A Task-Based Evaluation of Combined Set and Network Visualization

Peter Rodgers[a], Gem Stapleton[b], Bilal Alsallakh[c], Luana Michallef[d], Rob Baker[a], Simon Thompson[a]

[a] *University of Kent, Canterbury, UK*
[b] *University of Brighton, Brighton, UK*
g.e.stapleton@brighton.ac.uk
[c] *Vienna University of Technology, Vienna, Austria.*
[d] *Helsinki Institute for Information Technology, Aalto University, Helsinki, Finland*

Abstract

This paper addresses the problem of how best to visualize network data grouped into overlapping sets. We address it by evaluating various existing techniques alongside a new technique. Such data arise in many areas, including social network analysis, gene expression data, and crime analysis. We begin by investigating the strengths and weakness of four existing techniques, namely Bubble Sets, EulerView, KelpFusion, and LineSets, using principles from psychology and known layout guides. Using insights gained, we propose a new technique, SetNet, that may overcome limitations of earlier methods. We conducted a comparative crowdsourced user study to evaluate all five techniques based on tasks that require information from both the network and the sets. We established that EulerView and SetNet, both of which draw the sets first, yield significantly faster user responses than Bubble Sets, KelpFusion and LineSets, all of which draw the network first.

Keywords: Set visualization, graph visualization, combined visualization, clustering, networks.

1. Introduction

There has been a rapid rise in the volume of network data where the data items form overlapping groups. Data of this type arise in many situations such as in criminal investigations where networks represent relationships between those under investigation and groups represent organizations to which people belong or locations they frequent. Similar complex data occur in biological settings [18, 40] where the interaction between genes forms a network and genes with shared features are in common groups. In this paper, we focus on social networks, where networks represent connections between people, and groups represent shared interests [47].

Reflecting the importance of understanding such data, which we call *grouped network data*, there have been a number of techniques proposed for visualizing it, such as [3, 12, 35]. Existing layout methods typically draw either the data items first or the sets first, thus assigning *primacy* to one or the other. Primacy is acknowledged by Collins et al. [12], who introduce the concept of *spatial rights* in respect of the fact that spatial positioning can be "the most salient feature of possible visual encodings", backed up by [9]. Collins et al. state that set relations have spatial rights when the sets are the primary relations. Under these circumstances, a visualization of the sets is first found, after which the network is laid out. The network has primary spatial rights when it is visualized first, with the sets laid out afterwards.

Visualizing the sets first can lead to a poor network visualization. Similarly, drawing the network first can seriously compromise the layout of the sets. However, when primacy is assigned to the network, the visualization techniques can be applied more generally, in particular when the data items are required to take specific locations such as when data items are points on a map. At present, we have no insight into the relative effectiveness of visualization techniques that make different choices about which of the sets and the network has primary spatial rights. This paper addresses this gap in understanding by comparing the relative strengths of these approaches.

Fig. 1 illustrates the five visualization techniques that we evaluate. EulerView [41] and SetNet, introduced in this paper, both assign primary spatial rights to the sets but SetNet refines the layout of the sets to account for the network structure. Bubble Sets [12], KelpFusion [28] and LineSets [3] assign primary spatial rights to the network. The techniques visualize the sets in different ways, as explained in Sect. 3. All of the approaches represent networks with node-link diagrams. Our main contributions are:

- A theoretical analysis (Sect. 3) of Bubble Sets, EulerView, LineSets and KelpFusion using existing theories of effective set visualizations (Sect. 2).

- A novel technique, SetNet, that optimizes the set visualization using these theories (Sect. 4).

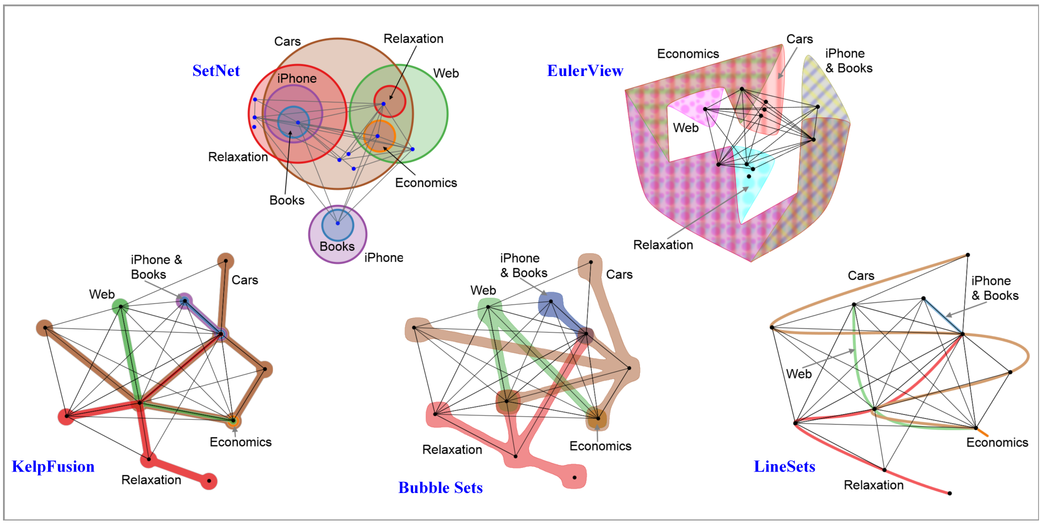

Figure 1: The five visualization techniques evaluated in this paper, depicting the same grouped network data about twitter networks.

- An empirical study (Sect. 5) to test the hypotheses that follow from the theoretical analysis. We evaluate the effectiveness of all five techniques when performing common tasks, derived from [38], that require both information from the networks and the sets. These tasks build on taxonomies for network-only tasks [2] and set-only tasks [5].

In Sect. 6 we present the results of our empirical evaluation, along with a discussion of our hypotheses and comparison with previous evaluations of grouped network data. Scalability issues associated with all five techniques are explored in Sect. 7. Sect. 8 concludes our work and summarizes the limitations of our study along with possibilities for future work. The Set-Net software is available from [1] along with all of the study material and a video preview of our work. This includes the diagrams, questions and the performance data collected. The study's diagrams are also available as supplementary material.

2. Principles in Psychology and Layout Guides

This section summarizes relevant theories and guides that allow us to analyze the qualities of the evaluated visualization techniques. We have omitted a detailed discussion around graph layout choices and their impact on comprehension; these are widely studied in contrast with similar choices for the visualization of sets.

2.1. Gestalt Principles and Well-Matchedness

The following is a summary of two general Gestalt principles [26] as well as the related principle of well-matchedness [19] that apply to the evaluated visualization techniques.

Principle of Proximity This principle states that people tend to perceive objects that are close together as part of the same group. Likewise, objects that are far apart are perceived to be in different groups. Visualizations that meet this principle can reduce the time it takes for people to distinguish groups because they process a smaller number of stimuli. Consequently, a visualization meeting this principle is expected to be easier to internalize and understand.

Applying this principle to grouped network data requires nodes in a common group to be located in close proximity to one another, while nodes not in that group should be located further away. Visualization techniques that give primary spatial rights to the network may not consider the principle of proximity: the placement of the nodes need not correspond to the groupings of the data items. By contrast, when the sets have primary spatial rights, it should be possible to locate data items in common groups so that they are in close proximity.

Fig. 2 illustrates this issue with two hand-drawn diagrams, where the nodes are represented by numbers to ease comparison. The diagram on the left adheres well to the principle of proximity: the nodes in common groups are drawn close to each other. By contrast, the diagram on the right has located the node '5' far from the other two nodes in the group called A.

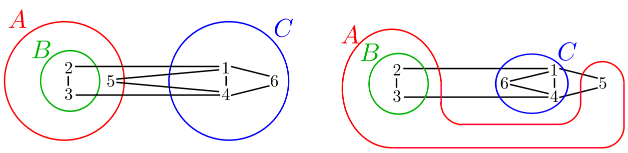

Figure 2: Well-matched diagrams and the principle of proximity.

Principle of Good Form There is a tendency for people to group together graphical objects that share some property, like colour, pattern, or shape. Similarly, graphical objects that have different properties will typically be considered, by people, as being in different groups. This suggests that different semantic constructs should be represented by different graphical objects. Graphs, when representing networks, meet this principle: items are all represented by nodes and relationships are all represented by edges. It would also seem sensible to represent sets with common syntax and colour could also be used if we want to give visual clues about semantic commonality or distinctness.

Principle of Well-Matchedness Gurr proposed that visualizations which are *well-matched* to their semantics are effective [19][1]. A notation is well-matched when its spatial relationships directly mirror the semantic relationships being visualized. In our context, a well-matched visualization of sets would ensure that subset, disjointness and intersection relations are represented by enclosure, non-overlap, and overlap respectively. In terms of networks, their representation using graphs can be considered well-matched: a connection at a syntactic level (node adjacency) corresponds to a connection at the semantic level. When combining visualizations of sets and networks, well-matchedness will be achieved when element containment and exclusion in sets at the semantic level is mirrored at the syntax level.

In Fig. 2 both diagrams are well-matched: subset ($B \subseteq A$) and disjointness ($A \cap C = \emptyset$ and $B \cap C = \emptyset$) are visualized by curve containment and non-overlap respectively; the nodes are inside the curves corresponding to the sets containing the represented elements. Syntactically, edges connect nodes that are semantically connected, and their absence indicates no semantic connection. In Fig. 2, for example, 1 is connected to 2 but not to 3.

2.2. Layout Guides for Region-Based Visualization

Sets are often visualized using closed curves whose spatial relations convey information about the sets' relationships. The regions within such curves represent the corresponding sets. Such notations are commonly known as Euler diagrams which are often incorrectly called Venn diagrams: in Venn diagrams, for each subset of the curves, there must be a non-empty region inside those curves but outside the remaining curves. Thus, every Venn diagram is an Euler diagram, but not vice versa. Euler diagrams support the interpretation of grouping information since elements in a common set are located in the same region [31].

[1] This is similar to epistemic fidelity theory described in [23].

There have been a number of empirical studies conducted to ascertain the effect of different Euler diagram layout choices on user comprehension which provide a starting point for effectively laying out Euler diagrams with graphs. We summarize these results, presented as a series of layout guides by Blake et al. [10]:

Well-formed Draw Euler diagrams that are well-formed [37]. The well-formedness properties include: only simple curves, sets represented by unique curves, no concurrency between curves, no 3-points, and curves that cross and do not brush[2]. The sixth well-formedness property is that diagrams must ensure that the regions which represent set intersections are connected. An Euler diagram is well-formed if it possess all six properties.

