

MULTI-USER NON-LINEAR ADAPTIVE MAGNIFICATION FOR SATELLITE IMAGERY AND GRAPH NETWORKS

By

Sean Christopher Kim

A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
Major Subject: COMPUTER SCIENCE

Examining Committee:

Barbara Cutler, Thesis Adviser

Charles Stewart, Member

David Mendonça, Member

Rensselaer Polytechnic Institute
Troy, New York

July 2014
(For Graduation August 2014)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

When dealing with emergency situations in the wake of a disaster where the infrastructure of a region is damaged, a wide range of expertise is needed to quickly and efficiently solve issues. A prototype system for training emergency response personnel was designed to foster a collaborative environment through a single visualization with multiple users. The traditional methods of providing more detail in geographic data when dealing with a single user do not work in situations requiring interactions from multiple users, as this requires the loss of detail elsewhere within the fixed resolution of a display. The data being displayed in this simulation consists of a graph network of nodes and edges on top of a underlying series of satellite images. The nodes and edges are drawn over corresponding physical locations on the satellite images. When interacting with this system, it is often helpful to zoom in and increase the level of detail for two primary reasons: it is difficult to distinguish geographic features at low resolution, and subsequently, it is difficult to distinguish between nodes when the screen distance between them is relatively small.

This thesis work presents a method for providing a focus plus context solution to this system, allowing for continuous visual information with a minimal amount of distortion. This method creates circular areas of high magnification which gradually fall off to a base level surrounding individual cursors. These different regions achieve the original goals of zooming by providing a magnified look at the satellite images and increasing the screen distance between nodes. By having these regions centered on cursors, multiple users can view data on a shared display with their own degree of magnification while still retaining the ability to view the majority of the surrounding data.

Contributions of this thesis include efficient implementations of rendering a graph network and text onto an underlying layer of satellite images, algorithms to perform the transformations of edges, vertices, and image data for rendering, and a preliminary feedback on the usability of these changes along with suggestions for a formal user study to be conducted as future work.

CHAPTER 1

INTRODUCTION

Animations of human characters are used heavily in video games, movies, and other fields. Creation of such animations is largely done by hand by artists using keyframing. In a keyframe animation, certain “key” parts of the animated sequence are specified, with the remaining frames filled in, or “tweened” using an automated interpolation method or manual frame addition. For 2D animation, this occurs as a series of images which are played back in order to produce the animation. In 3D, keyframe animations are performed on a 3D model.

3D models are described as a mesh, a collection of primitive polygons (i.e. quadrilaterals or triangles) which are stored as vertices. This mesh describes what is drawn, including any texture, color, and other material information. Along with the mesh, a skeleton, or rig, is stored. The rig describes a heirarchical structure of bones and accompanying joints. Each vertex is given a series of weights describing the impact each joint has on its transformation. This allows many vertices, and therefore many polygons, to be transformed at once in organized groups, simplifying the problem of animating the model to a matter of transforming the skeleton in the desired manner. To animate this 3D model, an artist specifies keyframes of the animation by positioning the skeleton. The stored keyframes, instead of an image, are the transformations of each joint at this frame in the animation, which a rendering or game engine can interpolate between to produce the final animation.

Specifying these animation frames is work intensive, taking up significant time and resources to produce for a single character. Recent work in animation generation seeks to automate this process, replacing the manual process with a procedural one. Physics based simulations can be used to produce controllers for the skeleton, determining joint positions and rotations for keyframes automatically. Not only does this reduce the effort involved in the creation process, but this also provides a basis for dynamic interaction between a character’s animation and the environment, which is not possible with manual keyframe animations.

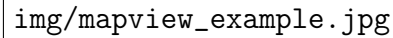


Figure 1.1: This example screenshot shows an example scenario of a residential node lacking water, by following the network back to the supply node, we can deduce that the overall problem is twofold, a lack of power to the water supply node and a damaged water pipe - both must be addressed to restore water to the demand node.

