DOI: 10.1111/cgf.12722

COMPUTER GRAPHICS *forum*
Volume 35 (2016), number 1 pp. 234–260

The State-of-the-Art of Set Visualization

Bilal Alsallakh[1], Luana Micallef[2,3], Wolfgang Aigner[1,4], Helwig Hauser[5], Silvia Miksch[1] and Peter Rodgers[3]

[1]Vienna University of Technology, Vienna, Austria
bilalsal@gmail.com, wolfgang.aigner@fhstp.ac.at, miksch@ifs.tuwien.ac.at
[2]Helsinki Institute for Information Technology HIIT, Aalto University, Finland
Luana.Micallef@hiit.fi
[3]University of Kent, Canterbury, United Kingdom
p.j.rodgers@kent.ac.uk
[4]St. Poelten University of Applied Sciences, St. Poelten, Austria
[5]University of Bergen, Bergen, Norway
Helwig.Hauser@UiB.no

Abstract

Sets comprise a generic data model that has been used in a variety of data analysis problems. Such problems involve analysing and visualizing set relations between multiple sets defined over the same collection of elements. However, visualizing sets is a non-trivial problem due to the large number of possible relations between them. We provide a systematic overview of state-of-the-art techniques for visualizing different kinds of set relations. We classify these techniques into six main categories according to the visual representations they use and the tasks they support. We compare the categories to provide guidance for choosing an appropriate technique for a given problem. Finally, we identify challenges in this area that need further research and propose possible directions to address these challenges. Further resources on set visualization are available at `http://www.setviz.net`.

Keywords: information visualization, visualization

ACM CCS: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces, F.4.1 [Theory of Computation]: Mathematical Logic—Set theory

1. Introduction

A common step in data analysis is to group data items into sets based on specific properties. For instance, Figure 1(a) shows how members of a social network are grouped according to their interests. Figure 1(b) shows how chemicals (dots) are contained in blood samples of different whales. Several relations between sets are possible, such as: containment, exclusion and intersection. Analysing these relations is key to gain information about the behaviour of the entities they represent. Such information might involve which set combinations are common (Figure 1c), and whether certain data features are responsible for this. As illustrated in Section 2.5, a variety of real-world concepts can be modelled using sets, including: club memberships, product features and employee skill sets. Example questions about such data are: whether certain clubs are exclusive to each other, whether a certain product feature is always present in combination with another one and whether specific skill combinations are highly paid.

Information visualization (InfoVis) offers many opportunities for analysing sets and their relations. A key challenge in visualizing sets is the potentially large number of possible relations between them (Section 2.4). Besides Venn and Euler diagrams, several InfoVis techniques were proposed to visualize sets using various representations. These techniques vary in their scalability limits and in the set-related tasks they support. We survey state-of-the-art techniques for visualizing sets.[1] After discussing several characteristics of set-typed data (Section 2) and tasks related to them (Section 3), we provide an overview and a categorization of these techniques (Section 4) based on the visual metaphors they use. In addition, we compare these techniques by their advantages and limitations, and by the tasks they support (Section 5). Finally, we identify challenges that require future research, along with possible opportunities to tackle them (Section 6).

[1]This paper extends a recent report on set visualization [AMA*14].

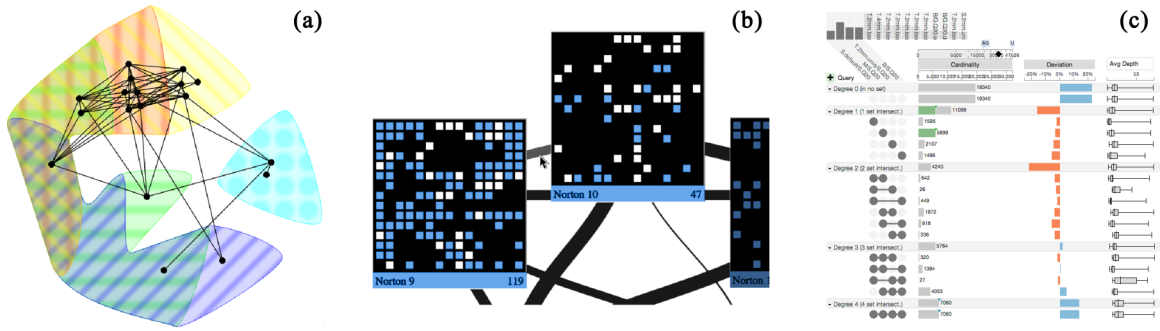

Figure 1: *Different set visualizations: (a) a combined Euler and node-link diagram [SAA09], (b) OnSet [SMDS14], (c) UpSet [LGS*14].*

2. Sets and Set-Typed Data

Sets have been traditionally studied by mathematicians and logicians as a foundational concept. A set is defined as a collection of unique objects, called the set elements. A key characteristic of this collection is that it does not impose an ordering of the elements. A family of sets, also called a set system, is a collection of subsets of a given set of elements. Such sets potentially overlap, making several relations between the sets possible such as containment, exclusion and intersection. Cantor formalized *set theory* [Can95] in the 19th century. This theory is concerned with various concepts related to sets, such as set algebra and set operations.

In data analysis, sets have been mainly treated as a collection of data points, such as a subset of rows in a data table. Such subsets are usually used to define training and validation sets, or to store the results of search and clustering algorithms. In addition, set-theoretic operations such as intersection, union, difference, complement, Cartesian product and the power set are extensively used in relational databases to query elements and join multiple data tables.

Despite the ubiquitous usage of sets in data analysis, sets have not commonly been treated as their own data type in InfoVis literature, unlike graphs and hierarchies. Set memberships are rather often abstracted and treated as separate Boolean attributes, as noted by Freiler *et al.* [FMH08]. Treating set families as an elementary data type contributes to a better understanding of their characteristics and the challenges associated with visualizing them. We refer to data that involve element-set memberships as *set-typed data* or *set-based data*. The data can also encompass additional attributes of the elements or the sets. In the following, we give examples of how set-typed data are represented and what special cases, specific features, similarity measures and tasks are associated with them.

2.1. Data representation

There are several ways to represent a set family on the data level, depending on the information available. One way is to explicitly represent the relations between the sets in the family. The data store the absolute or relative size of the intersection of these sets (Figure 2a). This representation is suited when no information about individual set elements is available. For example, when the sets represent events, relative sizes can be used to describe joint probabilities for these events.

When the number of elements in the set family is finite and their set memberships are available, three data structures for graphs can be used to represent these memberships. A multi-valued attribute can specify the sets to which each element belongs (Figure 2b), resembling adjacency lists. Alternatively, a table of element-set memberships can be used (Figure 2c), resembling an edge list. Boolean attributes representing the sets can also be used to specify which elements belong to them (Figure 2d), resembling an adjacency matrix. These representations illustrate a duality between the elements and the sets: by transposing the matrix, each set S can be treated as an element that belongs to the dual sets corresponding to the elements of S. Similarly, instead of representing set memberships for each element, adjacency lists can represent sets by extension, i.e. as lists of their elements.

Besides set membership, further attributes of the data elements might need to be involved in the analysis. For example, besides membership of different clubs (sets), information about club members (elements) might encompass their age and sex. Furthermore, attributes can be associated with the set memberships themselves, such as membership date for club members. Certain techniques support visualizing such set-dependent attributes (Section 5.1).

2.2. Scope and special cases

In general, the sets in a set family overlap, i.e. they have one or more intersection relations. When all sets are in an exclusion relation, they exhibit no overlap and define groupings over the respective elements. If such sets cover all the elements, they define a partitioning of the elements into classes. In such cases, the set memberships can be represented by one categorical attribute that stores these classes. When the sets exhibit both exclusion and inclusion relations, but no intersections, they define a hierarchy over their elements. We limit our survey to techniques for visualizing overlapping sets. While many of these techniques can also be applied to hierarchies, dedicated techniques [Sch11] are better suited for visualizing strict hierarchies.

A family of sets defined over a finite number of elements is equivalent to a hypergraph whose hyperedges represent the sets. A hypergraph is usually drawn either in subset standard (Section 4.2.1) or in edge standard (Section 4.3) [Mäk90].

In some cases, there are constraints on possible intersection relations between the sets. One example is when an element can belong

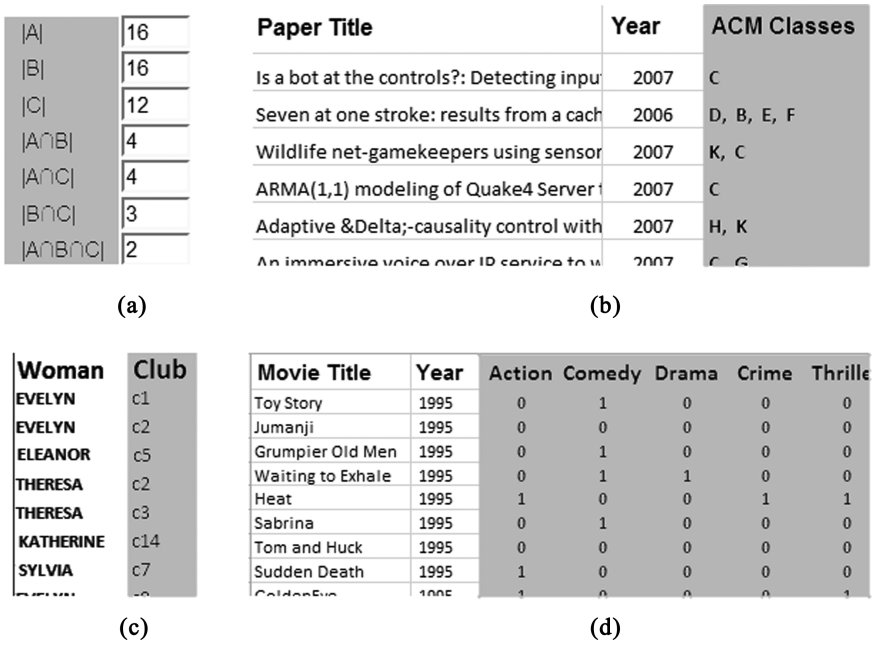

Figure 2: *Various forms of set-typed data: (a) the cardinality of set relations, (b) a multi-valued attribute (in grey), (c) a membership list, (d) Boolean attributes (in grey).*

to a maximum of $k < m$ sets from a family of m sets. Another example is when a set can intersect with k other sets at most. It is important to identify and exploit such special cases, as they can simplify the visualization.

2.3. Similarity measures

Many tasks related to set-typed data are concerned with finding which pairs of sets S_1 and S_2 exhibit higher similarity than other pairs, with regard to the number of shared elements between them $|S_1 \cap S_2|$. Several similarity measures between finite sets have been proposed in the literature. A symmetric measure was proposed by Jaccard [HHH*89]:

$$Jaccard(S_1, S_2) = |S_1 \cap S_2| / |S_1 \cup S_2|.$$

It has been employed in set visualization both explicitly to reveal set similarity as in Radial Sets (Figure 16) and implicitly for matrix reordering (Section 4.4). Tversky [Tve77] proposed a generalized index for set similarity that can replicate other measures by using different parameterizations.

$$Tversky(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cap S_2| + \alpha \cdot |S_1 \setminus S_2| + \beta \cdot |S_2 \setminus S_1|}.$$

Using different values for α and β, many similarity measures can be replicated such as Dice's [Dic45] and Tanimoto's [Tan58] coefficients. It is also possible to weigh shared elements differently when computing the similarity. For example, elements in $S_1 \cap S_2$ that only belong to S_1 and S_2 can be weighed higher than ones that are also members of other sets. Also, set-exclusive elements can be excluded when computing the denominator, especially in sparse families of sets that exhibit little overlap. An important issue with similarity measures is their sensitivity to the respective set sizes. Larger sets have higher probability of overlap, causing a bias in the above-mentioned measures. Applying the χ^2 statistic can eliminate such bias [AAMH13].

The choice of an appropriate similarity measure depends on the data and the information to be communicated by the visualization. Depending on whether the chosen measure is symmetric or not, and on the value range it takes (e.g. $[0, 1]$ or $[-1, 1]$), different visual variables are appropriate for encoding set similarity, such as size [KSB*09], colour [AAMH13], position [LLS05] or order [KLS07].

2.4. Combinatorics of sets

In order to choose an effective visualization of set-typed data, it is important to understand the combinatorics of sets. The exponential growth of relevant pieces of information about the data imposes severe limits on visualization techniques (Section 5). In the following, we illustrate aspects of set combinatorics that are relevant for visualization.

Given a family of n sets $F = \{S_1 \subseteq E, ..., S_n \subseteq E\}$ defined over m elements $E = \{e_1, ..., e_m\}$, the *set membership degree* of an element $e \in E$ can be defined as follows:

$$degree(e) = |S \in F : e \in S|. \quad (1)$$

This degree denotes the number of sets in the family the element belongs to. It corresponds to set cardinality in the dual representation of sets and elements. A related concept is the *exclusive membership* of certain sets or set intersections. For example, elements of degree

1 in a specific set belong exclusively to this set. Also, elements of degree 2 in an intersection between two sets belong exclusively to these two sets. Many analysis tasks (Section 3) and visualization designs (Section 4) are concerned with the element degrees and with exclusive membership.

An element of degree k can belong to one of $\binom{n}{k}$ possible combinations of k sets taken from F. We assume that elements $E_C \subseteq E$ in a combination of k sets $C = \{S_{i_1}, ..., S_{i_k}\} \subseteq F$ are all of degree k, and hence do not belong to other sets in F. Therefore, assuming $\{S_{i_{(k+1)}}, ..., S_{i_n}\} = F \setminus C$, the elements E_C in combination C are:

$$E_C = (S_{i_1} \cap ... \cap S_{i_k}) \cap (\bar{S}_{i_{(k+1)}} \cap ... \cap \bar{S}_{i_n}), \tag{2}$$

where $\bar{S} = E \setminus S$. These elements correspond to identical rows in the matrix representation of the set family, that all have 1 for $S_{i_1}, ..., S_{i_k}$ and 0 for $S_{j_1}, ..., S_{j_{(n-k)}}$. Hence, set combinations of all degrees define equivalence classes over the elements, as it is not possible to separate between elements in the same combination given the set information only.

The number of non-empty set combinations of all degrees in a family of sets is equal to the number of possible subsets of the set family:

$$|P(F)| = |\{X : X \subseteq F\}| = 2^n, \tag{3}$$

where $P(F)$ is the power set of F.

This is equal to the number of all possible distinct rows in the matrix representation of the set family, as each of the n bins in the row can be either 0 or 1. This number is also equal to the number of regions in a Venn diagram (Section 4.1), including the complementary region that surrounds all curves. Figure 14(f) illustrates all possible 16 combinations of all degrees between four sets. The number of combinations can be very large, even when the number of sets is in the order of tens. In practice, the majority of combinations are empty. Also, the number of non-empty combinations is always $\leq m$ (the number of elements), as each element can belong to exactly one combination. Nevertheless, the number of non-empty combinations can still be overwhelming.

Since set combinations of all degrees correspond to equivalence classes over the elements, they comprise the smallest units that can be used to build set-based queries, as illustrated in Figure 14(f). A set-based query returns certain elements based on their set memberships, by defining which of these units should be included in the result. The result comprises the union of all corresponding set combinations. This corresponds to a disjunction of conjunctions, as each set combinations defines a conjunction of the sets (Section 2). As each unit can either be selected for inclusion in the query result or not, the number of all possible set-based queries is equal to

$$|P(P(F))| = 2^{|P(F)|} = 2^{|2^n|}, \tag{4}$$

where $|P(F)|$ is the number of set combinations of all degrees (Equation 3). The number of all possible set-based queries can be very large even with a very small number of sets. For example, with $n = 4$ sets, the number is equal to 65 536.

Besides set-based queries, other queries are possible over the elements E, such as queries based on additional element attributes not related to their set memberships. The number of unique result sets over m elements is equal to

$$|P(E)| = 2^m,$$

where $P(E)$ is the power set of the set of all elements E. Usually, only a fraction of these query results can be captured using set-based queries.

