Interactive Design and Visualization of N-ary Relationships

Botong Qu
Oregon State University
Corvallis, OR 97331, USA

Prashant Kumar
Oregon State University
Corvallis, OR 97331, USA

Eugene Zhang
Oregon State University
Corvallis, OR 97331, USA

Pankaj Jaiswal
Oregon State University
Corvallis, OR 97331, USA

Laurel Cooper
Oregon State University
Corvallis, OR 97331, USA

Justin Elser
Oregon State University
Corvallis, OR 97331, USA

Yue Zhang
Oregon State University
Corvallis, OR 97331, USA

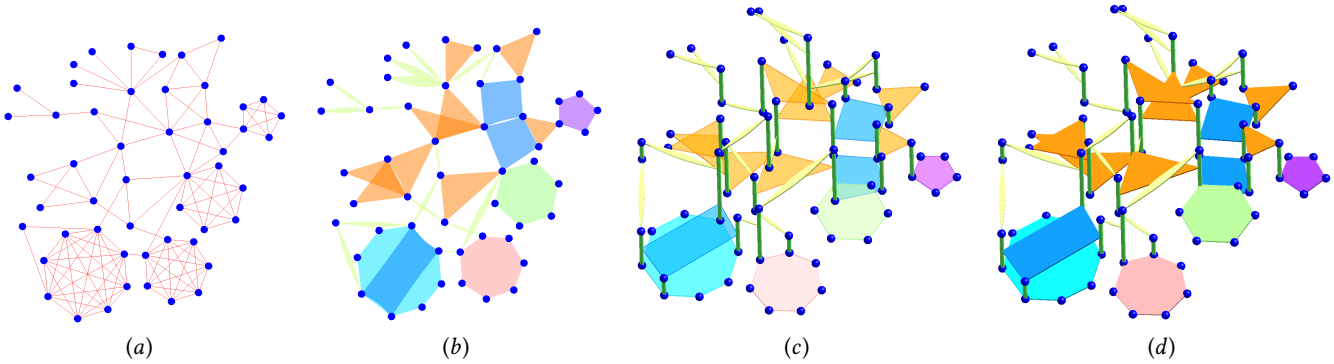


Figure 1: A social network data set is visualized using a graph layout algorithm (a). N-ary relationships in the data are represented as cliques, which can be difficult to recognize visually due to the increasing number of edges in the cliques. We address this challenge by using the notion of *CW complexes* (b), a mathematical concept that extends graphs. An N-ary relationship is visualized with an N-sided polygon, and a binary relationship with a digon. With translucent rendering, our visualization makes N-ary relationships more easily recognized, and their cardinality more pronounced. To deal with the large overlaps of the polygons, we develop a 3D layout method where overlapping polygons are placed at different layers (c) . In addition, we can render the polygons as opaque or translucent objects (d). Note that the colors for the polygons are based on the number of edges in the polygons (cardinality).

ABSTRACT

Graph and network visualization is a well-researched area. However, graphs are limited in that by definition they are designed to encode pairwise relationships between the nodes in the graph. In this paper, we strive for visualization of datasets that contain not only binary relationships between the nodes, but also higher-cardinality relationships (ternary, quaternary, quinary, senary, etc). While such higher-cardinality relationships can be treated as cliques (a complete graph of N nodes), visualization of cliques using graph visualization can lead to unnecessary visual cluttering due to all the pairwise edges inside each clique. In this paper, we develop a visualization for data that have relationships with cardinalities higher than two. By representing each N-ary relationship as an N-sided polygon, we turn the problem of visualizing such data sets into that of visualizing a two-dimensional complex, i.e. nodes, edges, and polygonal faces. This greatly reduces the number of edges needed to represent a clique and makes them as well as their cardinalities more easily recognized.

We develop a set of principles that measures the effectiveness of the visualization for two-dimensional complexes. Furthermore, we formulate our strategy with which the positions of the nodes in the complex and the orderings of the nodes inside each clique in the complex can be optimized. Furthermore, we allow the user to further improve the layout by moving a node or a polygon in 3D as well as changing the order of the nodes in a polygon. To demonstrate the effectiveness of our technique and system, we apply them to a social network and a gene dataset.

SA '17 Symposium on Visualization, November, 2017, Bangkok, Thailand

ACM ISBN 978-1-4503-5411-0/17/11.
https://doi.org/10.1145/3139295.3139314

CCS CONCEPTS

• **Human-centered computing → Graph drawings**; **Information visualization**;

KEYWORDS

Visualization, Graph visualization, N-ary relationships, Cliques

ACM Reference format:
Botong Qu, Prashant Kumar, Eugene Zhang, Pankaj Jaiswal, Laurel Cooper, Justin Elser, and Yue Zhang. 2017. Interactive Design and Visualization of N-ary Relationships. In *Proceedings of SA '17 Symposium on Visualization, Bangkok, Thailand, November, 2017,* 8 pages.
https://doi.org/10.1145/3139295.3139314

1 INTRODUCTION

Graph and network visualization has been well-researched by the visualization and graph layout communities. Graphs, by definition, include a set of nodes and a set of edges, where each edge encodes a binary relationship between a pair of nodes.