Smooth Curves Draw Euler diagrams with smooth curves [7].

Diverging Lines Draw Euler diagrams with diverging lines which means that the crossing angle at points of curve intersection approaches 90 degrees [7].

Shape Draw Euler diagrams with circles [10][3].

Symmetry Draw Euler diagrams with highly symmetrical curves [10].

Shape Discrimination Draw Euler diagrams such that the regions are discernable from the curves via their shape but not at the expense of symmetry [10].

Meeting the Shape guide implies that the Symmetry and Shape Discrimination guides are also met.

[2]For space reasons, we omit the formal definitions which are not necessary for this paper; they can be found in [43]. Briefly, the properties are as follows. Simple curves: no curve self-intersects. Unique curves: no set is represented by more than one curve. No concurrency: no pair of closed curves intersect in a non-discrete set of points. No 3-points: no points are passed through more than twice by the curves. Only crossings: whenever two curves intersect, they cross each other. Connected regions: the regions which represent set intersections, typically called zones, are connected components of the plane.

[3]It is observed in [10], when using circles to draw Euler diagrams, that the regions formed from intersecting circles have non-circular, non-smooth boundaries, promoting shape discriminability (i.e. the regions are discriminable from the circles). Moreover, the regions formed when intersecting rectangles are themselves rectilinear, inhibiting shape discriminability. The authors of [10] further observe, using similarity theory [13], that the reduced ability to discriminate implies the use of rectangles leads to worse performance as compared to circles. These insights support the results of the empirical study in [10], which establishes that circles outperform rectangles.

3. Visualizing Grouped Network Data

The primary purpose of this section is to introduce and theoretically analyze the four techniques that we evaluated: EulerView [41], Bubble Sets [12], LineSets [3], and KelpFusion [28]; they are illustrated in Fig. 1. Whilst the networks and sets were automatically drawn, we manually added the labels to all visualizations included in the paper.

3.1. EulerView (EV)

EulerView [41] uses closed curves to represent sets, assigning primacy to the sets. The method converts a description of the non-empty set intersections into a graph, called a *zone graph*, from which the Euler diagram is created. The method, in essence, fattens the edges of the zone graph and then uses these fattened edges to route the curves. The network is then visualized in the drawn Euler diagram. EulerView uses a colour fill inside the curves and is the only technique discussed here that also uses a texture to aid identification of the set intersections.

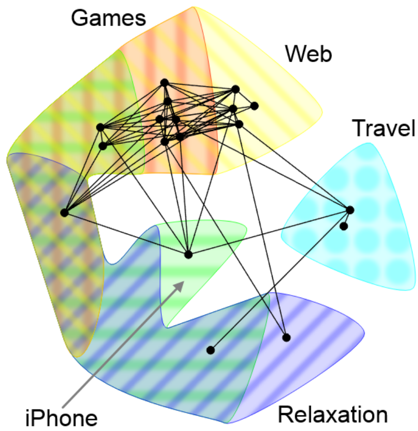

Figure 3: An EulerView visualization of grouped network data.

As EulerView assigns primacy to the sets, the principle of proximity is adhered to, as seen in Fig. 3. The data items inside only 'Games' and 'Web' are located in close proximity. One could also argue that the principle of good form is met by EulerView. Here, each set has its own colour and texture. The use of the texture through the different regions that, between them, represent the entire set is designed to help user interpretation. For example, in Fig. 3, the texture inside 'Relaxation' (diagonal lines) is crucial, in our opinion, for determining that there are three regions that make up this set.

EulerView also produces well-matched diagrams: in an Euler diagram, well-matchedness is achieved when all of the regions correspond precisely to the non-empty set intersections. However, the diagrams are far from well-formed: sets are not necessarily represented by unique curves; curves are nearly always concurrent; points can be passed through an arbitrary number of times by the curves; when curves meet there is no guarantee that they cross; and the region that represents the complement of the represented sets can be disconnected. Moreover, the curves do not diverge when they intersect and they are not circular or, generally, symmetric. Lastly, the regions to which the curves give rise take similar shapes to the curves so the shape discrimination guide is not met either.

Fig. 3 exhibits concurrency (e.g. Relaxation is concurrent with iPhone, Games and Web), a 4-point (e.g. where the four just mentioned curves all meet), and wherever any pair of curves meet they do not cross. None of the curves are circular and they generally lack symmetry. The regions in this diagram are often irregular shapes, like the curves.

3.2. Bubble Sets (BS)

Bubble Sets is a flexible technique because it can be applied to a variety of domains [12]. The method routes closed curves around the nodes of an already drawn graph to form 'bubbles', assigning primacy to the network. Fig. 4 shows an example, which represents the same data as Fig. 3. Thus, Bubble Sets can be seen as using Euler diagrams augmented with graphs to represent grouped network data.

Bubble Sets aim to satisfy the following:

(a) for each set, all set members are to be enclosed by the closed curve representing that set,
(b) for each set, non-set members, if possible, are to be excluded from the closed curve representing that set,
(c) where non-members appear within curve boundaries, visual and interactive hints can be given to clarify membership, and
(d) to provide rendering to allow for interactive adjustment.

Thus, from (b), we can see that Bubble Sets can, in principle, produce incorrect visualizations. As our empirical study is based on static layouts, we do not discuss (d) or the interactive aspect of (c) any further.

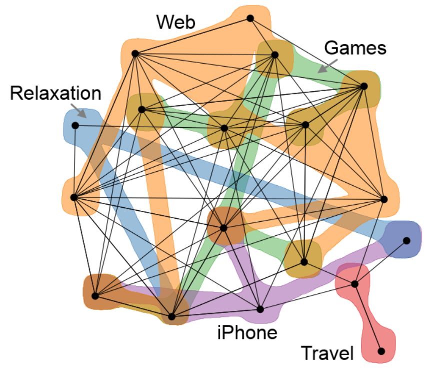

Figure 4: Routing curves around set members in Bubble Sets.

As Bubble Sets layout starts with an already drawn graph, there is no guarantee that visualizations will adhere to the principle of proximity. For example, in Fig. 4, the four data items that are in only the set 'Web' are not drawn in close proximity. The relationship between 'Games' and the other sets is unclear. Perhaps more obviously, Fig. 5 shows that the Bubble Sets technique can give rise to visualizations where set membership and exclusion is hard to identify. Even though a unique colour has been used for each set, in the overlaps the colours blend so identifying set membership could be difficult for users. Thus, whilst Bubble Sets attempt to adhere to the principle of good form, the technique has features that are potentially problematic.

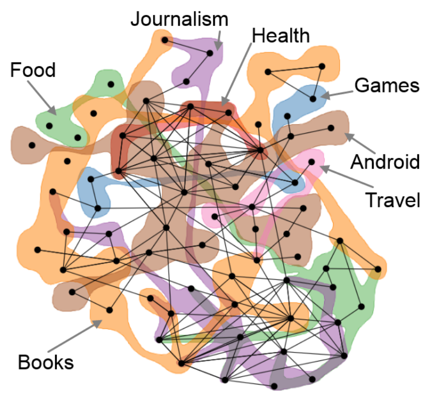

Figure 5: Problematic layout features of Bubble Sets.

Bubble Sets can fail to deliver well-matched diagrams in two respects. First, they can include additional regions that represent empty sets. This

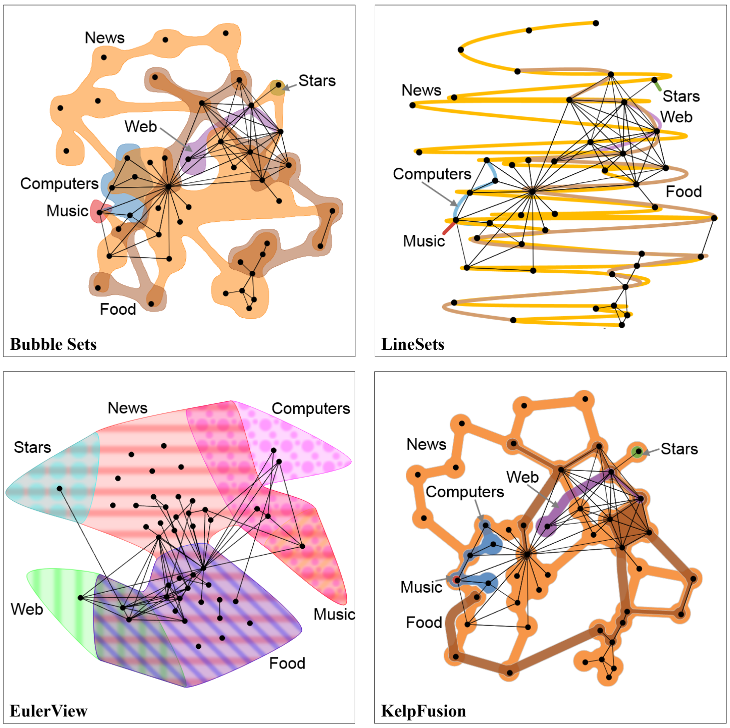

Figure 6: Visualization errors in Bubble Sets and LineSets, correctly displayed by EulerView and KelpFusion.

can be seen in Figs 4 and 5. However, as the data items are explicitly represented, the likelihood of confusion is perhaps reduced. More concerning, though, is the inclusion of data items in sets of which they are not members; this possibility is acknowledged by property (c), although attempts are made to avoid it (see property (b)). This non-well-matched feature will cause misinterpretation. Similar erroneous behaviour can be seen in Fig. 6: there is exactly one node inside only 'Computers' in the Bubble Sets visualization yet there should be two such nodes, which is correctly displayed in the EulerView visualization.

Bubble Sets can also fail to be well-formed. In our figures, the following are exhibited: concurrency between curves, points passed through more than twice by the curves, curves meeting but not crossing, and disconnected regions. For example, in Fig. 4, the region inside 'Web' and 'Games' but outside the other curves comprises multiple components and it is not immediately obvious how many data items are in the intersection. Also, the curves are not always diverging, they are not circular or, generally, symmetric. Lastly, the boundaries of the regions are similar in shape to the curves themselves so the shape discrimination guide is not met.

