# Untangling Euler Diagrams

Nathalie Henry Riche and Tim Dwyer

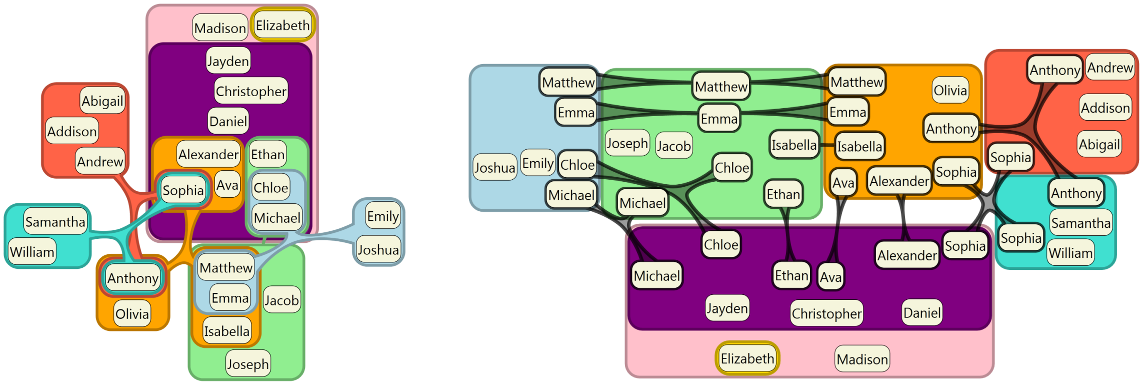


Fig. 1. Compact Rectangular Euler Diagram(left) and Euler Diagram with Duplications(right)

**Abstract**—In many common data analysis scenarios the data elements are logically grouped into sets. Venn and Euler style diagrams are a common visual representation of such set membership where the data elements are represented by labels or glyphs and sets are indicated by boundaries surrounding their members. Generating such diagrams automatically such that set regions do not intersect unless the corresponding sets have a non-empty intersection is a difficult problem. Further, it may be impossible in some cases if regions are required to be continuous and convex. Several approaches exist to draw such set regions using more complex shapes, however, the resulting diagrams can be difficult to interpret. In this paper we present two novel approaches for simplifying a complex collection of intersecting sets into a strict hierarchy that can be more easily automatically arranged and drawn (Figure 1). In the first approach, we use compact rectangular shapes for drawing each set, attempting to improve the readability of the set intersections. In the second approach, we avoid drawing intersecting set regions by duplicating elements belonging to multiple sets. We compared both of our techniques to the traditional non-convex region technique using five readability tasks. Our results show that the compact rectangular shapes technique was often preferred by experimental subjects even though the use of duplications dramatically improves the accuracy and performance time for most of our tasks. In addition to general set representation our techniques are also applicable to visualization of networks with intersecting clusters of nodes.

**Index Terms**—Information Visualization, Euler diagrams, Set Visualization, Graph Visualization

## 1 Introduction

Grouping data elements in sets (or clusters) is a common task in many analysis scenarios. For example, when analyzing documents, linguists often group words into semantic categories and topics. Similarly, when analyzing social networks, sociologists group people into communities and study their relationships. There is a wide range of techniques to compute sets (or clusters) based on similarity data [22]. The topic of this paper is visual representations of data elements such that their set membership is shown by region boundaries. When sets intersect in complex ways, this type of representation becomes a challenging problem in information visualization.

The common visual representation of sets are Venn and Euler style diagrams [14]. Venn diagrams represent all sets and their possible intersections with overlapping elliptical shapes. Euler diagrams are a relaxation of Venn diagrams in which the shapes corresponding to sets are not required to overlap if their corresponding intersection is empty. We identify two main challenges when drawing Euler diagrams:

1) *Complexity of set regions*. Gestalt theory [27] suggests that convexity of regions plays a key role in perception [23] and in our ability to complete shapes when partially occluded [28]. In addition, a few experimental results show that Euler Diagrams with convex shapes are more effective [3]. However, it can be a difficult challenge to draw Euler diagrams using convex set regions such that there are no overlaps between regions where the corresponding sets have an empty intersection [33].

2) *Drawing data elements*. Most work on drawing Euler diagrams focuses on classifying the sets in a particular dataset as drawable under constraints such as elliptical or convex regions [6]. Such work is rarely concerned with the problem of ensuring that sufficient space is provided inside the regions to show item labels or glyphs. Although there are applications (for example in biology) where only the sets themselves and their intersections need be shown [24], visually representing the data elements belonging to the sets is important in more general information visualization applications. For example, when analyzing communities in social networks or when studying articles grouped by keywords, it is important to identify which elements are in multiple sets.

Recent work in Information Visualization has attempted to address the challenge of drawing both sets and data elements. Simonetto *et al.* [31] describe how to automatically generate drawings with sets represented as non-convex regions as well as placing labelled elements inside these regions automatically. They demonstrate how their technique can draw previously undrawable Euler Diagrams. A second article from Collins *et al.* [7] presents a technique to generate set boundaries given a fixed layout of their elements. This technique can recompute boundaries around items involved in the same set effi-

- *Nathalie Riche is with Microsoft Research, E-mail: nath@microsoft.com.*
- *Tim Dwyer is with Microsoft Corp., E-mail: timdwyer@microsoft.com.*

*Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.*
*For information on obtaining reprints of this article, please send email to: tvcg@computer.org.*

ciently enough for interactive scenarios. While these techniques represent strong advances in the field, both of them can lead to non-convex and/or discontinuous shapes when many sets intersect in intricate ways (Figure 2). The use of color and texture [31] may help to convey the continuity of a given set region up to a limited number of different sets but the readability of such diagrams has not been well studied.

In this paper, we present two novel approaches to drawing set diagrams with labelled elements which break complex set intersections into a strict hierarchy that can be easily drawn with convex shapes (Figure 1). The intersections are then represented with additional links. These techniques leverage automatic constraint-based layout techniques to keep the links short and to produce readable, rectangular set and sub-set regions that are strictly non-overlapping except in the case of complete inclusion. Both techniques produce diagrams that are topologically Euler-like, but offer different approaches to simplifying the drawing of intersecting regions. The first approach: Compact Rectangular Euler Diagram (ComED), involves splitting regions involved in intersections and then folding them back together into a strict containment hierarchy with links between the split regions to show that they are connected. Our second technique: Euler Diagram with Duplications (DupED), takes inspiration from the *Semantic Substrate* technique [30] for placing the nodes of graphs with a grouping defined by the semantics of the application using non-overlapping rectangular regions. In DupED we avoid drawing intersecting sets by duplicating elements belonging to multiple sets. Duplicated elements have the same label and are connected by a link to indicate that they represent the same element [18]. We automatically arrange both ComED and DupED using a constraint-based graph-layout technique. Both techniques are therefore easily applied to the drawing of complex clustered-graphs, for example, see Figure 3.

To evaluate the readability of each technique, we ran two controlled experiments using five tasks testing the subjects' ability to quickly and accurately interpret the diagram. In the remainder of this article we describe previous work in set visualization, describe each of our techniques in more depth and present the controlled experiments we performed. We conclude by discussing the results of the study and the potential and limitations of each technique.

## 2 Related work

Set representations are used in many different fields; for example, to teach and demonstrate logic and set theory in schools [16], to analyze communities in social network analysis [29], to present results by topics in information retrieval [32], or to study groups of genes in bioinformatics [24]. While each field may have its unique terminology, in this article we use the generic term *set* as the logical group of *elements* and the term *set region* to designate its visual representation.

