diagrams.*** **In: Compendium of IEEE Information Visualization 2009 (IEEE InfoVis 2009). .**

Poster: Force-Directed Layout for Euler Diagrams

Luana Micallef* and Peter Rodgers†
University of Kent, UK

ABSTRACT

Euler diagrams are the only diagrams that intuitively represent containment, intersection and exclusion of data, but none of the current automatic diagram layout techniques produce good layouts in a reasonable time. We adopt a force-directed approach to automatically layout aesthetically pleasing Euler diagrams in a relatively fast time. A Java prototype demonstrates our novel force model.

KEYWORDS: Euler diagrams, Venn diagrams, diagram layout.

1 INTRODUCTION

Euler diagrams can intuitively represent containment, intersection, exclusion and other relationships between data sets. Consequently, these diagrams are used in a wide variety of application areas, such as biological visualization [5], and classification and querying of large databases [7]. As illustrated in Figure 1, these diagrams are composed of simple closed curves. As these curves meet, the plane is split up into zones that are uniquely identified by the curves that contain them.

Producing an aesthetically pleasing Euler diagram layout is a difficult task. Recently, approaches such as evolutionary optimization [5], hill climbing and aesthetic layout metrics [4] were adopted to automatically lay out these diagrams. Such methods are computationally expensive and thus, they are impractical to lay out diagrams with a large number of curves.

We adapt the well-known and successful force-directed approach used in graph drawing [2] to lay out Euler diagrams. For graphs, a system of springs (the edges) and electrically charged particles (the vertices) forms a force model, which is brought to stability in a sequence of iterations. Noting that closed curves in an Euler diagram can be represented as polygons, forces between vertices, lines, and the polygons themselves are used to produce nicely laid out diagrams. A major challenge, which is not a problem in graph layout, is the development of an appropriate force model that maintains the original structure of the diagram. By this we mean that all the zones that are in the original diagram are still in the laid out diagram, and that no new zones are added.

Force-directed placement has not previously been used to lay out Euler diagrams, but from the ongoing research described here, we have learnt that good layouts can be produced from previously generated diagrams in a reasonable time.

2 OUR APPROACH

Automatic generation methods for Euler diagrams such as [3, 6] produce difficult to comprehend representations (see Figure 1, left column). Thus, after generation, these representations have to be refined and improved by some other specific layout techniques.

In our case, our initial diagrams are generated by a technique based on that given in [6]. These diagrams are wellformed and thus, they have no poor structural features such as triple points, concurrency or disconnected zones [3]. As a result, the technique in [6] does not generate all possible diagrams.

Given an initial diagram, our approach simulates the effect of a group of attractive and repulsive forces for a number of iterations. The polygons are changed and the diagram is improved until finally, stability is reached. The main objectives of the forces are to: i) attain regular, smooth and similarly shaped polygons; ii) maintain the original structure of the diagram; and iii) ensure adequately sized polygons and zones.