It is important to take the large number of possible combinations into consideration when designing set visualizations. As explained throughout the survey (Section 4), some of the existing techniques aim to represent all of these combinations, while other techniques depict only aggregated information about set overlaps. Interaction plays a major role in reducing visual complexity, by enabling the exploration of certain details on demand. It is also important to find out if a given analysis problem restricts possible element-set memberships and set relations. Such constraints often limit the number of possible set combinations, allowing for simpler and more scalable visualizations for the given problem.

Similarly, it is challenging to design a search interface that supports all possible set-based queries.

Interactive visual feedback plays an important role in supporting the user with defining and refining set-based queries.

2.5. Modelling problems using sets—examples

The notion of sets, set relations and set operations is easy to understand, yet powerful, making set-typed data applicable to a variety of data analysis problems. The following list gives examples of how sets can be used to model problems from various domains.

- **Tags and multi-label classifications:** labelling a collection of items with tags is equal to defining multiple sets over these items. This applies to many classification problems, where an item can be classified into multiple classes at the same time.
- **Subscriptions:** many news and service providers allow users to receive new items and updates by subscribing to multiple areas of interests (AOIs) (often by selecting checkboxes). Each area of interest defines a set over the users who subscribe to it.
- **Voting:** in many voting scenarios, a voter can select one more candidates for a specific election. Each candidate defines a set over the voters who selected him/her.
- **Surveys:** some multiple-choice survey questions can be answered by selecting multiple answers (e.g. countries visited in the past). Each answer defines a set over the survey subjects who select this answer.
- **Probabilistic events:** sets are often used to reason about probabilities [Che11]. Each probabilistic event can be modelled as a subset of the sets of all possible outcomes.
- **Fuzzy clustering:** a cluster can be modelled as a set over the data points. In fuzzy clustering, one point can belong to multiple clusters, which results in set overlaps.

- **Query results:** items satisfying a query can be modelled as a set over all data items. It is possible to compare multiple query results by analysing the corresponding set relations.
- **Meta search:** performing the same query using different search engines results in multiple result sets. Meta search combines these results using set union, and allows exploring which elements were retrieved by which engines [Spo04].
- **Faceted search:** this search paradigm allows defining queries over data items based on a number of search facets [Tun09]. Each facet can be modelled as a set of the data item that satisfy its criterion [BKP12, AMR14, GLSS11].
- **Formal Concept Analysis (FCA)** [GWW99] makes use of concepts and their relations to reason about problems. These relations are analogous to set relations, when treating concepts as sets [BKP12].
- **Genomics:** to analyse how genes belong to individuals, each individual can be modelled as a set that contains specific genes. Several other genomics problems can be modelled using sets [LG14].

After modelling a data analysis problem using sets, it is important to think about tasks that need to be performed with these sets. Examples for this are given in the next section.

3. Common Tasks with Set-Typed Data

When designing a visualization of set-typed data, it is important to determine which tasks it needs to support. Here, we list general tasks addressed by the surveyed techniques, classified into the following categories.

3.1. Tasks related to elements

These tasks are concerned with the membership of the elements in the sets:

(A1) Find/Select elements that belong to a specific set.
(A2) Find sets containing a specific element.
(A3) Find/Select elements based on their set memberships: e.g. elements in A and in B but not in C.
(A4) Find/Select elements in a set with a specific set membership degree: e.g. elements exclusive to the set or that also belong to two other sets.
(A5) Filter out elements based on their set memberships.
(A6) Filter out elements based on their set membership degrees: e.g. filtering out elements exclusive to their sets, to focus on shared elements.
(A7) Create a new set that contains certain elements.

3.2. Tasks related to sets and set relations

These tasks are concerned with higher level reasoning about the sets without taking individual elements into account. Example tasks applied to sets A, B and C include:

(B1) Find out the number of sets in the set family.
(B2) Analyse inclusion relations: e.g. find out if a set A is fully included in B, or in $B \cap C$, or in $B \cup C$.
(B3) Analyse inclusion hierarchies: e.g. find out if A is included in B, and B in turn is included in C (and so on).
(B4) Analyse exclusion relations: e.g. find out if A does not intersect B, or $B \cap C$, or $B \cup C$.
(B5) Analyse intersection relations: e.g. find out if a certain pair of sets overlap, or if a certain group of sets overlap (i.e. have a non-empty intersection).
(B6) Identify intersections between k sets.
(B7) Identify the sets that constitute a certain intersection.
(B8) Identify set intersections contained in a specific set.
(B9) Identify the set with the largest/smallest number of pairwise set intersections.
(B10) Analyse and compare set- and intersection cardinalities: e.g. estimate $|A|$ or $|A \cap B|$, compare $|A|$ with $|B|$, or $|B \cap C|$, or $|B \cup C|$ and identify the set or set intersection with the largest or smallest cardinality.
(B11) Analyse and compare set similarities: e.g. find out which pairs of sets exhibit high or low similarity according to some similarity measure.
(B12) Analyse and compare set exclusiveness: e.g. find out if A contains more exclusive elements than B, or more elements shared with 1, 2 or 3 other sets.
(B13) Highlight specific sets, subsets or set relations: e.g. to emphasize them, and deemphasize the remaining data.
(B14) Create a new set using set-theoretic operations: e.g. create the complement of A, or $A \setminus B$ as a new set to compare with other sets.

3.3. Tasks related to element attributes

Set-typed data can encompass additional attributes of the elements. The following tasks are concerned with how the element memberships and attributes are interrelated:

(C1) Find out the attribute values of a certain element.
(C2) Find out the distribution of an attribute in a certain set or subset: this aims to understand how the attribute correlates with element membership of this set. Sometimes, the two attributes have a spatial reference and the elements are positioned accordingly as in maps or scatter plots (Section 4.2). In this case, the task supports estimating the spatial distribution of a set [DvKSW12].
(C3) Compare the attribute values between two sets or subsets: e.g. the attribute distributions in two sets can be compared against each other. Alternatively, summary values can be compared such as the mean, the median or the dominant category.
(C4) Analyse the set memberships for elements having certain attribute values: e.g. find out if these elements appear more frequently or less often in certain sets/subsets.
(C5) Create a new set out of elements that have certain attribute values: this set represents a query on the elements based on their attributes. Shneiderman emphasized the importance of supporting such queries in his task taxonomy [Shn96] and the role of set-theoretic operations to combine multiple constraints on the attribute values.

In the next section, we survey state-of-the-art techniques that address the generic tasks listed above. A number of other tasks are

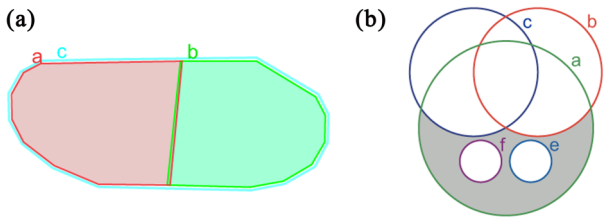

Figure 3: *(a) A well-matched Euler diagram that is not well-formed [RZF08], and (b) a not well-matched Euler diagram with shading that is well-formed [SFRH12].*

also concerned with set-typed data such as hierarchical clustering of the sets or the elements, comparing multiple instances of a set family, and analysing changes in the data over time. Such tasks are often application-specific and require dedicated techniques, and hence are not addressed explicitly in this survey.

4. A Survey of Set Visualization Techniques

This following section extends on our previous survey of set visualizations [AMA*14]. We classify set visualization techniques into six categories listed in Sections 4.1–4.6 according to the main visual representation they use for depicting set relations. The techniques in each category exhibit similar scalability and readability properties as well as design considerations. Also, certain tasks are better supported by a certain category of techniques as we discuss in Section 5. The following subsections list the six visual categories and describe the techniques in each of them. Certain techniques, however, might belong to multiple categories as explained in Section 5 and in Table 3. Available software implementations, demos and videos about the surveyed techniques are available at `http://www.setviz.net`.

4.1. Euler and Venn diagrams

Euler and Venn diagrams are among the oldest [Bar69] and most popular set visualizations. Sets are represented by labelled closed curves and set relations by the curve overlaps. Euler diagrams can depict any set inclusion, exclusion and intersection, but a Venn diagram must show all possible curve overlaps. The closed curves clearly indicate set membership [War12], as the perceptual tendency to organize space into regions is much stronger when indicated by closed curves than by proximity or similarity [Pal92]. Set relations are also easily visible, as the closed curves pop out preattentively, particularly when smooth [TS85, Kof35].

An Euler diagram is *well-matched* to what it represents when the spatial relationships of the curves precisely reflect the set relations [Gur99]. An Euler diagram should ideally be *well-formed* [RZP12], such that it has: (i) simple curves that meet at most at one point where they cross; (ii) every set is represented by at most one curve; (iii) every set relation is represented by at most one region. It is not always possible for a diagram to be well-matched and well-formed [SH14] (e.g. Figure 3), but a study indicates that well-matchedness is more important than well-formedness [CSR*14].

Euler diagrams with non-smooth curves or curves close to one another impede understanding [BR07]. Those drawn with circles are the most effective, followed by those with highly symmetric curves and regions whose shape is highly distinguishable from the curves [BSR*14]. The orientation of the diagrams does not impact understanding [BSR*12].

As shown in Table 1 and the next sections, various techniques have been devised to generate Euler diagrams with different properties and for different data types (see also, surveys on Venn [RW97] and Euler [Rod14] diagrams).

4.1.1. *Techniques for any or for specific set relations*

Techniques that draw well-matched diagrams for any set relation (e.g. [SRHZ11, SAA09, RZF08]) often produce not well-formed diagrams with non-smooth curves (e.g. Figure 3a). The smoothness, shape and closeness of the curves of the diagrams can be improved by other methods (e.g. [MR14c, FRM03]), but not well-formed diagrams are likely to remain not well-formed. Multiple curves for the same set can be used to draw well-matched Euler diagrams (e.g. [SFRH12]).

Techniques that draw well-formed Euler diagrams for any set relations often produce diagrams with smooth, highly symmetric curves, like circles. However, the diagrams might be not well-matched, and the unwanted regions are shaded (e.g. [SFRH12, Ven80]; Figure 3b) or left empty while other regions are filled with glyphs (e.g. [MDF12, Cla08]; Figure 5). Nonetheless, shading is less effective than well-matchedness with respect to human accuracy and time [CSR*14].

Other techniques generate an Euler diagram only for set relations for which a well-matched, well-formed diagram can be drawn (e.g. [SZHR11, FH02]).

4.1.2. *Techniques for area-proportional diagrams*

Euler diagrams can be *area-proportional*, such that the area of each region in the diagram is directly proportional to the cardinality of the relevant set relation. Techniques that draw these diagrams often use circles (e.g. [LM13, Wil12, CR05a]) to facilitate comprehension, but circles have limited degrees of freedom. An accurate area-proportional Venn diagram can be drawn with circles for only two-set data [Cho07]. Thus, misleading diagrams are often produced with circles; e.g. in Figure 4(a), the region with 1 is larger than that with 3.

Techniques using rectilinear [CR03], convex [RFSH10] or irregular [CR05b] polygons produce accurate diagrams for most data. However, these diagrams are often difficult to comprehend, as they are not well-formed and have non-smooth, non-symmetric curves (e.g. Figure 4b). Techniques using regular polygons [KMK*08] produce symmetric curves, but have the same limitations as those using circles.

eulerAPE [MR14b] uses ellipses to produce accurate diagrams for most data with smooth curves (see eulerAPE's evaluation for three-set data; also Figures 4c and d). Methods to accurately and

Table 1: *Features of implemented automatic drawing techniques for Euler and Venn diagrams.*

For any relation	No. of curves	Wellmatched matched	Wellformed formed	Smooth curves	Curve shape	Symmetric curves	Region shading	Area proportional	Cardinality glyphs	Example techniques
√	any	√			polygon					[SRHZ11, SAA09]
√	any	√		√	circle	√				[SFRH12] (no shading)
√	any		√	√	circle	√	√			[SFRH12] (shaded)
	any	√	√		polygon					[FH02]
	3	√	√	√	circle	√		√		[LM13, CR05a]
√	any			√	circle	√		√		[Wil12]
	3	√	√	√	ellipse	√		√		[MR14b]
√	any				polygon	√		√		[KMK*08]
	3	√			polygon			√		[RFSH10, CR03]
√	any	√			polygon			√		[CR05b]
√	1–3	√			circle polygon			√		[RHSF14]
	3		√	√	circle	√			√	[Cla08]
	3			√	ellipse	√			√	[MDF12]

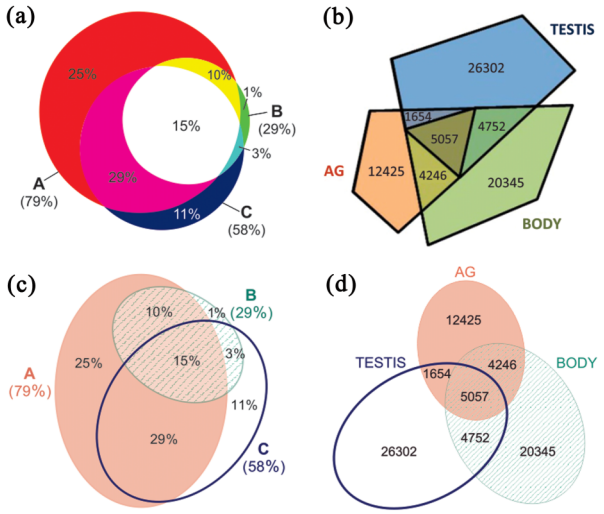

Figure 4: *Area-proportional Venn diagrams drawn with: (a) circles [LF06] using 3 Circle Venn [CR05a]; (b) polygons [BVT*13] using Convex Venn-3 [RFSH10]; (c)–(d) ellipses using eulerAPE [MR14b] for the data in (a) and (b), respectively.*

instantaneously compute the region areas of multiple intersecting ellipses are also available [MR14a].

4.1.3. *Techniques for Euler diagrams with glyphs*

Humans are biased to area judgement [CM84], so techniques are available to depict the set relation cardinalities by the number of glyphs in the regions and not the region areas. TwitterVenn [Cla08] draws such diagrams to depict tweets containing any of two or three user-selected words. *eulerGlyphs* [MDF12] draws similar diagrams with randomly or uniformly positioned glyphs and possibly area-proportional curves for Bayesian problems (e.g. Figure 5a). Differently sized and multi-attribute glyphs could be used to depict different associated quantities [Bra12] (Figure 5b).

4.1.4. *Other techniques*

SketchSet [WPS*11] and SpiderSketch [SDRP11] draw Euler diagrams from hand drawn sketches, with possibility to add shading or graph overlays. In SetFusion, users interact with a recommender system using a Venn diagram [PBT14]. Other techniques draw 3D Euler diagrams [FSR14] or Euler diagrams for reasoning [Sta05] (e.g. spider diagrams [HST05]), specifying constraints [SD08], defining ontologies [HSTC11, OHS*09] and proving theorems [UJ14, UJ12, UJSF12].

4.1.5. *Diagram design*

Euler diagrams come in different designs, but very few empirical studies have been conducted to assess these designs. A different colour per curve is often used. Curve interiors are often coloured with transparency to help identify the curves in which a region is located. These colours perceptually fuse at overlaps, so the colours of regions in the same curve often seem unrelated (Figure 4b) as when unrelated colours are used for regions in the same curve (Figure 4a). A weaving approach [LRS10] has been proposed to alleviate this problem. eulerAPE (Section 4.1.2) avoids colour fusion by using different visual feature channels such as colour, outline and texture for the curves, as in Figures 4(c)–(d), allowing one to easily focus on a specific curve [War12]. A recent study suggests that Euler diagrams whose curves have a coloured outline and no fill are easier to comprehend than ones whose curves have a black outline or a coloured fill with transparency [BSRH14].

4.1.6. *Euler diagram variants*

Several variations of Euler diagrams have been proposed for different purposes. Like Euler diagrams, these techniques use closed regions to represent the sets or subsets thereof.

