3D Exploration of Graph Layers via Vertex Cloning

James Abello[†], [*]
Rutgers University

Fred Hohman[‡], [*]
Georgia Institute of Technology

Duen Horng (Polo) Chau[§]
Georgia Institute of Technology

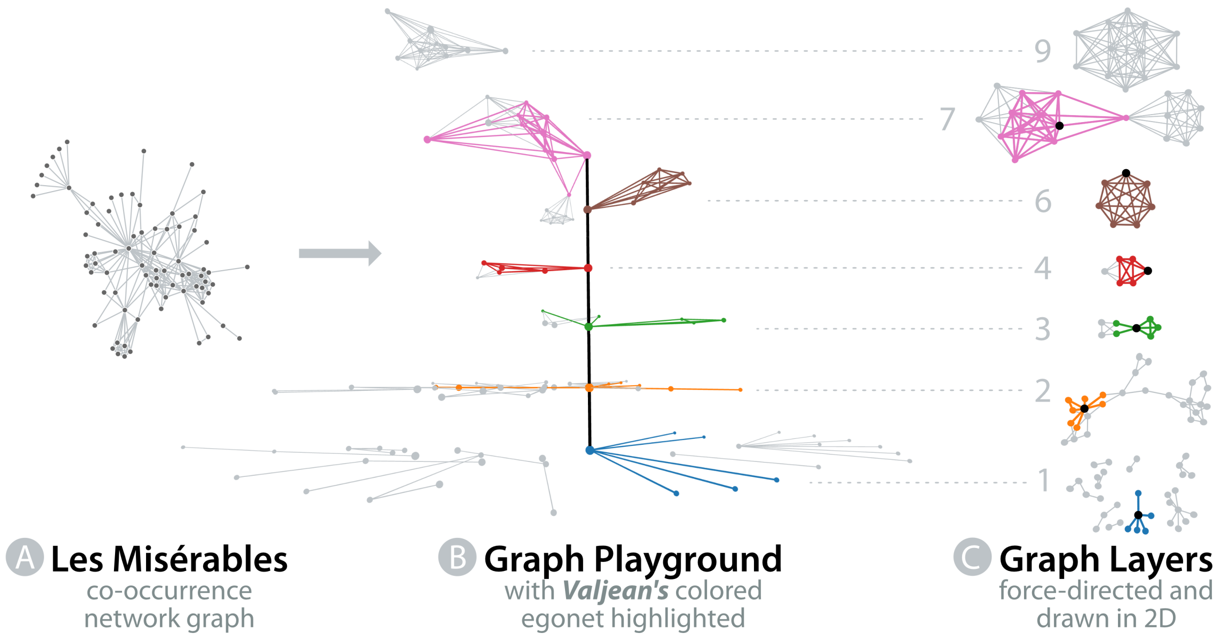
9
7
6
4
3
2
1
A Les Misérables
co-occurrence network graph
B Graph Playground
with Valjean's colored egonet highlighted
C Graph Layers
force-directed and drawn in 2D

Figure 1: (A) The *Les Misérables* co-occurrence network of the novel's characters, visualized using standard force-directed layout. (B) A *graph playground* created by applying fixed-point *edge decomposition*, producing *cloned vertices* that appear in multiple layers. The character *Valjean* appears in six layers; his clones are connected using a vertical black line, and his egonet is highlighted in every layer. (C) The graph playground layers separated and individually redrawn using force-directed layout in 2D, with Valjean's colored egonet still shown. Our method reveals interesting subgraph structures and distributes them into layers, e.g., stars in layer 1 (blue), and a clique in layer 6 (brown). Valjean's vertex is colored black in every layer he exists in (all layers except layer 9), highlighting his central role in the novel and his diverse participation in different graph patterns.

