Visualization of Large Graph in Immersive Environments

Samuel Guleff
Columbia University, Sg2665@columbia.edu

Abstract – Network and graph visualizations pose extreme challenges due to the density and interconnectedness of data. Many tools attempt to solve this through intelligent layout and interactivity with limited success. While applying the principles of network/graph layout and interactivity with an immersive environment (AR – Augmented Reality or VR – Virtual reality) we can immerse end users in a much larger set of data while interactively presenting valuable insights.

Keywords – Augmented Reality, Graphs, Networks, Virtual Reality, Visualizations

I. INTRODUCTION

Graph and network visualizations, due to limited visualization space and the density of nodes and edges, can cause plots to appear chaotic. Overlaying additional edge and node properties such as weight utilizing colors, weight and line thickness often add to the complexity of the visualization.

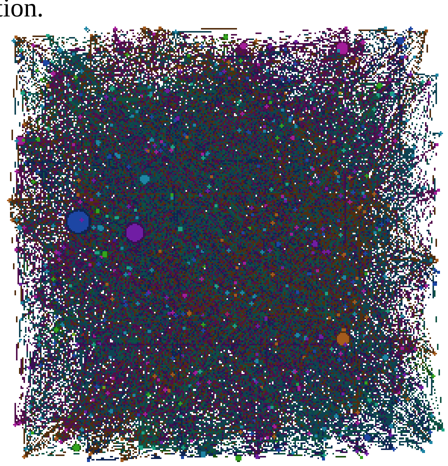

Figure 1-Arbitrary Graph Layout

Applying a force directed graph algorithm to graph can intelligently optimize graph layout allowing pleasing results with little to no oversight by users [1]. There are many advantages provided including flexibility, intuitive layout, simplicity, interactivity with the negatives of higher running time and poor local minima [2] It can provide interactivity with the graph as shown in figure3.

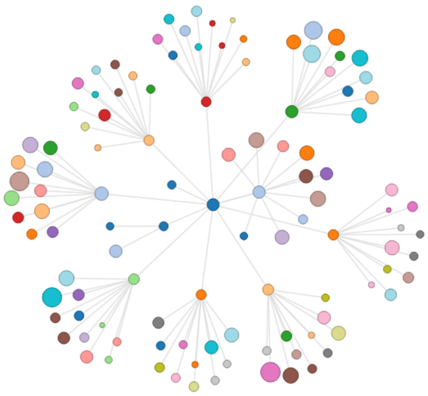

Figure 2-Force Directed Graph [3]

Further enhancements to graph layout have occurred as advancements to 3D software and hardware have become more mainstream. This has enabled even more graph properties to be displayed.

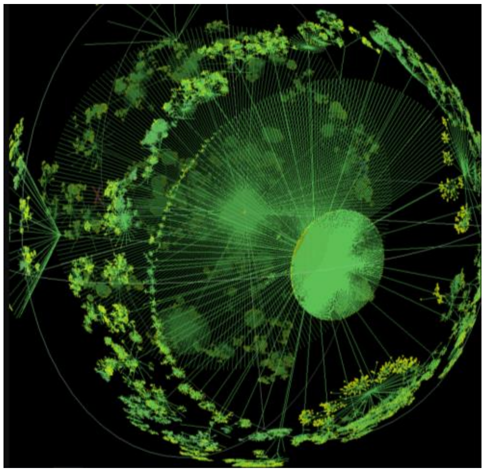

Figure 3-3D Graph Layout

The penultimate layout appears to be rendering graphs into an immersive environment, leveraging all the interactivity of the previous layouts while providing a nearly infinite canvas and numerous intuitive methods of interactivity, touch, gaze, and voice.

II. RELATED WORKS

- **AR/VR Hardware –** Microsoft is currently leading the research charge in Augmented Reality hardware providing a relatively inexpensive commercial prototype Hololens and SDK allowing companies and universities to advance research in AR [4]. As of the time of this writing Microsoft is currently working on 3rd generation hardware which will likely provide greater field of vision, better processing, and better pricing [5]. Additionally, the promise of AR innovation can be seen as Magic Leap has raised nearly $1.5Billion USD [6].
  Consumer VR has become relatively mainstream primarily for gaming as HTC, Oculus(Facebook) and Sony(PlayStation) have all released 1st Generation hardware focused on enthusiasts [7].
- **AR/VR Software –** Looker, a BI platform has taken up the lead in interactive immersive 3D Visualizations providing an API to integrate with both the Oculus Rift and HTC Vive [8].
- **AR/VR Research** – AR/VR research at universities has been applied to a variety of problems Manufacturing, architecture, surgical procedures, education, auto safety, field workers, art, archeological exploration, and geospatial data [9][10][11][12][13][14][15][16][17].
  There have been significant research into Visualizations to provide better insights into the bigger data sets that are provided today [18][19].