While graphs are a powerful representation of many phenomena from biology to social science, they are intrinsically limited in the sense that they are designed to capture binary relationships between the entities of interest, such as mutual friendships and opponents in an NBA game. However, in many applications an entity (node) may be involved in an N-ary relationship whose cardinality (N) is higher than two. For example, three chess players representing the same club are involved in a ternary relationship, which is different from the case where three players are involved in three binary relationships but not a ternary relationship, e.g. players A and B play chess, players A and C play golf, and players B and C play tennis.

N-ary relationships have been noted in many research domains that include social science, physical science and engineering applications. Students adjust to school lives by knowing their social circles and which clique to join. Engineering designs optimize length, width, and depth of an object. Physical and mathematical theories often rely on groups of parameters. Conventionally, to store these data structures, we utilize pairwise relationships and conduct searching and grouping on them during post processing.

Existing visualization techniques model N-ary relationships as *cliques* in a graph, i.e. each N-ary relationship is represented as a complete subgraph with $\frac{N(N-1)}{2}$ edges. While this approach works well for data with relatively few relationships and the maximum cardinality of all relationships is relatively low, it suffers from a number of challenges for more complex and more realistic data sets.

First, when the cardinalities of an N-ary relationship is high, this visualization involves a large number of edges, making it difficult to visually identify and verify whether a subgraph is a clique. For example, consider verifying whether a subgraph of 10 nodes form a clique. This would involve checking 45 edges between the nodes, which is a daunting task and can put a great mental load on the user. At the same time, it is difficult to mentally associate all the nodes in a clique as part of the same object. This is because these nodes are often obscured by all the edges incident to the nodes.

Second, such a visualization can often lead to confusion over whether N nodes are involved in an N-ary relationship or $\frac{N(N-1)}{2}$ binary relationships.

Third, when generating the optimal layout of the nodes, only information regarding the edges are considered (binary relationships). This is akin to the finite difference method in numerical simulation [14], where a quantity at a node is computed based on edges incident to the node, rather than the faces or cells. Such an approach naturally optimizes for binary relationships but disregard N-ary relationships.

All of the above highlight the fundamental flaw in a graph-based representation of data with N-ary relationships, that is, an N-ary relationship is not explicitly modeled, but instead, implicitly modeled as cliques.

In this paper, we address this problem by extending the notion of graphs to *CW complexes*. A zero-dimensional CW complex consists of a set of unconnected nodes, while a one-dimensional CW complex consists of a set of nodes and a set of edges each of which connects a pair of nodes. Note that a graph is a one-dimensional CW complex. A two-dimensional CW complex consists of nodes, edges, and polygons. The notion of an N-dimensional CW complex can be defined in a similar fashion.

By explicitly modeling each relationship in the visual display as a single object (a polygon in a two-dimensional CW-complex , it is easier to identify an N-ary relationship, understand its cardinality, and see the nodes involved in the relationship. Moreover, when generating optimal layouts of the data by moving the nodes, N-ary relationships are better modeled as a whole.

To the best of our knowledge, there has been relatively little research in achieving CW complex layout. In this paper, we explore the requirement and quality measures for CW complex layout. In addition, we provide a CW complex layout technique that allows a two-dimensional CW complex to be placed in a plane. To reduce the potential overlaps between polygons, we identify a set of layout editing operations including moving a node, moving a polygon, and changing the order of the nodes in a polygon. In addition, our system allows the input data to be filtered, which can result in better layout for the remaining nodes and polygons. Moreover, we provide an overlap-free approach by allowing the data to be in multiple planes. To avoid curved polygons due to different polygons involving the same node, we represent each node as a cylindrical tube, and polygons from different layers that are incident to the same node do not share the same physical point (Figure 1).

We have applied our technique to social network data and biological applications.

2 PREVIOUS WORK

There has been much work in graph layout algorithms, and we refer interested readers to the following surveys [11, 12, 15, 19].

N-ary relationships have found many applications [3]. Many algorithms exist for finding cliques in the data [26]. Open source tools such as Iridia (http://iridia.ulb.ac.be/ fmascia/graphvisualizer) demonstrate examples of cliques that have cardinality of 4 and 5. During the visualization of cliques or nodes with high-order cardinality, cluttering and edge cris-crossing can be seen. Lack of

clarity hinders domain scientists from seeing the N-ary relationships quickly or even from knowing the existence.

Klamt et al. [18] model N-ary relationships with the notion of *hypergraphs* and use line glyphs to represent such relationships. They have demonstrated their techniques on relatively small datasets with only a number of nodes and a handful of relationships. It is unclear how well their techniques scale to datasets with more realistic complexity. Moreover, it is not clear how well their technique can handle the case where the nodes in one relationship is a subset of the nodes of another relationship. Our CW complex visualization framework can handle data with larger complexity as well as the aforementioned relationships.

N-ary relationships have also been modeled as cliques in a graph. With this representation, nodes in the same clique are clustered during node placement. In addition, an envelop with color can be placed around the cluster to highlight the fact that these nodes are part of the same object (a clique) [8]. Force-based methods [17] have been used for this purpose. However, such an approach suffers from the overlapping cliques which share nodes, which we strive to address with the CW complex representation and user editing.

