Frost_Byte

An investigation into the visualization of permafrost data utilizing the Discovery Lab at the Arctic Regional Supercomputing Center and University of Alaska Fairbanks.

By:
Jordanna Chord
ARSC Summer Intern 2004

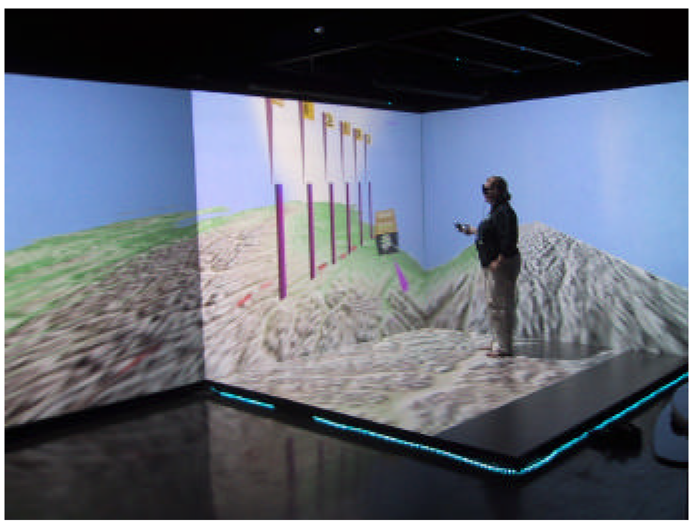

Table of Contents

Abstract

This report describes the efforts of two Arctic Regional Supercomputing Center Interns, Jordanna Chord and Patrick Webb, to visualize model and recorded permafrost data developed and collected throughout the state of Alaska. Frost_Byte serves to fill the void where no commercial application currently exists. It creates the first interactive virtual reality environment for viewing scientific data.

Defining the Problem

Scientists with the Geophysical Institute at UAF have been studying permafrost for over forty years. Permafrost, or perennial frozen ground, describes soil with a thickness from 50 meters to 650 meters that stays frozen year long.

The Permafrost Lab approached ARSC with the simple desire for a tool to visualize their data. In collaboration with staff at ARSC, it was decided to go beyond what had ever been done before in any available commercial application. The solution to display the permafrost data would be a virtual environment that would take the user and put them directly in Alaska. The application would make the user feel like they were actually in the state examining data with an added interactive element to add value to the already existing data.

Frost_Byte is the beginning of this new application. This project's goal was to take soil temperatures of known permafrost regions and create a visual model in a three-dimensional virtual environment to enhance visual understanding of data. In addition, the software developed must be expandable and easily modifiable. The products of this project must be able to serve as the foundation for the ultimate goal: a real-time model of permafrost with the ability to apply a scientific model to observe effects of natural climatic change, human interaction with the land, and other environmental changes such as wild fires or drought in a virtual reality environment.

Background

The Permafrost Lab was established in the 1960s by Professor Thomas E. Osterkamp. Other present members of PL are Assoc. Professor Vladimir E. Romanovsky, Visiting Research Scholars Gennadiy S. Tipenko and Dmitri O. Sergueev, Research Coordinator Irina Outkina, Graduate Students Tatiana Sazonova, Galina Ershova, Sergei Pokrovsky, and Dmitri Nikolsky. [1]

Figure 1: Dawson, AK: Abandoned Home affected by permafrost.

The active layer, usually the upper 30 to 50 cm, is the only portion that thaws every summer then completely refreezes during the winter. The average temperature of most permafrost is just below freezing[2]. This leaves it vulnerable to global climatic change causing a vast array of unknown environmental

[1] Permafrost Laboratory with Geophysics Institute Home Page; http://www.gi.alaska.edu/snowice/Permafrost-lab/
[2] Romanovsky, Vladimir; "How rapidly is permafrost changing and what are the impacts of these changes." ; Geophysical Institute UAF;

effects and potential damage to man-made structures constructed on the land. The effects of changing permafrost can already be seen in many Alaskan communities with homes whose foundations have cracked or buckling roads that must be repaved every other year. To better understand and predict these changes, the scientists in the Permafrost Lab have developed a model mapping permafrost temperatures and have spent years recording temperatures at various locations in Alaska.