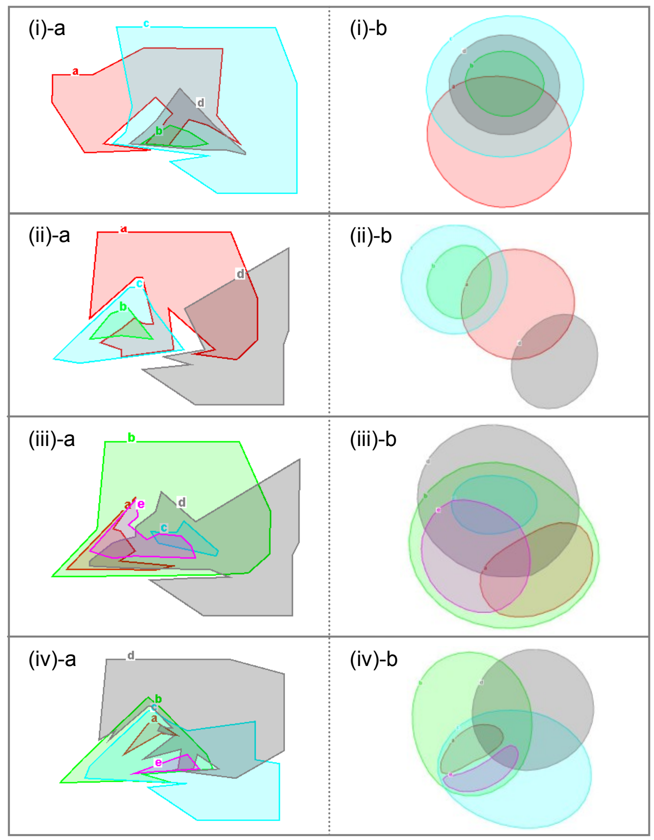

Figure 1: Initial 4 and 5-curve diagrams (left column) and their final layouts produced by our force-directed approach (right column)

2.1 Forces

Similarly to graph-based force-directed approaches, two types of forces are used in our model: attractive and repulsive forces. Attractive forces are directly proportional to the distance between two vertices, and repulsive forces are inversely proportional to the squared distance between two vertices.

Two forces attempt to attain smooth and regular polygons: **(a)** a repulsive force between every pair of vertices of the same polygon; and **(b)** an attractive force between every pair of adjacent vertices of the same polygon. Thus, as all of the vertices of a polygon move away from each other, the lines (acting as springs) attract neighbouring vertices so that a smooth polygon is formed.

*e-mail: lm304@kent.ac.uk, †e-mail: P.J.Rodgers@kent.ac.uk

However, since regular polygons might not always be appropriate to represent the required zones as the structure of the diagram must be maintained, a set of other forces that handle all the possible polygon-to-polygon relationships are required. Thus, if in the original diagram two polygons *do not intersect*, **(c)** vertices of the two different polygons repulse away from each other. If in the current layout these polygons intersect, **(d)** vertices (of the different polygons) that are currently breaking the structure attract. If in the original diagram polygons *intersect*, **(e)** vertices (of the different polygons) that are currently in the common zone repulse away from each other. This helps to ensure the existence of the zone. If in the current layout these polygons do not intersect, a special attractive force, **(f)** which is inversely proportional to the squared distance between vertices of the first and second polygon, is necessary to recover the lost zone. If in the original diagram, a polygon *contains* another polygon, **(g)** vertices of the two polygons repulse away from each other. If in the current diagram, the first polygon does not contain the second, **(h)** vertices (of the different polygons) that are invalidating the structure attract. Since there might be cases where a vertex of one polygon is closest to a point on a line between two vertices of a second polygon, the above mentioned forces are applied between the vertex of the first polygon and the closest point on the line of the second polygon. In this way, the system helps to ensure that none of the moving vertices invalidate the diagram structure.

The final set of forces are specifically designed to ensure adequately sized polygons and zones. Noting that the comprehension of diagrams consisting of various n-curve zones (n refers to the number of polygons in which the zone lies) can be improved by increasing the size of outer zones (which have a small n) and decreasing the size of the inner zones (which have a big n), the ideal size for every n-curve zone is calculated by $2^m/2^n$ (where m is the number of curves in the diagram). To increase the size of the zone, the size of the containing polygon is increased and the polygon is moved. To increase the size of the polygon, the repulsive force that handles its smoothness - force (a) - is progressively increased. To increase the actual zone size, **(i)** the polygons containing the zone are moved by an attractive force between their centroid and the zone centroid. If a new, unwanted, zone is introduced, **(j)** the polygons containing the new zone are moved away by a repulsive force between their centroid and the zone centroid. If alternatively a wanted zone is lost, **(k)** the polygons that should contain the zone are pushed towards each other by an attractive force between their centroids. Such corrective forces are essential to correct any flaws in the structure during the layout process. Still related to diagram aesthetics, **(l)** contained polygons are attracted to the centroid of the containing polygon or zone to centre internal curves.