III. DESIGN OVERVIEW

The primary goal of the design will be to build a modular plug in library that can handle all the basic functionality needed to communicate with external data source, provide external logging, store and analyze graphs in memory, render graphs in a spatial environment, provide metadata around graphs and node and edge properties, and provide interactivity through gaze, touch and voice.

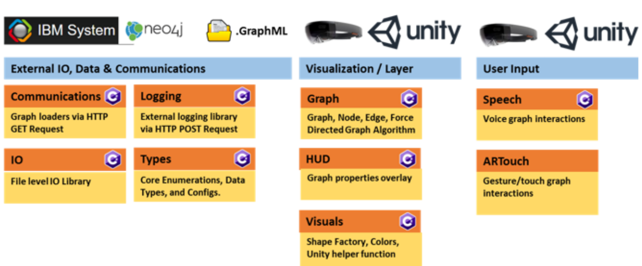

Figure 4- AR Graph Core Libraries

Graph Loading Mechanisms – As a preliminary loading mechanism the engine will use a http get command to URLs where GraphML standard files are stored. This allows the application to pull and process the Graphs which are represented as XML. Further developments to leverage the REST functionality of IBM SystemG and Neo4j will be developed as time allows. Once these standard interfaces are built the library should be extended to query these graph databases to leverage the computation power of the sever rather than relying on the lightweight processors which are likely to be placed in AR headsets.

External Logging – To extract, store and analyze performance and system logs the library leverages web POST requests to Postgrest which stores all the log data for future analytics.

In memory Graph Structure and Methods – The design pattern of the graph mimics a doubly linked list with each node containing a map of all incoming and outgoing edges. Edge objects contain pointers to start and end nodes. This architecture allows for lookup by key, traversal from node to edges, as well as looping across all notes or edges globally. Additional functionality has been partially developed to allow standard ANSI SQL commands to query edges, nodes, and properties. Additionally, as graphs might have nearly infinite properties, each edge or node contains a map of object type which can store all associated properties, as well as explicitly defined fields for common types e.g. Weight.

UX of Graphs – Graph nodes are represented by spheres and edges by hyperrectangles. Scales can be adjusted for weight and color can be adjusted for additional numerical field mappings which allows the graph to represent up to 6 dimensions x,y,z,t + 2. A simplistic force directed graph and random placement algoritm(s) have been developed to provide some initialization of the graph within a defined bounding box. Additional algorithms have been created to provide intelligent scaling and heatmapping of any numerical properties. Allowing the user to iterate rapidly over these visualizations is the core way end users will interact with the graph.

Efficient Graph Algorithms – A simplistic Force directed graph algorithm was developed to attempt to provide relative local grouping of nodes with their connected components. Additional a Breath First Search as well as a Depth First Search algorithm have been implemented to provide visual clues for the layout of the graph.

HUD of Metadata – As a user interacts with the graph the library should provide a HUD to display to the user interactive properties of the current graph and currently selected nodes or edges. As this was relatively cumbersome to develop it did not make it into the final build please see refinements from current work in section IV.

Graph Interactivity- The core library seeks to Provide as many intuitive ways to interact with the graph as possible. By iterating over a set of voice commands the user will be able to hide and show nodes and edges with specific sets of properties e.g. edges with properties above a set threshold, or densely connected nodes. This interactivity could be expanded nearly endlessly as additional commands and metrics are needed. Touch can provide a way to manipulate the graph as well as to explicitly select edges or nodes to view their properties.

IV. REFINEMENTS FROM CURRENT WORK

Graph gesture (touch) interactive is currently implemented in a rudimentary state, where nodes or edges which have been "captured" by a user's gaze are selectable. Once selected the properties of the graph entity are read to the user. This is a subpar experience as the properties list can contain vast amounts of data which would not allow the user to interpret this well. A UI to limit to specific subfields should be developed as well as a HUD (Heads Up Display) to show the relevant properties to the user as their gaze hovers over graph entities. Lastly more touch interactivity could be added such as touch-hold-drag interactivity to move nodes and edges around to optimize the layout of the graph.

Graph loading types were selected based on their relative ease to implement as such the XML formatted GraphX format as well as a Neo4j connector were built. Currently the Neo4j connector only handles simplistic graph types and the library is not generic enough to allow for the serialization of the more complex types. This library needs significant overhaul however it is no small task given the limited libraries that are currently available for Hololens applications which are deployed via Unity. Additional Graph Database technologies should be investigated as well such as IBM SystemG, or support for the Tinker Pop Stack.