The discovery lab is the main visualization tool for Frost_Byte. Used to "Visualize data in a three-dimensional, immersive environment. The MD Flex is a small room approximately 10' x 10' x 8' in size. Three walls and the floor are equipped to display stereoscopic images, which allow researchers to virtually immerse themselves in their data set."[3] Frost_Byte will continue the study of user interfaces and scientific data representation for virtual environments. Frost_Byte was developed with the assistance of Professor Glenn Chappell and ARSC staff Bill Brody, Jesse Niles, and Don Bahls.

Approach

Navigation and Environment

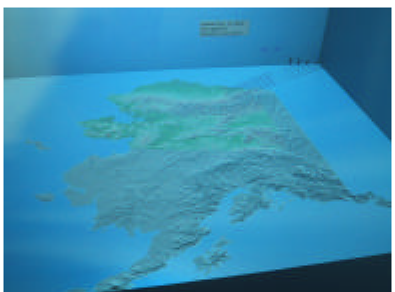

Figure 2: Aerial view of frost_Byte world.

The first task was to create the environment and determine how they could move about. It was decided early on that the user would be on a globe where Alaska is the only land mass. A satellite image of the state was textured onto a large rectangle and a blue sky texture was placed around the map to provide the user with a sense of direction and boundary. Now that the world existed; how was the user to navigate?

Navigation was made as simple as possible. Initially, three buttons on the wand, the main control for the virtual reality environment, were used for right, left, and forward motion. However, these are the only buttons and it was decided to reserve them for alternative interactions. A far more intuitive approach allowed the navigator to move in which ever direction the wand pointed. With navigation implemented, a rudimentary acceleration/travel speed method was added. As the user climbs in elevation so does the speed of travel. This allows the user to fly across the state faster at higher altitudes and move at a walking speed at ground level. This makes it easier to fly across large areas, but slow down to examine data close to the ground.

Figure 3: Wand: (Source www.isense.com)

[3] www.arsc.edu

User Interface

Figures 4-8: Three Views of the mini-map: Left-Top View of buttons to change days. Center – View of map, red marker highlights position in world. Right: User clicks a button on the mini-map with data wall behind. Bottom Row: Two views of the day display and season clock. Red indicates summer and blue winter, with a small black marker indicating what season is currently displayed.

Now that navigation is handled; how will the user know where they are? If they are down at ground level it will be impossible for them to know their location in relation to the whole state without flying to a higher elevation and looking down. With more generations growing up with electronic games; the user interface was shaped after what is often seen in first-person shooters. What is now called the virtual PDA (V-PDA), was developed. This miniature cube follows the user wherever he or she goes; can be moved in different locations around the person, and contains valuable information and tools for interaction with the environment. The V-PDA's main feature is an image of the state of Alaska with a marker, called the ant, indicating where the user is. The PDA cube may be rotated around to access other tools and will be used as the main access point for future features.

Some thought was put into how sound can enhance the user's experience. Although sound is built into most computers today; some users find sound effects useless, leaving speakers off. Thought was invested in how sound could enhance this project; although it was concluded to only use sound as a confirmation an action occurred. So if the user clicks on a button, they hear a clicking sound, or if they fly they hear air rushing by.

Visualizing Data

There are approximately one hundred locations where permafrost temperatures have been recorded for a period of about thirty years, with depths up to several meters. In addition, there is currently model data being developed of the same number of locations, expanding a period of two hundreds years, and with depths up to 1.3 kilometers.

How can Frost_Byte present this data in some meaningful matter. There are few traditional methods used to visualize permafrost, with most research sticking with graphs and tables showing trends in depths and temperature.