Dunne and Shneidermann [10] develop clique glyphs to highlight each clique in the graph as a data abstraction. While each clique is clearly distinguishable from other cliques, the nodes in the cliques are not explicitly shown. We seek to develop a framework in which both the nodes and the relationships are visible.

3 PROBLEM STATEMENT

In this section, we make rigorous the problem statement in our system. First, the input data to our system has the following format:

$$\mathbb{C} = \{E, R_2, R_3, ..., R_{|E|}\} \quad (1)$$

where E is the set of entities which may have their attributes including their importance, and $|E|$ is the number of entities in E.

R_k ($2 \leq k \leq |E|$) is the set of k-ary relationships among the entities in E. More specifically, a relationship $r \in R_k$ is a set $(e_1, e_2, ..., e_k)$ where $e_i \in E$ for $1 \leq i \leq k$ and $e_i \neq e_j$ for any $1 \leq i < j \leq k$. Note that the order of the entities in this relationship is insignificant, i.e., we do not consider permutations among the entities. As an example, a binary relationship $r = (e_1, e_2)$ is the same as $r = (e_2, e_1)$. Note that when $R_3 = R_4 = ... = R_{|V|} = \emptyset$, our data reduces to an undirected graph where each element in R_2 is an edge in the graph.

Recall that we model this type of data as a two-dimensional CW complex. Every entity in E is mapped to a node in the complex, while every relationship $r \in R_k$ is mapped to a k-sided polygon (two-dimensional complex). Our problem of visualizing the entity-relationship data is thus converted to that of visualizing a two-dimensional CW complex. We will use entities and nodes in an interchangeable fashion. This is true also for polygons and relationships.

We define the *degree* of a node to be the number of relationships that involve the node. In addition, two nodes are said to be *adjacent* if they are involved in at least one relationship.

In a graph, two edges may share one or zero common node. Moreover, in a *multi-graph*, there can be multiple edges between the same pair of nodes. In a CW complex, there are more cases due to the existence of N-ary relationships where $N > 2$:

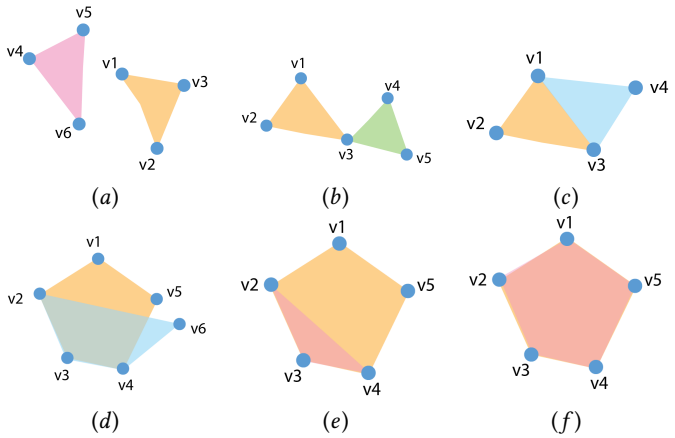


Figure 2: Example adjacent polygons scenarios: (a) unrelated, (b) nodely adjacent, (c) edgely adjacent, (d) partially overlapping, (e) enclosing, and (f) co-locating.

(1) Two polygons have no common nodes. In this case we refer to the polygons (relationships) as being *unrelated*.
(2) Two polygons share one common node. The two polygons are said to be *nodely adjacent*.
(3) Two polygons share two common nodes while one of the polygons has at least one extra node. The two polygons are said to be *edgely adjacent*.
(4) Two polygons share at least three common nodes but each has at least a node that is not part of the other polygon. In this case the polygons are said to be *partially overlapping*.
(5) The set of nodes of one polygon is a subset of the nodes of another polygon. In this case, the larger polygon *encloses* the smaller one.
(6) Note that as a special case of the aforementioned case, the two polygons involve exactly the same set of nodes. The two polygons are said to be *co-locating*.

See Figure 2 for examples. As with nodes, each relationship may have an importance property and other properties. The *cardinality* of a polygon refers to the number of nodes in the polygon, while the *degree* of the polygon refers to the number of relationships that have at least one common node with the relationship.

There are a number of tasks that are essential in applications of this type of data.

(1) Each node should be visually distinguishable from other nodes and can be highlighted so that its properties can be inspected.
(2) A node's importance should be clearly visible.
(3) Each polygon should be visually distinguishable from other polygons even when the two polygons share some nodes. A polygon can be highlighted for the user to inspect its properties.
(4) A polygon's cardinality should be easily determined visually. Moreover, the nodes belonging to a polygon should be easily identified.
(5) A polygon's importance should be clearly visible.

This naturally leads to following research questions:

(1) How to measure the quality of the visualization, i.e., what layout principles apply here?
(2) Based on these principles, how to generate an optimal layout given a dataset, i.e., node placement and ordering of the nodes inside a polygon?
(3) How to implement the strategy to produce interactive visualization?
(4) How to improve the design with user interaction, and what fundamental operations are needed to provide the user with sufficient flexibility?

We address these questions in the following sections.

4 CW COMPLEX LAYOUT PRINCIPLES

In this section we describe the CW complex layout principles that we have identified during this research. While some principles are common with graph layout, others are specific to CW complex layout.