### 2.1 Representing sets

The most common set representations are called Venn and Euler diagrams. Venn diagrams represent all possible combinations of sets regardless of whether a given set intersection contains elements. Euler diagrams have the additional constraint that set regions should not overlap if the intersection of those sets is empty. There are multiple precise definitions of Venn and Euler diagrams [14, 4, 21, 33] varying in their definition of the shapes allowed for the set regions [13] or the definition of their undrawable instances.

Consequently, many algorithms exist to draw different types of euler diagrams and datasets that meet the different requirements for drawability, for an overview see Chow [6]. Particularly, a number of systems have been developed representing sets with convex set regions such as VennMaster [24], DrawEuler, DrawVenn and VennCircles [5]. However, these applications draw only particular subsets of Euler and Venn diagrams subject to fairly strict limitations (e.g. planarity of the Euler Dual).

Recently, Simonetto *et al.* [31] proposed a more generic method to represent a larger subset of Euler diagrams, drawing a number of previously impossible cases and scaling to larger and more complex set combinations. However, when a large number of sets intersect and many intersections are empty, the set regions have complex shapes and an intricate arrangement making the diagrams difficult to interpret, e.g. Figure 2. In these cases, fundamental set operations such as comparing the number of elements of two sets or assessing their intersection may still be very difficult to perform. To our knowledge, there are very few empirical studies focused on the readability of Euler Diagrams. One study [3] evaluated how the smoothness of the line contour, the sizes of the set regions and the closeness of contours affect the complexity of set regions and the readability of diagrams. These preliminary results did not clearly show the impact of these factors and it is not clear how these factors were controlled.

We opt for another approach and consider the complexity of set regions as a primary factor impacting readability. We formed this hypothesis from empirical results in Gestalt psychology [27] showing that convexity of regions plays a key role in how we perceive shapes as being part of the foreground or the background [23] and how it impacts our ability to complete shapes [28] when partially occluded. In other words, we believe that readability of Euler Diagrams is impacted when it is difficult to guess what the shape of the set region is (due to occlusion or ambiguity of overlapping regions). Coloring regions and applying textures as presented in [31] may help the user perceive the distinct set regions but unfortunately these solutions do not scale very well to large numbers of sets particularly when they are highly intersecting. In this paper, we present two algorithms to simplify the representation of set regions. To evaluate these techniques, we propose a method to control the level of complexity of set arrangements (see Section 6.1).

### 2.2 Representing sets and their elements

Because Euler diagrams are difficult to draw and to read when many sets intersect, alternative representations have been developed such as ConSet [25] and ComVis [15]. These representations are very effective when the primary focus of users is to analyze relations between sets without reading individual data elements. However, in many cases visualizing both sets and their elements is important. Visualizing ele-

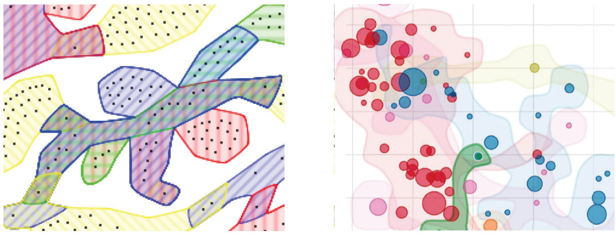


Fig. 2. Complex arrangement of sets in [31] and BubbleSets [7]. Note that the first method provides strict containment only for point-size elements, not textual or graphical labels; the second works with an existing placement and does not try to separate element boundaries to avoid intersections between set regions.

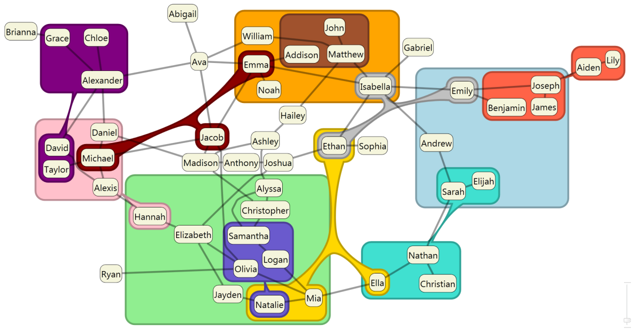


Fig. 3. A small (artificial) social network with primary "friend relations" represented as simple links and overlaid with intersecting set regions (representing, for example, affiliations) rendered using the ComEDtechnique.

ments adds an extra layer of complexity as intersecting regions must be large enough to contain the glyphs or labels representing those elements. For example, when analyzing multi-dimensional data with Scatterdice [12], identifying sets of elements sharing common properties is the main task, but elements have a fixed spatial layout induced by the scatterplot representation, highly constraining the drawing of sets. Other examples include maps or timeline representations in which set regions have to be represented as more or less complex shapes surrounding their members [7].

A related application is clustered graph visualization where the primary relations to be visualized are binary connections between elements while clusters (sets) are a second type of relationship that needs to be overlaid and, ideally, also considered in the layout of the diagram. See, for example, our rendering of such a graph in Figure 3. In these cases, sets are often overlayed on the graph representation, for example [9, 17]. In these representations, the problem of overlapping sets is rarely addressed. Eades *et al.* [11] proposed solutions to handle hierarchical sets, avoiding the case where a given element (node) belongs to multiple unrelated sets. Similarly, semantic substrates [30] demonstrates how laying out nodes according to their data attributes can help analysts discover insights about the data. However, this representation does not handle elements belonging to multiple sets, a common fact in real-world data.

A few techniques have used the duplication of elements to deal with overlapping sets. For example, Melancon *et al.* [26] duplicated elements to transform a direct acyclic graph into a hierarchical tree, represented it using treemaps [2]. Abello *et al.* [1] generate a clustering over a graph using biconnected components in order to provide semantic zoom navigation. They use duplication of articulation nodes (nodes that appear in multiple biconnected components) in order to obtain a strict hierarchy. Henry *et al.* [18] explored the use of duplications in NodeTrix, a hybrid of node-link and matrix representions for visualizing clustered graphs [19]. Generalizing the use of duplications in Euler diagrams and evaluating the readability of this representation compared to non-convex Euler diagrams has not been performed.

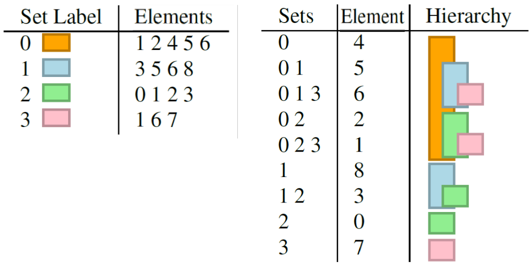
## 3 Two simple models for complex set intersections

Assessing the readability of Euler diagrams and identifying the factors that make them easy or difficult to interpret is a challenging problem [3]. Previous studies [3] suggested that factors such as smoothness of the boundaries or size of the set regions may affect the readability. Based on our empirical observations and results from the Gestalt theory [28], we believe that set intersections and complexity of the set regions are more likely to affect the readability of a diagram. In particular, we identify the following visual artefacts: (1) the *number of intersecting shapes* and how *their boundaries intersect or overlap*; (2) the *complexity and predictability of the set regions* (for instance a convex-hull is simple and predictable). In this section, we present two models that aim at improving these two readability issues.

### 3.1 Compact Rectangular Euler Diagram (ComED)

The basic idea of ComED is to split sets with intersections to produce a strict hierarchy which can be easily drawn with non-overlapping convex shapes (groups) and then to link up the split regions with lines. We argue that this technique minimizes the intersection of set regions' boundaries; i.e. non-intersecting group shapes cannot overlap, only the narrow links between them. Further, it guarantees that an element will be drawn strictly inside only those group boundaries corresponding to the sets to which it belongs. Figure 5(b) shows a simple (though highly intersecting) example to which we will refer again. We lay out the resulting diagram using constraint-based force-directed layout as described in section 5. Note that the term *group* is used to refer to the internal nodes in the hierarchy (the rectangular boundaries in our drawings).

