Fluid UI for HIGH-dimensional Analysis of Social Networks

Riku Takano
Tokyo Institute of Technology
rick.takano@gmail.com

Ken Wakita
Tokyo Institute of Technology
wakita@is.titech.ac.jp

ABSTRACT

Social Viewpoint Finder (SVF) is a visual analytics tool for social networks and complex networks. SVF lays out the input network data on a HIGH-dimensional (500-3,000 dimensional) Euclidean space, offers a simple, but unique dragging-based UI to trigger computation-intensive, high-dimensional rotation of the presented network, and let the user investigate the clustering structure of the network. This demonstration presents the effectiveness of SVF as well as the employed implementation techniques that enabled the fluid, complex user interaction. To achieve fluidness, SVF heavily relies on modern OpenGL technologies. It achieves massively parallel computing through the use of the compute shader, and graphic pipelines. A large volume of the network data and its layout information is stored in a shader storage buffer. A fragment shader-based, efficient object identification method is devised.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces; I.3.6 Computer Graphics: Methodology and Techniques

Author Keywords

Network Visualization; Interaction; Modern OpenGL

INTRODUCTION

The visual analytics system is a tool that allows us look into and reason about the data, and that assists us to discover scientific findings, to comprehend the business structure, and so on. The visual analytics tool is most effective when it takes over our mental and physical burdens, helps us focus on data analysis tasks, and efficiently and effectively guides us in understanding scientific phenomena, social activities, and business logics,

Social Viewpoint Finder (SVF) is a successful visual analytics tool for the user to navigate the complex, social networks and reason about its structure[1]. It depicts the topology of the complex/social network and offers a simple, but unique user-interaction method. The user can choose an arbitrary node of the network and move it to a random position by dragging. This dragging operation triggers and interesting visual effects

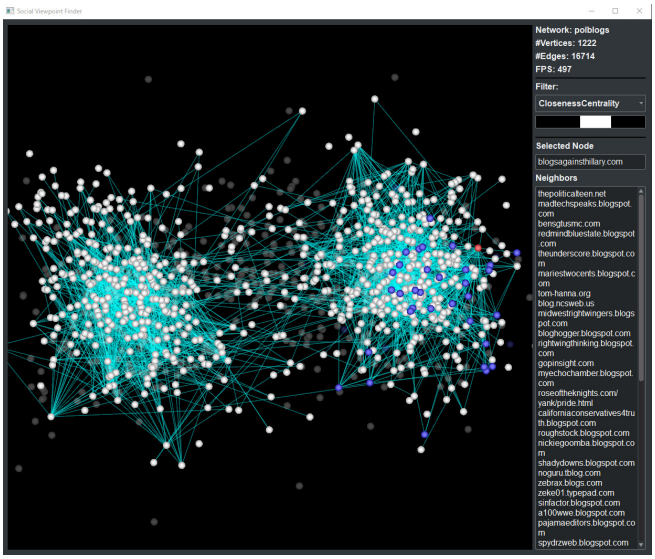

Figure 1. Social Viewpoint Finder (Analysis of the citation network among US political blogs). This video and more are available on our web site.

that often segregates the network into several subcomponents called *(community) clusters*. In other words, a series of node dragging operations is regarded a manual human effort of network clustering. For example, Figure 1 demonstrates a visual analytics process of a *citation network among US political blog sites*. At start, the visualization of the network is a *giant hair ball* as we commonly see in visualizations of other types of networks. The binary structure depicted in the figure was exposed, moving only a few random nodes by dragging (see more in our video). The two subcomponents correspond to the blog sites of the supporters of two political parties.

The heart of SVF is interactive, HIGH-dimensional visualization method called *active graph interface* (AGI), proposed by Hiroshi Hosobe[2]. AGI, places the network topology in a high-dimensional Euclidean space whose dimension can often be higher than 500. The interaction mechanism is based on *minimization of a non-linear cost function* and *high-dimensional rotation technique*. This combination of a heavy-weight computational task on the computer side and light-weight operating task of mouse dragging on the human side is ideal for successful visual analytics tools.