3.3. LineSets (LS)

LineSets were introduced to overcome problems associated with visual clutter that is seen in Euler diagrams [3]. As the name suggests, LineSets use lines to represent sets, rather than closed curves. Clutter associated with intersecting sets, arising from overlapping graphical objects, is reduced through the use of lines. LineSets assign primacy to the network and their layout begins with a distribution of already drawn nodes. An example can be seen in Fig. 7, which represents the same data as the EulerView and Bubble Sets visualizations in Figs 3 and 4. LineSets aim to avoid self-intersecting lines and to minimize bends in the (smooth) lines used to represent sets. As with Bubble Sets, the LineSets implementation has interactive features that we do not consider in our study. Moreover, the LineSets implementation does not draw lines for sets containing exactly one element. An example can be seen in Fig. 1, where the line representing 'Economics' was manually drawn after the rest of the visualization had been automatically produced.

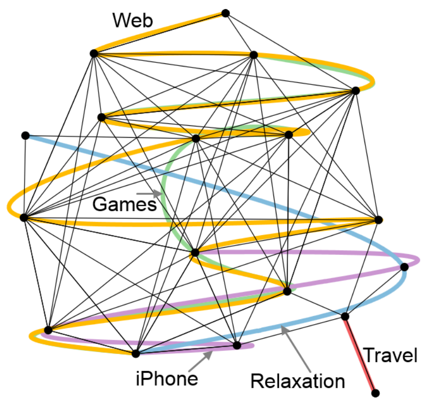

Figure 7: Using lines to visualize sets with LineSets.

As the LineSets technique starts with an already drawn graph there is no guarantee that visualizations adhere to the principle of proximity. This can be observed in Fig. 1, where the 'Web' line must join up three nodes that are not located in close proximity. Considering the principle of good form, LineSets use lines to represent *both* connections between items and set membership, which is against this principle. However, as colour is used to distinguish sets from connections it is unclear whether LineSets meet the principle of good form.

In terms of well-matchedness, the set theoretic properties of subset, disjointness and intersection are not visualized by spatial relationships. This is because the spatial positioning of lines is not important. Rather, it is the topological property of intersection (or otherwise) between lines that indicates shared membership or disjointness. For instance, in Fig. 6, 'Food' is a subset of 'News'. However, because lines do not permit direct visualization of containment, this aspect of LineSets is not well-matched. For intersection, one might expect lines connecting common elements to be visualized as a concurrent segment in order to be closer to well-matched. Lastly, whilst lines pass through nodes in the represented sets, they can also pass through nodes that are *not* in the set. This can be seen in Fig. 6, where one node in the LineSets visualization that should be in only the set 'Computers' is also on the line for 'News'. As LineSets are not based on Euler or Venn diagrams, we do not consider the guides in Section 2.2.

3.4. KelpFusion (KF)

KelpFusion is a blend of curve-based and line-based techniques for visualizing sets around an already drawn graph and, so, assigns primacy to the network. Sets are represented by fattened paths that can include cycles. An example can be seen in Fig. 6: the set 'News' is represented by the orange 'fattened' edges and contains cycles. Unlike LineSets, these fattened edges properly contain some of the graph edges that represent connections between the data items in the visualized set.

KelpFusion's adherence to the principles of proximity, good continuation, and good form is much the same as for LineSets. However, KelpFusion sometimes overcomes the non-well-matchedness of LineSets: by using fattened edges to represent sets, KelpFusion *can* exploit spatial relationships to assert subset, disjointness and intersection. However, the layout method does not ensure that well-matched visualizations are produced. This can readily be seen on the right of Fig. 6, where 'Food' is a non-contained subset of 'News'. Lastly, as KelpFusion does not directly use Euler diagrams as a basis, we do not compare its layout properties with the guides in Section 2.2.

3.5. Other Notable Techniques

An early attempt to combine sets and networks in a single visualization relied on first drawing an Euler diagram then placing a graph inside it [30], however the sets were often visualized with convoluted, difficult to follow curves. In addition, only limited kinds of set data could be shown as the system was limited to well-formed Euler diagrams. Compound graphs can be used to represent restricted kinds of grouped network data [8]. Graph clusters are visualized with transparent hulls by Santamaria and Theron [39]. However, the technique removes edges from the graph and it is not sufficiently sophisticated for arbitrary overlapping sets. Itoh et al. [24] proposed to overlay pie-like glyphs over the nodes in a graph to encode multiple categories. Each set is hence represented using disconnected regions that are linked by having the same colour. This causes difficulties with tasks that involve finding relations between sets such as T1, T3 and T4 in Section 5.3. A related class of techniques visualize grouping information over graphs using convex hulls, such as Vizster [22]. However, they do not support visualizing set overlaps.

Riche and Dwyer introduced two Euler diagram-based techniques, ComEd and DupEd, designed to visualize sets using simple regions and to draw individual data elements as text-annotated nodes [35]. Unlike the methods described in detail above, Riche's and Dwyer's two methods lay out the network with regard to the set structure. ComEd represents each set as one or more rectangles connected with concave curves, assigned a unique colour. The use of rectangles and irregular curves could impose cognitive difficulties in perceiving the sets, as they might give the impression of different semantics [45]. Another issue with ComEd is the artifacts caused by overlaps between the curves that connect the rectangles. These overlaps do not correspond to actual set intersections which impacts well-matchness properties, and might impair usability [37]. 'DupEd' uses only rectangles to represent sets which is a shape with good properties. However, the visualizations do not use the spatial properties of containment and disjointness to represent subset and set disjointness information, so the diagrams are not well-matched. This leads to elements being represented by duplicated nodes joined by edges to assert identity.

It is possible to visualize grouped network data using ComEd and DupEd by drawing edges between the nodes [35]. An incremental constrained graph layout, IPSep-CoLa [14], has been employed to create such a visualization by defining the grouping of the nodes into sets as constraints on the graph layout. This results in a balanced layout with respect to sets and to the network structure. Our new technique, SetNet, shares this property with ComEd, as explained in Sect. 4. However, instead of using the set information to adapt the graph layout, it adapts the set layout to account for the network information. Moreover, it uses circles to represent the sets, instead of rectangles connected with concave curves. Further techniques for visualizing similar data are described in [18, 40, 44] and a recent survey on set visualizations [5].

3.6. Previous Empirical Comparisons

Bubble Sets, KelpFusion and LineSets, which assign primacy to data items, have been subject to rigorous empirical evaluation. Alper et al. [3] evaluate LineSets as compared to Bubble Sets with three categories of grouped network data: 50 items, 3 sets and 5 set intersections; 100 items, 4 sets and 10 set intersections; and 200 items, 5 sets and 30 set intersections. They did not indicate how many connections existed between the items (or, therefore, how many edges were in the corresponding graphs). The tasks that users were asked to perform focused on set-theoretic properties and were of the form: *how many sets are there?*; *which of these two sets is largest?*; *which sets do both x and y like?*; and *how many elements of this set are also in these other sets?* LineSets were shown to yield significantly more correct answers and to permit significantly faster responses than Bubble Sets. However, the tasks did not utilize both the network and the set information in conjunction, which perhaps explains why there was no mention of the number of edges in the graphs.

Meulemans et al. evaluated KelpFusion against both LineSets and Bubble Sets [28]. Their study used visualizations of grouped data (with no network information) with the following characteristics: 4 sets with 15 to 49 items; 5 sets with 12 to 29 items. As this study focused on data items that were locations on a map, connection information through a graph-based visualization was not present. As such, the tasks again were focused on set-theoretic concepts, such as cardinality; e.g. *How many elements are there in these two sets?* (paraphrased here). The study found that KelpFusion and LineSets both yielded significantly better accuracy than Bubble Sets. With regards to completion time for the tasks, KelpFusion performed fastest, followed by LineSets and, lastly, Bubble Sets. Thus, overall, KelpFusion was the most effective visualization technique.

A further related study has been conducted to compare four techniques for displaying group information over graphs [25]: coloured nodes, Bubble

Sets, LineSets and GMap [17]. The error performance of these techniques was compared on ten tasks, grouped into: 4 group tasks, 2 network tasks, 3 combined group-network tasks, and one task on the memorability of node locations. Bubble Sets was the most effective technique for group tasks. Similar to our study, the combined group-network tasks involved questions on node degree in a specific group and tracing paths over groups, although the tasks themselves were different. In [25], each node belonged to only one group and, in addition, the groups were all disjoint. This meant that no set overlaps were present. By contrast, for our study the nodes belonged to multiple groups and there were richer relationships between the groups such as subset and intersection.

3.7. Summary

The visualization techniques all have limitations and the degree to which they impact comprehension is only partly known. Tables 1 and 2 summarize which of the principles and layout guides are met for the five evaluated visualization techniques[4] (SN is SetNet introduced in the next section). The entries are interpreted as follows: '×' the property is not possessed; '✓' the properties is possessed; '?' it is unclear whether the property is possessed. We have place a '?' against LineSets and KelpFusion for the well-matched property. In the case of LineSets, this is because they use lines for set containment and for network connections, which can be seen as breaking the property. However, the sets are coloured whilst the graph is not, so it is unclear whether this means that LineSets, are in fact, well-matched. In the case of KelpFusion, it is unclear whether the method ensures that the regions present in the diagrams are precisely those that represent non-empty sets. For this reason, we are unable to determine whether KelpFusion guarantees well-matchedness.

Table 1: How the evaluated techniques meet the general principles and guides.

	BS	EV	KF	LS	SN
Proximity	×	✓	×	×	✓
Good Form	✓	✓	?	?	✓
Well-Matchedness	×	✓	×	×	×

[4]It is known that no Euler diagram drawing method which is capable of visualizing any finite collection of sets can ensure all six well-formedness properties hold as well as being well-matched [16].

Table 2: How the evaluated region-based techniques meet the specific principles and guides.

	BS	EV	SN
Well-formed: Simple Curves	?	✓	✓
Well-formed: Unique Curves	?	×	×
Well-formed: No Concurrency	×	×	✓
Well-formed: No 3-points	×	×	✓
Well-formed: No Brushing Curves	×	×	✓
Well-formed: Connected Regions	×	×	✓
Smooth Curves	✓	✓	✓
Shape	×	×	✓
Diverging Lines	×	×	✓
Symmetry	×	×	✓

The empirical studies described in section 3.6 have two primary limitations. First, the compared visualization techniques all assign primacy to the data items. By contrast, the study in this paper compares five techniques, three of which assign primacy to the network and two that assign primacy to the sets. Second, the tasks undertaken by the participants in the first two studies focus on set-theoretic questions, with the exclusion of questions that rely on the combination of both the network connections. If one is only interested in information about the sets (or the network) then simpler visualization techniques can be used. The third study did not have any set overlaps. These are serious limitations. Thus, it is important to compare these visualization techniques in the context of tasks that require information from both the network and the sets when overlaps are present, as we do in this paper.

