A Multi-Level Middle-Out Cross-Zooming Approach for Large Graph Analytics

Pak Chung Wong[1], Patrick Mackey[2], Kristin A. Cook[3], Randall M. Rohrer[4], Harlan Foote[5], Mark A. Whiting[6]

[1,2,3,5,6] Pacific Northwest National Laboratory [4] U.S. Department of Defense

ABSTRACT

This paper presents a working graph analytics model that embraces the strengths of the traditional top-down and bottom-up approaches with a resilient crossover concept to exploit the vast middle-ground information overlooked by the two extreme analytical approaches. Our graph analytics model is co-developed by users and researchers, who carefully studied the functional requirements that reflect the critical thinking and interaction pattern of a real-life intelligence analyst. To evaluate the model, we implement a system prototype, known as GreenHornet, which allows our analysts to test the theory in practice, identify the technological and usage-related gaps in the model, and then adapt the new technology in their work space. The paper describes the implementation of GreenHornet and compares its strengths and weaknesses against the other prevailing models and tools.

KEYWORDS: Graph analytics, information visualization.

INDEX TERMS: H.1.2 [User/Machine Systems]: Human Information Processing – Visual Analytics; I.6.9 [Visualization]: information visualization.

1 INTRODUCTION

Graph analytics is a way of facilitating guided graph exploration through visual and interactive means. It has been among the most discussed topics in the annual IEEE Symposium on Visual Analytics Science and Technology (VAST) [17] in recent years. Unlike many graph visualization research efforts that focus predominantly on layout algorithms and rendering techniques, graph *analytics* research strives to provide an engaging interactive journey that bridges the gap from data to information to knowledge. Graph visualization still plays a vital role in building this analytical journey, as do database querying, graph mining, interactive interrogation, and human judgment and senses.

Until recently, there were two primary schools of thought when designing graph analytics tools: top-down and bottom-up. The top-down approach often provides an initial full view of the entire dataset and then gradually reaches out to the local details. The bottom-up approach frequently starts with seed nodes or a subset of nodes, and then builds the rest of the graph through associations. As we will discuss later, neither is ideal, nor is one better than the other. They are suited to different tasks or goals.

This paper presents a pragmatic solution that embraces the strengths of both top-down and bottom-up approaches with a resilient crossover concept to exploit the vast middle-ground information overlooked by the two extreme analytical approaches. We call it a *multi-level middle-out cross-zooming model for large graph analytics*. The model is co-developed by users and researchers, who carefully studied the functional requirements that reflect the critical thinking and interaction pattern of a real-life intelligence analyst. To evaluate the model, we implement a system prototype, known as GreenHornet, which allows our analysts to test the theory in practice, identify the technological and usage-related gaps in the model, and then adapt the new technology in their work space. Figure 1 sketches the fundamental conceptual differences among the three analytical models.

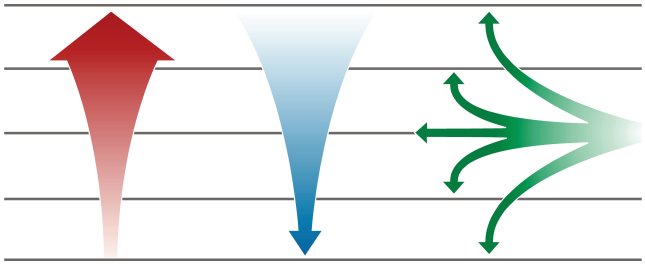

Figure 1: Bottom up (red), top down (blue), and middle out (green) graph analytics design concepts. The gray straight lines reflect different degrees of information details.

In laymen's terms, our model 1) generates a hierarchy of increasingly coarse layouts G_n, … G_i, ... G_1, G_o of a graph G, 2) selects one of the coarsened layouts G_i as an initial view of G, 3) applies a user query to G and visualizes both the selected and unselected graph nodes, and 4) allows the users to globally or locally, and jointly or independently, zoom in and out of both the selected and unselected nodes. So in essence, step 1 provides the required *multi-level* support, steps 2 and 3 incorporate the *middle-out* analytical concept (that characterizes the middle passages between the top-down and bottom-up approaches), and finally, step 4 reflects the *cross-zooming* capability across the multi-level hierarchy.

Perhaps our approach so closely resembles the natural problem-solving instincts that some might overlook its complexity and requirement for a successful implementation. Among the challenges is the fact that a database query often provides exact answers, whereas a multi-level graph visualization shows only coarse approximations. Difficulties arise when these two components have to support and facilitate each other. For example, for nodes within a coarsened graph that can aggregately contain both qualified and non-qualified leaf nodes, we must determine if these coarsened nodes should be put in the *foreground* or the *background*. (We use the terms *foreground* and *background* of a visualization to refer to two conceptual layers that display the selected and unselected nodes of a multi-level graph layout.) When a coarsened node is decomposed into nodes with finer resolutions, should the non-qualified leaf nodes stay in the foreground or fade into the background?

[1,2,3,5,6] P.O. Box 999, Richland, WA 99352. Email: {firstname, lastname}@pnl.gov.

[4] Suite: 6513, 9800 Savage Road, Fort Meade, MD 20755. Email: rohrer@acm.org

IEEE Symposium on Visual Analytics Science and Technology
October 12 - 13, Atlantic City, New Jersey, USA
978-1-4244-5283-5/09/$25.00 ©2009 IEEE

Challenges like these, and more, require not just a mathematical inquiry into the structure of the graph but also a translation of real-world context into an empirical and analytical journey that facilitates logical reasoning and stimulates critical thinking. The design and implementation of GreenHornet incorporate a wealth of insight and practicality from career analysts, who seek an optimal balance between tractability and usability of the tool. We do not suggest that our model design is the only viable solution to the problem, but instead is one of many potential solutions and one that has been tested and proven to work in a real-world environment.

2 Related Work

Like many visual analytics studies, the research and development of GreenHornet involves a number of interrelated topics and technologies. This section describes some of the related work.