(1) Nodes should not overlap or cluster
(2) A node should not be placed inside or on the boundary of any polygon that is not incident to the node.
(3) Every polygon should be as close to being regular as possible.
(4) The number of overlaps between polygons should be minimized.

If achieving Principle 3 is impractical, i.e., it is not possible to make all the polygons regular, at least the polygon should be convex. This principle not only increases the aesthetics of the polygon, it also helps the user distinguish between two overlapping polygons and see the cardinality of the polygon.

Principle 4 states that unrelated polygons should not overlap, while nodely and edgely adjacent relationship should only overlap at a node or an edge, respectively. Note that it is not possible to reduce the number of partially overlapping, enclosing, and co-locating relationships since there are algebraic intersections of the nodes of the polygons. By algebraic intersections, we mean two sets of nodes have common elements, each set represents one relationship. However, in case of partial overlapping relationships, the number of crossings between the edges of the two polygons should be minimized.

Based on these principles, we have come up with two approaches to CW complex layout, described in Sections 5 and 6.

5 PLANAR CW COMPLEX LAYOUT

In our approach, we seek to layout a CW complex on the plane that follows the aforementioned layout principles. This is achieved through a two-step pipeline. First, we automatically generate an initial layout by treating the data as a graph and reusing a force-based graph layout algorithm [17] to compute the locations of the nodes. Then a clique's outline is used to generate an N-sided polygon. Next, the user can further improve the quality of the CW complex layout with a set of editing operations that we have identified as being fundamental to CW complex editing. Given a layout, the complex is visualized as follows. Every node is visualized as a sphere centered at its computed location. Its color and reflectance material is specified by the user. Lights are given in the scene. Rendering point clouds as points can make it difficult to see the depth of

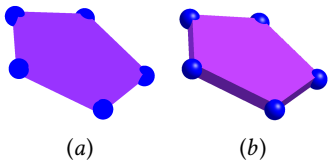


Figure 3: This figure shows the importance of rendering the nodes and polygons with lighting.

the points [4]. By rendering the nodes as spheres, highlights and shadows on the sphere can improve the perception of the nodes. Similarly, each polygon is rendered with a color and reflectance properties which, when interacting with light, can improve depth perception. Figure 3 compares the rendering without and with lights, or as points and as spheres. In addition, we implement depth peeling [2], a technique in which overlapping surfaces are rendered as translucent materials.

It seems rather counter-intuitive to compute the locations of the nodes by using the input data as the graph format. After all, we wish to use regular polygons instead of a "spaghetti" of edges inside the cliques in the graph. However, we have found through experiments that by using all the edges in a clique, the force-based layout method tends to have the outline of the clique in a nearly regular polygonal shape. As one might expect, the initial layout can

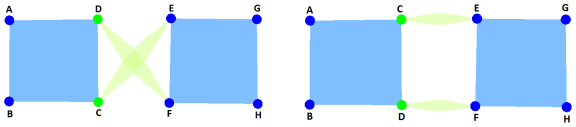


Figure 4: This figure demonstrates the importance of the polygon reconnection operation. In (a), a polygon having unnecessary overlaps in the initial layout cannot be easily fixed (b).

be rather suboptimal, with many unnecessary overlaps of polygons that are not adjacent. This is because the force-based method does not take into account the shape of the associated polygons when optimizing the location of the nodes. We address this by providing users with the capability of editing the layout of the CW complex as follows.

First, the user can choose to move a node. However, all the polygons incident to the node will be deformed due to this operation. Second, the user can also move, rotate, and resize a polygon. All of the adjacent polygons adjacent will be deformed.

While the above operations seem intuitive, they cannot change the order of the nodes in a polygon. Figure 4 (left) shows an example in which a sub-optimal order inside a triangle can make it difficult to generate a satisfactory layout. This inspires the following editing operation.

Third, the user can change the order of nodes in a polygon. By allowing the nodes in the polygon to be re-ordered, more degree of freedom is available (Figure 4 (right)).

In our system, polygon reconnection is achieved through pair swapping operations. More specifically, given a k-sided polygon

$\{e_1, e_2, ..., e_i, ..., e_j, ..., e_k\}$ where e_m has the location (x_m, y_m) for $1 \leq m \leq k$, we can swap e_i and e_j to generate a new polygon: $e_1, e_2, ..., e_j, ..., e_i, ..., e_k$. Moreover, the node e_i will take the coordinates (x_j, y_j) while the node e_j take the coordinates (x_i, y_i). Figure 4 shows the effect of performing a swap on a polygon. Note that all polygons incident to e_i and e_j are swapped.

While the aforementioned operations can help improve the layout by reducing visual overlaps between unrelated or adjacent polygons, they do not address polygons that are partially overlapping, enclosing, or co-locating since in these scenarios the visual overlaps are caused by the overlaps in the data, not the suboptimal layout. Therefore, we develop the fourth operation as follows.

Fourth, when overlapping polygons exist, the user can change the layout by sending the top polygon to the bottom, thus making the next-to-the-top polygon the new top polygon. As Figure 5 shows, different rendering order changes visibility of the polygons.

In addition to the above editing operations, our system allows users to filter their data, either based on nodes or polygons. We have found that the optimal layout for the whole data set is usually different from that of a subset. Figure 6 compares the optimal layout of the whole data set with one for a filtered subset.