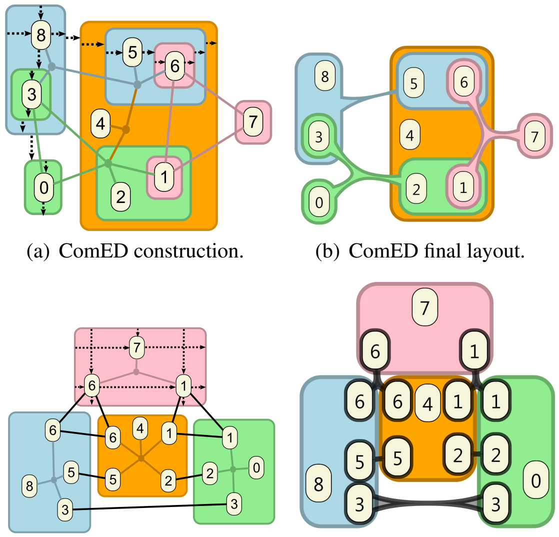
The algorithm for producing the strict hierarchy for a given set of intersecting sets could be thought of as a breadth-first traversal of the set-intersection graph. The difficulty lies in choosing the starting node for the traversal and for each pair of intersecting sets choosing which to split. Algorithm 1 is easily implemented without requiring an explicit computation of the intersection graph or traversal. Rather, it uses simple lexical sorting of set labels to apply a greedy heuristic which tends to split smaller sets in order to keep larger sets intact.



| Set Label | | Elements |
|---|---|---|
| 0 | 🟧 | 1 2 4 5 6 |
| 1 | 🟦 | 3 5 6 8 |
| 2 | 🟩 | 0 1 2 3 |
| 3 | 🟥 | 1 6 7 |

| Sets | Element | Hierarchy |
|---|---|---|
| 0 | 4 | |
| 0 1 | 5 | |
| 0 1 3 | 6 | |
| 0 2 | 2 | |
| 0 2 3 | 1 | |
| 1 | 8 | |
| 1 2 | 3 | |
| 2 | 0 | |
| 3 | 7 | |

Fig. 4. Figure 5(b) is produced according to the above algorithm as follows. First we sort the 4 sets and assign labels as shown in the left table. We then produce the list of lists of set labels for each element, sorted lexically, as shown on the right. We process this list to construct the hierarchy of groups and connect the groups with edges to produce the final ComED. Note that the lexical ordering of the sets also gives us the Z-ordering used for drawing set boundaries in the final drawing.



(a) ComED construction.

(b) ComED final layout.

(c) DupED construction.

(d) DupED final layout.

Fig. 5. Simple example of ComED and DupED representing the same dataset. Figures (a) and (c) show the underlying graph and arrows mark a subset of the constraints used to prevent overlap in the final layouts in figures (b) and (d), respectively.

Table 3.1 shows the sorted lists constructed by Algorithm 1 to produce Figure 5(b). Note that processing sets in order from largest to smallest is a heuristic and does not guarantee the least number of edges in the resulting diagram. Producing a ComED with as few edges as possible may be a difficult optimization problem and possibly NP-hard. In practice the method above seems to produce reasonable drawings and runs in $O(nm\log m)$ time for datasets with $n$ items and $m$ sets. Further analysis of this algorithm and the difficulty of finding optimal solutions is beyond the scope of this paper.

### 3.2 Euler Diagram with Duplications (DupED)

The principle of DupED is to avoid drawing any intersecting set regions and represent set regions with a simple rectangle. Though a more fitted hull could be easily drawn we find rectangles give a clean aesthetic. Overlapping set regions are allowed only in the case of a strict subset. Elements belonging to a single set are placed inside each set region. When a data element belongs to multiple sets, it is duplicated in each of the set regions. Figure 5(d) shows the DupED drawing of the sets shown in Figure 5(b). Algorithmically, finding nodes with

**Algorithm 1** Compact Rectangular Euler Diagram

```
Given a set of sets, assign a unique integer label to each set such that the labels increase as sets decrease in size (count of elements)
for each element do
    Construct a list of labels for all sets of which the element is a member
end for
Sort each list such that the labels increase monotonically
Sort the list of lists of set labels lexically
Construct a group for the first label in the first list
for each remaining label in the first list do
    Construct a group such that the group corresponding to each label is nested inside the previous group
    Add the first item as a child of the inner-most group
end for
for each remaining list of set labels do
    Construct groups for each of the sets in the list that differ from those in the previously processed list (nested as before)
    Add the item as a child of the inner-most group
end for
Connect with an edge all groups with the same label
```

multiple set parentage and choosing to duplicate them (if the entire set is not fully contained by the parent) is trivial.

## 4 Realizing the Models

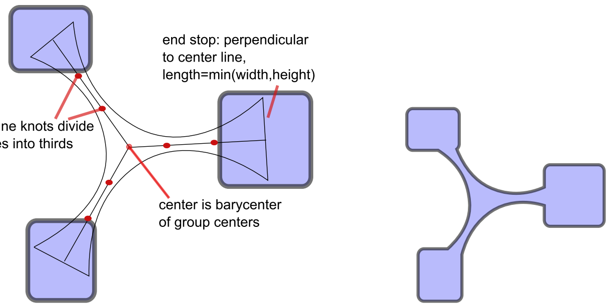
### 4.1 Link representation

In ComED we introduce links between all pairs of groups representing a single set and in DupED we introduce links between all pairs of duplicated elements. These links constitute an additional visual cue that the different graphical objects represent a unique element. Previous experiments [18] showed that these links were helpful even in the context of graph visualization, when other types of links are present.

Showing $n(n-1)/2$ links between all pairs of groups or elements can lead to a lot of clutter. As a result of our readability experiments, we decided to bundle links together to reduce clutter and better convey the unity of the sets. We used a simple technique to replace the multiple links with a single filled path centered on the barycenter of the connected groups or elements. To create the smooth contour, we draw lines from the barycenter to the center of each set and perpendicular segments to these inside each group. Splines are then computed using these construction lines. Figure 6 shows the details of generating the bundle shape with splines.

### 4.2 Layout

Both the ComED and DupED models provide methods for decomposing intersecting sets of items into a strict tree-like nesting hierarchy. Given such a hierarchy arranging the sets of items such that the hulls around the decomposed set-structures are strictly non-overlapping is a much more approachable problem. There is more than one layout method that could be applied to achieve a reasonable drawing. For example, treemaps are a simple and popular way of representing such hierarchies. However, we use a constraint-based layout method [8] that has two advantages over simple treemap arrangement. First, in the initial layout, links between split groups or duplicated elements are kept as short as possible. Second, it provides a strong guarantee that rectangular group and node boundaries can never overlap except when a node is a child of a group and that this will be achieved with as little displacement as possible from the underlying majorization-based layout.


end stop: perpendicular to center line, length=min(width,height)
spline knots divide lines into thirds
center is barycenter of group centers

(a) Link bundle construction.

(b) Union of element or group shapes with bundle path.

Fig. 6. The link-bundling technique used to connect split-groups or duplicated elements.

Our layout method for ComED and DupED is illustrated in Figure 5. First, we build a graph containing a node for each element and a node for each group. We then introduce edges from each group node to its member elements. Then, we add edges linking duplicated elements and duplicated groups as described above. An initial unconstrained majorization layout is then applied to this graph to unfold it, reducing crossings while keeping nodes well separated. Next, we generate separation constraints through a scan-line based method as described by Dwyer *et al.* [10] to preserve containment within rectangular groups and to prevent overlap between the rectangular boundaries where there is no containment relationship. The dotted arrows in Figures 5(a) and 5(c) show a subset of these constraints. Finally, we apply majorization-based layout subject to these constraints to minimize edge lengths while at the same time prevent unwanted overlaps. Full `C++` implementation of both the constraint generator and constraint-based layout methods is available from `http://adaptagrams.sf.net`.