Visualizing graphs and hierarchies has been a major research topic within the data visualization community since its conception in the early 1990s. The survey paper by Herman et al. [14] represents the most complete literature review up to 2000. The annual IEEE Information Visualization Conference (IEEE InfoVis) [16] and IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST) [17] continue to produce new results on various topics of graph visualization. The proceedings of the annual Graph Drawing Symposia [11], now in its seventeenth year, provide a wealth of information on the cutting-edge technology.

Layout-wise, both Di Battista et al. [6] and Sugiyama [22] present a number of layout classes for a graph. Newer graph layout techniques that use diverse computation and heuristic algorithms for different applications [4], [9], [18] continue to emerge. Detailed discussion of individual layouts is out of scope for this paper. While a node-link layout still represents the most popular approach to visualize a very large graph, some have used a matrix-based layout [6], [22] to accomplish different objectives of a very large graph visualization. Elsewhere, Aris and Shneiderman et al. [2] introduce a semantic substrates approach, which partitions graphs into smaller, more comprehensible ones that align to specific user tasks. Chen [5] uses the concept of *betweenness* to guide the exploration of scientific networks.

There is an ongoing community effort to speed up the drawing process by developing adaptive algorithms with complexity of $O(n^2)$ and beyond. Notable work in this area has been presented by Eades and Feng [7] and later Walshaw [24]. In addition to the cutting-edge visualization algorithms, the two papers provide resourceful clues and ideas for further improvements on their designs. Their references sections also present a wealth of information covering topics from graph partitioning to Laplacian Eigenvector computation. The design of GreenHornet is based partly on our prior multi-level graph visualization work on GreenMax [26] that addresses the challenges of navigating a large small-world graph progressively in interactive time.

Additionally, there are a number of hybrid techniques that use multiple visualization or interaction design approaches, and share a degree of similarity with the design of GreenHornet. Notable examples are van Ham and van Wijk [23] that uses both semantic and geometric distortions, Gansner et al. [8] that uses a customized fisheye lens, and Henry et al. [10] that integrate both matrix and node-link techniques into one visualization.

In graph analytics, the bottom-up approach, which builds an analytical graph by linking and association, remains the most popular choice in many commercial and academic tools such as Analyst's Notebook [15] and Jigsaw [21]. The top-down approach, which visualizes the entire graph and then navigates the details, is less widely known in the visual analytics community. One of the recent graph analytics implementations that support the top-down analytical approach is GreenMax [26].

GreenHornet combines the concepts of data query with graph visualization into one analytical platform. The idea of allowing users to interrogate both the selected and unselected nodes in different areas with different levels of detail using a seamlessly integrated visualization is less common in graph visualization literature. Many graph analytics tool, such as Starlight [20] and Jigsaw [21], provide a strong database query capability, but often applied to the foreground area (i.e., on the selected nodes) only. GreenHornet allows analysts to cross analyze both the foreground and background concurrently so that they can detect the expected in the foreground and discover the unexpected in the background.

3 Synthetic Data and Threat Stream Generator

To evaluate GreenHornet and demonstrate its capabilities in this paper, we created synthetic graphs using the Threat Stream Generator (TSG) [25] developed at the National Visualization and Analytics Center™ (NVAC™) [19]. TSG synthetic datasets have been used extensively to evaluate visual analytics applications, particularly when operational data is difficult for researchers to obtain; they have been used in the IEEE VAST symposium contests [13] since 2006. We select TSG because of its ability to easily define and generate multivariate data and to produce large quantities of data. TSG also provides flexibility in generating data according to various statistical distributions or user-defined rules.

For GreenHornet testing, we generated datasets ranging in size from 5,000 to 1.8 million records. The datasets are fictitious collaboration graphs, containing invented publications by computer science researchers. The dataset schema includes record fields such as primary and contributing authors, publication IDs, reference dates, publication types, and topic categories. The collaboration graph was generated using a simple scale-free, power-law form, featuring growth and preferential attachment [3]. We seeded our networks with five nodes and arbitrarily added initial links so that each author began with at least one collaborator. New authors were added one at a time, and collaborations were created proportional to the number of collaborators existing authors already had, following the probability of connection $p_i = k_i / \Sigma k_j$ where k_i is the degree of node i. When a graph had been generated of the appropriate size, we could then manually modify records and connections to implant interesting anomalies, such as groups of authors featuring abnormally high connectivity. Figure 2 shows a portion of a coarse view of a typical synthetic graph generated by TSG. The unconnected nodes and sub-graphs are plotted surrounding the connected graph in the center of the figure. The labels show the identities and contents of the graph nodes.

In addition to synthetic data examples, the paper also presents two use scenarios that involve real-life bioinformatics and web crawler graph data in Sections 6.2 and 6.3.

4 Top-Down, Bottom-Up, and Middle-Out

We compare and contrast the strengths and weaknesses of the top-down and bottom-up graph analytics designs and assert that the middle-out approach, as implemented in GreenHornet, is a promising and viable alternative to the two.

A top-down graph analytics approach often starts with a power overview of the entire graph (i.e., the whole) and then directs attention to the local details (i.e., the parts) through different means of interactive navigation. It does require some computational overhead to draw the initial large graph but does not need to create the multi-level hierarchy. Its biggest strength is

the whole graph visualization that provides, among other things, a reference for the follow-on interactions. But try as it may, it is not always possible to produce an effective layout of a large graph.