Lastly, of the four previously devised techniques that we evaluate, the only technique that assigns primacy to sets is EulerView. In section 3.1, we argued that there are significant problematic features of EulerView that could compromise its effectiveness as a visualization of grouped network data. This motivates our development of SetNet, which we introduce in the next section.

4. The SetNet Visualization Technique

It is not possible for any visualization technique that exploits Euler diagrams to meet all of the principles and guides in Table 2 whilst being able to visualize any grouped network data. However, it is certainly possible to meet more of them than existing techniques: our novel technique, SetNet (SN), meets 11 of them, as summarized in Tables 1 and 2.

4.1. The Method

To visualize the sets, SetNet builds on an Euler diagram drawing technique [42], called iCircles. The iCircles method draws a single or multiple circles to visualize each set. Since iCircles is capable of visualizing any finite collection of sets, it enables the representation of complex overlapping groups whilst preserving the simplicity of the region shapes. An extension of iCircles to include graphs has already been given in [36], which we further extend in this paper.

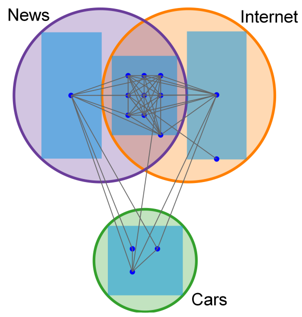

Figure 8: The initial layout.

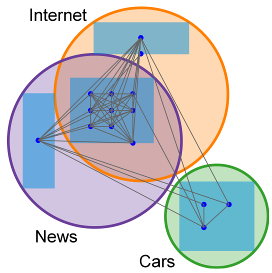

Figure 9: The layout after circle relocations.

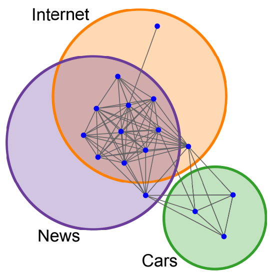

Figure 10: The final layout after optimizations.

Firstly, SetNet draws the circles using iCircles, giving the sets primary spatial rights. See [42] for details of how the iCircles tool works. To add the nodes, the largest rectangular area within each region is found. The nodes are drawn in a grid pattern within this region, giving an initial layout. Nodes outside all circles are placed at a suitable distance from all circles. Fig. 8 illustrates the process where the rectangular regions are superimposed.

The new aspect of SetNet is that it reduces the potential negative impact of set primacy on the network visualization[5]. The technique improves on [36] by adding a processing step. The aim is to adjust the layout of the Euler diagram to account for the inclusion of the nodes. The location of each circle is moved using a hill climbing search. On each iteration, the method tests alternative surrounding locations for each circle in turn and picks the best position based on a fitness score. If none of the alternative positions is better than the current location, the circle is not moved. The surrounding locations are eight points on a square with the circle at its centre. Iterations continue (with a decreasing size of square) until no more improvement can be found. Moves that place circles too close together or that change the relative topological arrangement of the circles are not performed.

To prevent the topological arrangement from changing, the system performs 'structure checking' each time a circle is moved. This test establishes whether the moved circle still intersects with all and only the same circles as before, for instance. A more detailed description on how structure checking is performed is in [46].

An outline of the algorithm used by SetNet is given here:

1. Parse the data to create a list of nodes, edges and regions to be drawn.
2. Send the list of all regions to iCircles and await reply.
3. For each circle given by iCircles, draw the circle on screen.
4. For each circle, start by setting the best score to be the fitness score (see below) for the circle with its current centre point and then:

    for each point in grid around circle centre:

        Move the circle centre to the point.

        If the structure of the diagram has not changed then calculate the fitness score. If this score is smaller, then this score becomes the best score and the point becomes the best circle centre, otherwise the best score and best circle centre are unchanged. Move the circle so that its center point is the best circle centre.

    Otherwise, keep the center point the same.

5. For each region[6] formed by the circles, find the largest rectangle in this region and arrange the relevant nodes in a grid pattern in this rectangle.

[5]Note that the techniques in [35] attempt to reduce the potential impact of assigning primacy to one component of the visualization as well.

[6]Here, by region we strictly mean zone.

6. For each edge, draw the edge between the relevant nodes.
7. For each region, apply a force directed layout to the nodes and edges in the region.

Fitness is measured by a weighted sum of two criteria: firstly, the square of the distance between connected regions and, secondly, a rating of the area of each region. The details of these calculations are given the in enumerated list below. Calculating region area may require finding the area of two circle intersections and finding the area left after the area of any contained circles are removed. The fitness score itself and the parameters for the aspects of the layout method were developed by ad-hoc experimentation, before running the SetNet layout method on the test data given in this paper.

A sketch of the algorithm used to compute the fitness score is given here:

1. **Fitness score:** This is defined to be $\mathit{ClosenessRating} + \mathit{AreaRating}$.
2. **Closeness Rating:** Each circle has an *individual closeness rating* and the sum of these is the closeness rating. Given a circle, to compute its individual closeness rating, for each region inside it and for each region that is topologically adjacent to the original region, calculate the distance between the largest rectangles inside them and square it; sum these distances for the circle.
3. **Area Rating:** Each circle has an *individual area rating* and the sum of these is the area rating. Given a circle, to compute its individual area rating, for each region inside it, calculate

$$\mathit{AreaOfRegion} - ((1 + \mathit{NumberOfNodesInRegion}) \times \mathit{AreaWeighting})$$

and that the sum of the square of these values. The area of each region is approximated by the area of its largest contained rectangle. The number *AreaWeighting* is a constant set to 200, a value derived through informal experimentation.

Continuing with the initial layout in Fig. 8, the result of this hill climbing search is depicted in Fig. 9. A key difference is that the region inside both 'News' and 'Internet' has enlarged, reflecting the fact that the number of nodes located here is relatively large. As a consequence, the two regions inside 'News' only and inside 'Internet' only respectively have reduced in size, congruent with the fact that they both contain fewer nodes. Once an improved layout has been determined, the final processing step is a force-directed layout of nodes. This is a standard spring embedder [15], with linear attractive forces and the addition of a repulsive node-circle force that pushes nodes away from circle borders. For each node and circle, a vector of inverse square distance is calculated (much like the node-node repulsion in the standard spring embedder). The distance is between the node centre and the nearest point on the circle to the node. The direction of the vector is away from the circle centre if the node is outside the circle, otherwise it is towards the circle centre. The node-circle force for a node is calculated from the sum of these vectors for all circles. We note that if the number of circles is less than the number of nodes (which is typically the case), then this force should be calculated more quickly than the node-node repulsion. So this new force, although adding some extra computation time, does not have a major adverse effect on performance.

When applying this force layout, nodes are not allowed to leave their region, but may move outside the starting rectangle. This is achieved by preventing node movements that would place them in the wrong region. As a result, nodes are inside precisely the curves corresponding to the sets that include the represented elements (unlike Bubble Sets) and that they do not cluster tightly on the inside of circles.

The final visualization of our running example can be seen in Fig. 10. The nodes have been moved so that the edge lengths are more uniform. However, this approach is not guaranteed to remove unnecessary edge crossings. It can also result in nodes being extremely close to edges with which they are not incident or close to circles. An example of a visualization produced using SetNet is in Fig. 11, which represents the same data as Fig. 6.

4.2. Meeting the Principles and Guides

We consider the principles and guides in the context of SetNet, summarized in Tables 1 and 2. Because the circles are used to group the nodes, data items within common set intersections are drawn within close proximity. For example, in Fig. 11, one can see that the nodes representing items in the set 'Food' (and not in any other sets) are located near to each other. In Fig. 11 it is hard to follow the edges at points where they intersect, in part because of the density of some of the crossing points. Whilst edge crossings are not always avoidable when laying out graphs, forcing nodes to take particular locations – which occurs when primary spatial rights are assigned to the curves – is likely to increase their occurrence.

The use of iCircles means that some data cannot be visualized in a well-matched fashion. Circles impose geometric constraints on the representation

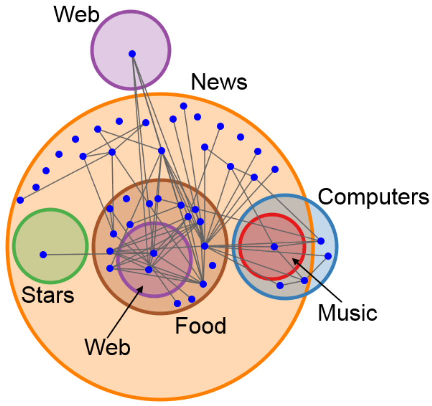

Figure 11: A complex network visualized using SetNet.

of sets which sometimes requires additional regions that represent empty sets to be included, thus violating well-matchedness. This occurs in Fig. 1, where 'Book' is drawn inside 'iPhone'. In fact, these two sets are equal (they contain exactly the same elements) and so the circles could be drawn concurrently, similar to the Bubble Sets, EulerView and LineSets visualizations in Fig. 1. However, to avoid breaking this well-formedness property, iCircles arbitrarily chooses one circle to be drawn inside the other.

The iCircles tool ensures that nearly all of the well-formedness properties hold: circles are simple, there is never concurrency, no points are passed through more than twice by the curves, whenever two curves meet they cross, and the regions representing the sets are connected. However, sometimes sets are represented with more than one circle. This phenomenon arises in Fig.11, where 'Web' is represented by two circles. This is not a feature specific to SetNet as such: when using circles, sometimes sets have to be represented using multiple circles which brings with it usability problems. However, circles have the advantage of being a cognitively beneficial shape. The principle of good form is utilized to reduce any potential negative impact, by using colour: for any given set, all circles representing it adopt the same colour and no other circles share that colour. The visualizations also have diverging lines wherever two circles intersect. The guideline to use circles is clearly met as are the symmetry and shape discrimination guides.

5. Hypotheses and Experiment Design

The aim of our study is to identify which of the existing visualization techniques, along with SetNet, allows users to most effectively obtain information about grouped network data. We judge 'most effective' in terms of task performance measuring accuracy and time; the tasks we use are given in Section 5.3. Of these two performance measures we view accuracy as more important than completion time, consistent with other researchers, such as [3]. Thus, we view one technique as outperforming another, overall, as follows:

1. if technique A has a significantly higher accuracy rate than technique B then A outperforms B,
2. if techniques A and B do not significantly differ in terms of accuracy but technique A has a significantly lower mean time than technique B then A outperforms B.