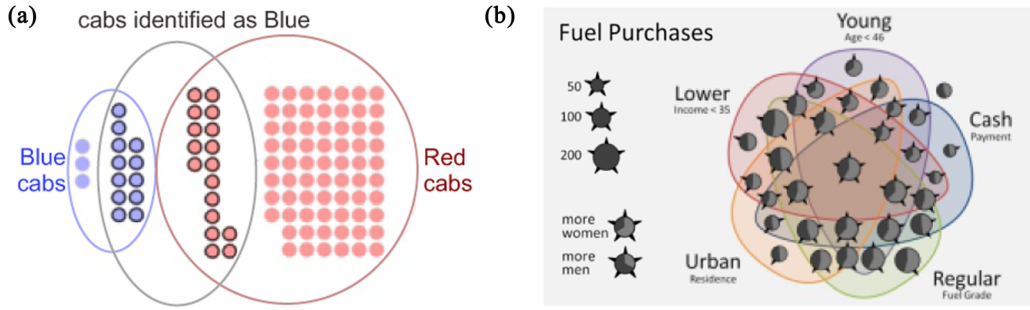

Figure 5: *Euler diagrams with glyphs: (a) eulerGlyphs [MDF12], (b) Brath's [Bra12].*

Missing Pieces [KSJ*06] use concentric rings for showing the results of three search engines (Figure 6a). The outer and middle rings include the elements retrieved by one or two engines, respectively. The inner ring includes elements retrieved by all three engines. The search results are represented as glyphs inside the respective regions and can be coloured to encode additional attributes. *Fan diagrams* [KLS07] use a similar layout to visualize three sets (Figure 18b). Instead of having a separate ring for pairwise overlaps, these overlaps are placed between the respective parts in the outer ring. Both techniques are limited to three sets.

Simonetto and Auber [SA08] proposed a method to handle cases where well-matched Euler diagrams cannot be drawn, by splitting or duplicating certain sets and subsets into disjoint parts, and connecting these parts using edges (Figure 6b).

ComED and *DupED* adopted similar ideas to *untangle* Euler diagrams and ensure their drawability. *ComED* splits a set into multiple rectangular parts, depending on how it overlaps with larger sets (Figure 6c). These parts are connected with hyperedges that preserve the continuity of the set regions, as in Euler diagrams. However, the hyperedges contain no elements and hence their mutual crossings represent no shared elements between the respective sets. The rectangular parts are arranged in a containment hierarchy that reveals several set relations. For example, in Figure 6(c) it is evident that all elements shared between the blue and the pink sets also belong to the green and purple sets. *DupED* creates separate rectangular regions for the sets, and duplicates the elements that belong to multiple sets. Multiple instances of the same element are linked with hyperedges (Figure 6d). It outperforms *ComED* in counting the sets, comparing their sizes and assessing their intersections. However, *ComED* scales significantly better in terms of visual complexity.

Despite using powerful visual properties for depicting set relations, large Euler diagrams (>6 curves) with various curve overlaps are hard to comprehend [ASHC14]. Interaction has been proposed to aid navigation in such diagrams [DKCN14]. Alternatively, different visual metaphors can be used for depicting sets and set relations, as explained in the next sections.

4.2. Overlays

In many scenarios, the set memberships are a secondary information in the data that needs to be analysed in the context of other data features. For example, when the elements have a spatial reference, they are often viewed on a map that provides context information about their locations. Other examples include points in a scatter plot or nodes in a graph. Several techniques have been proposed to augment set memberships over the elements in an existing visualization. These techniques can be classified into four categories according to the visual elements they use: regions (Section 4.2.1), lines (Section 4.2.2), glyphs and icons (Section 4.2.3) and a combination thereof (Section 4.2.4).

4.2.1. *Region-based overlay techniques*

These techniques surround the elements of a set with a closed curve that defines a region. One element can belong to multiple regions if it belongs to multiple sets. Colour is usually used to distinguish between the sets.

Bubble Sets [CPC09] constructs a contour (also named implicit surface) for every set so that it includes all of its elements and excludes all other elements if possible (Figure 7a). For this purpose, it computes an energy map over the pixels in the convex hull containing the set elements. In a second step, it applies the marching squares algorithm to compute the implicit surface from the map. The sets are assigned semi-transparent colours to reveal their overlaps and to keep the context visualization visible. Unlike Euler diagrams, two regions might overlap even if their sets share no elements. Such overlaps should be understood as artefacts that encode no information. An *inverse distance-based potential field* [VPF*14] was proposed to alleviate these artefacts but might result in disconnected regions (Figure 7b). *KelpFusion* (Section 4.2.4) also reduces these artefacts, without disconnected regions. Bubble sets were demonstrated to overlay set memberships over tens of elements in a scatter plot, a graph or a map. Depending on the extent of overlap, the technique can usefully visualize between 4 and 20 sets and still retain enough visibility of the context.

The *GMap* algorithm [GYK10] was used to visualize overlapping sets [GHK10, p. 5]. As with Bubble Sets, the algorithm computes the layout for each set separately by treating points not in the set as obstacles and by connecting the set elements with edges to avoid disconnected regions.

Texture splatting has been proposed to depict AOIs in software architecture diagrams [BT06]. Splatting is applied to a skeleton constructed from the diagram elements according to their size and

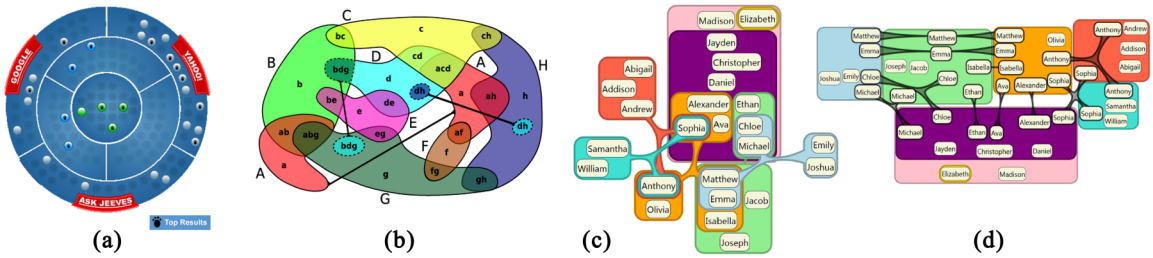

Figure 6: *Euler diagram variants: (a) Missing Pieces [KSJ*06], (b) Simonetto et al.'s [SA08], (c)–(d) ComED, DupED [HRD10].*

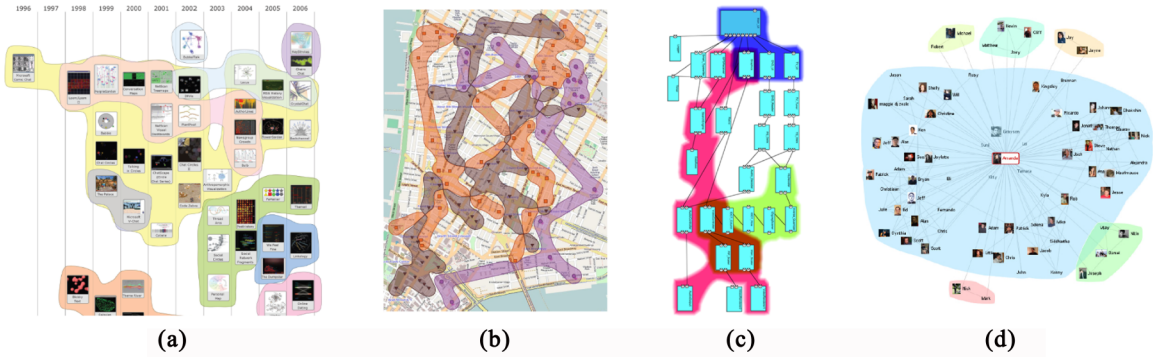

Figure 7: *Region-based overlay techniques: (a, b) Bubble Sets showing groups of items over a timeline [CPC09] and a map [VPF*14], (c) texture splatting to depict areas of interest [BT06], (d) convex hulls to depict clusters in Vizster [HB05].*

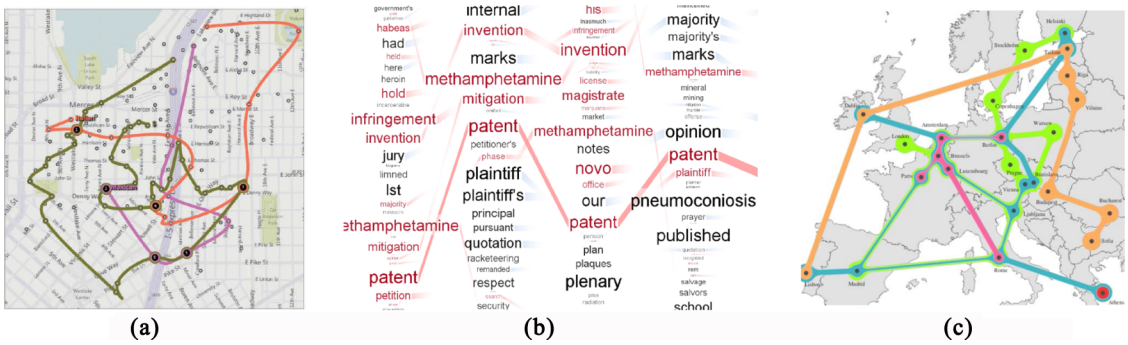

Figure 8: *Line-based overlays: (a) LineSets [AHRRC11], (b) Parallel Tag Clouds [CVW09], (c) Kelp Diagrams [DvKSW12].*

position. A post-processing step erases elements that incorrectly fall within a specific AOI. Overlaps between multiple AOIs are emphasized using subtractive colour blending which creates darker overlapping regions (Figure 7c). Splatting creates smooth boundaries and is applied there only, as it is computationally expensive.

In some cases, the elements in the underlying visualization do not need to be fixed at certain positions. For example, when visualizing sets over graphs, it is possible to move the nodes so that nodes in the same set lie closer together. *IPSep-CoLa* [DKM06] is an algorithm to compute incremental constrained graph layouts. It can be used to visualize sets over graphs, by defining grouping of the nodes into sets as layout constraints. This results in a balanced layout between the sets and the graph. The idea was used in *ComEd* (Section 4.1.6) to create overlapping grouping over graphs interactively [HRD10]. *eXamine* [DEKB*14] generates and displays set-based annotations as contours on top of a node-link diagram. It uses self-organizing maps to lay out nodes, links and contours in a unified way. *Vizster* [HB05] exploits the proximity of nodes in clusters computed by a graph-based clustering of a social network. It visualizes a cluster as a convex hull of the nodes and interpolates the hull boundaries using a cardinal spline (Figure 7d).

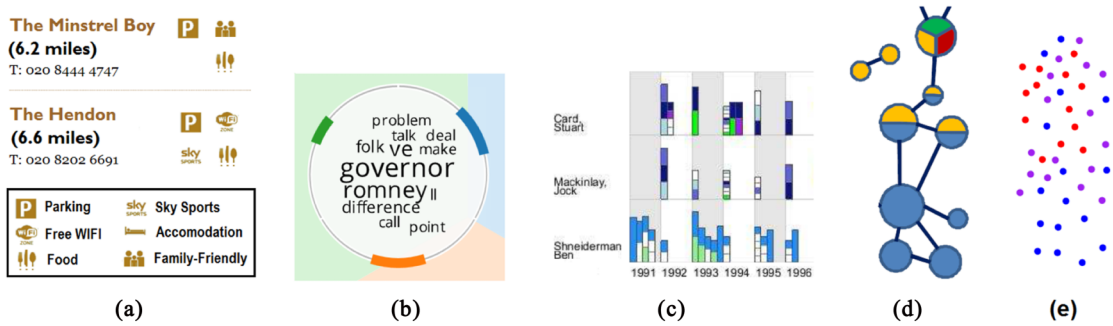

Figure 9: *Glyph and icon overlays: (a) an icon list [Kin], (b) DiTop [OSR*14], (c–d) colour-coded bars and nodes [SOTM06, IMMS09], (e) colour composition [HKvK*13].*

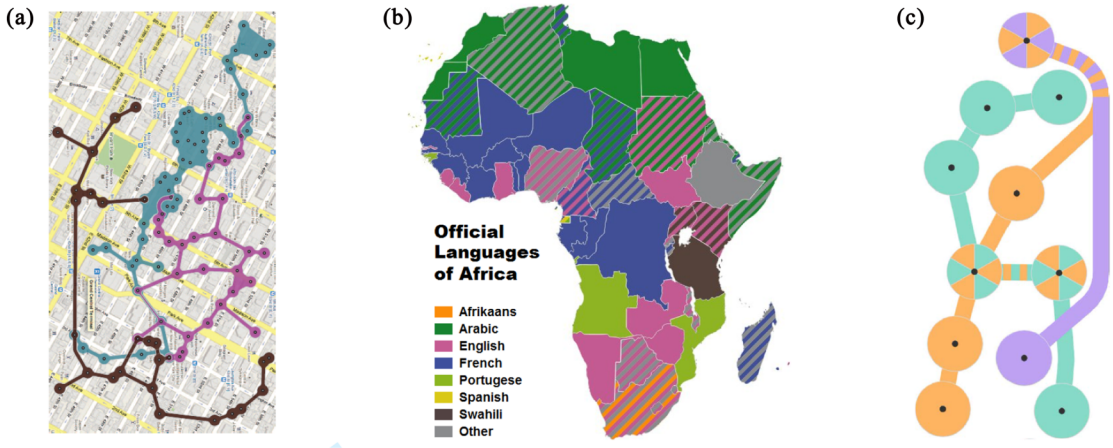

Figure 10: *Hybrid overlay techniques: (a) KelpFusion [MHRS*13] combines lines and regions, (b) region hatching as with colour-coded glyphs [Wik10], (c) a variation of Kelp Diagrams [DvKSW12] that uses both lines and glyphs.*

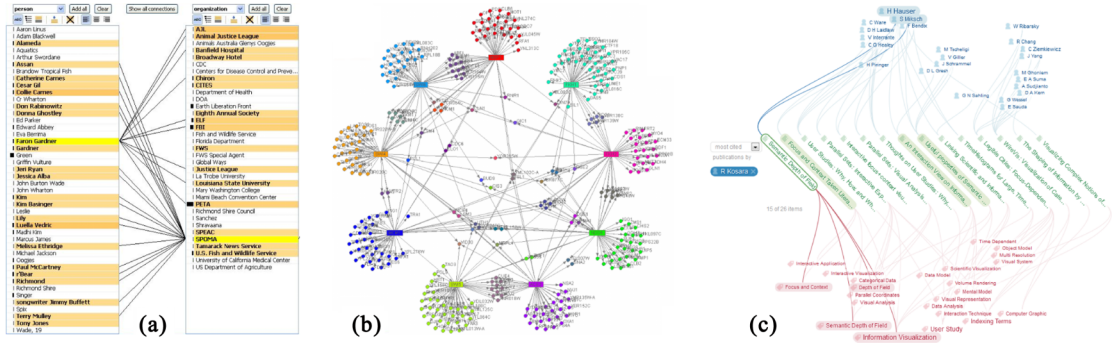

Figure 11: *Node-link techniques: (a) Jigsaw [SGL08], (b) anchored maps [Mis06], (c) PivotPaths [DHRRD12].*

The use of coloured regions to overlay sets facilitates perceiving objects within one region as one group, following Gestalt laws of closure. However, these regions might interfere with and compromise the perception of the underlying visualization. This can impose severe limits on the number of sets that can be overlaid before the visualization is cluttered. This limit depends on the complexity of both the set relations and the underlying visualization. Other overlays have been proposed to address these limitations, as explained next.

4.2.2. *Line-based overlay techniques*

To reduce the ink used in the overlay and the interference with the base visualization, many techniques use lines to represent set membership. Elements that belong to the same set are shown by being present on one or more connected lines.

LineSets [AHRRC11] computes a line for each set that passes through its elements (Figure 8a) using a travelling salesman

heuristic that minimizes the line length. This in turn reduces self-crossings and bends, making it easier to follow the line. The lines are drawn as piecewise Bézier splines of different colours. As with region-based methods, not all line crossings represent set overlaps. Actual overlaps are marked with concentric rings around the elements colour-coded according to the respective sets. Interaction makes certain lines salient, while the other lines are drawn thinner to reduce visual clutter. LineSets were shown to scale better than region-based methods and can overlay up to tens of sets over hundreds of elements. However, connecting set elements with a simple line imposes an ordering on them. This might be an undesirable artefact of the visualization if the elements have no inherent order.