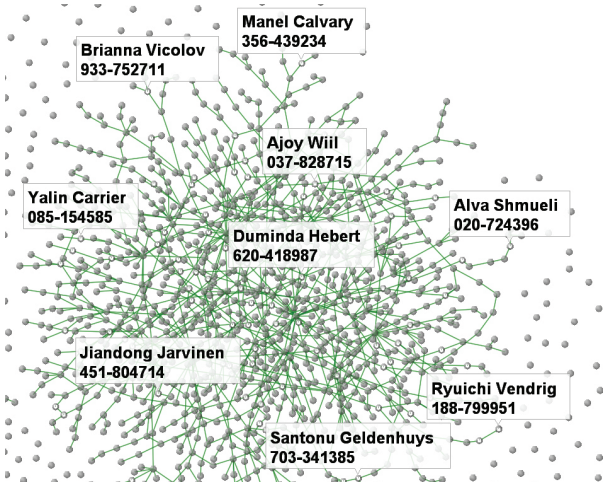

Figure 2: A coarse view of a typical synthetic graph generated by TSG. The labels show the identities and contents of the nodes.

A bottom-up approach frequently starts with seed nodes and/or links and then builds up the details through linkage and association. Compared to the top-down approach, the bottom-up approach requires a smaller visualization footprint and much less computational effort. The approach relies on the analysts to exploit the graph structure through questions and connections, and thus it is particularly effective at "detecting the expected." However, the approach may suffer when analysts fail to see beyond the horizon (because of the lack of an overall view) and is thus not an ideal approach for "discovering the unexpected."

A middle-out approach, as implemented in GreenHornet, primarily starts out with a coarsened view of the graph as illustrated in Figure 1. Because the entire approach is built on a multi-level hierarchy as described in Section 5.1, the analysts can capriciously and concurrently access both finer and coarser details of the underlying graph. As described later, the very flexible middle-out approach is designed to integrate the behaviors of both the top-down and bottom-up approaches (i.e., can perform like both of them) with additional features such as interactive cross-zooming to exploit the vast middle-ground of the graph hierarchy.

5 GREENHORNET

We continue our discussion on the design and development of GreenHornet. In particular, we explain the rationale behind the design of the three main analytical features (i.e., multi-level, middle-out, and cross-zooming) and address some of the engineering challenges ranging from effective user interface to change blindness problems of GreenHornet. Figure 3 shows an overview of our graph analytics model. Details of the individual model components are described in the following sections.

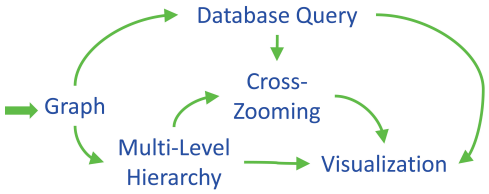

Figure 3: An overview of our multi-level, middle-out, cross-zooming graph analytics model.

5.1 Multi-Level Graph Hierarchy

Our analytical journey starts with a multi-level graph hierarchy. Given a graph G, GreenHornet progressively generates a hierarchy of increasingly coarse layouts G_{n-1}, … G_i, ... G_1, G_o that represent G in different resolutions. Many coarsening algorithms can be considered potential candidates for this task. We choose a customized algorithm that we previously developed for GreenMax [26], which is an interactive graph navigation tool for very large small-world graphs. GreenHornet stores the entire graph hierarchy in a file so it only has to be computed once.

A main requirement of a multi-level coarsening algorithm is to retain the most important structural features at each level. Reducing too many nodes at a time (and thus a shallower hierarchy) may cause a loss of too much detail to support an effective recovery later. Reducing too few (and thus a deeper hierarchy), on the other hand, may lead to unnecessary computation for many fairly similar graphs in the hierarchy. Our algorithm uses a coarsening approach known as *matching* that maintains an approximate 50% reduction rate at each level. This is done by merging the nodes with the least number of connections to the nodes they are connected with. The merging process continues until the 50% reduction rate is reached. The new coarser graph is guaranteed to have no less than half the graph nodes of the previous (finer) one.

Because GreenHornet is required to handle both connected and disconnected graphs (i.e., graphs with isolated sub-graphs), part of the matching strategy is to allow isolated nodes to be merged. GreenHornet always attempts to pair the isolated nodes with the closest neighbors in the current layout in order to minimize the distraction caused by the animation. Figure 4 shows an example of a multi-level graph hierarchy generated using our algorithm. Notice the "shape" of the graph remains intact in all three visualization levels.

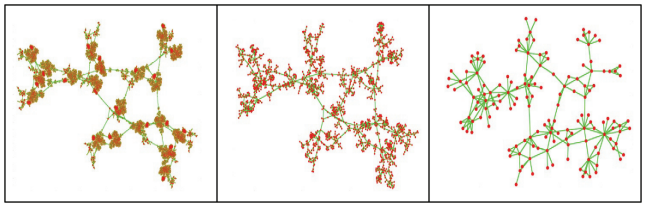

Figure 4: An example of a multi-level graph hierarchy. The three graphs (from left to right) represent the 1st, 5th, and 10th levels of the increasingly coarse graph hierarchy.

We have previously reported the GreenMax algorithm and implementation results in [26]. Computationally, it is shown that the total complexity of our coarsening algorithm remains in the order of $O(n)$. Interested readers are directed to work on GreenMax for more information.

5.2 A Guaranteed Visibility Initial View

After the multi-level graph hierarchy is generated, GreenHornet selects one of the coarsened graphs from the multi-level hierarchy based on the size of the display window and then draws the graph using a force-direct layout. The decision is a function of data size and number of available pixels. The motivation behind the selection of a coarsened layout (instead of a full-resolution one) is to ensure the graph is drawn with clarity and conciseness, given the size of the graph and the number of display pixels available. While seeking an ideal level for visualization is fundamentally elusive, users can interactively override GreenHornet's recommendation and start with a finer or coarser view. Based on the initial view of the graph, the analysts can now interrogate the graph by querying the database and analyzing the results.

The midway entry heuristic turns out to be a successful design and well-received by our analysts. Particularly when dealing with large graphs, it often means the difference between being able to see a glimpse of the graph or nothing at all. The relaxation of the requirement of drawing an entire graph also allows GreenHornet to tackle much larger graph problems.