## 5 Interaction

Interactions are demonstrated in the companion video of this paper available at `http://research.microsoft.com/~nath/EulerDiagrams`.

### 5.1 Interactive layout and set creation

Since the non-overlap/containment constraints are generated dynamically, the integrity of our ComED and DupED drawings is preserved even during interactive manipulation or editing of the groups and elements in the diagram.

In addition, the simplicity of our models allows interactive creation of sets. We implemented standard lasso selection to allow users to create sets interactively. ComED and DupED are recomputed after each set creation and the layout is automatically adjusted fast enough to provide responsive interaction on reasonably large diagrams (with hundreds of elements). From initial empirical tests, both techniques provide sufficiently stable configurations, in the sense that the arrangement of the existing set regions do not vary dramatically when adding new sets.

### 5.2 Interactive hybrid creation

DupED provides an extreme case of the use of duplications. Indeed, our current algorithm replaces all intersecting set regions (except when totally contained) by duplicated elements. However, we believe that in many cases, the most readable diagram is a hybrid version: some of the set regions replaced by duplicating elements while others still contained in overlapping regions. Then, identifying the right hybrid configuration and building it automatically becomes a challenging problem, out of scope of this paper. However, Figure 7 presents a simple technique allowing users to interactively create hybrid representations. This interaction technique can help disambiguating complex intersecting set regions by “untangling” them, smoothly transforming the intersection in a set of duplicated elements (see also the companion video).

## 6 Experiments

The two simple models presented both have advantages and drawbacks. ComED is a compact representation, designed to improve the readability of conventional “bubble-like” Euler diagrams. DupED is far less compact as it induces more elements (the duplicates) and introduces a layer of complexity: the additional links to mark duplications. Its advantages are that DupED produces a unique structure for

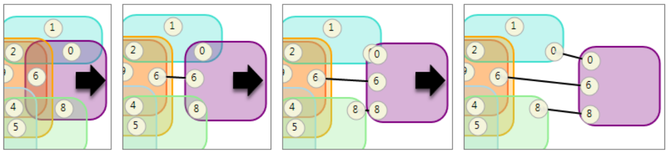
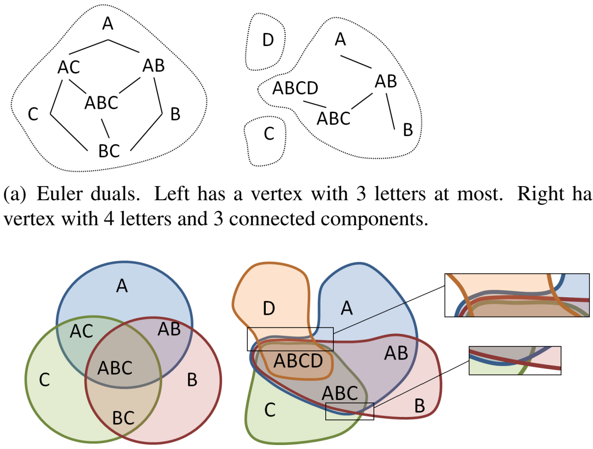


Fig. 7. Untangling sets interactively. From the initial configuration, the user drags the purple set to better understand how it intersects with others. As he drags the set, the areas of intersection are reduced until the elements have to be duplicated in both sets. The final configuration shows that the purple set shares one element with the blue set, one element with the green set and one with both the red and orange sets.



(a) Euler duals. Left has a vertex with 3 letters at most. Right has a vertex with 4 letters and 3 connected components.

(b) Corresponding Euler diagrams. Left is easy with a single 3-set intersection and no discontinuous regions. Right is harder with a 4-set intersection and two discontinuous regions (caused by empty intersections).

Fig. 8. Euler duals and their corresponding diagrams.

any given set input (there are multiple ways to draw overlapping set regions with ComED or other Euler diagrams) and rectangular set regions that are entirely disjoint.

To assess the readability of both techniques, we performed a first controlled experiment using 5 readability tasks measuring how users can perform general overview tasks (e.g. counting the overall number of sets) and more detailed tasks (e.g. assessing the number of elements in a given intersection). During this first experiment, we were surprised that our participants had difficulties learning ComED. Thus, we were curious to study how ComED performed compared to more conventional Euler diagram representations. For this reason, we performed a second controlled experiment comparing three different techniques: ComED, DupED and manually drawn Euler Diagrams.

### 6.1 Generating Euler diagrams of different difficulties

Generally, Euler diagrams increase in complexity as the number of sets participating in intersections increases until they can only be drawn with complex or discontinuous set regions. To control these properties and thus generate Euler diagrams with similar difficulties, we use the structure of corresponding *Euler duals* [6] to gauge complexity.

**Euler dual:** The Euler dual for a particular instance of intersecting sets is fairly straightforward. A vertex is defined for each topological region in the Euler diagram, i.e. a vertex for each non-empty intersection between two or more sets and also a vertex for any set with elements not found in any other set. An edge is defined in the graph between two vertices if the sets associated with each vertex differ by only one set. Figure 8 shows examples of Euler duals and their associated diagrams.

**Controlling the difficulty:** The *number of intersections* and the *number of sets involved in them* can be controlled via the generation of the vertices of the dual graph. For example, Figure 8(a) presents Euler duals with a vertex with 3 letters (left) and one with 4 letters (right) generating respectively a 3-set and a 4-set intersection (Figure 8(b)). The *complexity of the set regions* in a given Euler diagram can be controlled via the generation of the edges of the Euler dual. While a missing edge in the Euler dual may affect the convexity and regularity of set regions; disconnecting the Euler dual (*i.e.* breaking the dual in several connected components) ensures that set regions are discontinuous, introducing more complex set boundaries. Figure 8(a) illustrates the effect of breaking the dual in three connected components as well as the resulting diagrams (Figure 8(b)).

### 6.2 Method

Our first experiment compared performance of Compact Rectangular Euler Diagram (ComED) and Euler Diagram with Duplications (DupED). Our second experiment included Hand-Drawn Euler Diagram (DrawnED). For both, we used a within-subject design:

(Exp1) 2 *Vis* x 4 *Diff* x 5 *Tasks* x 3 repetitions.

(Exp2) 3 *Vis* x 4 *Diff* x 5 *Tasks* x 2 repetitions.

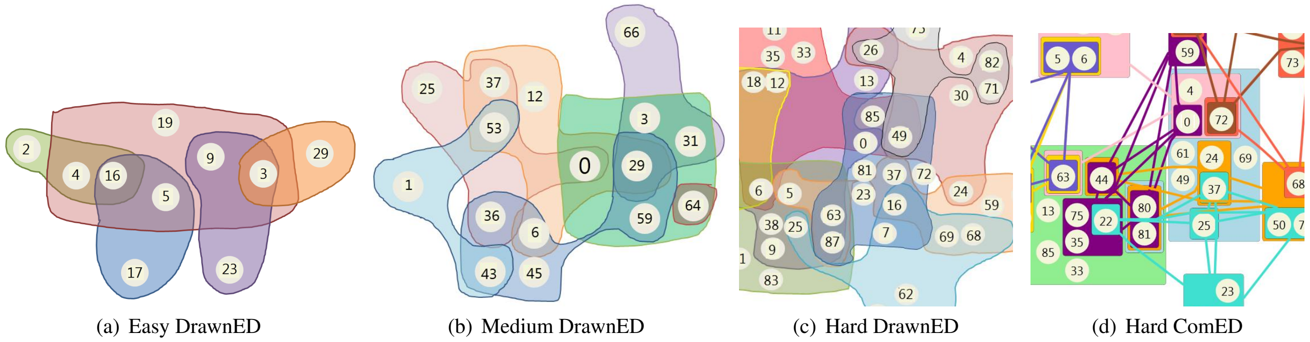
We used the same procedure for both experiments. We counterbalanced the order of the visualizations. The order of the tasks was fixed and we randomized the order of the datasets to avoid any memorization effect. Participants received training before each visualization. For each task, participants clicked on a button to indicate that they had finished reading the description of the task and were ready to begin, and pressed the space bar when they were done. The application recorded accuracy and completion time. To keep the study a reasonable length, we limited each task to a maximum of 40 sec. After the experiment, we collected user preferences and comments using a questionnaire. The study lasted approximately 60 min including training and post-experimental questionnaire.