Kelp Diagrams [DvKSW12] connect the elements in a set using a graph structure instead of a simple line. It surrounds each element with a circle clipped to its Voronoi cell to avoid overlaps. Then it computes a tangent visibility graph based on these clipped circles. Each set is computed as a minimum cost graph that connects its elements. This graph aims to capture the shape of a point set on a map. The graph links are routed so that they do not cross elements that do not belong to the respective set. Two design alternatives were proposed to draw overlapping links. Nesting draws the links over each other, with thinner links on top to ensure visibility (Figure 8c). The second design uses alternating stripes to indicate overlapping links (Figure 10c).

In some cases, the base visualization represents the elements of each set separately, and hence creates multiple instances of the same element. An example of this are the *parallel tag clouds* [CVW09] that represent multiple sets of tags (Figure 8b). This technique connects multiple instances of the same tag with a thick path line. To avoid clutter, only the two ends of the edge connecting a tag instance with its next occurrence are depicted. The full segment is shown only for selected tags on demand. While it is hard to follow the instances of an unselected tag, the depicted edge ends reveal if such instances exist or not in parallel clouds. The *context-preserving visual links* [SWS*11] are a generic technique that uses line overlays to link multiple instances of the same element in multiple coordinated views showing different visualizations. The layout algorithm routes the lines, preferably within white space, using a density map to minimize interference with the base visualizations.

Line-based overlays reduce the ink used to visualize the set information, at the cost of following the Gestalt law of closure. Also, the use of simple lines as in LineSets results in an artificial ordering of the elements, that might not be desirable. Finally, lines might interfere with the base visualization if it already uses lines for a different purpose, as in node-link diagrams. Glyphs and icons can offer an alternative for such cases, as explained next.

4.2.3. *Glyphs and icons*

In many cases, it is enough to represent set membership for the individual elements in the base visualization, without the need to represent each of the sets as a connected object. In this case, glyphs and icons can be used as simple overlays to represent the set memberships. Colour-coding is commonly used for this purpose: each set is assigned a colour from a qualitative (categorical) colour scale. Membership of multiple sets can be indicated using multiple icons (Figure 9a), colour-coded glyphs (Figures 9b–d), hatching (Figures 10b and c) or colour composition (Figure 9e), depending on how the base visualization represents individual elements.

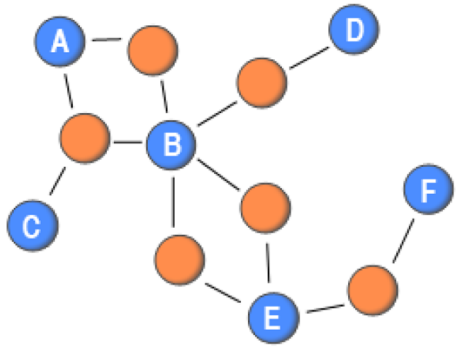

Figure 12: *An affiliation network [BH11].*

Pie-like glyphs are commonly used to compactly overlay set memberships when the elements are represented as circles, such as the nodes of a graph [IMMS09] (Figure 9d). *BiblioViz* [SOTM06] represents papers as bars in a timeline, and overlays coloured segments over the bars to represent multiple co-authors (Figure 9c). However, dividing circles or bars of different sizes into coloured segments might causes a bias regarding the order, size and spatial distribution of these segments. Hatching techniques can alleviate these effects (Figures 10b–c). Alternatively, set memberships can be indicated using separate colour-coded dots or icons within each element when its display area allows for this, as in *SchemaLine* [NXWW14] and *DiTop* [OSR*14] (Figure 9b).

Colour composition [HKvK*13] uses new colours (e.g. purple) to indicate membership of multiple sets (e.g. both red and blue groups as in Figure 9e). However, this is restricted to two or three sets, as it is otherwise hard to memorize all possible colour compositions.

The use of colour the inference of the spatial distribution of the sets. Instead of colour, icons can be used to indicate set memberships (Figure 9a). This is appropriate when the sets represent real-world concepts that have corresponding icons such as flags or common signs. However, without interaction, a serial scan might be needed to identify elements that belong to a certain set.

Other types of glyphs have also been devised for specific applications. Glyphs based on stacked bars [ZXQ15] or superimposed area charts [XDC*13] were proposed to augment a node-link diagram with set-based information about the nodes. Also, *MetaCrystal* [Spo04] uses coloured polygonal glyphs to represent meta search results, where both colour and the number of sides encode how many search engines retrieved a specific document. Finally, coloured pie-like glyphs were proposed to visualize fuzzy membership of overlapping communities in networks [VRW13].

Glyphs and icons offer a lightweight alternative to overlay set information on an underlying visualization in a minimal way. While not suited to encode set relations, they can be designed to effectively

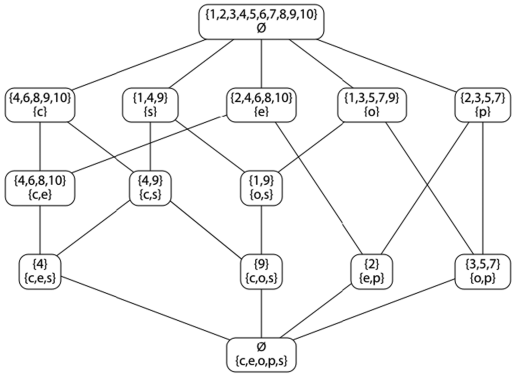

Figure 13: *A concept lattice [Epp06].*

encode element membership of the sets and to reveal the locations of all elements of a specific set in the underlying visualization.

4.2.4. *Hybrid overlays*

Lines, regions and glyphs have also been used in combination, to make advantage of their properties. Three possible types of combinations exist.

KelpFusion [MHRS*13] uses both lines and filled regions (Figure 10a) to bridge Bubble Sets (Section 4.2.1) and Kelp Diagrams (Section 4.2.2). It allows the graph connecting the points in a set to vary from a minimum spanning tree to the convex hull of a point set. Compared with Bubble Sets, KelpFusion uses less ink and alleviates artefacts caused by empty overlapping regions. Compared with line-based overlays, KelpFusion simplifies the visual representation by filling regions of high point density.

When the elements are represented as regions (such as countries on a map), hatching techniques can be used to indicate their set memberships as with coloured glyphs. Figure 10(b) shows how African countries are coloured according to their official languages. Each region is filled with colours that represent its set memberships.

Kelp Diagrams [DvKSW12] offer a variation that uses both lines and glyphs (Figure 10c). Colour-coded glyphs indicate element memberships of multiple set. Also, hatching indicates line segments shared between multiple sets.

Overlay techniques allow the analysis of how certain information and relations between the elements correlate with their set memberships. Alternatively, these correlations can be augmented with other visualizations that better emphasize the set information as in some of the techniques in the next sections.

4.3. Node-link diagrams

Both element-centric and set-centric visualization have been proposed based on node-link diagrams. Element-centric techniques model the membership relations between elements and sets as edges of a bipartite graph whose nodes represent the elements and the sets. Several techniques have been proposed to visualize such bipartite graphs.

A simple layout for bipartite graphs places the elements and the sets in two lists parallel to each other. *Jigsaw* [SGL08] uses this layout to show co-occurrence relations between different concepts in documents (Figure 11a). The lists can be sorted, filtered and coloured according to multiple criteria to show desired elements and sets. Interaction allows exploring set relations on demand to avoid visual clutter and to navigate into long lists. Schulz *et al.* [SJUS08] demonstrated techniques to reduce the clutter caused by crossing edges in such layouts using colour blending and a fisheye lens. Both systems allow exploring additional attributes of the elements using colour or additional columns.

Anchored maps [Mis06] use a circular layout to visualize bipartite graphs. The technique places the set nodes around a circle. The element nodes are placed depending on their set memberships (Figure 11b). Elements that belong exclusively to a set are placed as a bundle of nodes outside the circle, originating from the respective set node. Elements that are shared between multiple sets are placed within the circle, depending on their set memberships.

PivotPaths [DHRRD12] is designed to support strolling in multi-faceted information spaces. Its node-link layout can also be used to depict element-set memberships, by placing the set nodes in the middle line (Figure 11c). An element node is placed at a distance from the middle line that is proportional to its set membership degree. This allows the quick identification of elements that belong to multiple sets. The horizontal position of an element is computed as the mean of the set nodes it is connected to. The elements can be divided into two groups and placed at different sides of the middle line. Interaction allows inferring the set relations for a specific element or vice versa. *Eye diagrams* [CGF*09] use a radial arrangement of the elements to fit more elements in one screen. However, the number of set nodes is limited as they are placed on a line segment encircled by the elements.

Node-link diagrams can also be used to show the similarity between the sets as links of varying thicknesses. *OnSet* [SMDS14] (explained in the next section) allows showing links between the sets to encode their similarities (Figure 1b). *Circos* [KSB*09] uses a circular layout for the nodes, and stripes to encode shared elements between them.

Radial Sets [AAMH13] also uses links to show overlaps of a specific degree between multiple (Figure 16). To depict an overlap of degree 3 or higher, a meta-node of proportional size is created, with tapered links connecting this node to the respective sets. This resembles a hypergraph over the sets, drawn in edge standard [Mäk90]. As mentioned in Section 2.2, hypergraphs can also represent element-set relations, where each element defines a hyperedge over the sets it belongs to or vice versa. The Graph Drawing community proposed several methods for drawing hypergraphs [BCPS12, BVKM*10, KvKS09]. Drawing hypergraphs in *set standard* [Mäk90], results in Euler-like diagrams (Section 4.1). Drawing hypergraphs in *edge standard* [Mäk90] results in a two-mode network (Figure 12), where a dummy node is created for each hyperedge and connected with the graph nodes involved in this hyperedges. In the latter case, it is helpful to colour element nodes and set nodes differently as in affiliation networks [BH11].

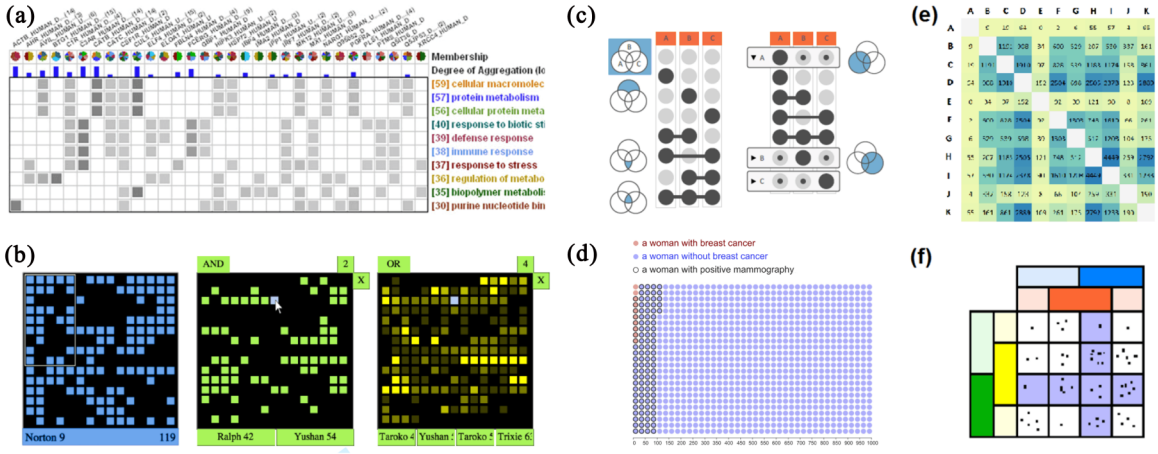

Figure 14: *Matrix-based techniques: (a) ConSet [KLS07], (b) OnSet [SMDS14], (c) UpSet [LGS*14], (d) frequency grid [MDF12], (e) similarity matrix [AAMH13], (f) the KMVQL layout [Huo08].*

Node-link diagrams are commonly used to facilitate reasoning about *Formal Concept Analysis* (Section 2.5). Dedicated layout algorithms were devised to visualize concept lattices [CDE06, EDB04, Wil07] (Figure 13). The layout is usually composed of multiple rows, each containing elements that belong to a specific number of concepts. Concept lattices have been applied to analyse frequent itemsets [BSH13]. Facettice [BKP12] allows interactive definition of concept lattices for faceted search. It uses glyphs to visualizes combinations of search facets, along with further attributes of the elements that satisfy these combinations.

Node-link diagrams are generally easy to understand. The use of nodes as visual objects allow encoding further information about the respective element or set, as in Jigsaw (Figure 11a) and Facettice [BKP12]. Also, besides showing set similarities as links between nodes, both OnSet (Section 4.4) and Radial Sets (Section 4.5) show information about individual or aggregated elements in these sets in their nodes. However, due to edge-crossing, node-link diagrams suffer from increasing clutter as the number of links increases. While layout and order algorithms can reduce this clutter, the diagram is often limited to tens of nodes having about one hundred of links.

4.4. Matrix-based techniques

Different methods have been proposed to visualize set memberships using matrices. These approaches are either element-centric or set-centric, and take advantage of the clear and flexible metaphor of matrices.

ConSet [KLS07] maps sets and elements to rows and columns, respectively. The cells encode set memberships (Figure 14a). The rows and columns are reorderable, as set and element names have no pre-defined order. The reordering can both simplify the matrix and reveal patterns in it, such as clusters of elements that exhibit similar set memberships. Several interactions and visual aids are possible with the matrix representation, such as the aggregation of elements or sets. Aggregated elements can be indicated visually using darker cells or additional bars. To facilitate inferring to which sets an element belongs, the cells can be coloured by unique set colours. Also, to facilitate inferring the elements that belong to a set, the respective cells can be connected with a line, instead of showing grid lines [ZKBS02].

OnSet [SMDS14] represents each set as a separate nearly square matrix whose cells encode which elements belong to the set (Figure 14b). Each element is represented by a unique cell position across all matrices. Hovering the mouse over an element highlights the respective pixels in the sets it belongs to. Drag and drop interactions allow aggregating multiple sets into one matrix using union or intersection (Figure 14b). Links of varying thicknesses can be displayed between matrices to represent their shared elements, with link thickness proportional to a similarity measure based on these overlaps. Hovering a link highlights these elements (Figure 1b). Representing sets as nearly square matrices facilitates perceiving them as containers of the elements, following Gestalt's law of closure [Wer38]. Moreover, the matrix can be divided into tiles to represent a hierarchy over the elements.

Frequency grids [MDF12] represent the elements as cells in a matrix, and places a glyph in each cell to encode the respective set memberships (Figure 14c). They facilitate element counting. However, they are limited to only a few overlap combinations between a small number of sets.

Set-centric techniques visualize relations between the sets. A matrix can alternatively depict how the sets overlap with each other, by representing the sets both as rows and as columns: Each cell contains a similarity measure between the respective sets (Section 2), encoded in colour as in a heatmap (Figure 14e). Each pair of sets corresponds to two cells in the matrix. Therefore, the matrix can fit two symmetric measures, or one asymmetric measure. The matrix can be reordered to reveal clusters of sets that exhibit high overlap with each other. Overlaps of higher degree can be represented by further dividing the rows or columns. This however results in a complex matrix that contains several redundancies.