Abstract

We use an iterative edge decomposition approach, derived from the popular iterative vertex peeling strategy, to globally split each vertex egonet (subgraph induced by a vertex and its neighbors) into a collection of edge-disjoint layers. Each layer is an edge maximal induced subgraph of minimum degree k that determines the layer density. This edge decomposition is derived completely from the overall network topology, and since each vertex can appear in multiple layers, we can associate to each vertex a vector profile that can be used to identify its different "roles" across the network. This allows us to explore a network's topology at different levels of granularity, e.g., per layer and across layers. This is only feasible by mapping simultaneously a vertex to a set of 3D coordinates (x, y, and z) where the third coordinate encodes the different layers a vertex belongs to. This is one of the few instances where 3D visualization enhances graph exploration and navigation in an arguably "natural" way: a graph now becomes a *3D graph playground* where a vertex plays a certain "role" per layer that is determined by the overall network topology. Our approach helps disentangle "hairball" looking embeddings produced by conventional 2D graph drawings.

Index Terms: G.2.2 [Discrete Mathematics]: Graph Theory—Graph algorithms; H.5.m [Information Interfaces and Presentation (e.g., HCI)]: Miscellaneous

1 Introduction

Graphs are everywhere, growing increasingly complex, and still lack meaningful large scale representations and interactive tools to support sensemaking. Typical approaches to visualize graphs include force directed layouts, vertex clusterings, and topological contractions to reduce visual complexity. Advanced analytic techniques like identifying vertex roles and diversity, graph motifs, and graph summarizations have been addressed in [2, 3], yet scalability and interaction are two pressing issues of central importance in "large" graph exploration systems. Extracting overall descriptive information about an unknown graph data set as it is being explored is a desirable feature that can amplify users ability to discover "out of the box" data features [4]. Edge decomposition algorithms based on fixed points of degree peeling have strong potential to aid users exploration of unknown graph data because these algorithms [1]:

- are iterative and scalable at a reasonable level of granularity that is determined by each layer density (linear in the number of edges per layer);

[†]abello@dimacs.rutgers.edu
[*]Authors contributed equally
[‡]fredhohman@gatech.edu
[§]polo@gatech.edu

- can be parameterized by time and space depending on the level of resources available to the user;
- discover subgraph patterns structurally similar or dissimilar to regular subgraphs;
- quantify the variety of "roles" a vertex can play in the overall network topology (vertex cloning).

We report our ongoing work in using edge decompositions as a central graph theoretical mechanism for 3D navigation, exploration, and large data sensemaking [1]. Figure 1 demonstrates our approach by decomposing and visualizing the *Les Misérables* network[1] into graph layers, and highlights within it the egonet decomposition of the central character *Valijean*. Our primary contributions are:

- Splitting graphs into fixed points of degree peeling (graph layers) using an edge decomposition, simultaneously revealing structure in graph layers and aspects of vertex diversity in 3D.
- A 3D web-browser Graph Playground tool that uses GPUs to offer users iterative edge decompositions suitable for exploration and navigation.

2 Graph Layers and Visual Vertex Cloning

Graph Layers. Iteratively removing vertices of minimum degree partitions the vertices of any graph $G=(V,E)$ into a collection of subsets, each of which is characterized by its iterative minimum degree in G, called the *peel value* of the subset. We call the subgraph induced by the subset of vertices with highest peel value the $EdgeCore$ of G, i.e., the highest edge *layer* of G. If $EdgeCore(G)$ is different from $E(G)$, remove the $EdgeCore$ from $E(G)$ and iterate. This process produces an edge decomposition of $E(G)$ into layers of decreasing density. Each such layer can be explored and analyzed, independent of other layers, by using methods that exploit the fact that each vertex in this layer has the same peel value, e.g., layer i contains vertices of peel value i. Interesting subgraph "patterns" found in a particular layer are seen in the global graph context by looking at the 3D volume generated by "clones" of the pattern vertices.

This iterative edge decomposition produces an ordered set of graph layers where lower layers consist of simpler graph structures (e.g., layer 1 consists of "trees" and "stars"). Higher layers consist of higher-order patterns, such as quasi-cliques, which are often candidates to be explored to find traditional local graph "communities" based on high density.