5.3 Database Query and Interactive Inquiry

Database query plays a significant role in the graph analytics cycle that GreenHornet supports. Analysts can bring graph entities to their attention (to the foreground) by 1) using a query panel such as one shown in Figure 5 to aggregately select or filter information or 2) clicking individual nodes to bring them into focus.

The query panel shown in Figure 5 is customized for the TSG-generated graph data. In general, GreenHornet allows analysts to conduct a context-sensitive keyword search at the top of the panel, time-period search using the two sliders right beneath the keyword search widgets, and filter binary attributes of the databases at the bottom of the panel.

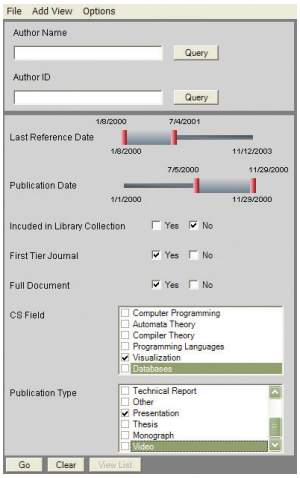

Figure 5: A data query panel for the scientific collaboration graph.

5.4 Visualization

We define key terms associated with our visualization design, highlight specific visualization features of GreenHornet, and explain the rationale behind our decision that resolves an inherent visualization ambiguity of the multi-level graph approximations.

Nodes of a coarsened graph are often compressed. We use the term *super-nodes* to refer to these compressed-nodes. A super-node of any coarsened graph may contain one or more super-nodes at finer levels, leave-nodes, and disconnected nodes.

We strive to refrain from over-applying different colors and icons to show different types of fine details in one visualization. Understanding that there is a major difference between a super-node and a leaf-node when we display the query results, we standardize a color-icon scheme to visualize the results as follows:

- A super-node is represented by a solid circle, whereas a leaf-node is represented by an empty circle.
- Nodes that satisfy the database query are painted in opaque red and conceptually placed in the foreground; otherwise, they are painted in semi-transparent gray and conceptually placed in the background.
- A super-node that contains both qualified and non-qualified nodes will be considered by GreenHornet as a qualified super-node. It is painted solid red and placed in the foreground of the visualization.

The design decision described in the last bullet was made based on the need to strike a balance between brevity and completeness of the multi-level visualization design. The described visualization has already provided a significant amount of information for analysts to grasp the essential structure and locality of the query results. Users can systematically visit individual nodes and drill down to the details instead of visualizing all the multi-level details all at once. In other words, the purpose of a solid red node in GreenHornet is not only to indicate a qualified node of a graph visualization but also to serve as a red flag that calls for further attention during a critical graph analytics journey.

5.5 Interactive Cross-Zooming

Throughout this paper, the term *cross-zooming* refers to the simultaneous, concurrent, sequential, or separate zooming of graph details that stretches across both the foreground/background boundary as well as the multi-level hierarchical boundary. All GreenHornet's zooming operations are done using animation to ensure proper understanding of how the graph evolves.

The real analytical challenge is to provide a *seamless* visualization at all times and hide the boundary artifact patterns that often distract our analysts. We described in Section 5.4 the visualization ambiguity caused by the graph approximations. The same issue has evolved into a larger problem when dealing with cross-zooming.

5.5.1 General Zooming Design Approach

The most basic zooming capability of GreenHornet is to show the multi-level graph hierarchy of the entire graph one level at a time. This multi-level browsing and navigation capability was previously implemented in our GreenMax [26] tool.

Additionally, users can zoom the entire foreground or background of GreenHornet separately. One scenario is to reduce the background details and free up space for foreground details.

An extension to the above zooming process is to adopt a heuristic to tie the foreground and background behaviors together. The idea is to allow GreenHornet to balance the amount of nodes to be displayed by automatically reducing the background details when the foreground details reach a certain limit.

Finally, users can also zoom individual nodes for finer details one at a time. We have experimented with a number of user interfaces, which we will discuss in Section 5.7, that satisfy the needs of different analytical patterns and habits.

5.5.2 Cross-Zooming Design Exceptions and Solutions

The general zooming design approach described in Section 5.5.1 works in all cases with the following exceptions.

When users increase the foreground resolution, GreenHornet splits all selected nodes into the nodes that are on the next level in the coarsening tree. Any of the child nodes that contain a selected node are considered selected, and therefore part of the foreground. If any of the new child nodes are directly connected to a super-node that is selected, they are considered part of the foreground, but not selected if they do not contain a selected node. If any of the new child nodes are neither selected nor connected directly to a foreground node (i.e., it is an island), then it is considered part of the background. If an individual selected super-node is clicked, this same process occurs, but for its child nodes only.

When users reduce the foreground resolution, GreenHornet first loops through all the visible super-nodes in the foreground and finds which ones are the least coarsened on the coarsening tree. Only the foreground nodes on the current least coarsened level are retracted. This is done so that a super-node that is near the top of the coarsening tree does not accidentally retract half the graph with it when only a small change is made in the slider.

When foreground nodes retract, they can retract background nodes with them, but background nodes will never retract a foreground node into them. Because of this, if there are some nodes in the foreground and some in the background, it will get to a point when GreenHornet can no longer retract any more background nodes, even if some of them still appear in the view. This is because GreenHornet would have to retract some of the

foreground nodes to get rid of them (because they are connected through the coarsening tree).

The design criteria of these zooming exceptions are more driven by user preferences than influenced by graph or logical theories. The key factor is to maintain behavioral consistency among the zooming operations and, at the same time, provide the necessary robustness needed for the analytical tasks.

5.6 Change Blindness Visualization

The overarching intent of separating the foreground and background visualization layers in GreenHornet is to allow the analysts to maintain focus on a selected set of graph entities when dealing with a large amount of graph information. It turns out that the foreground/background feature is not nearly enough to keep track of the changes during a graph analytics discourse because of a visual phenomenon known as change blindness [12].