KMVQL [Huo08] is a system to support formulating queries over a collection of items, by defining Boolean combinations of different

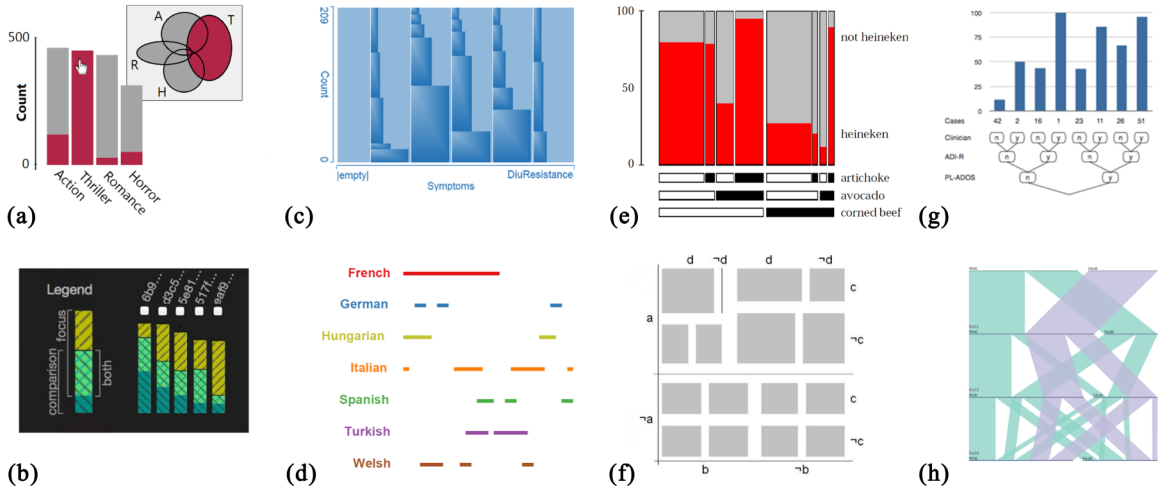

Figure 15: *Aggregation-based techniques: (a) an interactive bar chart [AAMH13], (b) SEEM [GSG*14], (c) Set'o'gram [FMH08], (d) a linear diagram [CSR*14, G*14], (e) Double-Decker plot [HSW00], (f) Mosaic displays [Hof00], (g) a binary tree along with quantitative values [Kos07], (h) Parallel Sets [KBH06],*

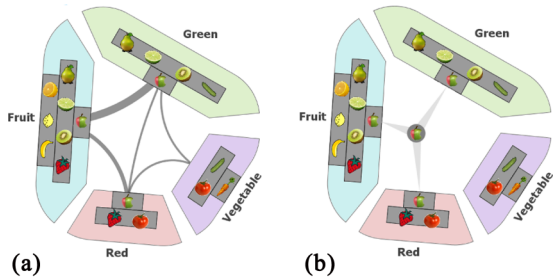

Figure 16: *Radial Sets [AAMH13] showing a breakdown of set elements by their degrees using grey histograms. The links show overlaps between pairs (a) or triples (b) of sets.*

search criteria. It encodes all possible 2^n combination of n sets in a matrix (Figure 14f). The user can click on a cell to include the elements it represents in the query result. Also, the cells can encode the frequency of elements in the respective set combination via colour.

UpSet [LGS*14] represents sets as matrix columns, with rows representing different set intersections. Dots are placed in cells that correspond to the sets involved in each intersection, and are connected with a straight line (Figure 14c). The matrix can also represent aggregates of set combinations as expandable rows whose dots encode the sets involved in each aggregate. UpSet also uses bars to represent the elements in each combination along with other information about them, as explained in the next section.

Matrices offer a clutter-free visual metaphor that can represent different aspects of set-typed data, and can generally scale better than node-link diagrams. However, these techniques are sensitive to the ordering of the rows and columns, which has major impact on the insights they can reveal. As we note in Section 6.2, this metaphor is not fully exploited for set visualization, which gives opportunities for developing new set visualization techniques based on matrices.

4.5. Aggregation-based techniques

When the number of elements is large, it becomes less feasible to depict and investigate how single elements belong to the sets. Following Shneiderman's visual information-seeking mantra [Shn96], many techniques provide an overview of such data first, and allow exploring details about certain elements on demand. These techniques employ frequency representations of set-typed data to show the number of elements in different sets and subsets. They aggregate multiple data elements into a single visual element that encodes this frequency.

Interactive bar charts have been used to depict the sizes of the sets and reveal the set overlaps as the bars are brushed [AAMH13]. Unlike traditional bar charts, an element can be aggregated in multiple bars, as it might belong to multiple sets. Clicking on one bar selects the elements in the respective set, and highlights the fraction that these elements represent in the other bars, revealing how certain pairs of sets overlap (Figure 15a). The selection can be refined further using set operations between new selection and previously selected elements, to investigate the overlaps between multiple sets. However, this chart does not readily reveal how the sets overlap and can only depict certain overlaps on demand.

SEEM [GSG*14] visualizes the relation of one focus set A with the other sets in the set system (Figure 15c). Each column in the visualization corresponds to one of these sets S_i and resembles a two-set Euler diagram showing $A \setminus S_i$, $A \cap S_i$ and $S_i \setminus A$ as stacked bars of corresponding sizes. The columns can be sorted by any of these quantities.

Set'o'gram [FMH08] is an extension to the interactive bar chart, designed for set-typed data. Instead of directly visualizing the

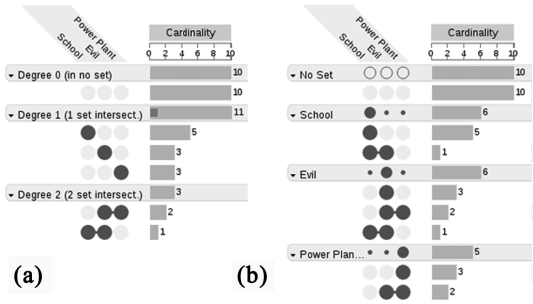

Figure 17: *Example grouping options in UpSet [LGS*14]: (a) grouping by element degree, (b) grouping by sets.*

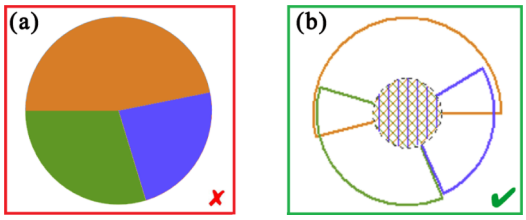

Figure 18: *(a) A pie chart of set sizes distorts their relation to the whole. (b) A Fan diagram [KLS07] showing set overlaps.*

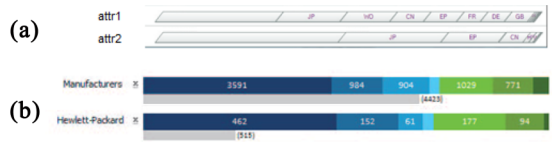

Figure 19: *Equal bar histograms to show set sizes: (a) parallel bars as special motives [WMLP12], (b) an additional bar to indicate the total number of elements [BCH*13].*

overlaps between the sets, a set'o'gram indicates how many elements in each set are shared with how many sets. For this purpose, it divides the bars representing the sets into sections that correspond to elements of different set membership degrees (Figure 15b). Starting from the bottom, the ith section in a bar represents elements in the respective set that are shared with $i-1$ other sets. The height of a section is proportional to the number of elements aggregated in it. Starting from the top, the sections are assigned increasing widths and are shaded along their diagonals to distinguish between successive sections and to reveal empty sections that have zero height. The sections can be selected and highlighted individually to reveal the sets involved in them.

The *Double-Decker plot* [HSW00] shows how multiple Boolean variables correlate. The upper part of this plot encodes the number of elements in each possible set combination by means of equal-height histograms (Figure 15e). This allows easy comparison between selected portions in different overlaps, as the respective bars in the upper part are of the same height. The histogram bars are arranged according to a hierarchy of set memberships which is depicted in the lower part by means of multiple rows. Starting from the bottom, row i is divided into 2^i parts that correspond to the different membership combinations of the sets $S_1..S_i$. This gives an overview of how the sets overlap, however, from the perspective of the set that defines the first partitioning level. The *set co-occurrence view* [Wit10] uses a similar plot to support set-typed data in the bargrams interface. This interface uses additional rows to show the possible values of other attributes and the frequencies of these values. *Linear diagrams* [CSR*14, G*14] allow for more generic divisions of the rows (Figure 15d). A circular version of these diagrams were proposed [ZXQ15] which plots set overlaps in a ring, in order to plot detailed information about selected elements in the inner area inside the ring.

Kosara [Kos07] proposed a redesign of Venn diagrams composed of two parts, as with Double-Decker plots. The lower part consists of a binary tree whose branches represent different set combinations (Figure 15g). The upper part is a bar chart that encodes quantitative information about the respective overlaps.

Mosaic displays [Hof00] is a space-filling technique that recursively partitions the space along the categories of multiple categorical variables (Figure 15f). To visualize set-typed data, set memberships can be treated as binary categorical variables [FMH08]. However, using both horizontal and vertical subdivisions makes it hard to relate display tiles that belong to the same set.

Parallel Sets [KBH06] can be applied to visualize set-typed data by treating set memberships as binary categorical variables. Each set is represented on a separate horizontal axis using two boxes of proportional size to represent both the elements that belong to the set and the remaining elements (Figure 15h). Up to four stripes connect the boxes between the two topmost axes to represent elements that fall in the respective set membership combinations. In the standard mode, the stripes are split further as they pass through the remaining axes, representing all possible set combinations. Unlike mosaic displays, Parallel Sets represent the elements of a set in one box only instead of several tiles. However, splitting the stripes increases them by a factor of 2, as with the mosaic tiles. Moreover, the stripes overlap, causing clutter with more than four sets. A bundled mode of the stripes reduces this clutter but causes stripe discontinuity.

Radial Sets [AAMH13] provide a more detailed overview of set-typed data than the above-mentioned techniques. The sets are depicted as non-overlapping regions with a radial arrangement. The elements are represented as histogram bars inside these regions, grouped by their degrees (Figure 16a). Overlaps between pairs of sets are represented as links of proportional thicknesses. Overlaps between triples of sets are represented by hyperedges between the respective regions (Figure 16b). A variety of interactions allow selecting elements based on their set memberships and attributes. Such queries can be defined iteratively, by combining multiple selections using set operations. Radial Sets use colour to indicate selected elements. When no elements are selected, colour can be used to encode aggregated attribute values of the elements aggregated in the histogram bars. This enables correlating element attributes with their set memberships.

InfoCrystal [Spo93] uses glyphs to represent all possible set overlaps. The set labels are placed on a circle and act as magnets on the glyphs to determine their placement. A follow-up work [Spo04] demonstrates the use of glyph sizes to encode overlap sizes, and the use of colour to encode the sets involved in the overlap.

Though the Double-Decker plot, Mosaic Displays, Parallel Sets, and InfoCrystal, represent all possible set overlaps, they are very limited in scalability and visual accuracy. The *overlap analysis view* [AAMH13] provided in the Radial Sets interface allows exploring all possible set overlaps using tabular lists. To allow for meaningful comparison, each list contains overlaps between a specific number of sets. The lists provide information about the overlaps such as size and disproportionality either graphically or in text.

UpSet [LGS*14] uses a bar chart to represent all possible set intersections (Figure 17). As explained in Section 4.4, the sets involved in each intersection are encoded via a row in a matrix. The length of the corresponding bar encodes the number of elements in the intersection. Other information can be depicted about these elements in additional columns (Figure 1c). One example is the disproportionality represented by each aggregate, computed by comparing actual and expected sizes of the respective set intersections. Another example are boxplots that show attribute distributions, which enable the analysis of element-set memberships in relation to element attributes. Three key features of UpSet are sorting, grouping and querying by various set-based and attribute-based criteria. For example, the rows can be sorted and grouped according to element degree so that an equal number of sets is involved in the bars of each group (Figure 17a). The groups are indicated by creating collapsible rows showing aggregated information about the union of its elements, in the same fashion as with any other row. Set-based queries can be defined also by adding a new row and specifying set membership criteria using the matrix dots. The query results are represented as an expandable group containing all set intersections that match the specified criteria, in the same fashion as with other groups. Certain grouping modes and set-based queries might result in duplicate rows. For example, grouping by set causes each intersection between multiple sets to be duplicated in the respective groups (Figure 17b).

In some cases, there is a need to provide a compact overview of set sizes as part of an information-dense interface. A common mistake is to show the set sizes via a pie chart, as the chart categories are not mutually exclusive and do not sum up as parts of a complete whole (Figure 18a). Fan diagrams (Section 4.1.6) address this issue by explicitly visualizing the overlaps between three categories (Figure 18b). To handle more sets, stacked bars with special motifs indicating possible overlaps [WMLP12] were proposed (Figure 19a). Showing the total number of elements helps quantifying the degree of overlap [BCH*13] (Figure 19b).

Aggregation-based techniques are highly scalable with the number of elements, but vary in their scalability with the number of sets. Except for mosaic displays (Figure 15c), these techniques might aggregate the same element in multiple visual elements, depending on the sets it belongs to. Some visualizations indicate this element redundancy explicitly, as with the links in Radial Sets and the collocated bars in Double-Decker plots. Interaction is needed to investigate which elements are present in multiple sets, and to obtain detailed information about selected elements.

4.6. Scatter plots and other techniques

Other techniques to visualize set-based information were proposed, that do not naturally fall under the five visual categories listed in Sections 4.1–4.5.

Scatter plots have been proposed to visualize similarity between sets, by representing sets as points in a 2D plane. A *scatter view* [LLS05] visualizes the similarity between a specific set, and the rest of the sets. It depicts two asymmetric similarity measures against each other to find which set is closer to the reference set both in overlap intensity and completeness. A *cluster view* [LLS05] was proposed to reveal clusters of similar sets by placing them closer together, similar to how multi-dimensional scaling operates (Figure 20a). *Correspondence analysis (CA)* [Gre84] has been used to visualize two-mode social networks by treating them as binary contingency tables [BH11]. Figure 20(c) depicts the CA plot for the southern women data set (Figure 2c). The plot contains points both for sets and elements. Edges can be optionally overlaid between the sets and elements. Close element points usually correspond to similar set memberships. Close set points usually correspond to high overlap.

All three scatter plot–based techniques described above suffer from two problems that limit their applicability to set-typed data. First, using dots to represent sets does not emphasize them as containers of elements. Secondly, set similarity measures do not define a distance function, which make the above-described 2D projections problematic: close points could be produced for two disjoint sets, if both of them are similar to a third set.

Bicentric Diagrams [PB15] were developed for bifocal network analysis. The visualization uses concentric circles to shows the neighbourhood of two focal nodes A and B (Figure 20b). Each inner circle represents immediate neighbours of the respective focal node. Each outer circle represents indirect neighbours at graph distance of 2 of the respective focal node. The intersections between the above four circles represent shared nodes between the respective neighbourhoods. Except for inner circles, each pair of circles intersect in two points, making it possible to split the shared nodes into two groups (e.g. based on their connectivity). PivotPaths (Figure 11c) also provide a similar bifocal mode for exploring faceted data [DHRRD12].

Further set visualization techniques would be possible, employing different visual metaphors than the ones presented so far. Combined and hybrid techniques can also be devised. Such techniques might be suited for specific problems (such as bifocal network analysis), or special forms of set-typed data (Section 6.2).

5. Comparison and Findings

To provide guidance on applying the surveyed set visualization techniques to a given problem, we compare the techniques according to the following three aspects.