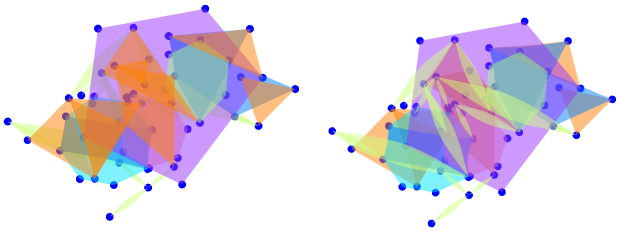


Figure 5: This figure compares the results of different rendering orders: (top) Polygons are rendered after edges, the triangles are rendered last so that they have the highest rendering priority. (bottom) The edges are rendered last so they are the most visible ones.

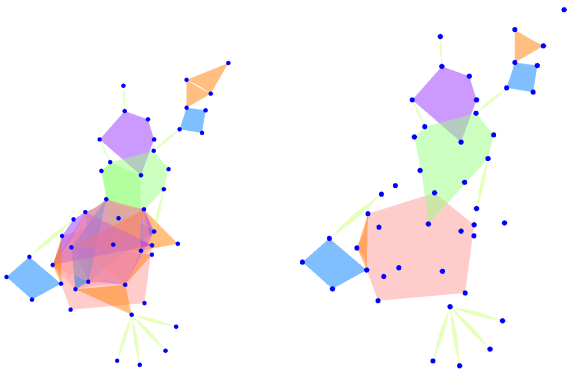


Figure 6: This figure compares the optimal layout for a dataset (left) and that of a filtered subset (right). Notice that visualizing a subset of relationships can lead to a layout that better shows the sebset.

6 MULTI-PLANE CW COMPLEX LAYOUT

While our editing operations can improve the planar layouts of CW complexes, it is usually not possible to completely remove visual overlaps among polygons. Depth peeling allows multiple visually overlapping polygons to be visible. However, we observe that the effectiveness of depth peeling decreases as the number of overlapping polygons increases.

To address this limitation of 2D visualization, we allow the polygons to be placed in 3D, which has the promise of removing the spatial overlaps between polygons that are partially overlapping or enclosing. The viewer can now inspect the layout in 3D with a much richer set of viewpoints to understand the data.

However, partially overlapping polygons, enclosing polygons, and co-locating polygons must be deformed to create spatial separation due to their common nodes. While having curved polygons may be visually pleasing, it can be difficult to see the cardinality of the polygon due to the curved boundary of the polygon. We wish to keep the polygons flat. To do so, we change the representation of a node from a sphere to a cylindrical tube (Figure 1: (c)-(d)). Polygons that have any overlaps in the planar layout are now placed in different but parallel planes and remain flat. Their shared node is visually represented as different nodes on the same tube.

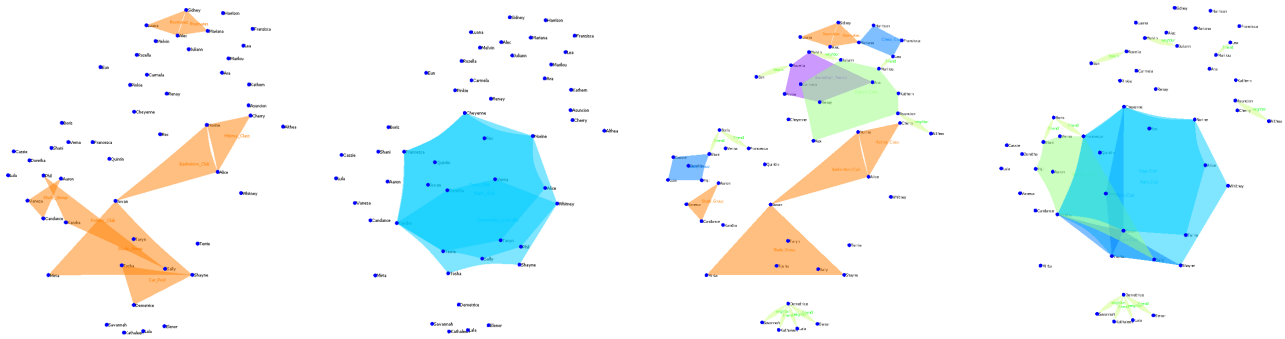
The 3D layout is generated by first computing the planar layout (Section 5). Next, we compute a height for each polygon so that no spatial overlaps can occur between any pair of polygons. This is achieved with a greedy approach. We put the first polygon in the plane $z = 0$. Then we iteratively place the remaining polygons in some plane as follows. Given a new polygon to be placed, we test for each existing plane whether by adding this polygon to the plane a spatial overlap with an already placed polygon can occur (collision). If such a collision does not occur, the polygon will be placed in the plane. Otherwise, the polygon will be tested for the next available plane. If all existing planes collide with the polygon, a new plane will be created at $z = N + 1$ where N is the height of the highest plane.

This leads to a 3D layout free of spatial overlaps between polygons.

7 APPLICATIONS

We have applied our analysis to two applications with N-ary relationships in order to demonstrate the utility of our CW complex visualization approach.