**Participants and Apparatus** We recruited 18 participants, 9 for each experiment, screened to include people with general computer experience, balanced for age and gender, and not color-blind. For Experiment 1, 6 males and 3 females participated, with an age ranging from 21 to 47. For Experiment 2, 5 males and 4 females participated, with age ranging from 25 to 40. In both experiments, participants used a 3.00 GHz dual-core PC with 4 GB of RAM, running Windows Vista, and using 21” monitors at a resolution of 1600x1200.

**Visualizations** We used the algorithm described in the previous section to create both the ComED and DupED diagrams. We did not have link bundling for our studies (see Figure 9). This feature was implemented *a posteriori* from our participants’ feedback.

For Experiment 2, we manually drew DrawnED using Powerpoint and Photoshop. We used the layout of the elements created using ComED and drew the corresponding set regions around these. We iterated several tims and when drawing the set regions, we avoided sharp angles, keeping the contour as regular as possible. Our goal was to create shapes similar to the one created using BubbleSet [7]. We were careful to make the shape boundaries as predictable as possible, avoiding overlap of boundaries. However, creating such shapes is difficult when representing discontinuous set regions. Initially, we used a transparent color for each set. However, since we felt it was very difficult to distinguish these colors when multiple sets intersected, we added a border with a solid non transparent color. Figure 9 shows examples of DrawnED.

**Datasets** We controlled the number of sets, number of elements, number of 2-set, 3-set and 4-set intersections as well as the number of discontinuous set regions. In addition, for each difficulty, we had a minimal instance (one element in each set and intersection) and one with additional elements. The one with more elements was created to evaluate the effect of multiple duplication links in DupED. For experiment 2, we decided to stress all techniques. We raised the difficulty to medium and difficult datasets. We also increased the number of additional elements.



(a) Easy DrawnED (b) Medium DrawnED (c) Hard DrawnED (d) Hard ComED

Fig. 9. Example of the three dataset difficulties we used during the experiments. (d) shows ComED without bundling as tested in our experiment.

We created multiple instances of datasets to avoid memorization. In Experiment 1, we had 3 instances of each of the 4 difficulties. In Experiment 2, we reduced to 2 instances of each difficulty to keep the experiment to a reasonable time. Table 1 lists the values we used the generate the Euler duals and corresponding Euler diagrams per difficulty. Figure 9 shows an example of each difficulty.

| Exp1 | sets | elements | 2-set | 3-set | 4-set | disc. |
|---|---|---|---|---|---|---|
| (D1) Easy min. | 4-5 | ˜ 10 | ˜ 3 | 3 | 0 | 1 |
| (D2) Easy add. | 4-5 | ˜ 15 | ˜ 3 | 3 | 0 | 1 |
| (D3) Med min. | 6-7 | ˜ 15 | ˜ 4 | 2 | 1 | 1 |
| (D4) Med add. | 6-7 | ˜ 25 | ˜ 4 | 2 | 1 | 1 |
| Exp2 | | | | | | |
| (D3) Med min. | 6-7 | ˜ 15 | ˜ 4 | 2 | 1 | 1 |
| (D4) Med add. | 6-7 | ˜ 35 | ˜ 4 | 2 | 1 | 1 |
| (D5) Hard min. | 8-9 | ˜ 25 | ˜ 6 | 3 | 2 | 3 |
| (D6) Hard add. | 8-9 | ˜ 45 | ˜ 6 | 3 | 2 | 3 |

Table 1. Parameters used to generate Euler diagrams per difficulty.

**Tasks** We selected five readability tasks focusing on both sets and elements. We attempted to capture both the overview of the diagram and the readability of detailed portions such as set intersections. For each task, participants had to select from multiple choices.
(SetCount) What is the total number of sets?
(SetComparison) Given A and B, which set contains more elements?
(SetIntersection) How many elements are in the intersection of A and B.
(EltCount) What is the total number of elements?
(EltMembership) Which set(s) contain element 0?

## 6.3 Hypotheses

In both experiments, we hypothesized that DupED would outperform the other techniques for tasks centered on sets (H1) as they provide a simpler boundary for the sets. However, we believed that the other techniques should outperform DupED for tasks centered on elements (H2) since they do not introduce duplicated elements. We believed that DupED would decrease in performance for the datasets with a high number of duplicated elements (H3), duplication links causing clutter and degrading the readability.

During experiment 1, we were surprised that the training for ComED was much longer than for DupED. Many participants had trouble identifying the sets and several commented on how hard it was to understand. Thus, we formed the hypothesis that ComED may, in fact, be significantly less readable than regular Euler diagrams with non-convex hulls such as BubbleSets [7]. For this reason, we decided to run a second experiment introducing a third visualization technique: Hand-Drawn Euler Diagram (DrawnED). We hypothesized that participants would prefer DrawnED since they are more familiar with it but that ComED would outperform it (H4).

In addition, since several participants reported that it was easy to only rely on the colors of the sets, we decided to raise the overall size and difficulty of the datasets. Finally, we took advantage of this second experiment to further explore the robustness of DupED by adding more duplication links (from +50% additional elements to +100% additional elements).

## 6.4 Results

In this section, we report the results of both experiments for each task. Figure 10 contains the details of the ANOVA for experiment 2 as well as the mean accuracy and time for each technique. Complete analysis for both experiment is available in [20]. Table 11 presents a summary of the results.

### SetCount

**Accuracy:** Wilcoxon's test and Friedman's test do not reveal any significant difference between *Vis* in both experiments. The accuracy for all the techniques is affected by the difficulty of the datasets and drops from 30% between experiment 1 and 2.

**Time:** ANOVA reveals a significant difference between techniques in both experiments. Post-hoc comparison reveals that DupED is faster than DrawnED, which is faster than ComED. DupED is about twice faster than the other two techniques for this task. It also reveals that DrawnED is about 20% faster than ComED.

**Preference:** Participants mostly preferred DupED for counting the sets (16/18 for both experiments). In the second experiment, almost all of the participants favored ComED over DrawnED, commenting that it was difficult to differentiate colors in DrawnED, especially when transparent and overlapping with each other. Several participants stated that "finding the set boundaries in there is a nightmare!" and two of them mentionned that it was difficult to "predict where the shape goes".

### SetComparison

**Accuracy:** Wilcoxon's test and Friedman's test reveal a significant difference between techniques for both experiments (Wilcoxon $p < .01$, Fiedman $p < .001$). DupED is more accurate than both other techniques. There is no difference between ComED and DrawnED. Results in experiment 1 shows that DupED is about 20% more accurate than ComED for this task.

**Time:** ANOVA also reveals a significant difference in performance time between techniques for both experiments. DupED is faster than both other techniques. ANOVA reveals a significant difference between *Diff* as well as an interaction *Diff* x *Vis*. As the difficulty increases, all techniques decrease in performances. DupED is particularly affected when the number of duplicated elements increase.