When expanded, each node provides its id and name. If it is either a supply or transshipment node, it has a corresponding status which indicates if it is currently supplying its resource to its edges. If a node requires a particular resource to function, it shows the current ratio of the supply and demand for that resource as a percentage. Edges show their current status: **damaged** or **status OK** as well as their ids and two corresponding node ids. If either a node or an edge is damaged, it is unable to transmit its resource.

When interacting with the system, users can toggle different parts of the display to simplify the data being presented. The underlying road network and satellite

images can be turned off, as well as the nodes and edges of a particular resource. Users can also increase the level of detail of the application, effectively zooming in and out on the satellite images. This allows the users to see the a more detailed view of the satellite images and also reduces the number of graph elements on the screen.

This entire application is designed to work with a system developed by Tyler Sammann and Chris Stuetzle, and which allows multiple users to interact with a system with mouse cursors and laser pointers [?]. These various input devices are generalized as cursors which have a generalized set of gestures, allowing for collaboration on a single screen between multiple users. In the context of the viewer application, this functionality allows the users to expand and collapse the different graph elements, navigate to different areas of the map, change the global level of detail, and toggle the visibility of different elements as described above.

Systems that use multiple cursors are designed to foster collaborative problem solving and data exploration. Having multiple people interact with a system all at once allows people do explore different areas of the data set at the same time where normally a single person would control what everyone using the system sees. By having users work on a single shared screen instead of multiple displays, each individual user can more easily see what other people are looking at or modifying.

Previous demonstrations of this application revealed that the current functionality was lacking a number of features that would allow for greater information visibility. Essentially, users are unable to view a particular region in a higher level of detail without possibly disturbing what the other users are viewing. Figure ?? presents an example of this issue.

A system allowing for multiple areas of magnification when integrated with the previously mentioned multi-cursors could be useful in other contexts. If examining any sort of large graph based data set, being able to view different focal areas in higher detail would allow users to navigate data sets easier while still viewing the overall graph structure.



Figure 1.2: This diagram shows a screenshot with two cursors in different regions of the application. The red arrow cursor is in the top left corner and the blue arrow cursor is in the bottom right corner. If either cursor performs a global zoom on their location to get more detail, the other cursor is unable to view their data.

1.1 Focus Plus Context

Overview plus detail, zooming, and focus plus context are all methods within information visualization for displaying data. They each seek to present varying levels of data in a format that facilitates understanding. Overview plus detail visualizations are similar to physical maps of cities. The overall view of a city is shown on the big map, but a higher detail region of the city is shown in a separate, enclosed area (Figure ??). This sort of visualization is easy to create, and is most common for static media, as certain areas inherently require more detail than others when dealing with static data. Being able to interactively change the level of detail is the principle behind zooming visualizations (Figure ??). This method is well suited for a single user looking to view different levels of detail at different points in time. It is also relatively easy to implement and does not cause distortions in the data being displayed. Focus plus context systems are useful for systems where a user needs both a local and global view at the same time. Some loss of data can occur in these visualizations, but by using a single screen, the user does not need to mentally combine the data of a overview plus detail visualization. The first two formats separate data in two different ways: overview plus detail interfaces separate data spatially, while zooming separates data temporally. Focus plus context uses a visual effect to display continuous information on a single screen [?].

The goal of focus plus context interfaces is to reduce the cognitive load of managing multiple different views of a single system. Theoretically, this would improve user performance with regards to understanding and utilizing the data being presented. The foundation for the focus plus context interfaces was established by Furnas in 1986 [?]. He described a “generalized fisheye view” formula is described which calculates a user’s “degree of interest” (DOI) [?] (Equation ??). The degree of interest, given a data element x and focus y , is the *a priori interest* (API) in the data element minus the distance between the data element and the focus, D . The term fisheye in photography refers to a strong visual distortion which has an extremely wide angle. This photographic technique allows for more detail in the resulting image by introducing distortion. The “generalized fisheye view” states that objects closer to the focal point are more important than ones further away, paralleling the