5.1. Comparison by what is represented

Set-typed data can encompass information about sets and their relations, elements and their set memberships and other element attributes. The surveyed techniques differ by the type of information they represent:

- **Representing set information only:** These techniques provide no information about the individual elements. This

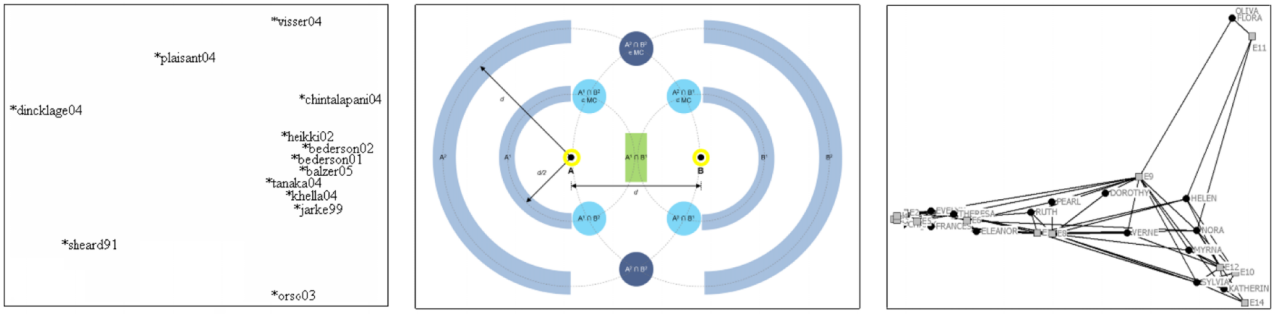

Figure 20: *Scatter plots and other techniques: (a) Cluster view [LLS05], (b) Bicentric Diagrams [PB15], (c) correspondence analysis view of the southern women data set [BH11].*

Table 2: *Selected strengths and weaknesses of the visual categories (Section 4).*

Category	Strengths	Weaknesses
Euler-based diagrams	Intuitive when well-matched (little training is required). Represent all standard set relations compactly.	Limited to few sets due to clutter and drawability issues. Desired properties not always possible (e.g. convexity).
Overlays	Emphasize element and set distributions according to other data features (e.g. map locations).	Often limited in the number of elements and sets. Undesired layout artefacts (overlaps, crossing, shapes, etc.).
Node-link diagrams	Visually emphasize the elements as individual objects. Show clusters of elements having similar set memberships.	Limited scalability due to edge crossings. No representation of set relations in element-set diagrams.
Matrix-based techniques	Fairly scalable both in the number of elements and sets. Do not suffer from edge crossings or topological constraints.	Limited in the set relations they can represent. Revealed membership patterns are sensitive to ordering.
Aggregation-based	Highly scalable in the number of elements. Some techniques can show how attributes correlate with set membership.	Usually, do not emphasize sets and elements as objects. Limited in the set relations they can represent.
Scatter plots	Show clusters of sets according to mutual similarity. Clutter free and scalable when showing sets only.	Do not represent standard set relations. Dots are often perceived as elements not as sets.

includes standard Euler diagrams that represent set relations, as well as matrices, node-link diagrams and scatter plots that represent set similarities.

- **Representing individual elements explicitly**: Examples are Euler diagrams with glyphs, overlays, element-set node-link diagrams, membership matrices and frequency grids. Further element attributes can often be represented using additional visual features or additional columns.
- **Representing element aggregates**: As discussed in Section 4.5, such techniques depict groups of elements, possibly along with relations between these groups. Some techniques (e.g. Double-Decker, UpSet and Radial Sets) can depict aggregated attribute values for group elements.

The techniques vary also in the set relations they represent explicitly. Euler diagrams show inclusion, exclusion and intersection relations. Scatter plot–based and some aggregation-based techniques (e.g. Set'o'grams) do not represent these relations explicitly. Other aggregation-based, node-link and matrix-based techniques represent certain set relations only (usually set intersections).

Finally, certain techniques show multiple instances of the same element according to the sets it belongs to. Examples for this include the DupED version of untangled Euler diagrams (Figure 6d) and parallel tag clouds (Figure 8b). Also, membership matrices fill multiple cells for the same element (Figure 14a). Visual duplicates allow set-dependent attributes (Section 2.1) to be shown, e.g. different tag frequencies or ranks in multiple clouds.

5.2. Comparison of general strengths and weaknesses

Each of the techniques categories listed in Section 2 has advantages and limitations associated with the visual representation it employs. Table 2 summarizes the major ones that generally apply to the techniques in the respective category. However, it should be noted that individual techniques have their own advantages and limitations,

Table 3: *Visual categories of selected techniques from Section 4.*

Technique	Euler-based	Overlay	Node-link	Matrix	Aggregation	Other
Euler diagrams	•					
ComED	•		•			
DupED	•		•			
BubbleSets	•	•				
LineSets		•				
Kelp diagrams		•	•			
KelpFusion		•	•			
Icon lists		•				
Linked lists			•			
Anchored maps			•			
PivotPaths			•			
ConSet				•		
OnSet			•	•		
Frequency grids				•		
Similarity matrix				•		
KMVQL				•	•	
Mosaic displays					•	
Double-Decker plot					•	
Sets'o'grams					•	
Radial Sets			•		•	
Parallel Sets					•	
Linear diagram					•	
UpSet				•	•	
MetaCrystal		•			•	
Scatter view						•
Bicentric Diagrams			•			•

and might belong to multiple categories (Table 3). For more details, refer to Section 4 and to the respective articles.

5.3. Comparison by supported tasks and scalability

The surveyed techniques differ in the tasks (Section 3) they support. Table 4 provides an overview of the tasks supported by a representative subset of techniques from all surveyed categories. The task support was either indicated by the authors or judged by us based on published work. We indicate whether the task is supported fully, partially or through interaction only. Partial support means that the technique is not always effective for the respective task, or support the tasks to a limited extent (e.g. with few sets only). In addition, we give a rough estimate of the scalability of the techniques, both in the number of sets and in the number of elements, when applicable. Actual scalability limits depend on the complexity of the specific data set, such as overlap strength and skewness in the set sizes.

The comparison matrix in Table 4 reveals how the techniques in the same category tend to have similar task support characteristics. As expected, this demonstrates the decisive influence of the visual encoding used by a technique on the types of tasks it supports. Note that certain techniques belong to multiple categories (Table 3). The matrix also reveals that certain techniques depend heavily on interaction in supporting their tasks.

Clearly, there is no single technique that supports all tasks. The choice of the technique to use for a specific problem requires extensive analysis of the problem domain and its data characteristics. This is important to determine the tasks that need to be supported and the actual scalability requirements.

6. Future Challenges and Opportunities

The techniques surveyed in Section 4 demonstrate the significant advances made in the past decade in visualizing sets and set-typed data. Nevertheless, research in this area is still in early stages, with many open problems and challenges that need to be addressed in the future. In the following, we give some of these problems and provide a list of unexplored research directions that could help in addressing them.

6.1. Open problems

The following problems need further research to improve on state-of-the-art techniques. Some of the issues we list are specific to certain techniques, while others are more generic in set visualization. In addition, some problems are concerned with complicated forms of set-typed data.

Generating Euler diagrams with specific properties: There are no generic tools that indicate, for a given input, whether it is possible or not to generate diagrams that are well-matched, well-formed, area-proportional, and/or use certain shapes (e.g. circles or

Table 4: *Comparison of selected techniques from Section 4 by the tasks they support (Sect. 3).*

	Technique	Element-related Tasks							Set-related Tasks													Attribute-related Tasks					Scalability	
		A1	A2	A3	A4	A5	A6	A7	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	C1	C2	C3	C4	C5	in # of sets	in # of elements
Euler-based	Euler diagrams	●	●	●	○			○	○	●	●	●	●	○	●	●	○	●	○	○	n/a	○	○	○	○			about 10	hundreds / ∞
	ComED	●	●	●	○			●	○	●	○	○	○	●	○	○				○	○	●						10 to 20	hundreds
	DupED	●	○	○	○			●	●	●	●	●	○			○				○	○	●						about 10	tens
Overlays	BubbleSets	●	●	○	○			●	○	○	○	○	○	○		○	○				○		○	○				about 10	tens
	LineSets	●	●	○	○			●	○					○		○				○	○		○					10 to 100	hundreds
	Colored glyphs	○	●	○	○				○				○	○		○				○	○					○		10 to 20	hundreds
	Icon lists		●					●												○			●			○	○	tens	large list
Node-link	Linked lists	🖱	🖱	🖱	🖱	🖱	🖱	🖱	●												○		●	○	○	🖱	🖱	hundreds	hundreds
	Anchored maps	○	○		○		○		●				○		○	○		○		●	🖱		🖱					20 to 50	hundreds
	PivotPaths	🖱	🖱		🖱		🖱		●					○						🖱	●		🖱					50 to 100	hundreds
Matrix	ConSet	●	●	○		🖱	🖱	🖱	●			🖱	🖱	○							🖱	🖱	●					about 100	about 100
	OnSet	●	🖱	🖱					●			🖱	●		○	○	○	●	○	●	○	●	🖱	🖱					tens	hundreds
	Frequency grids	●	●	○	○	○	○	○	○	●			●	●	●	●	○	●		●	●		🖱					3 to 5	hundreds
	Overlap matrix								●			●	●			●	●	○	○	○	●							about 100	not applicable
	KMVQL	🖱		🖱					●			○	○	○	●	●					●				●			4 to 6	not applicable
Aggregation	Mosaic displays			🖱					●			○	○	○	●	●		○		○	●				●			up to 4 sets	large (agg.)
	Double-Decker	○		●					●	🖱	🖱		●	○	●	●		●		●	●			●	●			4 to 6	large (agg)
	Sets'o'grams	🖱		🖱	●	🖱	●		●				🖱	○	🖱	🖱				○						🖱	🖱	50 to 100	large (agg.)
	Radial Sets	🖱		🖱	●	🖱	●		●			●	●	●	🖱	🖱	●	●	●	●	●	🖱		🖱	●	🖱	🖱	20 to 30	large (agg.)
	UpSet	🖱	🖱	🖱	●	🖱	●		●			○	●	●	●	●	○	●		🖱	●	🖱		●	●	🖱		20 to 30	large (agg.)
Scatter	Scatter view								○										○									hundreds	not applicable
	Cluster view								○										●									hundreds	not applicable

● Task is supported

○ Task is partially supported

🖱 Task requires interaction

A1: Find/Select elements of a specific set
A2: Find sets containing a specific element
A3: Find/Select elements by set memberships
A4: Find/Select elements by their degrees
A5: Filter out elements by set memberships
A6: Filter out elements by their degrees
A7: Create a set out of certain elements

B1: Find the number of sets in a family
B2/3: Inclusion relations / hierarchies
B4/5: Exclusion / intersection relations
B6: Identify intersections between k sets
B7: identify sets involved in an overlap
B8: Identify intersections of a set
B9: Identify the set with largest / smallest number of pair-wise set intersections
B10: Analyze & compare cardinalities
B11: Analyze & compare set similarities
B12: Analyze & compare set exclusiveness
B13: Highlight specific sets, subsets, etc.
B14: create a set by set-theoretic operation

C1: Find an element's attribute values
C2: Attribute distribution in a set / subset
C3: Compare attribute values between subsets
C4: Set memberships for specific attr. values
C5: Create a set of elements by attributes

convex polygons). Rodgers [Rod14] elaborated on related open research questions in generating Euler diagrams. Tools that determine whether a diagram can be drawn with desired properties and propose alternative solutions to non-drawable cases (e.g. using shading or approximate areas) would improve the quality of the generated Euler diagrams and their applicability in various domains. In this regard, a high-level algorithm has been proposed to determine the drawability of a well-formed diagram and generates the diagram in that case [FFH08], but no implementation is available yet.

Scalability: As Table 4 shows, it is not always possible to support tasks if they have particular scalability requirements. Moreover, the scalability of certain techniques is severely limited, such as overlays. Improving upon these limits is necessary to address various real-world problems that involve a large number of sets and/or elements.

The role of ordering: By definition, set-typed data impose no inherent ordering neither on the elements nor the sets. However, the order in which sets and elements are depicted has a significant impact on the patterns and relations revealed by the visualization. Though reordering problems are usually NP-complete, a lot of work has been done for reordering generic matrices and node-link diagrams to reveal clusters and/or reduce clutter. This work needs to be revisited from a sets perspective, e.g. by incorporating set-related data features such as element degrees. Also, more work is needed on the role of ordering in aggregation-based techniques.

Evaluation: There is a clear lack of empirical user studies that assess the effectiveness of different techniques in performing different tasks. Some comparative studies focus on techniques from the same category, such as Euler diagrams [BSRH14] or overlays over a specific visualization [AHRRC11, MHRS*13]. Few studies compare techniques from different categories. A recent study [CSR*14] assessed Venn diagrams with shading, well-formed Euler diagrams with shading, not well-formed but well-matched Euler diagrams and linear diagrams (Section 4.5). Tasks involved set intersection, inclusion and exclusion. The linear diagram outperformed all the three Euler diagrams variants with respect to accuracy and time.

More evaluation work is needed to determine which techniques work well for which data characteristics and tasks, and to steer future research towards promising directions.

Visualizing sets in the context of other data types: Overlay techniques reveal set memberships of elements placed according to other data features. However, they offer limited possibilities as the layout of the overlays cannot influence the element placement. Designing set-aware visualizations can improve on this. For example, a set-aware graph layout would compute a node placement that reduces edge crossing and produces convex-shaped overlays at the same time x. Further work is needed to visualize sets over elements in a timeline, a tree or a multi-variate visualization.

Comparing multiple set families: In many scenarios, multiple instances of a set family are compared (e.g. how skill overlaps change across different companies). With few sets, small multiples of Euler diagrams help in comparing the set relations between the respective set families (Figure 21). For example, the comparison

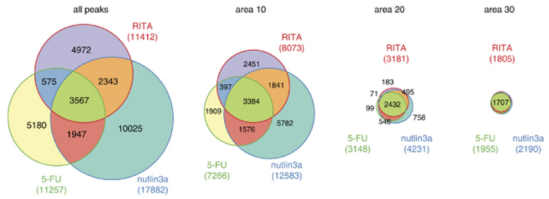

Figure 21: *Multiple set families [NSL*12].*

might involve finding which set relations or attribute values change most/least across the different families. Dedicated techniques are needed to support such comparison tasks in a scalable way in the number of sets and families. The linear layout of UpSet [LGS*14] (Section 4.5) provides a good starting point for this, as multiple columns can be created for multiple sets families.

Time-varying set-typed data: As with many types of data, set-typed data can vary over time. For example, in *evolutionary set theory* [TAON09] set memberships might change over time, leading to changes in set relations. Analysing these changes is a powerful tool to study the dynamics of set structured populations [TAON09]. Also, the attribute values of the elements might change over time even with static set memberships. Visualizing changes in set-typed data is challenging, as the data are already complex. Bubble Sets [CPC09] allow smooth recomputation of set overlays, making them suited to track the spatial distribution of set elements e.g. in an animated scatter plot. A technique similar to Parallel Sets was proposed to visualize object-group changes over multiple time steps [vLBA*12], however, allowing an element to belong to one set at a time.

Visualizing fuzzy and uncertain set memberships: Real-world data typically involve uncertainty that result in fuzzy set memberships. *disk diagram* [PP10] is a technique for analysing fuzzy data using interactive visualization of fuzzy set operations. More work on both analytical and visual methods is needed to communicate the fuzziness in the data and study its effect on various set-related tasks.

6.2. Possible opportunities

Next, we list ideas and research directions that could improve on existing set visualization techniques.

Interaction: Many of the techniques we surveyed make limited or no use of interaction to support the analysis tasks. Interaction opens new possibilities for addressing various challenges with analysing and visualizing set-typed data. For example, when generating Euler diagrams, the user could specify certain constraints and properties or choose where to take a compromise when they are not satisfiable. Interactivity makes simplifying complex visualizations possible by showing certain information on demand and selecting certain parts to explore in more detail. It also facilitates various comparisons within one set family or across multiple families. Interaction allows influencing matrix reordering, e.g. to restrict changes to certain rows or columns. Likewise, it helps in exploring large set-typed data at

multiple levels of detail and in applying appropriate data reduction methods. Finally, intuitive interactions allow sets to be defined and combined using Boolean operations and enables performing multi-faceted search over a set of elements.

Coordinated multiple views: can reduce the complexity of the data by showing information at multiple levels of detail. This can also provide complementary perspectives on the data (e.g. overlap matrix + spatial set distribution) to enrich the analysis.

Small multiples: could provide solutions to visualizations that are severely limited in the number of sets, such as Euler diagrams, Mosaic Displays or Double-Decker plots. They can also be used to compare, for instance, data with certain attribute values to determine if they correlate with certain set relations or membership patterns.

Hybrid representations: might be useful in certain cases, especially when the sets can be semantically divided in two groups. An example, in a 3×3 matrix of three sets, each cell can additionally depict how its elements belong to another group of sets by using a different visualization such as an Euler diagram. Another example is combining glyphs with frequency-based representations to visualize both the sets involved in an overlap and the overlap size.

Matrix-based representations: are not fully exploited for visualizing set-typed data. They are relatively simple and clutter-free, and fairly scalable in the number of rows and columns. Moreover, there are several possibilities to encode multiple values in a matrix cell [ABHR*13]. This can be employed to show aggregated information on the elements and their attributes, as with aggregation-based techniques.

Analytical methods can transform large set-typed data into volumes suited for visualization and still preserving the most important information. In particular, several aggregations of the elements are possible based on their set-memberships, degrees and attribute values. Similarly, intuitive set-operations can be used, e.g. to aggregate multiple sets, or to replace a large family of sets with a smaller family over the same elements.