The image of the state had superimposed on it approximate locations of the data points, the initial plan was to split the state along this line, setting the eastern portion at a higher elevation. Between the differences, a wall representing the temperatures at their points is displayed with the height of the wall determined by depth; and the color representing temperature.

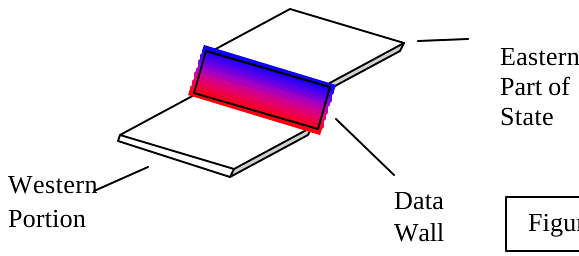

Figure 9: Example of first

It was realized however that mapping the wall along the split would be extremely difficult because the data points are located with longitude and latitude, and there was no easy way to accurately create the cut so that the data points would create a wall that would fit the cut. In addition there are some points located such that they don't fit in with one data wall. It was decided then to create a data wall, but have it project out of a flat ground. This allows for points that do not fit along the wall to be represented independently.

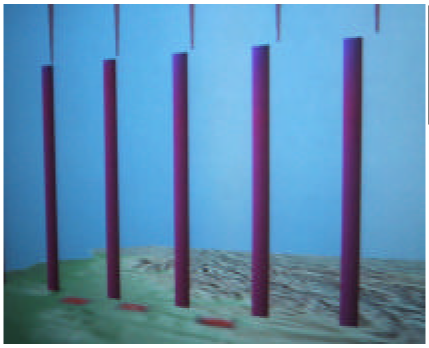

Figure 10: First evolution of the data wall.

The wall itself uses color to represent temperature. This poses several difficulties. Permafrost temperatures vary slightly, often with temperatures changing less than .01 Celsius through depths or days. First the color scale was red to blue with red representing the maximum temperature of 40 Celsius and blue representing -30 Celsius. The wall can look like one solid color although temperatures may be changing enough for scientific interest. Each depths segment is blended together to create a smooth transition between colors. With temperatures varying only slightly and the level of freezing ending up in the middle of our color scale, it was difficult to determine when soil thawed and refroze. To solve this problem anything above freezing was colored a shade of orange while a green color scale was applied to frozen soil. This provided a dramatic color change at the freezing temperature level allowing for users to see whether soil refreezes after it thaws. This color mapping is created with a single method that creates a texture. If the user would like to focus on a smaller temperature range only two values must be changed in the source code.

The data wall pictured above displays segments located where the model data was located. Because this is a model and not recorded data, Assoc. Professor Vladimir E. Romanovsky informed the team that Frost_Byte would be more valuable if it interpolated permafrost between segments by taking an average of the temperatures on both sides. This modification created the final design of the data wall.

Figures 11-13: Three Views of the data wall. Left: Winter approaches and permafrost refreezes. Center: Data wall with season dial and display of day. Right: Summer time, data reveals thawing of permafrost.

Finally, how the data was to be accessed was addressed. Currently there exists a data file with temperatures for seventy thousand days. The first method allowed the user to skip through the days using buttons to skip one, one hundred, or one thousand days. However, when the user got to higher values the program stalled because data was being read from the file took a long time. The newest approach reads in a user specified number of days at load time. They can then play through them like a movie, or skip through individual days. This allows for large quantities of data to be viewed encompassing a specified period of time allowing the user to see changes in permafrost.

Summary/Conclusion

The final piece of software will serve as the foundation for future expansion. By using C++ and an object orientated style changes by future programmers will be effortless. The innovative user interface has proven useful. A user is able to track their position using the virtual PDA, and make changes by using buttons on the other sides of the cube. The user can lose the cube if they place it behind them, but until this is tested further by users, it's unknown whether this will be a problem.

