Meteorological Visualization using MATLAB and a Gaming Engine

Joseph Kelzenberg
Student
Department of Electrical and Computer Engineering
St. Cloud State University
720 Fourth Ave. South ECC 211
St. Cloud, MN 56377
jmkelzenberg@stcloudstate.edu

Mark C. Petzold
Associate Professor
Department of Electrical and Computer Engineering
St. Cloud State University
720 Fourth Ave. South ECC 211
St. Cloud, MN 56377
mcpetzold@stcloudstate.edu

Abstract

The goal of this project is to advance the visualization of meteorological weather data, both model data and actual measurements. This would allow better representations of weather for research and instruction in meteorology. The way in which weather data is displayed has been mostly flat representations of 3D data. With the advent of 3D technology such as the Oculus Rift, 3D projector, and 3D television, it is possible to represent data in a truly 3D environment, which leads into a more realistic understanding of weather rather than a 2D understanding.

The most common form of data in Meteorology is the NetCDF format. MATLAB[1] is capable of working with NetCDF files. Using MATLAB we have created an interface that that allows the user to select the data that they want to display and then MATLAB is used as a preprocessor front end to manage the data, which leaves many possibilities for modifying the data. For instance, the professor can modify data to show difference in outcomes of a storm if the temperature or pressure is changed. The data is then sent over a TCP connection to Unity3D[2]. Unity3D, using scripts written in C#, uses the data to create a 3D model and display it through an Oculus Rift. Different types of data can be presented in various ways in Unity3D. Currently temperature data is shown using particle color and wind data is shown in particle movement. The user can then move through the environment with the XBOX 360 controller. We currently have a basic framework that can easily be adapted and increase its functionality. Because we are using a TCP protocol to communicate between MATLAB and Unity3D, the data can be selected and processed once, and then be made available to multiple users on different platforms.

The interface between MATLAB and Unity3D has been the biggest issue so far. We used the TCP protocol because of its ease of use and the ability to expand the number of computers the

interface in MATLAB can reach. Since the current software runs both MATLAB and Unity3D on one computer, other immediate possibilities for communication between the two platforms are JSON or XML files. A file could be utilized in future iterations for communication, but features such as quickly updating the data or two-way communication between MATLAB and Unity3D would be more complex when using a JSON or XML format. Having a TCP connection makes those processes easier, and would allow for direct updating and interaction between students and faculty in the visualization environment.

The way to write the data across the TCP connection is by bytes which makes reconstruction on the Unity3D side a bit more complex, as the data set has to be reassembled on the receiving end. This requires sizes and labels to be transmitted so that the data can be reassembled in Unity3D. The labels can be used as command codes in Unity3D, which allow us to select the way we reconstruct the data. We can tell it what the data set represents and use an appropriate visualization model for that data when it is rendered in the environment. Given the options available for data representation in Unity3D, there are many ways of representing data visually. In some cases, collections of data can be rendered using similar structures. For example, temperature and wind can be shown together through the movement of colored particles. Temperature can also be shown as relief maps, while wind by itself can be shown by either arrows or moving particles.

This project is based on the Unity3D game engine, so it is very easy to adapt it to different display technologies. These technologies can range from a standard video monitor and using a keyboard to interact with the data, to using an Oculus Rift or other head-mounted display and XBOX 360 controller for control, or at the high end side using an immersive visualization room (an immersive CAVE for example) with an interactive wand. Our poster and demonstration will show our progress so far with this project using an Oculus Rift.

Future work with include improvements and enhancements to the visualization environment including a more complete catalog of data types and their common visualizations, as well as enhancements in how the instructor or student can interact with the environment through the use of technologies such as the Kinect.

[1] MATLAB Release 2015a, The MathWorks, Inc., Natick, Massachusetts, United States. Available: http://www.mathworks.com.

[2] Unity 3D Game Engine, Unity Technologies, San Francisco, California, United States. Available: http://www.unity3d.com.