Identifying special cases and forms of set-typed data: For example, when the sets exhibit no intersection relations, treemaps would be a natural choice to visualize their containment hierarchy. Another example that arise in voting analysis, is when each element belongs to a constant number of sets, e.g. exactly to three sets out of 10. Such set memberships can be represented using three categorical variables which result in $\binom{10}{3} = 120$ non-redundant overlap combinations (many of them potentially empty). This is significantly lower than $2^{10} = 1024$ possible overlaps in the general case, and can be handled by categorical visualization techniques such as Parallel Sets.

Many other special cases can be identified in practical applications such as very sparse membership matrix, skewed or two-mode distribution of membership degrees, etc. The characteristics of these cases need to be studied extensively, e.g. to identify if they satisfy certain Euler diagram drawability properties, can simplify existing visualization techniques, allow for new forms of visual representations or overlays, or lend themselves to new ways of aggregation.

7. Conclusion

The powerful and generic concepts of set theory make sets and set relations essential data models in many data analysis scenarios. Unlike common data types in InfoVis such as graphs and trees, sets have been largely treated as data containers to group related elements or to illustrate overlaps between two or three groups. Nevertheless, a number of techniques have been devised to visualize sets and data related to them in the past decade. By emphasizing the notion of set-typed data, we have identified their specific characteristics as well as several measures and tasks commonly associated with this data type in visualization.

We have surveyed relevant literature on visualization techniques that can be applied to address these characteristics and tasks related to set-typed data, and have classified these techniques into six categories, according to the main visual representation they use for depicting set relations. For each technique, we have analysed which tasks it supports and its scalability with respect to the number of sets and elements. We have also outlined the general advantages and disadvantages of each representation, and which information they can represent from the data. This provides guidance for designers of set visualizations in choosing appropriate techniques for their data and tasks. Finally, we have examined major open problems in the area, and discussed various ideas that are worth investigating as opportunities to address open problems or to improve on state-of-the-art techniques. A visual browser of the surveyed techniques along with additional resources are available at `http://www.setviz.net`.

Acknowledgements

We thank the authors of the surveyed techniques for providing supplementary materials about this work. This work was partially supported by the Austrian Federal Ministry of Science, Research, and Economy via CVAST, a Laura Bassi Centre of Excellence (No. 822746) and the Austrian Science Fund (FWF) via the KAVA-Time project (No. P25489).

References

[AAMH13] Alsallakh B., Aigner W., Miksch S., Hauser H.: Radial Sets: Interactive visual analysis of large overlapping sets. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2496–2505.

[ABHR*13] Alper B., Bach B., Henry Riche N., Isenberg T., Fekete J.-D.: Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France, 2013), ACM, pp. 483–492.

[AHRRC11] Alper B., Henry Riche N., Ramos G., Czerwinski M.: Design study of LineSets, a novel set visualization technique.

IEEE Transactions on Visualization and Computer Graphics 17, 12 (2011), 2259–2267.

[AMA*14] Alsallakh B., Micallef L., Aigner W., Hauser H., Miksch S., Rodgers P.: Visualizing sets and set-typed data: State-of-the-art and future challenges. In *Proceedings of Eurographics Conference on Visualization (EuroVis) State of the Art Reports* (Swansea, UK, 2014), Eurographics.

[AMR14] Alsallakh B., Miksch S., Rauber A.: Towards a visualization of multi-faceted search results. In *Proceedings of the DL 2014 Workshop on Knowledge Maps and Information Retrieval (KMIR)* (London, UK, 2014), CEUR Workshop Proceedings.

[ASHC14] Alqadah M., Stapleton G., Howse J., Chapman P.: Evaluating the impact of clutter in Euler diagrams. In *Proceedings of vvConference on the Diagrammatic Representation and Inference (Diagrams), LNCS 8578* (Melbourne, Australia, 2014), pp. 108–122.

[Bar69] Baron M. E.: A note on the historical development of logic diagrams: Leibniz, Euler and Venn. *Mathematical Gazette 53*, 384 (1969), 113–125.

[BCH*13] Basole R. C., Clear T., Hu M., Mehrotra H., Stasko J.: Understanding interfirm relationships in business ecosystems with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), IEEE, 2526–2535.

[BCPS12] Brandes U., Cornelsen S., Pampel B., Sallaberry A.: Path-based supports for hypergraphs. *Journal of Discrete Algorithms 14* (2012), Elsevier, 248–261.

[BH11] Borgatti S. P., Halgin D. S.: Analyzing affiliation networks. In *The SAGE Handbook of Social Network Analysis* (Thousand Oaks, USA, 2011), SAGE, pp. 417–433.

[BKP12] Bach B., Kammer D., Polowinski J.: Facettice: Integrating faceted navigation and concept lattices for visual data exploration. In *Compendium of IEEE Information Visualization (InfoVis)* (Seattle, USA, 2012).

[BR07] Benoy F., Rodgers P.: Evaluating the comprehension of Euler diagrams. In *Proceedings of International Conference on Information Visualization (IV)* (London, UK, 2007), pp. 771–780.

[Bra12] Brath R.: Multi-attribute glyphs on Venn and Euler diagrams to represent data and aid visual decoding. In *Proceedings of International Workshop on Euler Diagrams* (Canterbury, UK, 2012), pp. 122–129.

[BSH13] Bothorel G., Serrurier M., Hurter C.: Visualization of frequent itemsets with nested circular layout and bundling algorithm. In *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, B. Li, F. Porikli, V. Zordan, J. Klosowski, S. Coquillart, X. Luo, M. Chen and D. Gotz (Eds.), vol. *8034* of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2013), pp. 396–405. http://dx.doi.org/10.1007/978-3-642-41939-3_38.

[BSR*12] Blake A., Stapleton G., Rodgers P., Cheek L., Howse J.: Does the orientation of an Euler diagram affect user comprehension? In *International Workshop on Visual Languages and Computing (VLC)* (Miami Beach, FL, USA, 2012), vol. 18, pp. 185–190.

[BSR*14] Blake A., Stapleton G., Rodgers P., Cheek L., Howse J.: The impact of shape on the perception of Euler diagrams. In *Proceedings of International Conference on the Theory and Application of Diagrams* (Melbourne, Australia, 2014), ACM.

[BSRH14] Blake A., Stapleton G., Rodgers P., Howse J.: How should we use colour in Euler diagrams? In *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction* (2014), ACM, p. 149.

[BT06] Byelas H., Telea A.: Visualization of areas of interest in software architecture diagrams. In *Proceedings of ACM Symposium on Software Visualization (SOFTVIS)* (Brighton, UK, 2006), ACM, pp. 105–114.

[BVKM*10] Buchin K., Van Kreveld M., Meijer H., Speckmann B., Verbeek K.: On planar supports for hypergraphs. In *Proceedings of Internatioanl Symposium on Graph Drawing, LNCS Vol. 5849* (Konstanz, Germany, 2010), Springer, pp. 345–356.

[BVT*13] Bailey N. W., Veltsos P., Tan Y.-F., Millar A. H., Ritchie M. G., Simmons L. W.: Tissue-specific transcriptomics in the field cricket teleogryllus oceanicus. *G3: Genes, Genomes, Genetics 3*, 2 (2013), 225–230.

[Can95] Cantor G.: Beiträge zur Begründung der transfiniten Mengenlehre. *Mathematische Annalen 46*, 4 (1895), Springer, 481–512.

[CDE06] Cole R., Ducrou J., Eklund P.: Automated layout of small lattices using layer diagrams. In *Formal Concept Analysis*, vol. *3874* of *Lecture Notes in Computer Science*, R. Missaoui and J. Schmidt (Eds.). Springer, Berlin, Germany, (2006), pp. 291–305. http://dx.doi.org/10.1007/11671404_20.

[CGF*09] Caldas J., Gehlenborg N., Faisal A., Brazma A., Kaski S.: Probabilistic retrieval and visualization of biologically relevant microarray experiments. *BMC Bioinformatics 10*, Suppl 13, BioMed Central Ltd., (2009), P1.

[Che11] Cheng P. C.-H.: Probably good diagrams for learning: Representational epistemic recodification of probability theory. *Topics in Cognitive Science 3*, 3 (2011), Wiley Online Library, 475–498.

[Cho07] Chow S. C.: Generating and Drawing Area-Proportional Venn and Euler Diagrams. PhD thesis, University of Victoria, Victoria, BC, Canada, 2007.

[Cla08] Clark J.: Twitter Venn. http://www.neoformix.com/2008/TwitterVenn.html, 2008. [Online (Jul. 2015)]

[CM84] Cleveland W. S., McGill R.: Graphical perception: Theory, experimentation, and application to the development of

graphical methods. *Journal of the American Statistical Association 79*, 387 (1984), 531–554.

[CPC09] Collins C., Penn G., Carpendale S.: Bubble Sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1009–1016.

[CR03] Chow S., Ruskey F.: Drawing area-proportional Venn and Euler diagrams. In *Proceedings of International Symposium on Graph Drawing* Berlin, Germany, (Perugia, Italy, 2003), Springer, pp. 466–477.

[CR05a] Chow S., Rodgers P.: Constructing area-proportional Venn and Euler diagrams with three circles. In *Proceedings of International Workshop on Euler Diagrams* (Paris, France, 2005).

[CR05b] Chow S., Ruskey F.: Towards a general solution to drawing area-proportional Euler diagrams. *Electronic Notes in Theoretical Computer Science 134* (2005), 3–18.

[CSR*14] Chapman P., Stapleton G., Rodgers P., Micallef L., Blake A.: Visualizing sets: An empirical comparison of diagram types. In *Diagrammatic Representation and Inference (Diagrams), LNCS* (Melbourne, Australia, 2014), pp. 146–160. http://dx.doi.org/10.1007/978-3-662-44043-8_18.

[CVW09] Collins C., Viegas F. B., Wattenberg M.: Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST)* (Salt Lake City, USA, 2009), IEEE, pp. 91–98.

[DEKB*14] Dinkla K., El-Kebir M., Bucur C.-I., Siderius M., Smit M. J., Westenberg M. A., Klau G. W.: examine: Exploring annotated modules in networks. *BMC Bioinformatics 15*, 1, BioMed Central Ltd, (London, UK, 2014), 201.

[DHRRD12] Dörk M., Henry Riche N., Ramos G., Dumais S.: PivotPaths: Strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), IEEE, 2709–2718.

[Dic45] Dice L. R.: Measures of the amount of ecologic association between species. *Ecology 26*, 3 (1945), JSTOR, 297–302.

[DKCN14] Delaney A., Kow E., Chapman P., Nicholson J.: Generating and navigating large Euler diagrams. In *Proceedings of the 4th International Workshop on Euler Diagrams, CEUR-WS. org Vol. 1244* (Melbourne, Australia, 2014), pp. 23–31.

[DKM06] Dwyer T., Koren Y., Marriott K.: IPSep-CoLa: An incremental procedure for separation constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 821–828.

[DvKSW12] Dinkla K., van Kreveld M., Speckmann B., Westenberg M.: Kelp diagrams: Point set membership visualization. *Computer Graphics Forum 31*, 3 (2012), 875–884.

[EDB04] Eklund P., Ducrou J., Brawn P.: Concept lattices for information visualization: Can novices read line-diagrams? In *Concept Lattices*, vol. 2961 of *LNCS*, P. Eklund (Eds.). Springer Berlin Heidelberg (2004), pp. 57–73. http://dx.doi.org/10.1007/978-3-540-24651-0_7.

[Epp06] Eppstein D.: Concept lattice, 2006. [Online (Jul., 2015)]. http://en.wikipedia.org/wiki/Formal_concept_analysis#mediaviewer/File:Concept_lattice.svg.

[FFH08] Flower J., Fish A., Howse J.: Euler diagram generation. *Journal of Visual Languages and Computing 19*, 6 (2008), 675–694.

[FH02] Flower J., Howse J.: Generating Euler diagrams. In *Diagrammatic Representation and Inference (Diagrams), LNCS*, vol. 2317. Springer, Berlin, Germany, (Callaway Gardens, GA, USA, 2002), pp. 285–285.

[FMH08] Freiler W., Matkovic K., Hauser H.: Interactive visual analysis of set-typed data. *IEEE Transactions on Visualization and Computer Graphics 14* 6 (2008), 1340–1347.

[FRM03] Flower J., Rodgers P., Mutton P.: Layout metrics for Euler diagrams. In *Proceedings of International Conference Information Visualisation (IV)* (London, UK, 2003), pp. 272–280.

[FSR14] Flower J., Stapleton G., Rodgers P.: On the drawability of 3D Venn and Euler diagrams. *Journal of Visual Languages and Computing, Special Issue on Visualization and Reasoning using Euler Diagrams 25*, 1 (2014), 186–209.

[G*14] Gottfried B., et al.: Set space diagrams. *Journal of Visual Languages & Computing*. (2014), Elsevier, 518–532.

[GHK10] Gansner E. R., Hu Y., Kobourov S. G.: Visualizing graphs and clusters as maps. *IEEE Computer Graphics and Applications 30*, 6 (2010), 54–66.

[GLSS11] Geymayer T., Lex A., Streit M., Schmalstieg D.: Visualizing the effects of logically combined filters. In *Proceedings of International Conference on Information Visualisation (IV)* (London, UK, 2011), IEEE, pp. 47–52.

[Gre84] Greenacre M. J.: *Theory and Applications of Correspondence Analysis*. (Waltham, MA, USA, 1984), Academic Press.

[GSG*14] Gove R., Saxe J., Gold S., Long A., Bergamo G.: SEEM: A scalable visualization for comparing multiple large sets of attributes for malware analysis. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security (VizSec)* (Paris, France, 2014), ACM, pp. 72–79.

[Gur99] Gurr C. A.: Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing 10*, 4 (1999), 317–342.

[GWW99] Ganter B., Wille R., Wille R.: *Formal Concept Analysis*, vol. *284*. Springer Berlin Germany, 1999.

[GYK10] Gansner R., Yifan H., Kobourov S.: GMap: Visualizing graphs and clusters as maps. In *Proceedings of IEEE Pacific Visualization Symposium* (Taipei, Taiwan, 2010), pp. 201–208.

[HB05] Heer J., Boyd D.: Vizster: Visualizing online social networks. In *Proceedings of IEEE Symposium on Information Visualization (INFOVIS)* (Minneapolis, USA, 2005), IEEE, pp. 32–39.

[HHH*89] Hamers L., Hemeryck Y., Herweyers G., Janssen M., Keters H., Rousseau R., Vanhoutte A.: Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula. *Information Processing & Management 25*, 3, (1989), Elsevier, 315–318.

[HKvK*13] Hurtado F., Korman M., van Kreveld M., Löffler M., Sacristán V., Silveira R. I., Speckmann B.: Colored spanning graphs for set visualization. In *Proceedings of International Symposium on Graph Drawing* (Bordeaux, France, 2013), Springer, pp. 280–291.

[Hof00] Hofmann H.: Exploring categorical data: Interactive mosaic plots. *Metrika 51*, 1 (2000), Springer, 11–26.

[HRD10] Henry Riche N., Dwyer T.: Untangling Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1090–1099.

[HST05] Howse J., Stapleton G., Taylor J.: Spider diagrams. *London Mathematical Society Journal of Computation and Mathematics 8* (2005), 145–194.

[HSTC11] Howse J., Stapleton G., Taylor K., Chapman P.: Visualizing ontologies: A case study. In *Proceedings of 10th International Semantic Web Conference (ISWC)* (Bonn, Germany, 2011), pp. 257–272.

[HSW00] Hofmann H., Siebes A. P., Wilhelm A. F.: Visualizing association rules with interactive mosaic plots. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (Boston, MA, USA, 2000), ACM, pp. 227–235.

[Huo08] Huo J.: KMVQL: A visual query interface based on Karnaugh map. In *Proceedings of International Working Conference on Advanced Visual Interfaces (AVI)* (Napoli, Italy, 2008), ACM, pp. 243–250.