When an earlier version of GreenHornet was presented to our analysts for evaluation, they soon expressed a strong desire for an additional feature that marks down the locations of the most recent changes in the visualization, which may be brought by zooming and/or querying. When a large number of similar visualization entries are displayed on screen, our analysts sometimes experience change blindness and fail to detect all the changes by looking at the still visualization after the change animation is completed.

Change blindness in visualization can be caused by a number of cognitive and perception factors. But the amount of changes at any one time, in our case, plays a significant role for the visual phenomenon. Following the same design principle of refraining from over-applying different colors and icons to show different types of fine details in one visualization, we decide to paint a light yellow halo icon behind every node that has just expanded or retracted. The light yellow color was selected to easily blend in with the rest of the visualization. We have also tried motion, such as jittering vibration, but found that extremely distracting in our environment. The halo icons fade away right before another set of new changes (and the corresponding halo icons) arrives. Figures 6d and 6e depict the use of the change blindness detection icons.

5.7 Human-Visualization Interaction

GreenHornet provides a fully interactive response time [1] when navigating a large small-world graph with hundreds of thousands of nodes on a desktop computer. In other words, the animation of the visualization would be completely smooth instead of flickering, and the analysts would not need to wait for longer than interactive response time for the results to show up on screen.

The above interactive response time goal can be accomplished solely through computational and algorithmic means by and within the computer. However, to ensure that analysts take full advantage of this response time guarantee and subsequently enrich their analytics experience, a carefully arranged interface will be required to facilitate the communication between the computer, the visualization, and the human users.

Bear in mind that our analysts are free to access any graph nodes in the visualization, in both foreground and background areas, and across the multi-level hierarchical boundaries. Our goal is to minimize the amount of hand-movement and reduce analyst fatigue when they carry out all the above operations for an extended period of time.

Another design restriction we have to deal with is that we can only use existing desktop hardware, which includes a keyboard and mouse. So our interactive design options are narrowed down to 1) mouse clicking on button widgets, 2) mouse dragging on slider widgets, and 3) mouse scrolling wheel.

One of the most frequent uses of GreenHornet is to support the zooming of both the foreground and background at the same time. Instead of using one slider widget for the foreground and another one for the background, we chose a slider to control the foreground and the mouse scrolling wheel to control the background. Our design decision is based on the observation that the analysts will need more precision to manipulate the foreground resolution. A slider, which can interactively show precise readings, does a much better job than a mouse scrolling button. On the other hand, the background often requires attention when the analysts need more visualization spaces for the fine foreground details. The mouse scrolling button can effectively deliver that kind of prompt action.

Finally, the main reason that we chose a combination of a slider and scroll mouse to navigate the multi-level hierarchy is to allow the analysts to control both foreground and background by moving the slider and scrolling the mouse button at the same time without panning the mouse (i.e., switching from one widget to the other one).

5.8 System Engineering

The front-end of GreenHornet is implemented using Microsoft C# with .net framework and Microsoft DirectX 9 graphics. The underlying graph analytics and computation library is implemented using Microsoft C++ to ensure optimal performance of the library. No commercial tools or libraries are used in the GreenHornet implementation.

6 Demo Examples

We present three demo examples to illustrate the ideas and the potential use of GreenHornet. The first example uses a medium-sized synthetic graph with about 50K nodes generated by TSG, as described in Section 3. The other two involve a protein-network and a web-crawler graph. Our analysts normally use GreenHornet to analyze large graphs with hundreds of thousands to a million nodes on a desktop computer. But the size of the underlying data, in this case, has less to do with the messages that we want to convey by our demo examples.

6.1 TSG Database

In this example, we first attempt to use the top-down approach to visualize the TSG-generated graph in full detail. While GreenHornet is capable of drawing the graph in Figure 6a, none of the graph details are visible from the figure. We then switch to the bottom-up approach and start with a query to find the relationships of all the authors under the "visualization" and "presentation" categories in the databases. The results are shown in Figure 6b with the red dots indicating the selected nodes. The background view, in this case, offers very little help to show the associations of the selected (red) nodes. Clearly, neither approach performs well in this example.

We then apply GreenHornet's middle-out approach to provide a coarse view of the graph in Figure 6c. While we can see more details of the coarse graph, most of the unselected (gray) nodes in the background do not provide critical information in this particular view. We further reduce the background resolution and simultaneously increase the foreground resolution of the selected (red) nodes all the way to the leaf-node level. Figure 6d finally reveals the clear connections among the red nodes. Now we turn on the labels of the selected nodes and reveal their identities in Figure 6e. We can also examine the link data, which describes the publication information that ties the two red nodes together.