**Preference:** Participants mostly preferred DupED for comparing the number of elements of two sets (16/18 for both experiments). All participants favored ComED over DrawnED for this task.

| ANOVA exp 2 | *Vis* | *Diff* | *Vis*x *Diff* |
|---|---|---|---|
| | F(2,16) | F(3,24) | F(6,48) |
| All tasks. | 15.98*** | 53.85*** | 2.67* |
| SetCount. | 67.66*** | 10.82*** | 0.8 |
| SetComp. | 48.16*** | 24.58*** | 3.64** |
| SetInter. | 01.78 | 08.15** | 3.92** |
| EltCount. | 11.95*** | 30.51*** | 1.04 |
| EltMember. | 19.14*** | 29.35*** | 5.82*** |
| | ***$p < .001$ | **$p < .01$ | *$p < .05$ |

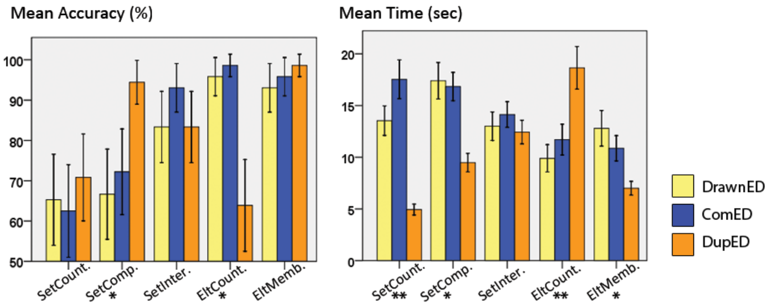


Fig. 10. ANOVA and mean accuracy and time for the three techniques compared in experiment 2.

| Task | Accuracy | Time | Preference |
|---|---|---|---|
| SetCount | $DupED = ComED = DrawnED$ | $DupED < DrawnED < ComED$ | $DupED > ComED > DrawnED$ |
| SetComparison | $DupED > ComED = DrawnED$ | $DupED < DrawnED < ComED$ | $DupED > ComED > DrawnED$ |
| SetIntersection | $DupED > ComED = DrawnED$ | $DupED < ComED = DrawnED$ | $DupED >= ComED > DrawnED$ |
| EltCount | $ComED = DrawnED > DupED$ | $DrawnED < ComED < DupED$ | $ComED > DrawnED > DupED$ |
| EltMembership | $ComED = DrawnED = DupED$ | $DupED < ComED = DrawnED$ | $DupED >= ComED > DrawnED$ |

Fig. 11. Table summary of the results.

### SetIntersection

**Accuracy:** Wilcoxon's test and Friedman's test do not reveal any significant difference between *Vis* in both experiments. In experiment 2, Friedman's test reveals a significant difference in accuracy when splitting the results by *Diff*. In the most difficult case (D6), Wilcoxon's test reveals that ComED is more accurate than both other techniques ($p < .05$).

**Time:** ANOVA only reveals a significant difference for experiment 1, showing that DupED is faster than ComED. As *Diff* increases, all techniques are affected. Splitting the results by *Diff* for experiment 2 reveals that DupED is strongly affected by the number of duplicated elements. In the most difficult case, DupED is about 30% slower than the two other techniques.

**Preference:** In experiment 1, participants were divided between both techniques (4/9 preferred DupED, 2/9 preferred ComED, 3/9 ranked both techniques as equivalent). In experiment 2, participants mostly preferred DupED to both other techniques (7/9). All participants favored ComED over DrawnED for this task.

### EltCount

**Accuracy:**Wilcoxon's test and Friedman's test show a significant difference between *Vis* in both experiments (Wilcoxon $p < 0.01$, Friedman $p < .001$). In experiment 1, ComED is about 20% more accurate than DupED. In experiment 2, both ComED and DrawnED outperform DupED. No difference is shown between ComED and DrawnED.

**Time:** ANOVA reveals a significant difference between *Vis* for both experiments. In experiment 1, results show that ComED performs faster than DupED. In experiment 2, post-hoc comparison shows that both DrawnED and ComED perform twice faster than DupED and that DrawnED is faster than ComED for this task. ANOVA also reveals a significant interaction *Diff* x *Vis* in both experiment. As expected, DupED is strongly affected when the number of duplicated elements increase.

**Preference:** Participants uninamously disliked DupED for counting the elements, many stating that the duplication links were an extra burden for this task. In experiment 2, 7/9 participants ranked ComED above or equal to DrawnED despite the significant superiority of DrawnED in completion time. When asked why they preferred ComED for counting elements, participants explained that the overlapping shapes were distracting, one participant commenting that "it was difficult to tune out the fuzzy shapes".

### EltMembership

**Accuracy:** Wilcoxon's test and Friedman's test do not reveal any significant difference between techniques for both experiments.

**Time:** ANOVA reveals a significant difference between *Vis* for both experiments. Contrary to our hypothesis, post-hoc comparison shows that DupED is faster than both other techniques even when the number of duplicated elements is high. In experiment 2, results show that DupED is about 40% faster than both other techniques. There is no significant difference between DrawnED and ComED. However, ANOVA reveals a significant interaction *Vis* x *Diff*. DupED is particularly affected when the number of duplicated elements increases and DrawnED is strongly affected as the dataset difficulty increases. For the most difficult dataset (D6), DupED performs twice as fast as DrawnED.

**Preference:** In experiment 1, 7/9 participants preferred DupED to ComED explaining that following the duplication links was easier than counting the set boundaries. In experiment 2, in which the number of duplicated elements significantly increased, participants were more divided. 6/9 favored ComED and 3/9 favored DupED. They expressed either their preference for containment: "I just look at the colors in a single eye movelement from the element to the top of the screen" or explained that following the duplication links was more efficient. None of them preferred DrawnED.

## 7 Discussion

Our results showed that DupED outperformed the other techniques for two of the set tasks as we expected (H1). However, *the cluttering caused by the duplication links seems to have affected the SetIntersection task* (in which participants had to count the number of links between two sets). As expected, ComED and DrawnED outperformed DupED for counting elements but *surprisingly not when identifying the element membersip* (H2).

Overall, we were surprised that *the number of additional elements did not impact more strongly on the overall performance of DupED* (H3), especially in experiment 2 as we raised the complexity significantly. We observed, though, that this clutter impacted the preferences as, contrary to our first experiment, very few participants mentioned that DupED was "cleaner" or easier to read than the other techniques.

Surprisingly, *ComED did not significantly outperform DrawnED for SetIntersection and ElementMembership* (H4). We believe that this is due to the clutter caused by the links between split regions. This led us to implement the link-bundling method described in section 4.1. In addition, for ElementMembership, we observed several participants counting the set colors instead of using the set containment. While we briefly explained the strategy during the participant training, we believe that ComED requires more practice.

Contrary to our expectations, *ComED was unanimously preferred over DrawnED* and favored to DupED in several cases. Our participants commented that ComED worked best for the most complex cases, especially when the number of duplicated elements was high in DupED and the set arrangement intricates with DrawnED. One participant ranked ComED as his favorite technique for all conditions, mentionning that it "worked" for him and that he could "tune his eyes to see the sets in different layers by color". However, his quantitative results did not differ significantly from the others.

### 7.1 Limitations

The controlled experiments we performed suffer from a number of limitations and should only be considered as an initial exploration of the readability of set visualization techniques. First, as we implemented link-bundling as a result of our studies, further experiments are required to explore the impact of this feature. While we believe that the bundling would not impact strongly the results for the small datasets we tested, we feel it is an important feature to scale to larger and more complex set arrangements (see Figure 13). Second, we highlight the need to further compare the readability of set visualization techniques such as weaving techniques and texture-based techniques. Finally, in the following section, we give anecdotal evidence to show how our techniques scale to complex set arrangements. However, further controlled experiments with larger and more complex datasets are required to formally assess the scalability of these techniques.