Otherwise, there is no overall performance difference between the two techniques.

Our discussions in the previous sections lead us to the following hypotheses:

- **H1** Of the techniques that assign primary spatial rights to the network, we expect KelpFusion to outperform LineSets which, in turn, will outperform Bubble Sets.
- **H2** Of the techniques that assign primary spatial rights to the sets, we expect SetNet to outperform EulerView.
- **H3** Of non-Euler diagram-based techniques, we expect KelpFusion to outperform LineSets.
- **H4** Of the Euler diagram-based techniques, we expect SetNet and EulerView to outperform Bubble Sets.

H1 and H3 are derived from the previous empirical studies discussed in Section 3.6. H2 and H4 are derived from Tables 1 and 2. These hypotheses directly address our research problem of how to best visualize network data that are grouped into overlapping sets.

It is unclear which of the techniques will perform best overall in terms of accuracy and time. Our experiment design, described below, will allow us to provide a ranking of the visualization methods in terms of both time and accuracy data. To compare the relative effectiveness of these visualization methods, we adopted a between-groups design, with five groups each answering 16 questions: 4 question types × 4 repetitions. Of these 16 questions, four were used for training (one of each question type), leaving 12 questions for analysis (three of each question type). Using a between-groups design meant participants only needed to be trained to interpret one visualization technique. It would be unfeasible to train participants to interpret five different visualizations, as would be required if a within-groups design was used.

5.1. Data for Visualization

We derived eight data sets from Twitter ego-networks, obtained from the SNAP network data set collection [27]. This collection contains ego-networks of 1000 Twitter users. The nodes of each network are users connected to the ego user. The edges correspond to 'follow' relationships between these users. Additionally, a number of user groups (i.e. social circles) are available for each network, representing users that subscribed to the same Twitter list. We selected eight ego-networks that contain at least three groups, with varying number of nodes (people) and edges (follows relations), summarized in Table 3; the task types mentioned in the table are explained later. In Table 3, the *i*-set intersection rows indicate the number of sets that contain data items that are in exactly *i* sets[7]. We excluded people that belong to no group in order to reduce the complexity of the respective graphs and to focus on set information defined by the social circles as this is a key aspect of all the visualization techniques.

Table 3: Data set sizes used in the study.

Data Set	1	2	3	4	5	6	7	8
Nodes	15	45	11	19	15	16	18	64
Edges	83	121	42	85	59	119	77	162
Sets	3	6	6	7	3	3	5	7
1-set intersections	2	3	3	4	3	3	4	6
2-set intersections	3	3	1	0	2	1	2	5
3-set intersections	0	2	2	1	1	0	1	0
4-set intersections	0	0	1	0	0	0	1	0
Total intersections	5	8	7	5	6	4	7	11
Task Type	T1 T3	T2 T4	T1 T3	T2 T3	T1 T4	T2 T4	T2 T3	T1 T4

[7]As with the study in [28], our data includes up to a maximum of 4-set intersections.

As study participants need not be familiar with Twitter, the questions used in the study were not presented in this context. Moreover, to avoid any possibility of previous knowledge of the data impacting the results, all set names were changed, but keeping a real-world scenario: people in a social network and their interests. For example, in Fig. 1, all of the visualizations show that there is only one person interested in both 'Books' and 'iPhone' and this person has connections with five other people[8].

5.2. Visualizations Used in the Study

Using the eight data sets, we generated visualizations using each of the five techniques; all of the diagrams are available from [1] as well as in the supplementary material. The process of producing the visualizations was kept as consistent as possible across techniques, as follows:

Graphs The graphs were drawn in black, except for SetNet which has blue nodes and black edges. In order to avoid problems of clutter in all visualization methods, no node labels were included as they were not necessary to complete the tasks. For the techniques that assign primary spatial rights to the network, the graphs were laid out using Gephi's ForceAtlas 2 layout algorithm [6]; the same graph layout was used for these three techniques. Gephi was not used for EulerView or SetNet, which assign primary spatial rights to the sets; they use their own graph layout method.

Sets The colour assigned to sets was the same across visualizations, except for EulerView which has a colour selector over which we had no control. For the remaining techniques, a palette of colours was generated using a qualitative colour scale from colorbrewer2.org [20].

Set Names The names were all coloured black and they were manually placed closest to the curve or line that they were labelling. If there was a possibility of ambiguity when placing the names (i.e. potential for confusion about the syntactic item being labelled), an arrow was added to indicate the syntactic item being named. When two or more equal sets were represented by a single curve or line, the label was the conjunction of the names of the sets, such as 'iPhone & Books' in Fig. 1 (see EulerView, Bubble Sets, LineSets and KelpFusion).

[8] Although 'follows' is a directed relationship, the visualizations just use undirected edges to indicate that a connection exists between people. This decision reduced visual complexity and removed the need to teach participants the meaning of *directed* edges.

For each visualization technique, in order to generate 16 visualizations, the original eight visualizations were rotated (as with LineSets in [3]). The rotation was chosen to be 140 degrees, an oblique value randomly generated between 60 and 320. The eight rotated visualizations had different names for the sets than those used in the original version, in order to reduce learning effect. We expected that participants would not recognize the rotated visualizations, evidenced in part by Plimmer et al.'s study were participants saw rotated graphs but, when asked, were unaware of this fact [33]. Lastly, all of the visualizations in this paper, except for Figs 2, 8, 9 and 10, are scaled versions of those used in the study, although some labels have been resized and moved for readability in the paper.

For SetNet and EulerView, the sets, nodes, and edges were completely automatically generated; the SetNet software was developed by us and the EulerView software was provided to us by the developers. For BubbleSets, KelpFusion, and LineSets, the nodes and the edges were generated using Gephi and exported as SVG files. The node coordinates were provided to Bubble Sets and LineSets using the available software. The developers of KelpFusion produced the diagrams used in the study, after we have them the Gephi-generated graphs. After generation, the Gephi-generated nodes and edges were laid over the generated Bubble Sets, LineSets, and KelpFusion diagrams.

We note that Bubble Sets, KelpFusion, and LineSets have been implemented in an interactive environment. However, in this study we used static visualizations since many uses of visualizations are not interactive, such as diagrams used in printed documents or on websites. Using static visualizations also allowed us to evaluate all five visualization techniques, as not all methods are integrated into interactive software. If we were to mix interactive and static visualizations, the results of the study would become less clear as any effects might be due to the visualization, the interface or the functionality provided. Finally, by using static visualizations, we can be confident that the effects we see are a result of only the visualization.

5.3. Tasks to be Performed by Participants

Previous work has provided a classification of tasks on grouped network data, according to the information required to solve them: group-only, group-node, group-link and group-network [38]. Our tasks fall into the group-link (T1 and T3 described below) and group-network class (T2 and T4). This means that each of our tasks required participants to use both the set and network information to perform the task, justified by the fact that this is the type of information visualized by the notations in the study. We did not include any tasks from the group-only or group-node classes. If one is only interested in information about sets (with corresponding tasks falling in to the group-only class) then one does not need to visualize the network. A similar point holds for group-node tasks. We take this view because simpler visualizations (e.g. those that do not visualize the network, only the sets) can be used for simpler tasks. Hence, group-link and group-network tasks are those most appropriate for evaluating visualizations of grouped network data.

We felt that it was important to have a diversity of group-link and group-network tasks in order to provide a rounded insight into the relative performance of each visualization technique. Moreover, we also wanted our choice of tasks to be practically relevant. With this in mind, we appealed to [47], which presents tasks that people need to perform when analyzing social networks. Inspired by these tasks, we devised four types of questions that arise when extracting information about *both* the network and the sets:

T1 *How many people with interests in X have n connections to other people?* This task is about examining how many direct connections people have, measuring their network activity with a focus on particular interest groups.

T2 *What are the interests of the person who, if removed, leaves n people disconnected from all the other people?* This task identifies the interests of key players in the social network [11].

T3 *How many direct connections are there between people interested in X and people interested in Y?* This measures the social cohesion between two groups [34].

T4 *What is the fewest number of people you need to pass through to get from people interested in X to people interested in Y? (Do not include the people at the start and end of the path).* This identifies the shortest path between two interest groups.

An instance of each question type is as follows:

T1 How many people with interests in only Hifi or people with interests in all of Hifi, Internet and Android have exactly three connections to other people?

T2 What are the interests of the person who, if removed, leaves at least two people disconnected from all the other people?

T3 How many direct connections are there between people interested in all of Economics, Cars and Web and people interested in all of Relaxation, Cars and Web?

T4 What is the fewest number of people you need to pass through to get from people interested in only Web to people interested in only Computers? (Do not include the people at the start and end of the path)

T3 was asked of the visualizations in Fig. 1. The assignment of tasks to data sets is given in Table 3, where data sets 1 and 2 were used for training the participants in the study. The questions were designed by the investigators so that they required cognitive effort to answer and so that the answer was unique. Typically this left very little choice of question. All of the questions used in the study are available online: the actual study can be taken at [1].

All of the questions were presented as multiple choice, with four possible answers. The unique correct answer could be selected by participants using radio buttons in our data-collection software. When using multiple choice questions, there is always a possibility that participants may guess the answer. It is anticipated that correct guesses (and, therefore, incorrect guesses) are approximately evenly distributed across the groups, thus not impacting on the statistical analysis of the collected data in substantial way. In addition, if guessing was rife amongst participants, we would expect the accuracy rates to be approximately 25%. As we shall see later, the lowest accuracy rate for five visualizations was 58.8% which is well above 25%.

5.4. Data Collection Method

We adopted a crowdsourcing approach, using Amazon Mechanical Turk (MTurk) [32] to automatically out-source tasks. There is evidence that crowdsourcing is a valid approach for collecting data because this method has now gained recognition within the scientific community [21, 32]. The tasks, called HITs (Human Intelligence Tasks), are completed by anonymous participants who are paid on completing the HIT. The HITs were based on the templates provided by Micallef et al. [29].

Every question, in the training and main study phases, was displayed on a separate page of the HIT. Participants were asked to "answer questions without delay". Previous pages could not be viewed and subsequent pages were not revealed until the question on the current page was answered. Unlike the training questions, in the main study the questions were randomly sequenced.