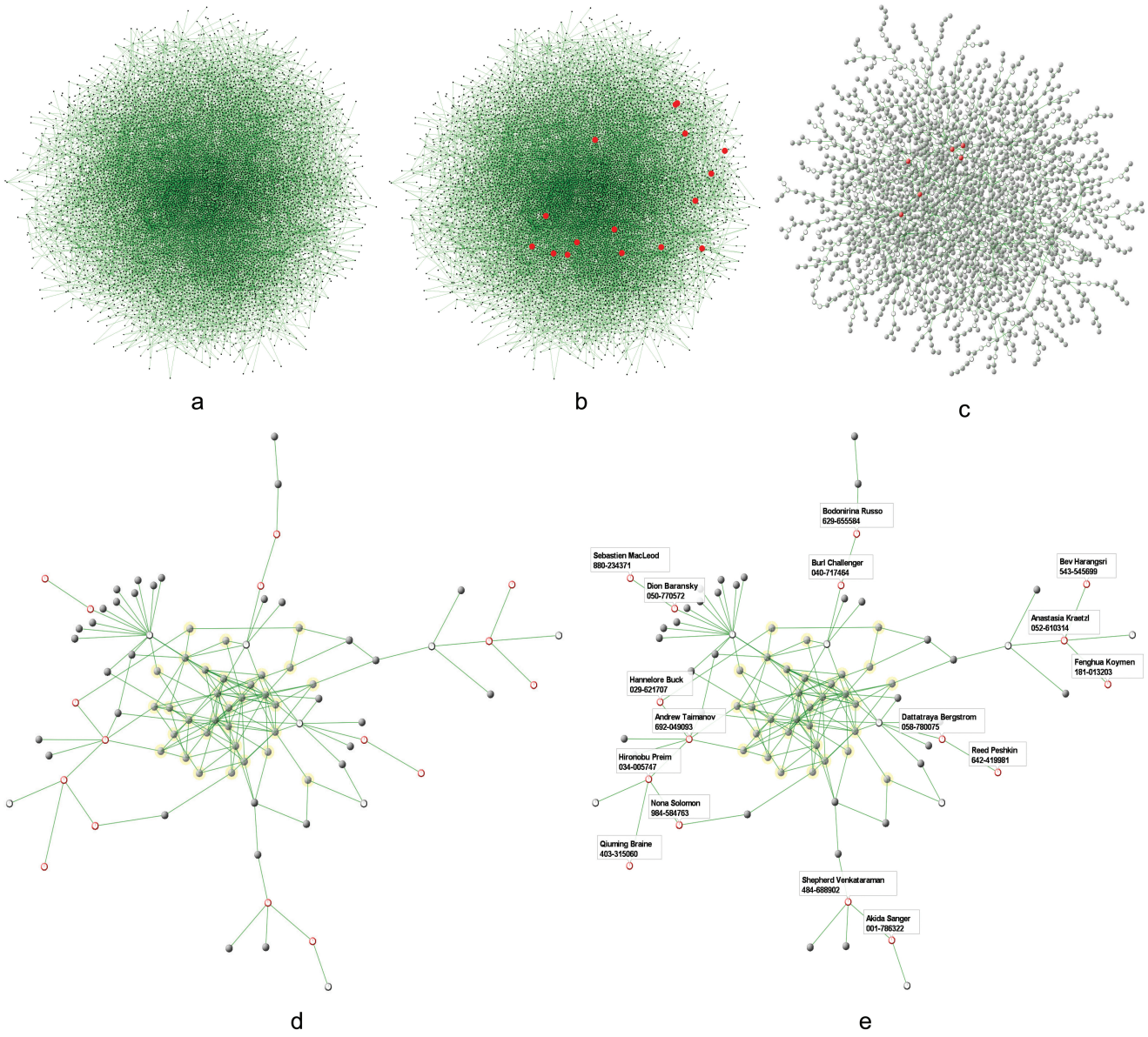

Figure 6: a) A full-resolution view of the graph. b) Query results on "visualization" and "presentation" are shown in red. c) A coarse view of the graph. d) Increase the foreground resolution and decrease the background resolution. e) Turn on the labels of all the foreground nodes for inspection.

The important message we want to convey from this example is that GreenHornet provides a lightweight, scalable graph analytics solution for analysts to harness their large graph data without losing the capabilities and benefits of graph visualization.

6.2 Bioinformatics Network

In the second example, we invited a bioinformaticist to use GreenHornet to analyze a bioinformatics graph. The targeted dataset is a genome-level protein network of an entire set of 4,242 proteins encoded by the purple, nonsulfur bacterium *Rhodobacter sphaeroides*. As expected, the brute-force visualization approach in Figure 7a does not provide a lot of useful information for the analysis. Then during the course of cross-zooming the graph, she noticed the emergence of four sets of leave-nodes as annotated by the arrows in Figure 7b. The fact that these nodes reach leaf-level (i.e., empty circle, not solid nodes) all about the same time indicates a starburst structure in each of the four clusters. Based on this observation, she brought the clusters to the foreground and at the same time reduced the resolution of the background in Figure 7c. Upon further analysis of the clusters and cross-examination of them with the other bioinformatics tools, she found that the four clusters contained different types of proteins that power the transport of material across the cellular membranes using the energy molecule ATP.

The important message delivered by this example is that the animation of the multi-level cross-zooming process, as implemented in GreenHornet, can be very useful to discover structures that could otherwise be difficult to detect from a still visualization.

6.3 Web Crawler

In the final example, a cybersecurity analyst investigated network information collected from a web crawler. The relatively small network is shown in Figure 8a. The analyst used the database

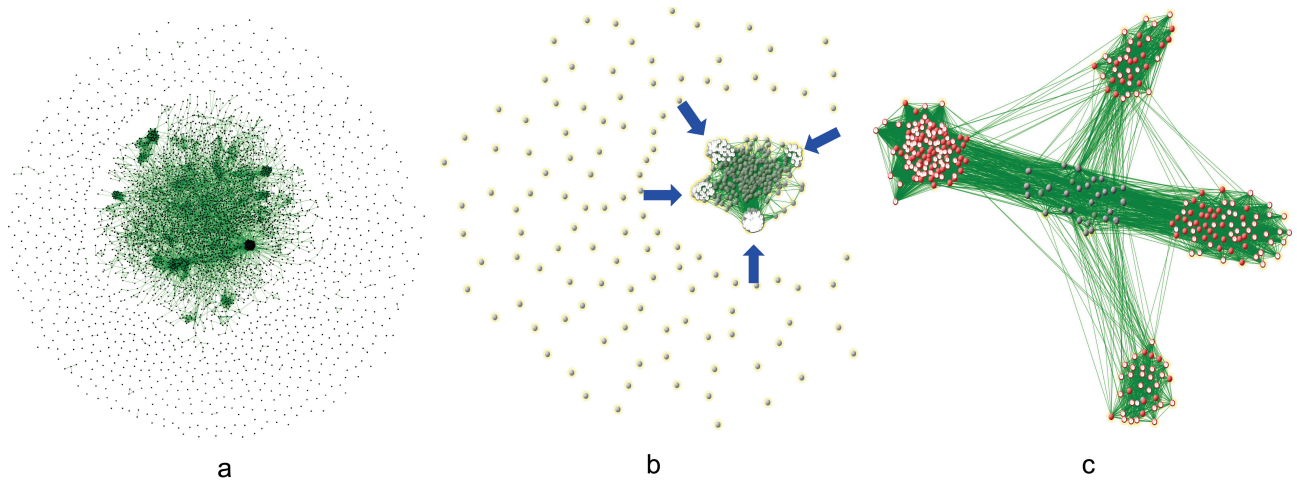

Figure 7: a) A full resolution view of the protein network. b) A coarsened view of the same network, notice the appearance of the four node clusters at this resolution level. c) The clusters are moved to the foreground of the visualization for examination.