### 7.2 Scalability

While these experiments showed that DupED is an effective technique to show relatively simple set intersections, we believe ComED is promising for larger and more complex diagrams. For DrawnED, the complexity of the set intersections is the factor that cause the technique not to scale. As observed in the "complex" set arrangements of our experiment, participants commented that ComED worked the best for complex cases in which the transparent overlapping set regions of DrawnED were "really indiscernible".

The issue of scale is not one of algorithmic complexity: the algorithms used to split groups (ComED) or duplicate elements (DupED) and our layout method easily run in a few seconds for graphs with thousands of nodes and hundreds of groups (detailed algorithmic analysis is beyond the scope of this paper). However, anecdotal evidence suggests that ComED scales significantly better than DupED in terms of visual complexity.

To further investigate the scalability of both ComED and DupED, we visualized two larger datasets: Figure 12 shows the top 100 movies and their 1174 actors (from IMDB); Figure 13 shows a tag cloud of the top 200 words in the ten tragedies of Shakespeare. Figure 12 shows that several thousands of elements may be easily visualized if the set intersections are relatively simple. Figure 12 has only 61 nodes but the complex intersections of the ten sets is really testing the limits of readability of ComED and DupED is really no longer useful due to the number of complex links between duplicated elements. However, we find the ComED result for such large and complex examples compelling and already useful. A more sophisticated edge bundling method that avoids unnecessary intersections would further improve readability of such complex examples.

## 8 Conclusion and Future Work

Our goal in this paper was to provide techniques to improve the readability of Euler diagrams visualizing both sets and set elements. Many applications rely on visualizing both sets and their elements: when analyzing text it is important to visualize both the keywords and the documents they are part of; when analyzing social networks, it is required to visualize people's name and the communities they belong too; or in more general applications such as exploring pictures, it may be important to visualize both images and their topics.

We proposed two novel approaches. The first approach we proposed (ComED) aims at simplifying the shapes of set regions by splitting set regions into compact rectangular shapes connected by links. The intention is to make them more predictable and to limit intersections between boundaries. The second technique we presented entirely avoided set intersections and, instead, duplicated elements in multiple sets. Results of our studies show that both techniques are promising. DupED was found to be more readable than ComED, but ComED was preferred by many participants. Since ComED produces more compact drawings, we believe that it scales better to diagrams with more sets and elements as we briefly show in the previous section.

While our results did not show that ComED significantly outperformed Hand-Drawn Euler Diagram, ComED was unanimously preferred. In addition, we believe that the link bundling feature we implemented based on study feedback improves the readability of ComED. It would be interesting to study further how different types of bundling techniques affect the readability. Many other refinements are possible, to both the generation of ComED and DupED models and their visual representations. For example, rectangular group boundaries look "tidy" and are well suited to our layout technique but we could easily draw more fitted boundaries such as a smoothed convex hull.

These visualizations also invite more interaction. We are currently planning to allow users to interactively manipulate the center of the link bundle and to enable the interactive creation of more hybrids. Since we identified advantages and disadvantages to both techniques, hybridization seems to be a promising direction.

## References

[1] J. Abello, F. van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12:669–676, 2006.

[2] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, October 2002.

[3] F. Benoy and P. Rodgers. Evaluating the comprehension of euler diagrams. In *IEEE IV*, pages 771–780, 2007.

[4] A. F. Blackwell, K. Marriott, and A. Shimojima, editors. *Diagrammatic Representation and Inference, Third International Conference, Diagrams 2004, Cambridge, UK, March 22-24, 2004, Proceedings*, volume 2980 of *Lecture Notes in Computer Science*. Springer, 2004.

[5] S. Chow. http://webhome.cs.uvic.ca/ schow/.

[6] S. Chow. *Generating and drawing area-proportional euler and venn diagrams*. PhD thesis, University of Victoria, Canada, 2007.

[7] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE TVCG*, 15:1009–1016, 2009.

[8] T. Dwyer, Y. Koren, and K. Marriott. Ipsep-cola: An incremental procedure for separation constraint layout of graphs. *IEEE TVCG*, 12:821–828, 2006.

[9] T. Dwyer, K. Marriott, F. Schreiber, P. Stuckey, M. Woodward, and M. Wybrow. Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE TVCG*, 14(6):1293–1300, 2008.

[10] T. Dwyer, K. Marriott, and P. J. Stuckey. Fast node overlap removal. In *Proc. 13$^{th}$ Intl. Symp. Graph Drawing (GD'05)*, volume 3843 of *LNCS*, pages 153–164. Springer, 2006.

[11] P. Eades and Q.-W. Feng. Multilevel visualization of clustered graphs. In *Proc. Graph Drawing, GD*, number 1190 in LNCS, pages 101–112, Berlin, Germany, 1996. Springer-Verlag.

[12] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE TVCG*, 14(6):1141–1148, 2008.

[13] A. Fish and G. Stapleton. Defining euler diagrams: choices and consequences. *Euler Diagrams Workshop*, 2005.

[14] J. Flower and J. Howse. Generating euler diagrams. In *DIAGRAMS '02: Proc. of the 2nd International Conference on Diagrammatic Representation and Inference*, pages 61–75. Springer-Verlag, 2002.

[15] W. Freiler, K. Matkovic, and H. Hauser. Interactive visual analysis of set-typed data. *IEEE TVCG*, 14(6):1340–1347, 2008.

[16] E. M. Hammer. *Logic and Visual Information*. CSLI Publications, 1995.

[17] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Proc. Intl. Symp. Information Visualization (Infovis'05)*. IEEE, 2005.