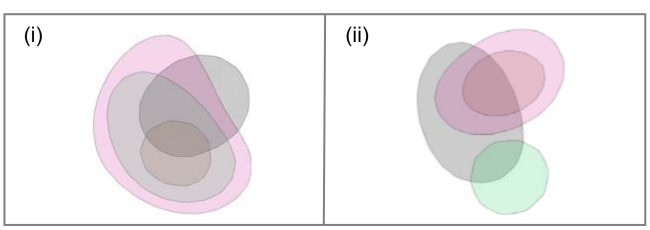

Figure 2: Final layouts produced by [4]

3 RESULTS

We implemented a Java prototype for our force-directed layout approach and we tested it using various 3, 4 and 5-curve wellformed diagrams generated by [6]. Figure 1 illustrates some before and after layouts. The final layout for all the 9 3-curve and all the 114 4-curve diagrams maintained the original description and we judged them to have acceptable layout. On average, 3-curve diagrams were laid out in 7 seconds[1] and 4-curve diagrams were laid out in 26 seconds. For 5-curve diagrams, 208 out of 342 final layouts successfully maintained the original description and by our judgement they had an acceptable layout. On average, 5-curve diagrams were laid out in 77 seconds.

4 CONCLUSIONS AND FUTURE WORK

We have developed a novel force model to produce aesthetically pleasing and comprehensible 3, 4 and 5-curve Euler diagram layouts, relatively quickly. This is still ongoing research and hence, refinements to the force model and algorithm are likely to achieve further improvements.

A main drawback of this approach is that for some cases, particularly dense diagrams, the interacting forces become unmanageable and thus, it is difficult to design a force model that handles all possible diagrams. However, results indicate that it works for all 3, 4-curve and most 5-curve diagrams.

An issue with this work is the evaluation of the quality of the final layouts. Various aesthetic metrics have been developed for graph drawing, but few are available for Euler diagrams. Only one empirical study has been carried out [1] and from this, the need for criteria to avoid polygon jaggedness, inequality of zone areas and curve closeness, has been identified. Although these criteria are dealt with by our force model, further evaluation based on empirical studies and general information visualization and perceptual principles [8], needs to be conducted. It is possible to compare layouts (i)-b and (ii)-b in Figure 1 (produced by our approach) with the corresponding layouts (i) and (ii) in Figure 2 (produced by [4]). We believe that the layouts produced by our force model are more aesthetically pleasing and comprehensible.

Currently, only wellformed diagrams are handled. Since generation algorithms can produce non-wellformed diagrams, current forces in our model could be modified and new ones could be added, to ensure the successful layout of such diagrams.

The code is still experimental and efficiency is currently not a primary concern. Clearly, many efficiency gains can be made by general code speedups and by applying known optimization methods used in other force-directed approaches.

ACKNOWLEDGMENTS

Work partially funded by the EPSRC, grant ref EP/E010393/1.

REFERENCES

[1] F. Benoy and P. Rodgers. Evaluating the Comprehension of Euler Diagrams. *Proc. IV 2007*, pages 771–778. IEEE.

[2] P. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 42:149–160, 1984.

[3] J. Flower and J. Howse. Generating Euler Diagrams. *Proc. Diagrams 2002, LNCS 2317*, pages 61–75. Springer.

[4] J. Flower, P. Rodgers, and P. Mutton. Layout Metrics for Euler Diagrams. *Proc. IV 2003*, pages 272–280. IEEE.

[5] H. A. Kestler, et al. VennMaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics*, 9(1), 2008.

[6] P. Rodgers, L. Zhang, G. Stapleton, and A. Fish. Embedding Wellformed Euler Diagrams. *Proc. IV 2008*, pages 585–593. IEEE.

[7] A. Verroust and M.-L. Viaud. Ensuring the Drawability of Extended Euler Diagrams for up to 8 Sets. *Diagrams 2004, LNCS 2980*, pp. 128–141. Springer.

[8] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[1] An Intel Core 2 Duo CPU E7200 @ 2.53GHz with 3.23GB RAM running Microsoft Windows XP Professional was used.