Social Network Complex Visualization: In this example, we make use of a synthetic social network dataset in which a group of individuals are connected through friendship circles, daily activities, and hobbies. Individuals and groups are represented by nodes and polygons, respectively. Our visualization can be used to answer questions regarding individuals as well as groups. In this case, we strive to clearly show the size and influential power of each group. The size of a group is measured in terms of the number of individuals in the group (cardinality of a polygon). Study groups, sewing clubs, and car pool participants tend to have a relatively high cardinality. The influential power is measured by the total number of other groups that share at least one member (degree of a polygon). We conjecture that if a group of high influential power initiates a social activity, the spread range will be the biggest



(a) (b) (c) (d)

Figure 7: In (a) and (b) we show the groups (polygons) with a size (cardinality) of three (a) and seven (b). In contrast, in (c) and (d) we show the groups with influential power (degree) of $1-8$ (c) and $12-18$ (d). It is interesting to notice that larger groups tend to have more influential power (polygons in (b) are also in (d)). Such insights are difficult to validate with traditional graph based visualization.

comparing with other groups. Or, if a member of this group is having a infectious disease while participating group activities, the disease has a high probability of being spread to the entire network of people.

Figure 7 shows the groups in terms of its size (a-b) and influence power (c-d). It is interesting to observe that larger groups tend to be more influential. Such insights, while intuitive, is difficult to validate with traditional graph-based visualization techniques. Further applications include disease modeling in the community of humans, animals and plants with reference to the geographic co-location and movement of the subjects, interactions-groups and their environment.

Gene Interaction Complex Visualization: We also apply our visualization to biological visualization. In gene visualization, when we have a very complicated and dense network, the visualization can only give viewers an impression of the structure. Most often, only the special nodes with more number of edges stand out and are noticeable, where as some subtle interactions get lost in the maze. For example, in Figure 8 a, we visualize a real gene interaction data from Arabidopsis thaliana (Source: [1]), where each edge represents an interaction determined experimentally. The locations of nodes of the polygon are calculated by a force-directed method. As we can see, the relationship between interacting genes, which should construct a group is missing from this graph (Figure 8a).

However, using CW complexes visualization directly suffers serious overlapping problems since the algebraic interactions between polygons often occur. In order to solve that problem, we apply filters to open the visualization in an inside out approach to analyze each layer separately. A similar approach was adopted to normalize the visualization of complex data plots and data arrays by the CIRCOS and Hive Plots [20, 21]. As Figure 9 shows, most N-ary relationships have two or three nodes (Figure 9 (a-b)) compared to the rarely occurring polygons with quaternary and quinary relationships (Figure 9 (c)).

Note that we only assigned five vertices to a polygon when these vertices construct a clique. This explains why polygons with higher cardinalities are rare. Meanwhile, The study of these genes shows clear N-ary relationships, e.g. the quaternary relationships in Figure 8 (b). The Arabidopsis thaliana genes AT5G03150 (JKD, JACKDAW), AT3G54220 (SCR, SCARECROW), AT4G37650 (SHR, SHORT ROOT) and AT1G03840 (MGP, MAGPIE) encode for proteins that bear transcription factor function. Based on the experiments, the encoded proteins are known to interact physically with each other [16, 24]) and play an important role in plant root development and radial patterning. Similarly, the cells which connect the genes AT2G35940, AT1G75410, AT2G23760, AT1G62360, AT4G32040, AT2G30400, AT4G34610, and AT1G23380 play a role in the maintenance of active shoot apical meristem where new cells are formed and later committed to making new organs such as leaves and reproductive parts and cell parts such as the secondary cell wall [13, 22].

When using the degree filter to open the visualization results, we can find N-ary relationships with different adjacency. We find even though the cardinality of the polygon is relatively high (the quad has a cardinality 4), the degree of this polygon can be low, which means the genes inside this quaternary relationships may not have much interactions with other genes. This implies that either there is not much experimental evidence, or the information still needs curation, or these N-ary relationships are yet to be analyzed in future experiment.

Though we can only infer gene-gene interactions through these networks as shown in the Figures 8 and 9, when combined and complemented with directed acyclic graphs of Ontology-based knowledge domains [5–7, 23, 27], such as molecular functions, biological processes, pathways, location in a cell, plant body and the developmental stage, environment [9, 25] and phenotypes the visualization methods described here become powerful analytics-based tools in the hands of biologists for making interpretations, building hypothesis for future validation in the laboratory.

8 CONCLUSION AND FUTURE WORK

In this research, we introduce the problem of CW complex layout and visualization, which is an extension of the graph layout problem. Instead of treating our data as cliques and using graph layout