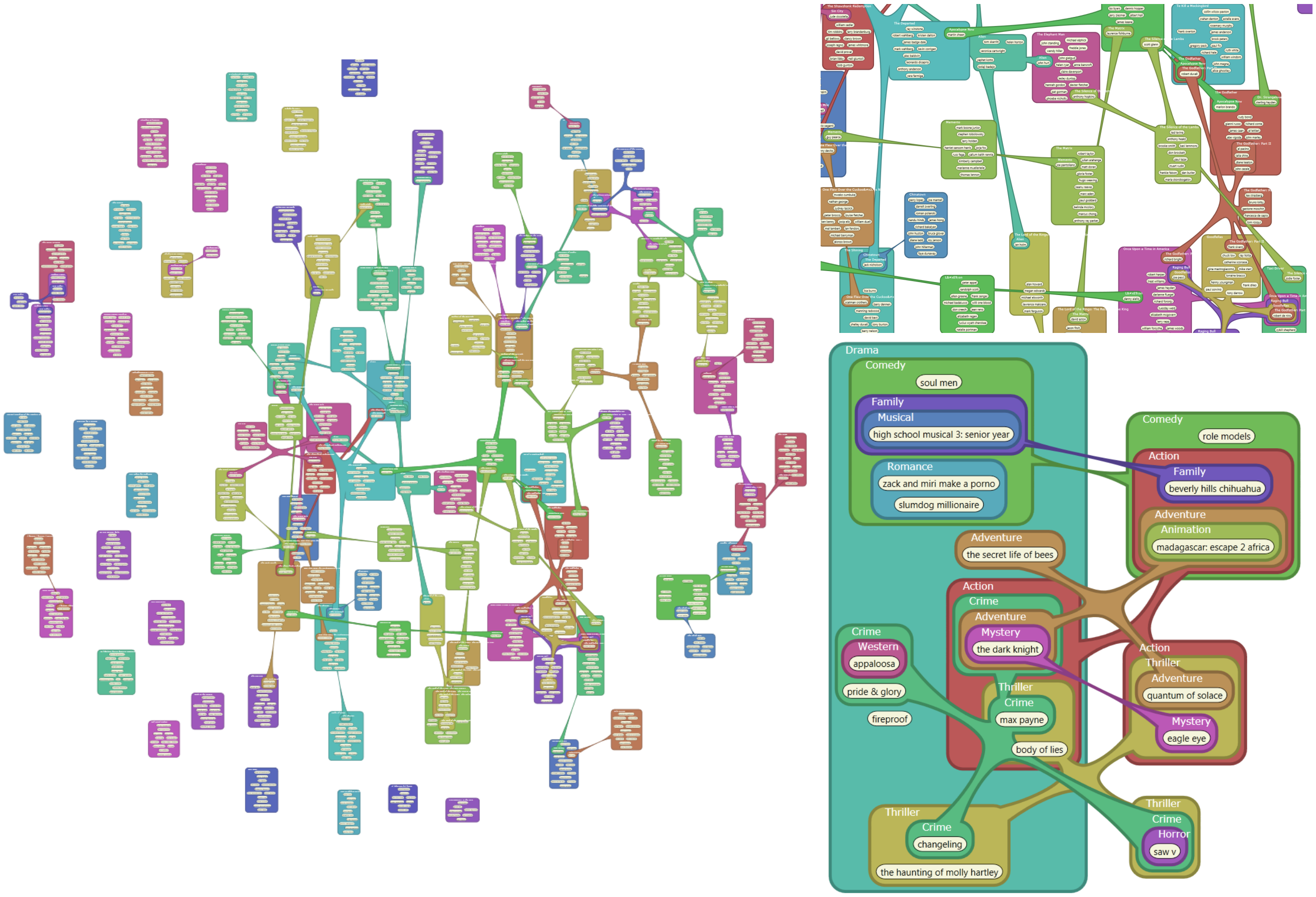
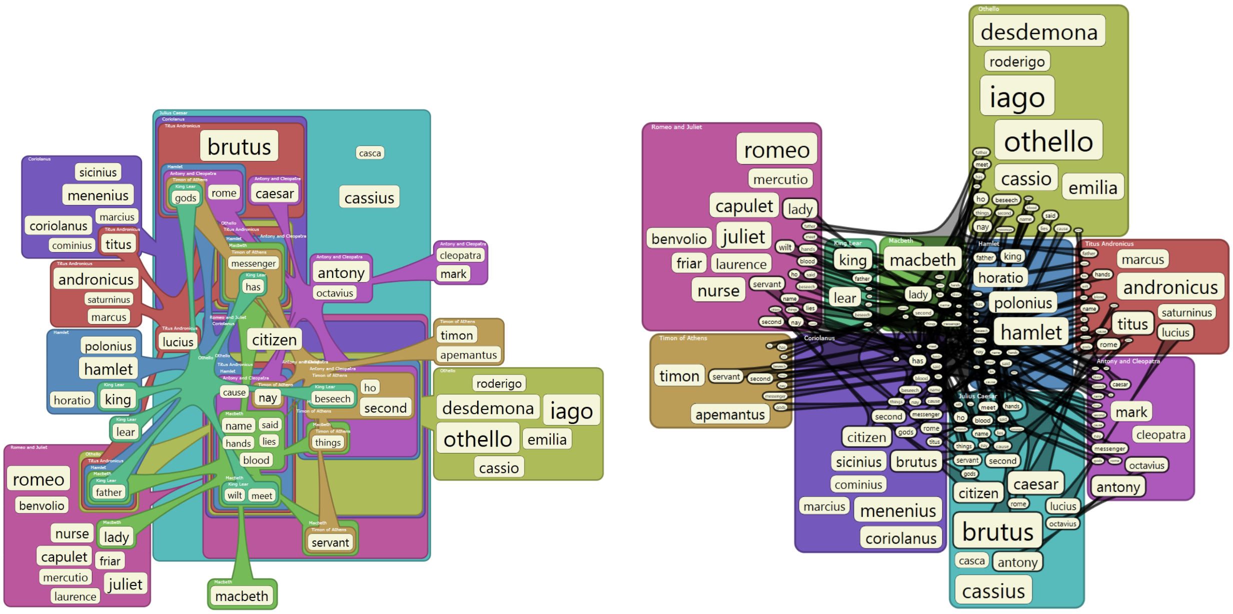


Fig. 12. Larger examples showing the scalability of ComED to larger and more complex datasets extracted from IMDB. On the left, ComED showing actors in the top 100 ranked movies, 1174 items (actors) and 100 sets (movies). The top right figure shows a close-up view. The bottom right figure is a ComED showing the top 20 ranked movies by Genre.



(a) ComED showing frequency across all plays.

(b) DupED where, since the words appear once per group, we can see the frequency each word is used within each play

Fig. 13. Tag cloud examples of ComED and DupED representing the same dataset: the most frequently used words in ten plays by Shakespeare. Words used in all 10 plays were removed since their set membership can be trivially expressed in a separate list, leaving 61 words. Font-size is set to the squareroot of frequency. Although these examples are larger and more complex than those tested in our experiment, our feeling is that ComED is significantly more readable with large and complex datasets.

[18] N. Henry, A. Bezerianos, and J.-D. Fekete. Improving the readability of clustered social networks using node duplication. *IEEE TVCG*, 14(6):1317–1324, 2008.

[19] N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE TVCG*, 13(6):1302–1309, 2007.

[20] N. Henry-Riche and T. Dwyer. Untangling euler diagrams. *Technical Report*, 2010. http://research.microsoft.com/ nath/EulerDiagrams.

[21] J. Howse, F. Molina, J. Taylor, S. Kent, and J. Gil. Spider diagrams: A diagrammatic reasoning system. *Journal of Visual Languages and Computing*, 12(3):299–324, 2001.

[22] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[23] G. Kanizsa and W. Gerbino. Convexity and symmetry in figure-ground organization. In M. H. ed., editor, *Vision and artifact*. M Henle ed., New York: Springer, 1976.

[24] H. Kestler, A. Muller, J. Kraus, M. Buchholz, T. Gress, H. Liu, D. Kane, B. Zeeberg, and J. Weinstein. Vennmaster: Area-proportional euler diagrams for functional go analysis of microarrays. *BMC Bioinformatics*, 9(1), 2008.

[25] B. H. Kim, B. Lee, and J. Seo. Visualizing set concordance with permutation matrices and fan diagrams. *Interacting with Computers*, 19(5-6):630–643, 2007.

[26] P.-Y. Koenig, G. Melancon, C. Bohan, and B. Gautier. Combining dagmaps and sugiyama layout for the navigation of hierarchical data. *IEEE IV*, 0:447–452, 2007.

[27] K. Koffka. *Principles of Gestalt psychology*. Oxford, England: Harcourt, Brace, 1935.

[28] Z. Liu, D. W. Jacobs, and R. Basri. The role of convexity in perceptual completion: beyond good continuation. *Vision Research*, 39(25):4244 – 4257, 1999.

[29] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[30] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE TVCG*, 12:733–740, 2006.

[31] P. Simonetto, D. Auber, and D. Archambault. Fully automatic visualisation of overlapping sets. *Comput. Graph. Forum*, 28(3):967–974, 2009.

[32] A. Spoerri. Infocrystal: a visual tool for information retrieval. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 150–157, Washington, DC, USA, 1993. IEEE Computer Society.

[33] A. Verroust and M.-L. Viaud. Ensuring the drawability of extended euler diagrams for up to 8 sets. In *LNAI 2980*, pages 128–141, 2003.