Visual Vertex Cloning. Whereas each edge is assigned only one peel value, a vertex can appear in multiple layers if it is connected to multiple edges that belong in different layers. We call vertices that appear in multiple layers *cloned vertices*. Therefore, not only does this edge decomposition algorithm assign peel values to edges, it also produces a vector profile for each vertex describing what layers that vertex exists in. The Shannon entropy of a normalized version of this vector profile has been used as a measure of vertex diversity [1]. This helps reveal the various roles a vertex can play in the overall graph. This is seen in Figure 1B, where a vertical black line connects all the clones of the character (vertex) *Valjean* in the *Les Misérables* graph. *Valjean* exists in layers 1,2,3,4,6, and 7, and since this graph does not produce layers 5 and 8, the only layer *Valjean* does not appear in is layer 9. Figure 1B also highlights *Valjean's* egonet in every layer in color, crystallizing his importance and diversity in the entire graph.

3 Graph Playground System Design

Our prototype system uses a Python back end to compute the edge decomposition of network graphs. The implementation draws the

[1]`http://www-cs-faculty.stanford.edu/~uno/sgb.html`
[2]`http://snap.stanford.edu/data/egonets-Facebook.html`

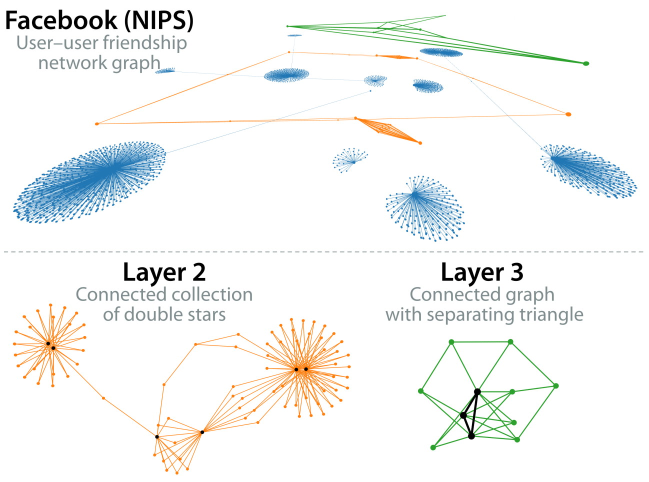
Facebook (NIPS)
User–user friendship network graph
Layer 2
Connected collection of double stars
Layer 3
Connected graph with separating triangle

Figure 2: A larger graph playground of a Facebook user-user friendship network (2,888 vertices, 2,981 edges)[2]. This graph decomposes into 3-layers and highlights the difference in structure between vertices belonging in layer 1 (blue, collection of stars), layer 2 (yellow, connected collection of double stars), and layer 3 (green, connected graph with a separating triangle).

graph using a standard force-directed 2D layout (giving vertices x and y-coordinates), performs the edge decomposition (i.e., assigning each edge a single layer number, which is used to calculate the z-coordinate), and finally uses a WebGL wrapper to render the 3D structure in the web browser. Rendering 3D graphs using GPUs promotes real-time interactivity, for movement controls like zooming, panning, and rotation.

4 Conclusions and Ongoing Work

Our work uses an iterative edge decomposition algorithm to globally split graphs into a collection of edge disjoint layers. We then visualize these graph playgrounds in 3D, deriving the z-coordinate as a function of the layer assignment. This edge decomposition algorithm is largely unexplored for visualization and shows strong potential for graph sensemaking. We postulate the 3D representation of these structures is useful, and a synchronization between 3D and traditional 2D views provide new ways for deriving insights.

We will further experiment with 3D representations to create visually scalable representations of power-law graphs with millions of vertices and edges (much larger than the other graph playgrounds shown above), potentially using summary views as seen in [4]. We plan to evaluate how our 3D structures could enhance and complement existing 2D graph analytic and visualization systems.

Acknowledgments

This work was supported by NSF grants IIS-1563816, IIS-1563971 & TWC-1526254.

References

[1] J. Abello and F. Queyroi. Network decomposition into fixed points of degree peeling. *Social Network Analysis and Mining*, 4(1):1–14, 2014.

[2] C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *CHI*, pp. 3247–3256. ACM, 2013.

[3] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: structural role extraction & mining in large graphs. In *KDD*, pp. 1231–1239. ACM, 2012.

[4] S. Van den Elzen and J. J. Van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE TVCG*, 20(12):2310–2319, 2014.