There is little control in MTurk over who participates in the study and, so, some participants may fail to give questions their full attention or have difficulties with the language; we call these participants *inattentive*. In an attempt to avoid language issues, a system qualification was used, allowing only participation from workers based in the USA with a HIT approval rate of 95%. Another recognized technique for identifying participants who cannot understand the language used is to include questions that require careful reading, yet are very simple to answer. In our study, we included two such questions which asked participants to click on a specific area in the diagram, whilst still presenting the participants with (redundant) radio buttons for typical answers as seen for the 12 main study questions; these catch questions appeared as the first and seventh questions after the training phase. Participants were classified as inattentive if they clicked a radio button on either of the two inattentive participant-identifying questions. All data obtained from inattentive participants was removed before analysis.

5.5. Experiment Execution

A pilot study ran with 25 participants of which one was inattentive. The pilot study proved the experimental design to be robust, with a few minor changes made to the wording of the questions. We also changed one question (not the diagram) as it appeared to be too hard with a 29% success rate, close to the number of correct answers that might be expected with random guesses. A further 500 participants took part in the main study. All participants were randomly allocated to one of the five groups in equal numbers. There were 32 inattentive participants and the data from a further participant was corrupted, leaving 467 participants in total, whose demographics were as follows:

- **gender:** 259 M, 208 F,
- **location:** 467 in USA,
- **age range, in years:** 18 to 70, 32.4 mean,
- **qualification level:** 17 not stated, 5 some high school, 40 high school graduate, 140 some college, 46 associates degree, 162 Bachelors degree, 0 Masters degree, 57 doctorate degree.

In each group, the number of participants was: Bubble Sets: 91, EulerView: 96, KelpFusion: 95, LineSets: 92, and SetNet: 93.

Each participant who either started or completed one of our HITs had their ID recorded, allowing us to prevent multiple participation. The participants performed the experiment at a time of their choosing and in a setting of their choosing. They were told that the experiment would take approximately 20 minutes, based on the time taken during the pilot study. All participants were paid \$1 for taking part; \$1 for 20 minutes work is higher than is typical for MTurk workers [32]. The data for the main study was collected over two days, with HITs made available in sequential batches.

Participants were instructed to read the questions carefully and were advised that they had to answer “some key questions” correctly in order to be paid (i.e. not classified as inattentive). Moreover, they were told that the first four pages of the HIT were training (the first phase of the experiment). During training, participants attempted questions. For every training question, the answer was explained to the participant after they had attempted it. After completing the training, participants entered the main study phase. During this main phase, participants were not given the correct answer to the problems. They were provided with a URL where the correct answers would be revealed after a certain date (this date was one we were confident would be after the data collection was completed).

6. Results

We analyzed accuracy and time data to establish which of our hypotheses hold and to rank the visualization techniques. We regarded p-values of less than 0.05 as significant. A secondary analysis, by task type, is also presented.

6.1. Accuracy

The following results are based on $500 - 33 = 467$ participants each answering twelve questions, giving $467 \times 12 = 5604$ observations. There were a total of 1866 errors, giving an accuracy rate of 66.7%. The mean accuracies for the visualization techniques are in Table 4, along with the standard deviations, where the abbreviations correspond to the visualization techniques. Whilst these data are not normal, the skewness is -0.65; given the size of the data set, it is robust to conduct an ANOVA. We found a significant effect of visualization technique ($p = 0.036$). Performing a pairwise comparison, using a Tukey test revealed the only significant difference in accuracy was between EulerView and LineSets: participants were significantly more accurate with EulerView than with LineSets. Fig. 12 is an interval plot for the accuracy means, showing that the only non-overlapping pair of confidence intervals (and, therefore, only significant difference) is between EulerView and LineSets. On average, participants were nearly 70% accurate with EulerView, dropping to nearly 59% accurate with LineSets.

Table 4: Average accuracy results for all tasks.

	BS	KF	LS	EV	SN
Mean	68.0	68.2	58.8	69.9	68.6
Std Dev.	29.1	26.4	26.4	27.0	24.4

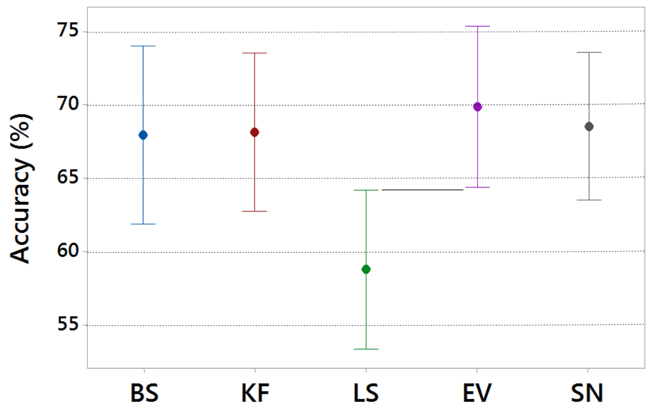

Figure 12: Mean accuracy with 95% confidence intervals.

6.2. Time

Consistent with Meulemans et al. [28], we only analyze the correct answers. Removing incorrect responses left 3738 observations (Bubble Sets: 742, EulerView: 805, KelpFusion: 777, LineSets: 649, and SetNet: 765). The grand mean for this reduced data set is 39.3 seconds per question (standard deviation: 29.2). The average completion times by technique are given in seconds in Table 5, along with the standard deviations.

In order to establish whether there is significant overall variation across visualization methods, we conducted an ANOVA. The analysis is performed using $\log_{10}$(time); whilst normality is not achieved, the skewness of the data is -0.19 so our analysis is robust. We found a significant effect of visualization technique ($p = 0.003$). Performing a pairwise overall comparison, using a Tukey test revealed that EulerView and SetNet significantly outperform

Table 5: Average time results for all tasks.

	BS	KF	LS	EV	SN
Mean	41.8	41.6	44.05	33.9	36.1
Std Dev.	31.9	27.8	36.9	24.4	23.5

Bubble Sets, KelpFusion, and LineSets. Fig. 13 shows an interval plot for the mean times: as faster is 'better', we can clearly see that EulerView and SetNet outperform Bubble Sets, LineSets and KelpFusion. Within these two groups of techniques, there are no significant differences. The mean times in Table 5 indicate the scale of the differences.

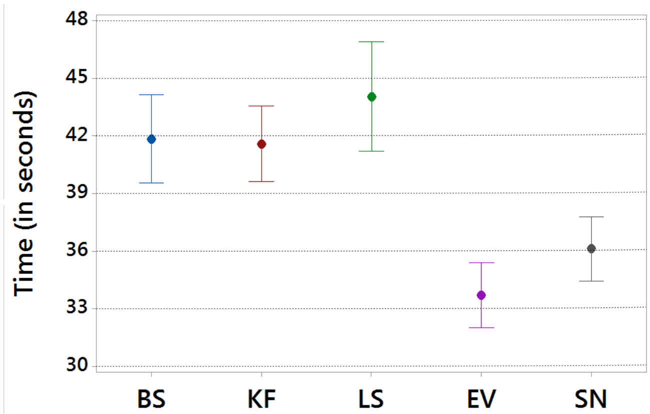

Figure 13: Mean time with 95% confidence intervals.

6.3. Interactions with Task Type

Here we present an indication of relative performance by task type is now given which should, however, be treated with caution: the study only contained three questions of each type. As a result, it is not scientifically robust to draw general conclusions about the interaction between visualization technique and question type. To identify *potential* significant differences in general, we make use of plots (Figs 14 and 15) showing 95% confidence intervals for the means, broken down by question type. Any pair of confidence intervals that do not overlap illustrate a significant difference in *our data*, taking $p = 0.05$.

Fig. 14 shows 95% confidence intervals for the mean accuracy rates. It can be seen that no significant differences in accuracy exist for question types

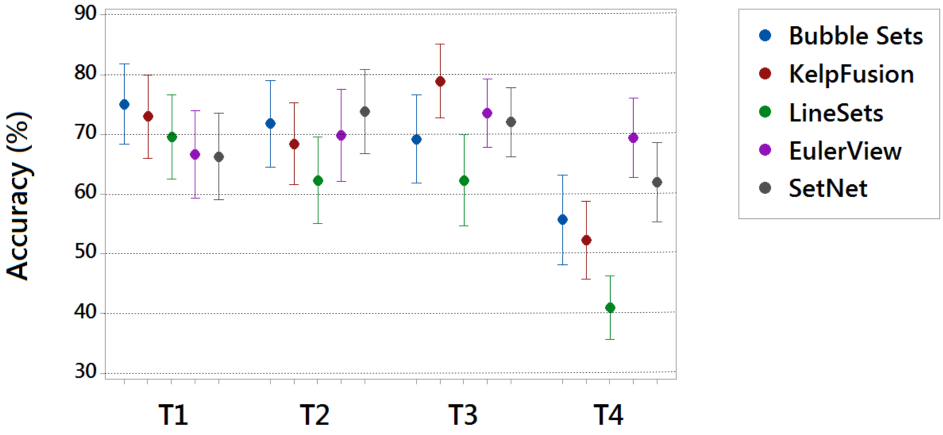

Figure 14: Mean accuracy by task with 95% confidence intervals.

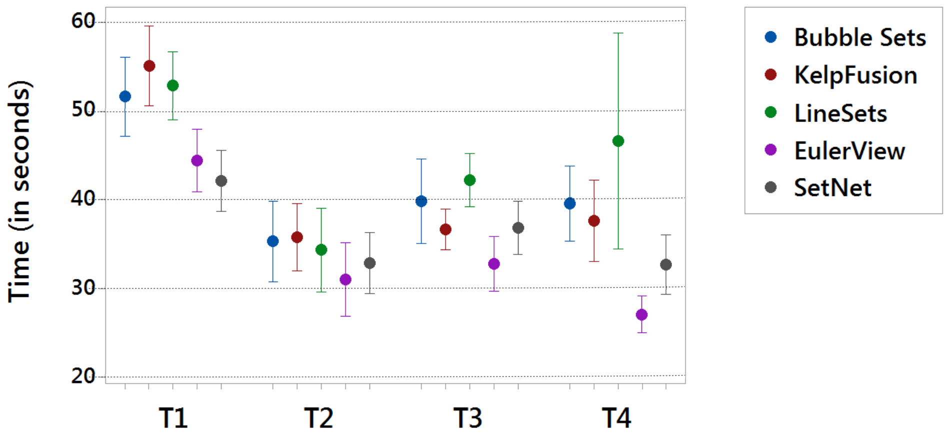

Figure 15: Mean time by task with 95% confidence intervals.