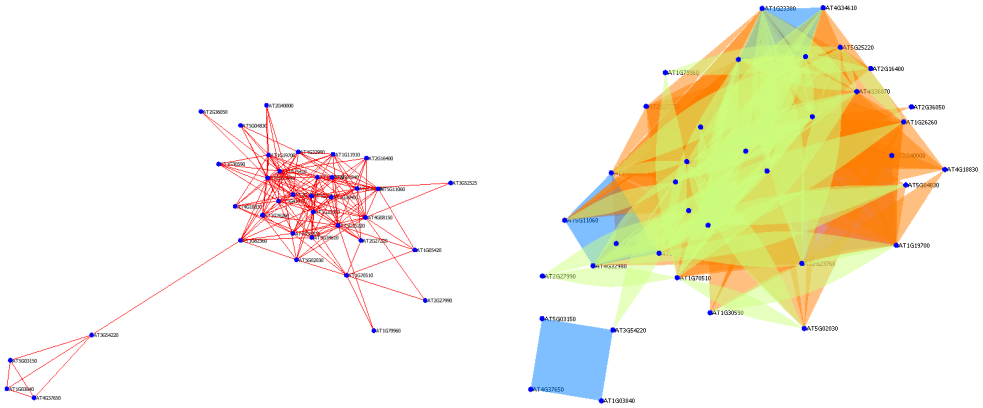


Figure 8: (a) Traditional graph visualization of the gene intersection data set. (b) The planar CW complex layout which suffers serious overlapping among polygons caused by algebraic intersections.

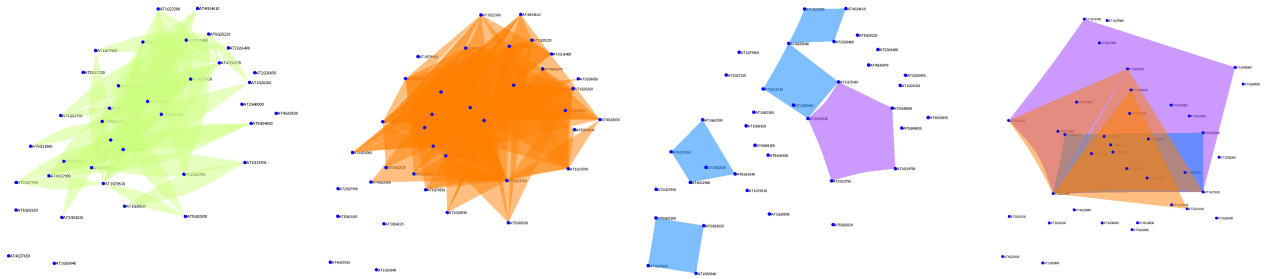


Figure 9: The gene interaction data with different filters: (a) binary relationships, (b) ternary relations, and (c) quaternary and quinary relationships. In (d) we show N-ary relationships with degrees from 34 to 44, highest degree among all polygons. Note that in this application the cardinality of a polygon does not seem to correlate with the degree of a polygon as in the social network example.

methods, we treat each N-ary relationship in the data as a polygon. This leads to a much heightened sense that an N-ary relationship being a single object. Furthermore, it is easier to see the cardinality and degree of an N-ary relationship.

We have also developed a number of principles that our research has shown to be important guidance for judging the quality of a CW complex layout. In addition, we build a framework that allows a CW complex to be visualized either in a single plane with possible spatial overlaps, or multiple parallel planes without spatial overlaps. We also provide a set of editing operations to improve the layout, including changing the order of nodes in a polygon, the display order of a polygon in a stack of overlapping polygons, as well as filtering the data.

Our method is not without limitations. For very large graphs, we have observed that the force-based graph method that we use to get an initial CW complex layout is far from being satisfactory. As such, the amount of user work involved to improve the layout makes editing impractical. In the future, we plan to investigate automatic techniques to generate an optimal layout.

While using 3D visualization can help with the overlapping polygon problem, finding good viewpoints becomes essential. We will investigate this further in our future research in this area.

We have applied our techniques to social network data and biological data. In the future, we will continue working with domain scientists in these areas for additional evaluations. Furthermore, we plan to explore additional applications such as power supply networks and road networks.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable suggestions. This work is supported by US National Science Foundation awards 1340112, 1566236, and 1619383.

REFERENCES

[1] arabidopsis.org. 2009. Tair Protein Interaction. ftp://ftp.arabidopsis.org/home/tair/Proteins/Protein_interaction_data/TairProteinInteraction.20090527.txt. (2009).

[2] L. Bavoil and K. Myers. 2008. *Order independent transparency with dual depth peeling.* Technical Report. NVIDIA Developer SDK 10. http://developer.download.nvidia.com/SDK/10\/opengl/src/dual_depth_peeling/doc/DualDepthPeeling.pdf

[3] A. R. BENSON, D. F. GLEICH, and J. LESKOVEC. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.

[4] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovačević. 2017. Fast Resampling of 3D Point Clouds via Graphs. *arXiv preprint arXiv:1702.06397* (2017).

[5] Gene Ontology Consortium and others. 2004. The Gene Ontology (GO) database and informatics resource. *Nucleic acids research* 32, suppl 1 (2004), D258–D261.

[6] Laurel Cooper and Pankaj Jaiswal. 2016. The Plant Ontology: A Tool for Plant Genomics. *Plant Bioinformatics: Methods and Protocols* (2016), 89–114.

[7] Laurel Cooper, Ramona L Walls, Justin Elser, Maria A Gandolfo, Dennis W Stevenson, Barry Smith, Justin Preece, Balaji Athreya, Christopher J Mungall, Stefan Rensing, and others. 2013. The plant ontology as a tool for comparative plant anatomy and genomic analyses. *Plant and Cell Physiology* 54, 2 (2013), e1–e1.

[8] G. L. CROMAR, A. ZHAO, A. YANG, and J. PARKINSON. 2015. Hyperscape: Visualization for complex biological networks. *Bioinformatics* 31, 20 (2015), 3390–3391.