panel to query the graph, and the results are shown in Figure 8b. Unfortunately, the labels of the selected nodes are too close to each other for effective comparison and interpretation. To address the graph labeling problem, the analyst reduces the resolution of the background nodes and frees up enough space for the foreground visualization to spread out, as depicted in Figure 8c. The analyst can now examine the link structure of the nodes and read the information provided by the corresponding labels.

The message here is clear: we do not need a very large graph to create a difficult graph analytics situation. Many of these problems, however, can be addressed by our multi-level middle-out cross-zooming approach with only a few keystrokes or mouse clicks.

7 Strengths and Weaknesses of GreenHornet

We summarize the strengths and weaknesses of our graph analytics model and GreenHornet in the context of a real-world application.

Our middle-out graph analytics model is flexible. GreenHornet is designed to integrate the strengths of both top-down and bottom-up approaches with additional features such as the multi-level graph hierarchy to explore the middle-ground details. For example, users can take the bottom-up approach to analyze a graph together with a graph overview usually available only to the top-down approach.

Our middle-out model is powerful. It provides a guaranteed visibility initial view of graphs with up to a million nodes through the use of a very fast and effective coarsening algorithm. The relaxation of the requirement of drawing an entire graph allows GreenHornet to tackle a much larger graph problem.

The dual-resolution design of the foreground and background layers is both unique and innovative. Users can interactively and dynamically balance the need for foreground details without losing the overall perspective of the background information.

The information displayed in the foreground is the result of either a *known-known* or *known-unknown* query. The prefix "known" refers to the known-contents or known-structures queried by the users. The other side of the foreground visualization, i.e., the background layer, becomes the territory of either *unknown-known* or *unknown-unknown*. Unlike the bottom-up approach that works only on the selected nodes (i.e., the foreground), GreenHornet supports the interrogation of both foreground and background layers and thus enables the "detection of the expected and the discovery of the unexpected."

While we are pleased with the design of our middle-out graph analytics model, the GreenHornet implementation itself is not without weaknesses. First, GreenHornet currently only supports a

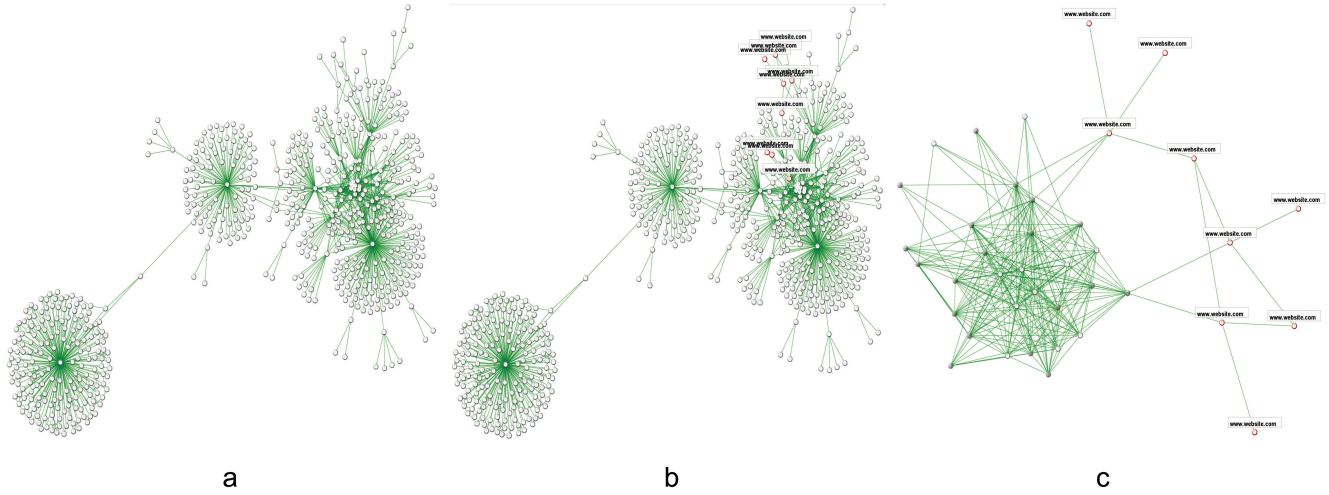

Figure 8: a) A full resolution view of the web-crawler graph. b) Query results with overlapping labels. c) Reduce background resolution and allow the foreground information to spread out for inspecting.

very basic graph link information query. A more powerful link query capability would require a new visualization design and possibly a new definition of a multi-level graph hierarchy, which focuses mainly on link merging.

Second, although GreenHornet is good at what it is designed for, it lacks the feature richness of prevailing tools such as Analyst's Notebook to make it a full-blown graph analytics tool.

Finally, the default 50% reduction rate of our multi-level graph hierarchy implementation, while theoretically and computationally sound, may be too rigid for some applications where the users need to access the very fine features between two hierarchical levels.

8 Future Work

We have at least two potential extensions of GreenHornet in the pipeline. The first one involves a similar middle-out concept but is being applied to a matrix-based visualization platform (instead of the graph-based approach we have presented in this paper). We are not debating the strengths and weaknesses between a matrix-based and a graph-based visualization here. Our goal instead is to allow the two very popular graph visualization techniques to complement each other and make GreenHornet an even stronger tool.