1 and 2. For type 3, KelpFusion outperforms LineSets. For type 4, Bubble Sets, EulerView and SetNet outperform LineSets and, in addition, EulerView outperforms KelpFusion. Fig. 15 shows the 95% confidence intervals for the mean times. For type 1, SetNet outperforms Bubble Sets, KelpFusion and LineSets. No significant differences exist for type 2 questions. For type 3, EulerView and KelpFusion outperform LineSets. Lastly, for type 4, EulerView outperforms all other methods. In all cases, either EulerView or SetNet are not significantly different from the visualization technique with the highest mean accuracy or the visualization technique with the lowest mean time,

supporting the results concerning overall performance. All of these observations require further study before any firm conclusions can be drawn about which techniques significantly differ when performing specific tasks.

6.4. Relationship to Hypotheses and Discussion

In terms of our hypotheses from Section 5, we reach the following conclusions, based on the *overall* accuracy and time analysis:

1. We do not have sufficient evidence to accept H1. No pair of KelpFusion, LineSets and Bubble Sets are significantly different in terms of accuracy or time.
2. We do not have sufficient evidence to accept H2. SetNet and EulerView are not significantly different in terms of accuracy or time.
3. We do not have sufficient evidence to accept H3, again because KelpFusion and LineSets did not yield significantly different performance.
4. We can accept H4: both EulerView and SetNet outperform Bubble Sets. Whilst they are not significantly different in terms of accuracy, the analysis of time data allows us to accept this hypothesis.

We conclude that EulerView and SetNet outperform Bubble Sets, KelpFusion, and LineSets, for the group-link and group-network tasks undertaken in our study, when static visualizations are used.

These results indicate that both EulerView and SetNet improve performance for grouped network data tasks. This suggests that Euler diagrams are an effective visualization for conveying information about overlapping sets. Moreover, our results suggest that assigning spatial rights to the sets is beneficial for the effectiveness of the visualizations, as compared to assigning spatial rights to the network. Lastly, with reference to Tables 1 and 2, the only feature that distinguishes both EulerView and SetNet from the other techniques, but not from each other, is the principle of proximity. We speculate, therefore, that adhering to this principle when visualizing grouped network data may be particularly important.

It is perhaps surprising that only one of our four hypotheses is supported by our analysis, as they were informed by previous empirical studies (albeit with different tasks performed by participants) and insights gained from perceptual theories. It is interesting to recall that these previous studies were inconsistent with their findings. One found Bubble Sets outperformed LineSets [24] whereas another found that KelpFusion outperformed LineSets which, in turn, outperformed Bubble Sets [28]. Our study did not reveal any significant difference between these three techniques. It may be that the type of task is crucial when deciding between these techniques.

Of particular note, though, is that our tasks differed significantly from those used in previous studies [3, 25, 28]. All of our questions required the participants to access information about *both* the sets and the network. The tasks used in [3, 28] did not include any network or group-network tasks. Thus, our results suggest that, as the complexity of the task increases, the difference in effectiveness of Bubble Sets, KelpFusion and LineSets becomes insignificant. Not being able to accept H3 perhaps has a similar explanation. Therefore, we have demonstrated that the previous results do not generalize to group-link and group-network tasks.

Likewise, since these techniques are visualizing rich information (data items, intersecting sets, and networks) our results point towards limitations of the perceptual theories discussed in Section 2. These theories consider aspects of visualizations in isolation, such as the syntax used to group items. Moreover, those discussed in Section 2.1 do not fully take into account graphical features, including those that are geometric, which can have a profound impact on the usability of visualizations. Thus, our results suggest that even if a notation has properties known to be beneficial, the importance of finding an effective layout when visualizing data is, unsurprisingly, important. We deduce that, whilst visualization design should respect perceptual theories, the importance of graphical properties should not be underestimated.

In the case of [25], the set visualization was simpler to the one we used as there were no group overlaps. That study included two tasks that use both group and network information. The first of these tasks presented a visualization which highlighted the groups of interest, so the only cognitive effort required to perform the task arose from interpreting the network. For the other task, there was no significant difference between Bubble Sets and Line Sets in their simpler setting of only visualizing disjoint sets, congruent with our study.

Concerning H2, the result that EulerView and SetNet are not significantly different is particularly surprising. The layout guides indicate that SetNet should visualize sets more effectively than EulerView (at least on the basis of the summary in Tables 1 and 2). However, the results suggest that SetNet may be overly compromising the layout of the network. The use of circles limits the spatial arrangement of the curves and the shapes of the regions to which the curves give rise. These limitations may be forcing the network to take a layout which is less effective than in EulerView, thus cancelling out any benefit of meeting more Euler diagram layout guides described in Section 2. The force model applied to the graph in SetNet and the initial placement of the nodes could also be improved. In addition, the well-formedness conditions are not equal in their effect on understanding. Visualizing any given set with more than one curve, as in SetNet, has been shown to be worse than concurrency [37], present in EulerView; three quarters of the SetNet diagrams used more than one curve for a set whereas none of the EulerView diagrams did so. Also, EulerView uses texture alongside colour, so this extra encoding may aid understanding.

Concerning H4, we have seen that SetNet and EulerView outperform Bubble Sets. Again, this significant result would suggest that the layout of the sets has more impact on performance than the layout of the network, since SetNet and EulerView assign primacy to the sets whereas Bubble Sets assigns primacy to the network.

With regard to the previous study of Meulemans et al. [28], their key findings when comparing KelpFusion with LineSets and Bubble Sets were as follows:

1. "Using a graph holds up well compared to a single continuous path technique." Meulemans et al. had hypothesized that LineSets would have advantages over the "branching graph" geometry seen in KelpFusion, but this was not supported by their study.
2. "Using a mix of hulls and links has benefits over a single concave hull technique." In particular, their study found that KelpFusion (which uses a "mix of hulls and links") outperformed Bubble Sets which uses "single concave hulls".

Our study supports their first key finding, as we did not establish any significant difference between KelpFusion and LineSets. Unlike our study, Meulemans et al. found KelpFusion outperforms LineSets in terms of completion time. Their second key finding is *not supported* by our study: their finding does not generalize to the tasks considered in this paper. We observed no significant difference between Bubble Sets and KelpFusion. Moreover, EulerView represents sets using hulls which are often concave and outperformed KelpFusion.

Meulemans et al. indicated that they limited the number of data items in their study (to 49 items) and that they did not ensure strong spatial grouping of the nodes. They conjectured that increasing the number of elements and ensuring stronger spatial grouping of the nodes would be "likely to confirm the benefits of KelpFusion [over Bubble Sets and LineSets]". We have partly refuted this conjecture in that our study had up to 64 data items and we saw no significant differences between these three techniques. However, as we used a stand-alone tool to visualize the network, no attempt was made to ensure stronger spatial grouping. Moreover, our questions required the participants to use both the network and sets, so it remains open to explore the truth of this conjecture with the simpler tasks used in Meulemans et al.'s study.

7. Scalability of the Techniques

The five techniques that we have evaluated represent the state-of-the-art in set and network visualization methods currently available. They are all capable of providing effective visualizations of sets and networks, but as the data grows in size, this effectiveness can be diminished. Two key questions relating to scalability, for any visualization technique, are:

- How well does the technique scale in terms of being able to *theoretically* visualize large data?
- How well does the technique scale in terms of being able to *effectively* visualize large data?

All five techniques evaluated in this paper can, theoretically, visualize arbitrarily large data in every sense: there is no theoretical limit to the number of sets, nodes and edges that can be visualized[9]. For instance, our method – SetNet – exploits the iCircles technique that is theoretically proven to be able to visualize any finite collection of sets [42]. From this, it follows that any network data can be overlayed in the drawn Euler diagram.

Of course, from a practical perspective, the second question is more important than the first: if a visualization of a particular data set is not effective, preventing users from gaining an understanding of their data, then its usefulness (for that particular data) is questionable. The effectiveness of all five techniques can diminish as the size of the data increases. Here,

[9]We acknowledge that in practice the techniques will not necessarily be able to produce visualizations in acceptable run times due to the computational complexity of the drawing methods employed.

effectiveness can be linked with the level of *clutter* exhibited by the visualization under consideration, and other properties already discussed such as well-formedness. Clutter arises from the sets and the network being visualized, including the interplay between the two (such as how many data items are in each set intersection).

Regarding set visualization, Euler diagrams – the underling notation for SetNet, Bubble Sets and EulerView – exhibit high levels of clutter when there are lots of intersections between the sets. Increasing the number of sets being visualized does not necessarily cause scalability problems. Indeed, it is increasing levels of clutter in Euler diagrams that has been shown to have a negative impact on task performance [4]. Therefore, Euler diagrams can sometimes scale well to large numbers of sets, provided the intersections present do not yield highly cluttered diagrams.

We expect that clutter manifests in a similar way when using KelpFusion to visualize sets; KelpFusion uses surfaces with multiple boundaries to represent sets, generalizing the use of closed curves and their interiors seen in Euler diagrams. We conjecture that increased clutter, arising from many intersecting sets, will lead to a decrease in effectiveness, as has been shown for Euler diagrams. Establishing a measure of clutter and the impact of clutter on effectiveness when using surfaces with multiple boundaries to visualize sets remains an interesting avenue of future work, in part because it will provide insight about when KelpFusion scales to large data.

LineSets exploit a rather different way of visualizing the sets, using lines rather than closed curves or surfaces. They require each set-line to pass through all the data items (nodes) that lie in the represented set. As a consequence, data items in many sets are passed through by many lines, leading to a manifestation of clutter. Currently, there is no formal understanding about what constitutes a cluttered LineSets diagram, but there is likely to be a relationship between what people perceived to be cluttered and the number of data items in each set intersection. Again, establishing a measure of clutter and its relationship with task performance is an interesting problem for future work.

Scalability issues also arises through the number of data items and the number of links between them in the network. When graphs are sparse (i.e. the number of edges is low compared to the number of vertices), the visualization techniques can scale well. However, as the number of edge crossings increases, the effectiveness of the visualization is likely to decrease. Obtaining layouts of the graph with few edge crossings is complicated by the fact that, when using closed curves or surfaces, the location of nodes may be constrained by the visualization of the sets. Of course, significant research has been undertaken to evaluate layout properties of graphs and their impact on task performance. However, layout choices for graphs in the context of set visualization is unexplored.