[9] Andrew R Deans, Suzanna E Lewis, Eva Huala, Salvatore S Anzaldo, Michael Ashburner, James P Balhoff, David C Blackburn, Judith A Blake, J Gordon Burleigh, Bruno Chanet, and others. 2015. Finding our way through phenotypes. *PLoS Biol* 13, 1 (2015), e1002033.

[10] Cody Dunne and Ben Shneiderman. 2013. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 3247–3256. DOI: http://dx.doi.org/10.1145/2470654.2466444

[11] Helen Gibson, Joe Faith, and Paul Vickers. 2013. A survey of two-dimensional graph layout techniques for information visualisation. *Information Visualization* 12, 3-4 (2013), 324–357. DOI: http://dx.doi.org/10.1177/1473871612455749

[12] Palash Goyal and Emilio Ferrara. 2017. Graph Embedding Techniques, Applications, and Performance: A Survey. *CoRR* abs/1705.02801 (2017). http://arxiv.org/abs/1705.02801

[13] Jana Hackbusch, Klaus Richter, Judith Müller, Francesco Salamini, and Joachim F Uhrig. 2005. A central role of Arabidopsis thaliana ovate family proteins in networking and subcellular localization of 3-aa loop extension homeodomain proteins. *Proceedings of the National Academy of Sciences of the United States of America* 102, 13 (2005), 4908–4912.

[14] Richard W Hamming. 1986. *Numerical Methods for Scientists and Engineers (2Nd Ed.)*. Dover Publications, Inc., New York, NY, USA.

[15] Ivan Herman, Guy Melançon, and M. Scott Marshall. 2000. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 6, 1 (Jan. 2000), 24–43. DOI: http://dx.doi.org/10.1109/2945.841119

[16] Yoshinori Hirano, Masahiro Nakagawa, Tomoe Suyama, Kohji Murase, Maya Shirakawa, Seiji Takayama, Tai-ping Sun, and Toshio Hakoshima. 2017. Structure of the SHR–SCR heterodimer bound to the BIRD/IDD transcriptional factor JKD. *Nature Plants* 3 (2017), 17010.

[17] Yifan Hu. 2005. Efficient, high-quality force-directed graph drawing. *Mathematica Journal* 10, 1 (2005), 37–71.

[18] Steffen Klamt, Utz-Uwe Haus, and Fabian Theis. 2009. Hypergraphs and Cellular Networks. *PLoS Comput Biol* 5, 5 (29 May 2009), e1000385+. DOI: http://dx.doi.org/10.1371/journal.pcbi.1000385

[19] Ales Komarek, Jakub Pavlik, and Vladimir Sobeslav. 2015. *Network Visualization Survey*. Springer International Publishing, Cham, 275–284. DOI: http://dx.doi.org/10.1007/978-3-319-24306-1_27

[20] Martin Krzywinski, Inanc Birol, Steven JM Jones, and Marco A Marra. 2012. Hive plotsâĂŤrational approach to visualizing networks. *Briefings in bioinformatics* 13, 5 (2012), 627–644.

[21] Martin Krzywinski, Jacqueline Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. 2009. Circos: an information aesthetic for comparative genomics. *Genome research* 19, 9 (2009), 1639–1645.

[22] Yuanyuan Liu and Carl J Douglas. 2015. A role for OVATE FAMILY PROTEIN1 (OFP1) and OFP4 in a BLH6-KNAT7 multi-protein complex regulating secondary cell wall formation in Arabidopsis thaliana. *Plant signaling & behavior* 10, 7 (2015), e1033126.

[23] Anika Oellrich, Ramona L Walls, Ethalinda KS Cannon, Steven B Cannon, Laurel Cooper, Jack Gardiner, Georgios V Gkoutos, Lisa Harper, Mingze He, Robert Hoehndorf, and others. 2015. An ontology approach to comparative phenomics in plants. *Plant methods* 11, 1 (2015), 10.

[24] Hiromi Ogasawara, Ryuji Kaimi, Joseph Colasanti, and Akiko Kozaki. 2011. Activity of transcription factor JACKDAW is essential for SHR/SCR-dependent activation of SCARECROW and MAGPIE and is modulated by reciprocal interactions with MAGPIE, SCARECROW and SHORT ROOT. *Plant molecular biology* 77, 4-5 (2011), 489–499.

[25] Anne E Thessen, Daniel E Bunker, Pier Luigi Buttigieg, Laurel D Cooper, Wasila M Dahdul, Sami Domisch, Nico M Franz, Pankaj Jaiswal, Carolyn J Lawrence-Dill, Peter E Midford, and others. 2015. Emerging semantics to link phenotype and environment. *PeerJ* 3 (2015), e1470.

[26] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. 2006. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theorectical Computer Science* 363, 1 (2006), 28–42.

[27] Ramona L Walls, Balaji Athreya, Laurel Cooper, Justin Elser, Maria A Gandolfo, Pankaj Jaiswal, Christopher J Mungall, Justin Preece, Stefan Rensing, Barry Smith, and others. 2012. Ontologies as integrative tools for plant science. *American journal of botany* 99, 8 (2012), 1263–1275.