Further refinements could be applied to load smaller subsections of the graph and use either a gesture motion to traversed from the currently loaded nodes to the next section of the graph, or as a user walked it could have a bounding box defined and nodes and edges that fell out of the bounding box could be removed from the rendering and more edges and nodes could be loaded. An outer bounding box could be created to connect back to the external data source as needed to pull more of the graph back into memory, which would allow a nearly infinite graph space to be slowly loaded. By maintaining a larger graph in memory and only rendering a smaller subset of the graph we could effectively present a much cleaner UI to the end user. This optimization would require significant development work

V. CONCLUSION

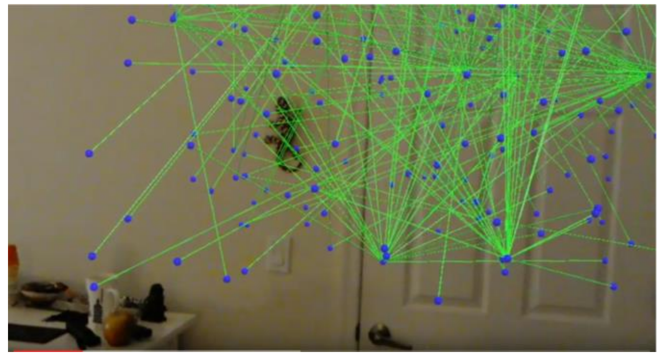

Figure 5- Final Visualization

The library provides basic connectivity to Neo4j databases as well as GraphX standard graph types, an ETL framework to load the graph data into memory, and preliminary ways to move nodes in rendered space. The library provides graph interactivity through gaze, tough, speech recognition and Many refinements, refactoring and optimizations would be needed to productionalize this code.

Offering an interactive user interface to parse through dense graph data is no trivial task. The core library that was developed seeks to simplify the end users task to mine for patterns in the data by allowing for quick interactions on the layout of the graph, scaling size and color dynamically on a variety of graph, edge and node properties. This interactive workflow with some patience should allow an experienced user to look for anomalies in the data visually in size or color space.

VI. ACKNOWLEDGMENTS

Professor Ching-Yung Lin and IBM Research for guidance and use of a Hololens for research and development purposes.

VII. REFRENCES

[1] Spring Embedders and Force Directed Graph Drawing Algorithms https://arxiv.org/abs/1201.3011
[2] Force Directed Graph Drawing https://en.wikipedia.org/wiki/Force-directed_graph_drawing
[3] JS Demo of Force Directed Graphs http://projects.flowingdata.com/tut/interactive_network_demo/
[4] Microsoft Hololens https://www.microsoft.com/microsoft-hololens/en-us

[5] Microsoft Gen 3 Hololens http://www.kitguru.net/tech-news/matthew-wilson/microsoft-reportedly-scraps-hololens-2-in-favour-of-skipping-to-gen-3/
[6] Magic Leap
https://www.crunchbase.com/organization/magic-leap#/entity
[7] Consumer VR After 1 Year
https://techcrunch.com/2017/03/28/as-the-oculus-rift-turns-one-facebook-is-still-responsible-for-where-vr-goes-next/
[8] Looker Immersive VR Data Visualizations
https://looker.com/blog/creation-of-lookvr
[9] Touring Machine:
http://link.springer.com/article/10.1007/BF01682023
[10] Manufacturing
http://ieeexplore.ieee.org/abstract/document/183317/
[11] Architecture
http://www.columbia.edu/cu/gsapp/BT/RESEARCH/PAPERS/ar-asce.html
[12] Surgical
http://www.sciencedirect.com/science/article/pii/S0278239197906893
[13] Auto Safety http://dl.acm.org/citation.cfm?id=1105175
[14] Field Workers
http://link.springer.com/article/10.1007/s00779-008-0204-5
[15] Art http://dl.acm.org/citation.cfm?id=985060
[16] Geospatial Data
http://www.mitpressjournals.org/doi/abs/10.1162/1054746021470577
[17] Archeological Exploration
http://ieeexplore.ieee.org/abstract/document/1028726/
[18] Immersive and Collaborative Data Visualization Using VR Platforms
https://arxiv.org/ftp/arxiv/papers/1410/1410.7670.pdf
[19] Novel Approaches to Data Visualization
https://www.microsoft.com/en-us/research/wp-content/uploads/2012/04/djorgovski.pdf?from=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fum%2Fredmond%2Fevents%2Fescience2012%2Fdjorgovski.pdf

VIII. AUTHOR INFORMATION

Sam Guleff, MS Candidate Data Science, Columbia University.