We have spent a significant amount of effort to identify the optimal human-visualization interaction that supports GreenHornet, as described in Section 5.7. Our current solution, however, is based on the restriction of using only mouse and keyboard. We envision that the users can take advantage of a more powerful interface, such as a two-handed control, that manages multiple focal-points (and analyzes different scenarios) simultaneously.

9 Conclusion

The paper introduces the concept of a multi-level middle-out cross-zooming model for very large graph analytics applications. We show that our model, as implemented in GreenHornet, is a promising and viable alternative to the more traditional bottom-up and top-down graph analytics models. The paper also addresses a number of general visual analytics issues, such as change blindness and human-visualization interface, surrounding the development of a graph analytics tool. The three demo examples illustrate the ideas and the potential use of the tool. We will continue the development of GreenHornet with new features such as those suggested in the future work discussion.

Acknowledgements

This work has been supported by the United States Federal Government. The Pacific Northwest National Laboratory is managed for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC05-76RL01830.

References

[1] Marshall Abrams, "Measurement of Interactive Response Time," *ACM SIGCOMM Computer Communication Review*, vol. 9, issue 1, pages 10-24, ACM Press, January 1979.

[2] Aleks Aris and Ben Shneiderman, "Designing Semantic Substrates for Visual Network Exploration," *Information Visualization*, vol. 4, no. 6, pages 1-20, Winter 2007.

[3] Rěka Albert and Albert-Lászlo´ Barabási, "Statistical Mechanics of Complex Networks", *Reviews of Modern Physics*, vol.74, pages 47-97, January 2002.

[4] Ulrik Brandes and Christian Pick, "An Experimental Study on Distance-Based Graph Drawing," *Proceedings 16th International Symposium on Graph Drawing (GD2008)*, pages 218-229, Springer-Verlag, 2009.

[5] Chaomei Chen, "The Centrality of Pivotal Points in the Evolution of Scientific Networks," *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI) 2005*, pages 98-105, ACM Press, 2005.

[6] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.

[7] Peter Eades and Qing-Wen Feng, "Multilevel Visualization of Clustered Graphs," *Proceedings of the Symposium on Graph Drawing, Lecture Notes in Computer Science*, vol. 1190, pages 101-112, Springer-Verlag, 1996.

[8] Emden R. Gansner, Yehuda Koren, Stephen C. North, "Topological Fisheye Views for Visualizing Large Graphs," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 11, no. 4, pages 457-468, IEEE CS Press, 2005.

[9] Stefan Hachul and Michael Jünger, **"**Large-Graph Layout Algorithms at Work: An Experimental Study." *Journal of Graph Algorithms and Applications*, vol. 11, no. 2, pages 345-369, 2007.

[10] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin, "NodeTrix: A Hybrid Visualization of Social Networks," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 13, no. 6, pages 1302-1309, IEEE CS Press, 2007.

[11] GD2009, 17th International Symposium on Graph Drawing 2009, http://facweb.cs.depaul.edu/gd2009/cfp.asp.

[12] J. Grimes, "On the failure to detect changes in scenes across saccades", in K. Akins, editor, *Vancouver Studies in Cognitive Science: Vol. 2: Perception*, pages. 89-110, New York: Oxford University Press, 1996.

[13] Georges Grinstein, Catherine Plaisant, Sharon Laskowski, Theresa O'Connell, Jean Scholtz, and Mark Whiting, "VAST 2008 Challenge: Introducing Mini-Challenges," *Proceedings IEEE Symposium on Visual Analytics Science and Technology (VAST) 2008*, pages 195-196, IEEE CS Press, 19-24 October 2008.

[14] Ivan Herman, Guy Melancon, and M. Scott Marshall, "Graph Visualization and Navigation in Information Visualization: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pages 24-43, IEEE CS Press, 2000.

[15] i2 Analyst's Notebook, http://www.i2inc.com.

[16] IEEE Information Visualization Conference (IEEE InfoVis), http://vis.computer.org/VisWeek2009/infovis/.

[17] IEEE Symposium on Visual Analytics Science and Technology (VAST) 2009, http://vis.computer.org/VisWeek2009/vast.

[18] Andreas Noack, "Energy Models for Graph Clustering," *Journal of Graph Algorithms and Applications*, vol. 11, no. 2, pages 453-480, 2007.

[19] NVAC, National Visualization and Analytics Center, http://nvac.pnl.gov.

[20] Starlight, http://starlight.pnl.gov.

[21] John Stasko, Carsten Görg, and Zhicheng Liu, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization," *Information Visualization*, vol. 7, no. 2, pp. 118-132, Palgrave Magellan, 2008.

[22] Kozo Sugiyama, *Graph Drawing and Applications*, World Scientific Publishing, 2002.

[23] Frank van Ham and Jarke van Wijk, "Interactive Visualization of Small World Graphs," *Proceedings IEEE Symposium on Information Visualization (InfoVis) 2004*, pages 199-206, IEEE CS Press, 2004.

[24] Chris Walshaw, "A Multilevel Algorithm for Force-Directed Graph-Drawing," *Journal of Graph Algorithms and Application*s, vol. 7, no. 3, pages 253-285, 2003.

[25] Mark A. Whiting, Wendy Cowley, Jereme Haack, Doug Love, Stephen Tratz, Caroline Varley, and Kim Wiessner, "Threat Stream Data Generator: Creating the Known Unknowns for Test and Evaluation of Visual Analytics Tools," *Proceedings 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods For information Visualization, BELIV '06*, pages 1-3, ACM Press, May 23 2006.

[26] Pak Chung Wong, Harlan Foote, Patrick Mackey, George Chin Jr., Heidi Sofia, and Jim Thomas, "A Dynamic Multiscale Magnifying Tool for Exploring Large Sparse Graphs," *Information Visualization*, vol. 7, no. 2, pages 105-117, Palgrave Macmillan, June 2008.