[IMMS09] Itoh T., Muelder C., Ma K.-L., Sese J.: A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *Proceedings of IEEE Pacific Visualization Symposium (PacificVis)* (Beijing, China, 2009), pp. 121–128.

[KBH06] Kosara R., Bendix F., Hauser H.: Parallel Sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 558–568.

[Kin] King G.: Find a proper pub. [Online (Jul. 2015)]. http://www.findaproperpub.co.uk/.

[KLS07] Kim B., Lee B., Seo J.: Visualizing set concordance with permutation matrices and fan diagrams. *Interacting with Computers 19*, 5–6 (2007), Oxford University Press, 630–643.

[KMK*08] Kestler H. A., Müller A., Kraus J. M., Buchholz M., Gress T. M., Liu H., Kane D. W., Zeeberg B. R., Weinstein J. N.: VennMaster: Area-proportional Euler diagrams for functional GO analysis of microarrays. *BMC Bioinformatics 9* (2008), 67.

[Kof35] Koffka K.: *Principles of Gestalt Psychology*. Harcourt Brace, New York, NY, USA, 1935.

[Kos07] Kosara R.: Autism diagnosis accuracy—Visualization redesign. [Online (Jul. 2015)]. http://eagereyes.org/criticism/autism-diagnosis-accuracy, 2007.

[KSB*09] Krzywinski M., Schein J., Birol İ., Connors J., Gascoyne R., Horsman D., Jones S. J., Marra M. A.: Circos: An information aesthetic for comparative genomics. *Genome Research 19*, 9 (2009), Cold Spring Harbor Lab, 1639–1645.

[KSJ*06] Koshman S., Spink A., Jansen B. J., Blakely C., Weber J.: Metasearch result visualization: An exploratory study. In *Proceedings of Canadian Association for Information Science Conference* (Canada, 2006).

[KvKS09] Kaufmann M., van Kreveld M., Speckmann B.: Subdivision drawings of hypergraphs. In *Proceedings of Internatioanl Symposium on Graph Drawing, LNCS vol. 5417* (Chicago, USA, 2009), Springer, pp. 396–407.

[LF06] Lenz O., Fornoni A.: Chronic kidney disease care delivered by us family medicine and internal medicine trainees: Results from an online survey. *BMC Medicine 4* (2006), 30–407.

[LG14] Lex A., Gehlenborg N.: Points of view: Sets and intersections. *Nature Methods 11*, 8 (2014), Nature Publishing Group, 779–779.

[LGS*14] Lex A., Gehlenborg N., Strobelt H., Vuillemot R., Pfister H.: Upset: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1983–1992.

[LLS05] Liu X., Luo M., Shneiderman B.: Visualization of sets. Unpublished manuscript (2005), Citeseer.

[LM13] Littlefield K., Monroe M.: Venn Diagram Plotter, Biological MS Data and Software Distribution Center. http://omics.pnl.gov/software/VennDiagramPlotter.php, 2013. [Online (Jul. 2015)].

[LRS10] Luboschik M., Radloff A., Schumann H.: A new weaving technique for handling overlapping regions. In *Proceedings of International Working Conference on Advanced Visual Interfaces (AVI)* (Rome, Italy, 2010), ACM, pp. 25–32.

[Mäk90] Mäkinen E.: How to draw a hypergraph. *International Journal of Computer Mathematics 34*, 3–4 (1990), Taylor & Francis, 177–185.

[MDF12] Micallef L., Dragicevic P., Fekete J.-D.: Assessing the effect of visualizations on Bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2536–2545.

[MHRS*13] Meulemans W., Henry Riche N., Speckmann B., Alper B., Dwyer T.: KelpFusion: A hybrid set visualization technique. *IEEE Transactions on Visualization and Computer Graphics 19*, 11 (2013), 1846–1858. doi: 10.1109/TVCG.2013.76.

[Mis06] Misue K.: Drawing bipartite graphs as anchored maps. In *Proceedings of Asia-Pacific Symposium on Information Visualisation (APVIS)* (Sydney, Australia, 2006), Australian Computer Society, Inc., pp. 169–177.

[MR14a] Micallef L., Rodgers P.: Computing the region areas of Euler diagrams drawn with three ellipses. In *Proceedings of the 4th International Workshop on Euler Diagrams, CEUR-WS. org vol. 1244* (Melbourne, Australia, 2014), 1–15.

[MR14b] Micallef L., Rodgers P.: eulerAPE: Drawing area-proportional 3-Venn diagrams using ellipses. *PLoS ONE 9*, 7 (2014), e101717. http://www.eulerdiagrams.org/eulerAPE.

[MR14c] Micallef L., Rodgers P.: eulerForce: Force-directed layout for Euler diagrams. *Journal of Visual Languages and Computing 25*, 6 (2014), 924–934.

[NSL*12] Nikulenkov F., Spinnler C., Li H., Tonelli C., Shi Y., Turunen M., Kivioja T., Ignatiev I., Kel A., Taipale J., Selivanova G.: Insights into p53 transcriptional function via genome-wide chromatin occupancy and gene expression analysis. *Cell Death and Differentiation 19* (2012), 1992–2002.

[NXWW14] Nguyen P. H., Xu K., Walker R., Wong B.: Schemaline: Timeline visualization for sensemaking. In *Proceedings of 18th International Conference on Information Visualisation (IV)* (Paris, France, July 2014), pp. 225–233, doi: 10.1109/IV.2014.14.

[OHS*09] Oliver I., Howse J., Stapleton G., Nuutila E., Törmä S.: Visualizing and specifying ontologies using diagrammatic logics. In *Proceedings of the 5th Australasian Ontology Workshop* (Melbourne, Australia, 2009), 3–12.

[OSR*14] Oelke D., Strobelt H., Rohrdantz C., Gurevych I., Deussen O.: Comparative exploration of document collections: A visual analytics approach. *Computer Graphics Forum 33*, 3 (2014), Wiley Online Library, 201–210.

[Pal92] Palmer S. E.: Common region: A new principle of perceptual grouping. *Cognitive Psychology 24*, 3 (1992), 436–447.

[PB15] Park H., Basole R. C.: Bicentric diagrams: Design of a graph-based relational set visualization technique. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)—Extended abstracts* (Seoul, Korea, 2015), ACM, pp. 1815–1820.

[PBT14] Parra D., Brusilovsky P., Trattner C.: See what you want to see: Visual user-driven approach for hybrid recommendation. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (Island of Madeira, Portugal, 2014), 235–240.

[PP10] Park Y., Park J.: Disk diagram: An interactive visualization technique of fuzzy set operations for the analysis of fuzzy data. *Information Visualization 9*, 3 (2010), SAGE Publications, 220–232.

[RFSH10] Rodgers P., Flower J., Stapleton G., Howse J.: Drawing area-proportional Venn-3 diagrams with convex polygons. In *Diagrammatic Representation and Inference (Diagrams), LNCS*, vol. 6170. Springer, Berlin, Germany, (Portland, OR, USA, 2010), pp. 54–68.

[RHSF14] Rodgers P., Howse J., Stapleton G., Flower J.: Drawing area-proportional Euler diagrams representing up to three sets. *IEEE Transactions on Visualization and Computer Graphics 20*, 1 (2014), 56–69.

[Rod14] Rodgers P.: A survey of Euler diagrams. *Journal of Visual Languages and Computing - Special Issue on Visualization and Reasoning using Euler Diagrams 25*, (2014), Elsevier, 134–155.

[RW97] Ruskey F., Weston M.: A survey of Venn diagrams. *Electronic Journal of Combinatoric 4* (1997), Dynamic Survey DS5 (revised in 2001 and 2005).

[RZF08] Rodgers P., Zhang L., Fish A.: General Euler diagram generation. In *Diagrammatic Representation and Inference (Diagrams), LNCS*, vol. 5223. Springer, Berlin, Germany, (Herrsching, Germany, 2008), pp. 13–27.

[RZP12] Rodgers P., Zhang L., Purchase H.: Wellformedness properties in Euler diagrams: Which should be used? *IEEE Transactions on Visualization and Computer Graphics 18*, 7 (2012), 1089–1100.

[SA08] Simonetto P., Auber D.: Visualise undrawable Euler diagrams. In *Proceedings of International Conference Information Visualisation (IV)* (London, UK, 2008), IEEE, pp. 594–599.

[SAA09] Simonetto P., Auber D., Archambault D.: Fully automatic visualisation of overlapping sets. *Computer Graphics Forum 28*, 3 (2009), 967–974.

[Sch11] Schulz H.-J.: Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications 31*, 6 (2011), 11–15.

[SD08] Stapleton G., Delaney A.: Evaluating and generalizing constraint diagrams. *Journal of Visual Languages and Computing 19*, 4 (2008), 499–521.

[SDRP11] Stapleton G., Delaney A., Rodgers P., Plimmer B.: Recognising sketches of Euler diagrams augmented with graphs. In *Proceedings of International Workshop on Visual Languages and Computing (VLC)* (Florence, Italy, 2011), Florence, Italy, vol. 17, pp. 182–196.

[SFRH12] Stapleton G., Flower J., Rodgers P., Howse J.: Automatically drawing Euler diagrams with circles. *Journal of Visual Languages and Computing 23*, 3 (2012), 163–193.

[SGL08] Stasko J., Görg C., Liu Z.: Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (2008), SAGE Publications, 118–132.

[SH14] Sathiyanarayanan M., Howse J.: Well-matchedness in Euler diagrams. In *Proceedings of the 4th International Workshop on Euler Diagrams,* CEUR-WS.org *vol. 1244* (Melbourne, Australia, 2014), 16–22.

[Shn96] Shneiderman B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages* (1996), IEEE, pp. 336–343.

[SJUS08] Schulz H.-J., John M., Unger A., Schumann H.: Visual analysis of bipartite biological networks. In *Proceedings of EG Workshop on Visual Computing for Biomedicine* (Delft, Netherlands, 2008).

[SMDS14] Sadana R., Major T., Dove A., Stasko J.: Onset: A visualization technique for large-scale binary set data. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), IEEE, 1993–2002.

[SOTM06] Shen Z., Ogawa M., Teoh S. T., Ma K.-L.: BiblioViz: A system for visualizing bibliography information. In *Proceedings of International Asia-Pacific Symposium on Visualization (APVIS)* (Tokyo, Japan, 2006), pp. 93–102.

[Spo93] Spoerri A.: InfoCrystal: A visual tool for information retrieval. In *Proceedings of IEEE Visualization* (San Jose, CA, USA, 1993), pp. 150–157.

[Spo04] Spoerri A.: MetaCrystal: Visual interface for meta searching. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)—Extended Abstracts* (2004), vol. 24. no. 29, ACM, pp. 1558–1558.

[SRHZ11] Stapleton G., Rodgers P., Howse J., Zhang L.: Inductively generating Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics 17*, 1 (2011), 88–100.

[Sta05] Stapleton G.: A survey of reasoning systems based on Euler diagrams. *Electronic Notes in Theoretical Computer Science 134* (2005), 127–151.

[SWS*11] Steinberger M., Waldner M., Streit M., Lex A., Schmalstieg D.: Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2249–2258.

[SZHR11] Stapleton G., Zhang L., Howse J., Rodgers P.: Drawing Euler diagrams with circles: The theory of piercings. *IEEE Transactions on Visualization and Computer Graphics 17*, 7 (2011), 1020–1032.

[Tan58] Tanimoto T.: *An Elementary Mathematical Theory of Classification and Prediction*. IBM Internal Report, 1958.

[TAON09] Tarnita C. E., Antal T., Ohtsuki H., Nowak M. A.: Evolutionary dynamics in set structured populations. *Proceedings of the National Academy of Sciences 106*, 21 (2009), 8601–8604. http://www.pnas.org/content/106/21/8601.abstract, http://arxiv.org/abs/http://www.pnas.org/content/106/21/8601.full.pdf arXiv:http://www.pnas.org/content/106/21/8601.full.pdf, http://dx.doi.org/10.1073/pnas.0903019106.

[TS85] Treisman A., Souther J.: Search asymmetry: A diagnostic for preattentive processing of separable features. *Journal of Experimental Psychology: General 114*, 3 (1985), 285–310.

[Tun09] Tunkelang D.: Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services 1*, 1, (2009), Morgan & Claypool Publishers, 1–80.

[Tve77] Tversky A.: Features of similarity. *Psychological Review 84*, 4, (1977), American Psychological Association, 327–352.

[UJ12] Urbas M., Jamnik M.: Diabelli: A heterogeneous proof system. In *Diagrammatic Representation and Inference (Diagrams), LNCS*, vol. 7364. Springer, Manchester, UK, (2012), pp. 559–566.

[UJ14] Urbas M., Jamnik M.: A framework for heterogeneous reasoning in formal and informal domains. In *Diagrammatic Representation and Inference (Diagrams), LNCS 8578* (Melbourne, Australia, 2014), pp. 277–292.

[UJSF12] Urbas M., Jamnik M., Stapleton G., Flower J.: Speedith: A diagrammatic reasoner for spider diagrams. In *Diagrammatic Representation and Inference (Diagrams), LNCS*, vol. 7352. Springer, Manchester, UK, (2012), pp. 163–177.

[Ven80] Venn J.: On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 10*, 59 (1880), 1–18.

[vLBA*12] von Landesberger T., Bremm S., Andrienko N., Andrienko G., Tekusova M.: Visual analytics methods for categoric spatio-temporal data. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)* (Seattle, USA, 2012), IEEE, pp. 183–192.

[VPF*14] Vihrovs J., Prūsis K., Freivalds K., Ručevskis P., Krebs V.: An inverse distance-based potential field function for overlapping point set visualization. In *Proceedings of International Conference on Information Visualization Theory and Applications (IVAPP)* (Lisbon, Portugal, 2014), SCITEPRESS, pp. 29–38.

[VRW13] Vehlow C., Reinhardt T., Weiskopf D.: Visualizing fuzzy overlapping communities in networks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), IEEE, 2486–2495.

[War12] Ware C.: *Information Visualization: Perception for Design* (3rd edition). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.

[Wer38] Wertheimer M.: Laws of organization in perceptual forms. In *A Sourcebook of Gestalt Psychology*. W. D. Ellis (Eds.). Routledge and Kegan Paul (London, UK, 1938), pp. 71–88.

[Wik10] Wikimedia Commons: Official languages in Africa, 2010. [Online (Jul. 2015)]. http://en.wikipedia.org/wiki/File:Official_languages_in_Africa.svg.

[Wil07] Wille R.: The basic theorem on labelled line diagrams of finite concept lattices. In *Formal Concept Analysis*, S. Kuznetsov and S. Schmidt (Eds.), vol. 4390 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg (2007), pp. 303–312. http://dx.doi.org/10.1007/978-3-540-70901-5_19.

[Wil12] Wilkinson L.: Exact and approximate area-proportional circular Venn and Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics 18*, 2 (2012), 321–331.

[Wit10] Wittenburg K.: Setting the bar for set-valued attributes. In *Proceedings of International Conference on Advanced Visual Interfaces (AVI)* (Rome, Italy, 2010), ACM, pp. 253–256.

[WMLP12] Wittenburg K., Malizia A., Lupo L., Pekhteryev G.: Visualizing set-valued attributes in parallel with equal-height histograms. In *Proceedings of International Working Conference on Advanced Visual Interfaces (AVI)* (Capri, Italy, 2012), ACM, pp. 632–635.

[WPS*11] Wang M., Plimmer B., Schmieder P., Stapleton G., Rodgers P., Delaney A.: SketchSet: Creating Euler diagrams using pen or mouse. In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (Pittsburgh, PA, USA, 2011), pp. 75–82.

[XDC*13] Xu P., Du F., Cao N., Shi C., Zhou H., Qu H.: Visual analysis of set relations in a graph. *Computer Graphics Forum 32*, 3 (2013), 61–70.

[ZKBS02] Ziegler E., Kunz C., Botsch V., Schneeberger J.: Visualizing and exploring large networked information spaces with Matrix Browser. In *Proceedings of International Conference Information Visualisation (IV)* (London, UK, 2002), IEEE, pp. 361–366.

[ZXQ15] Zhou H., Xu P., Qu H.: Visualization of bipartite relations between graphs and sets. *Journal of Visualization* Springer, *18*, 2 (2015), 1–14.