Highly relevant to our work is the interplay between the network and the sets. There is currently no understanding about how the layout properties arising from visualizing sets and networks together impacts visual clutter or task performance as data increase in size. We expect that useful insights about diagram clutter stand to be provided by examining how the graph edges pass through closed curves (in the case of Euler diagrams), surfaces (for KelpFusion), or lines (for LineSets) and how many are nodes present in each set intersection. By developing a rounded theory of clutter for these combined visualizations of sets and networks, we can begin to empirically explore the effect of increasing levels of clutter on task performance. As a result, we will gain important insight into the relative scalability of these five techniques.

8. Conclusion and Future Work

Using perceptual theories and empirically supported layout criteria, we analyzed four existing visualization techniques, namely BubbleSets, EulerView, LineSets and KelpFusion, alongside a new technique, SetNet, for visualizing grouped network data. SetNet, like EulerView, assigns primacy to sets but it possesses more features known to aid comprehension. We performed a controlled experiment to compare user task performance, in terms of accuracy and time, of these five techniques using real social network data displayed as static visualizations. Key results for our study are summarized as follows:

(a) EulerView and SetNet outperformed Bubble Sets, KelpFusion and LineSets overall when participants undertook group-link and group-network tasks.
(b) Visualizations that assign primary spatial rights to sets outperform those which assign primary spatial rights to the network.
(c) Adhering to the principle of proximity is seen as particularly important for effectiveness.
(d) Contrary to previous studies, we found no significant differences between Bubble Sets, KelpFusion and LineSets, indicating that earlier

results do not generalize to group-link and group-network tasks – i.e. tasks which use both the network and the sets, for which these visualization techniques were designed.

As with any empirical study, these results are valid within the constraints imposed by the study. In our case, the results are valid for the task types undertaken by our pool of participants drawn from the general population. It is an interesting avenue of future work to see whether these results hold when the participants are from a pool of technically trained people.

The automated layout of this complex type of data can be further improved. This may be assisted by additional studies on diagrammatic features that are highlighted by our results as potentially impacting on task performance such as texture and node layout within sets. It would also be interesting to devise a new approach that attempts to visualize both the set and network simultaneously, thus assigning primary spatial rights to neither the network or the sets. This is motivated by the indication that, whilst SetNet was perceived to better visualize the sets than EulerView (based on the analysis in Section 3), it overly compromised the layout of the network. In addition, a network layout method that explicitly optimizes graph layout guides (such as reducing crossings) could be implemented in place of the current force-directed approach.

We would like to see further studies exploring the generality of these results using different data and different tasks as well as alternative visualization techniques. In many cases, the assignment of primary spatial rights to either the sets or network may be task or application dependent. The generality of our results could be further investigated, with this choice in mind. It would also be interesting to perform further empirical studies to explore the use of multiple unjoined circles, as with SetNet, as compared to the use of multiple joined convex shapes as in [35].

Exploring interaction with the visualization techniques is also a valuable avenue for future research. Examples of interactive features include zooming, filtering, highlighting and providing details on demand. The ability of visualization techniques to support these features will be important for their wide-scale applicability, allowing one to exploit the value of interaction and enable new analysis possibilities.

Acknowledgements We would like to thank: Wouter Meulemans for generating all of the KelpFusion images used in the study; Basak Alper and Paolo Simonetto for providing us with the source code for LineSets and EulerView respectively; and Mohamad Tayssir Alkowatly for help with setting up the framework that enabled us to run EulerView. We are also grateful to the participants who took part in our study.

[1] Setnet software and study materials. www.eulerdiagrams.org/setnet, accessed January 2016.

[2] J-W Ahn, C. Plaisant, and B. Shneiderman. A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):365–376, 2014.

[3] B. Alper, N. Henry Riche, G. Ramos, and M. Czerwinski. Design study of LineSets, a novel set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, 2011.

[4] M. Alqadah, G. Stapleton, J. Howse, and P. Chapman. Evaluating the impact of clutter in Euler diagrams. In *8th International Conference on the Theory and Application of Diagrams*, pages 109–123. Springer, 2014.

[5] B.Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. Visualizing sets and set-typed data: State-of-the-art and future challenges. In *Eurographics conference on Visualization (EuroVis)State of The Art Reports*, pages 1–21. Eurographics Association, 2014.

[6] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media*, pages 361–362, 2009.

[7] F. Benoy and P. Rodgers. Evaluating the comprehension of Euler diagrams. In *11th International Conference on Information Visualization*, pages 771–778. IEEE Computer Society, 2007.

[8] F. Bertault and M. Mirka. An algorithm for drawing compound graphs. In *Graph Drawing*, pages 197–204, 1999.

[9] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* University of Wisconsin Press, 1983.

[10] A. Blake, G. Stapleton, P. Rodgers, L. Cheek, and J. Howse. The impact of shape on the perception of Euler diagrams. In *8th International Conference on the Theory and Application of Diagrams*, pages 124–138. Springer, 2014.

[11] S. Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

[12] C. Collins, G. Penn, and M. Sheelagh T. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[13] J. Duncan and G. Humphreys. Visual search and stimulus similarity. *Psychological review*, 96(3):433, 1989.

[14] T. Dwyer, Y. Koren, and K. Marriott. IPSep-CoLa:an incremental procedure for separation constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):821–828, 2006.

[15] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42(11):149–160, 1984.

[16] J. Flower and J. Howse. Generating Euler diagrams. In *2nd International Conference on the Theory and Application of Diagrams*, pages 61–75, Georgia, USA, 2002. Springer.

[17] R. Gansner., H. Yifan, and S. Kobourov. GMap: Visualizing graphs and clusters as maps. In *IEEE Pacific Visualization Symposium*, pages 201–208, 2010.

[18] G. Grothaus, A. Mufti, and T. Murali. Automatic layout and visualization of biclusters. *Algorithms for Molecular Biology*, 1(15):1–11, 2006.

[19] C. Gurr. Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing*, 10(4):317–342, 1999.

[20] M. Harrower and C.Brewer. ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[21] J. Heer and M. Bostock. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *ACM Conference on Human Factors in Computing Systems*, pages 203–212, 2010.

[22] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *IEEE Symposium on Information Visualization*, pages 32–39. IEEE, 2005.

[23] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13(3):259–290, 2002.

[24] T. Itoh, C. Muelder, M. Kwan-Liu, and J. Sese. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *IEEE Pacific Visualization Symposium*, pages 121–128, 2009.

[25] R. Jianu, A. Rusu, Y. Hu, and D. Taggart. How to display group information on node-link diagrams: an evaluation. *IEEE Transactions onVisualization and Computer Graphics*, 20(11):1530–1541, 2014.

[26] K. Koffka. *Principles of Gestalt Pschology.* Lund Humphries, 1935.

[27] J. Leskovec. Stanford large network dataset collection. http://snap.stanford. edu/data/index. html, 2014. Accessed January 2014.

[28] W. Meulemans, N. Henry Riche, B. Speckmann, B. Alper, and T. Dwyer. Kelpfusion: A hybrid set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1846–1858, 2013.

[29] L. Micallef, P. Dragicevic, and J.-D. Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2536–2545, 2012.

[30] P. Mutton, P. Rodgers, and J. Flower. Drawing graphs in Euler diagrams. In *3rd International Conference on the Theory and Application of Diagrams*, pages 66–81. Springer, March 2004.

[31] S. E. Palmer, J. L. Brooks, and R. Nelson. When does grouping happen? *Acta Psychologia*, 114(3):311–330, 2003.

[32] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

[33] B. Plimmer, A. Morrison, and H. Knoche. Exploring the effects of colouring graph diagrams on people of various backgrounds. In *8th International Conference on the Theory and Application of Diagrams*, pages 176–189. Springer, 2014.

[34] C. Reffay and T. Chanier. How social network analysis can help to measure cohesion in collaborative distance-learning. In *Designing for Change in Networked Learning Environments Computer-Supported Collaborative Learning*, pages 343–352. Springer, 2003.

[35] N. Riche and T. Dwyer. Untangling Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1090–1099, 2010.

[36] P. Rodgers, R. Baker, and S. Thompson. Multi-level visualization of concurrent and distributed computation in Erlang. In *Visual Languages and Computing*, pages 156–161. Knowledge Systems Institute, 2013.

[37] P. Rodgers, L. Zhang, and H. Purchase. Wellformedness properties in Euler diagrams: Which should be used? *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1089–1100, 2012.

[38] B. Saket, P. Simonetto, and G. Kobourov. Group-level graph visualization taxonomy. In *Eurographics Conference on Visualization*, pages 85–89. Eurographics Association, 2014.

[39] R. Santamaría and R. Therón. Overlapping clustered graphs: co-authorship networks visualization. In *Smart Graphics*, pages 190–199. Springer, 2008.

[40] R. Santamaría, R. Therón, and L. Quintales. BicOverlapper: A tool for bicluster visualization. *Bioinformatics*, 24(9):1212–1213, 2008.

[41] P. Simonetto. *Visualisation of Overlapping Sets and Clusters with Euler Diagrams*. PhD thesis, Université Bordeaux, 2012.

[42] G. Stapleton, J. Flower, P. Rodgers, and J. Howse. Automatically drawing Euler diagrams with circles. *Journal of Visual Languages and Computing*, 23(3):163–193, 2012.

[43] G. Stapleton, P. Rodgers, J. Howse, and J. Taylor. Properties of Euler diagrams. In *Proceedings of Layout of Software Engineering Diagrams*, pages 2–16. EASST, 2007.

[44] M. Sun, P. Mi, C. North, and N. Ramakrishnan. Biset: Semantic edge bundling with biclusters for sensemaking. *IEEE Transactions onVisualization and Computer Graphics*, 22(1):310–319, 2016.

[45] J. Wagemans, J. Elder, M. Kubovy, S. Palmer, M. Peterson, and M. Singh. A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organisation. *Computer Vision, Graphics, and Image Processing*, 31:156–177, 1985.

[46] M. Wang, B. Plimmer, P. Schmieder, G. Stapleton, P. Rodgers, and A. Delaney. SketchSet: Creating Euler diagrams using pen or mouse. In *IEEE Symposium on Visual Languages and Computing*, pages 75–82. IEEE, 2011.

[47] S. Wasserman and K. Faust. *Social Network Analysis*. CUP, 1994.