A challenge of SVF implementation is to realize quick processing of heavy-weight computational task to catch up with series of light-weight operations issued by the human actions. As mentioned by Card and others, a highly continuous, light-

IUI'18 Companion March 7–11, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-5571-1/18/03.
DOI: https://doi.org/10.1145/3180308.3180336

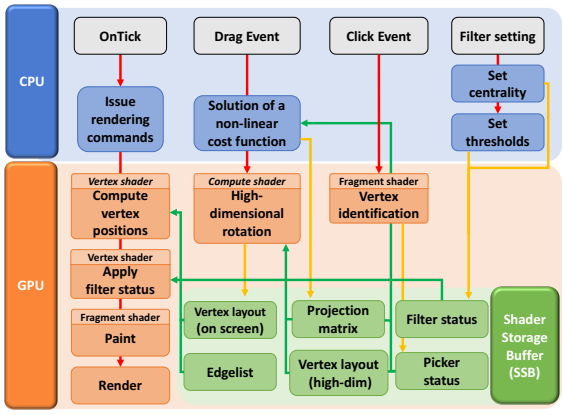

Figure 2. Implementation Scheme

weight dialogue for the such as mouse dragging and animation demands less than 100 milliseconds response time[3].

The purpose of the demo are twofold: firstly it demonstrates such seemingly impossible combination is technically feasible, and secondly to present the outline of the implementation techniques. SVF can process social networks that contains hundreds of thousands of connections placed on 500-dimensional Euclidean space and response time is less than 8 milliseconds on a moderate desktop PC (Intel 4.0GHz Core i7, 8GB DRAM, PCI Express x16 3rd gen., NVidia GTX 770). The employed technologies are *computation shader* for massively parallel computation, *shader storage buffer* for storage of a large volume of data on the GPU, vertex and fragment shaders for fast network rendering and smart object identification purposes.

IMPLEMENTATION

In order to let the SVF user enjoy fluid, visual analytics experience, acceleration of the costly computational task after every most dragging is important. Acceleration gain was primarily achieved by the use of massively parallel computing capability of the *compute shader*. Equally important issue was careful assignment of the computational tasks to stages of the graphic pipeline and data allocation to shader storage buffer, which allows the shaders to share data structures.

The computation of the SVF's interaction consists of four tasks (see Figure 2). The first thing, after the user starts dragging a node is identification of the network vertex that corresponds to the node being clicked. The window coordinate of the node is passed to the graphic pipeline. The graphic pipeline identifies ID of the clicked node and store its ID in SSB, which is retrieved by the (CPU-side) application. Secondly, after each drag event, the application calculates the high-dimensional rotation parameters from the mouse position. As this task is fully-serialized, optimization of a non-linear cost function, it is pursued by the application. Thirdly, the compute shader takes the rotation parameters and converts them to an updated projection matrix which is used to compute their two-dimensional positions and store them in the SSB. This task is where high-dimensional rotation takes place. Finally, the graphic pipeline reads in vertex coordinates from the SSB and draws the visualization.

Table 1. Comparison of the frame rates of CPU-based and Shader-based implementations

Dataset	V	E	CPU-based	Proposal
USPol	1,222	16,714	57	331.1
Math	3,608	48,315	26	293.3
Internet	22,463	48,436	3.8	228.8
Enron	33,696	180,811	1.9	162.6

Vertex Identification stage conceptually attempts to find the vertex who has the same coordinate as the mouse position. In our implementation, this computation is performed by a non-drawing fragment shader. If the pixel of the fragment shader equals the mouse position, the vertex ID associated with the fragment is stored in the SSB. This implementation is very efficient because it takes advantage of per-pixel parallelism of the fragment shader.

Rotation of the high-dimensional network positions is a computation-intensive task but is highly parallelizable one as it is essentially matrix multiplication of a huge matrix that holds the high-dimensional coordinates of the vertices and smaller projection matrix. This computation is performed by the compute shader, which gives the CUDA-like GPGPU power to shader programming. The two-dimensional vertex-coordinates of are saved in SSB.

Filter and Rendering the network is achieved by the graphic pipeline. Filtering the vertices and edges is established by conditional drawing at the vertex shader and rendering is done by the fragment shader.

Result: Table 1 compares the performance of SVF against our previous CPU-based implementation [1]. The previous implementation was capable to handle network dataset that consists of upto a few thousand vertices. When it is fed with a larger dataset like Internet and Enron, its frame rate dropped down to a 1.9-3.8. The response time longer than 250 ms gives shaky animation and is very uncomfortable for the SVF user. On the other hand, the newer implementation hits more than 160 fps; a less than 7 ms response time.

SUMMARY

In this work, the use of modern OpenGL shader techniques successfully combined a heavy-weight computational task and light-weight user interface, and made SVF an comfortable and ideal visual analytics tool for complex networks.

REFERENCES

1. K. Wakita, M. Takami, and H. Hosobe. Interactive high-dimensional visualization of social graphs. In *IEEE PacificVis '15*.
2. H. Hosobe. A high-dimensional approach to interactive graph visualization. In *ACM SAC '04*
3. S. K. Card, G. G. Robertson, and J. D. Mackinlay. The Information Visualizer, an Information Workspace. In *ACM CHI '91*.