Figure 1.3: This old map of Louisiana from 1853 showing a detailed view of a streets of New Orleans is an example of an overview plus detail visualization [?].

fact that in a fisheye visualization, objects close to the focal point are less distorted.

$$DOI(x, y) = API(x) - D(x, y) \quad (1.1)$$

Focus plus context systems often perform some sort of distortion to present the data. Further work by Furnas within focus plus context systems highlighted the crucial fact that distortion based visualizations achieve a seamless display by requiring the user to understand that data is being modified in some manner [?]. All such distortions perform some sort of filtering of the data, as the distortion functions are presented on media with finite limitations. As information gets further distorted, it is essentially filtered out, as it can no longer be discerned by the users. Distortion



img/zoom_interface.jpg

Figure 1.4: An image of a common zooming interface for map data, Google Maps [?].

causes geometric distortion among the data, altering the positions and shapes, while maintaining topological continuity. This property is potentially desirable, as our application features satellite images, which should remain topologically continuous [?]. By remaining topologically continuous, users would still be able to discern the relative distance between any two points on the map, as there are no regions of completely missing data.

Due to the nature of the data that we are working with, techniques that introduce some sort of distortion to achieve a focus plus context visualization are particularly relevant to the work presented in this thesis. Leung and Apperley provide an overview of such techniques [?]. Among the relevant applications are the Polyfocal Display [?], the Perspective Wall [?], and the Graphical Fisheye View [?]. These techniques are further explored in Chapter ?? . Leung and Apperley classifies

such techniques as magnification functions. The different functions can be further divided into techniques that have continuous magnification functions and those with noncontinuous piecewise functions. The different functions are unified into a single metaphor of a stretchable sheet on a rigid frame. By increasing the amount of data displayed in a single area, i.e. “stretching” the sheet to display information, other areas must “shrink” and display less data. The overall distortion effect that is then displayed is a result of the stretching and shrinking of the display. This stretchable rubber metaphor holds up even for views with multiple foci, though it is paramount that the balance between magnification and demagnification is maintained, otherwise the “frame” of the sheet would be deformed. At the core of [?] is the idea that all of these distortion techniques are similar, and are purely dependent upon a single magnification function. We will expand upon this idea of creating a magnification function for our visualization later on in Chapter ??.

1.2 Thesis Outline and Contributions

As mentioned earlier, the first draft of the infrastructure visualization program has the ability to recognize simultaneous input from multiple keyboards, computer mice, and lasers by individual participants, but only supported zooming on a global scale. This loss of information when performing a global zoom is a major hindrance to a system which should be used by multiple people at the same time. If diagnosing a central problem for discrete graph elements in different geographic regions, trying to get more information about the area around a node is impossible with the current model without disrupting the other users. The main goals in designing this new system were to provide users the ability to view different regions of data at a variable level of detail without altering a majority of the global context. Because this visualization may be viewed by people who are not directly interacting with the system, solutions that cause too much loss of global context should be avoided. If we change the global context, we run into the same issue of obscuring other regions of high importance.

The first step in creating a solution which met our needs was the design and implementation of a proof-of-concept system. An investigation of previous methods

of focus plus context visualizations was performed to formulate the design behind our solution, detailed in Chapter ???. The concepts learned from this system were then applied to the pre-existing application. The visualization required many changes to allow for the new feature to be implemented and a large bulk of time was spent on upgrading to modern OpenGL to also improve performance; these changes are detailed in Chapter ??. Two separate magnification functions were created to affect both the satellite imagery and the graph network. These functions, described in Chapter ??, cause the graph elements to stay in the same geographic location, and are easily adjustable for different parameters. The speed of the new system with regards to rendering and performing the magnification functions was measured and recorded in Chapter ??. Chapter ?? also includes an informal survey of untrained user responses was carried out to gauge the basic usability of the new magnification system. Finally, further improvements and changes to the system are discussed in Chapter ??.

References