The data wall has gone through several transformations, and may go through several more. Current time restrictions have limited some design elements to be left out until future additions, although effort was put forth towards discovering improved implementations. The data wall is the main scientific element and currently can be loaded with a user requested number of days of data from an input file. The most complete scientific data at this time available is about 70,000 days of temperatures to one kilometer depths for six points located in the northern coastal region of Alaska. This can be loaded and viewed by the user with darker greens indicating cold temperatures and a bright orange indicating a hot temperature. Currently the location of the wall is calculated by mapping the longitudes and latitudes directly to the world, and making a translation to place it into its proper place. The current map used to display the world did not have enough information available to properly calculate latitudes and longitudes, leaving this feature only partially implemented. To make it easier to add other data walls it will be necessary to reconcile this.

We have concluded that some features must still be added to the data wall to make this software additionally helpful. They include:

- A dark line splitting the depths to differentiate them.
- Ability to query the wall by pointing at it for scientific data.
- Input data from file using binary to increase read speed.
- Improve mapping of data points

The final program does meet all the stated goals. The beginnings of the very first interactive virtual reality visualization of data were successfully implemented. The user may navigate about a virtual environment to observe changes in permafrost temperatures while using a unique and helpful user interface. Sound has been integrated to provide confirmation that tasks have been completed, and all beta runs by new users have proven that the software is understandable and usable.

Appendix

Description of files and classes and directions for Frost_Byte.

FILENAME: ***README***
#
AUTHOR: Jordanna Chord, Patrick Webb
CREATED: 06/09/04

It requires VR Juggler 2.0 and a configuration similar to that of the Discovery Lab at ARSC (Arctic Region Supercomputing Center, University of Alaska Fairbanks).

FILES
-----

jugglerUser.h / jugglerUser.cpp

Header and source files for the jugglerUser class. This is an interface for VR input devices. It returns TRANSF types (see below).

vecpos.h / transf.h

Header files for the TRANSF transformations package, by Glenn G. Chappell and Chris Hartman.

tfogl.h

An experimental openGL interface for TRANSF.

frostByte.h / frostByte.cpp

Header and source files for the frostByte class, that runs the main navigation and drawing for the permafrost model.

main.cpp

Main program for frostByte. Starts the VR Juggler kerneland launches an instance of frostByte.

Makefile

Unix makefile for frostByte. Requires GNU make ("gmake").

run_frostByte

Executable script to run frostByte with the proper VR Juggler configuration.

minimap.h / minimap.cpp

Header and source files for the minimap application. This manages the small map that the user uses to navigate the world.

simplereadrgb.h / simplereadrgb.cpp

Header and source files for reading images for textures. In conjunction with member function readfile and class texture_data in Frost_Byte.h/.cpp, textures are managed for the world. These may be used independent from this program. Be aware that simplereadrgb only handles uncompressed .rgb files in the SGI format.

texture_data.h

Header file for storing GLuint variables for textures. All textures used in Frost_Byte are initialized here.

data.h / data.cpp

Header and source files for class data. Data handles reading input from file and rendering the main data wall. Currently, the rendering of the data wall handles making extra transformations to place the wall into the correct position. It is dependent on the use of the initial six data points provided by modelInputFixed.txt

bitmapPrint.cpp

Source file for function to print bitmap text to screen.

dateSelect.h/dateSelect.cpp

Header and source file for dateSelect class. Creates and controls minimap buttons that allow navigation through dates.

wandPointer.h / wandPointer.cpp

Creates a pointer tool at the end of the wand. Sends message to user when a clickable object is being pointed at.

Misc. Input Files/Folders used by Frost_byte
----

modelinputFixed.txt

Input file for permafrost data. Must be in format as described in header of readinputModel method in data.h

ak1.rgb, alaskaPermafrost1024.rgb, chevron.rgb, dateFace.rgb, emptyFace.rgb playbtn.rgb, stopbtn.rgb,

.rgb files used for textures in Frost_Byte.

wavs/

Folder where all sounds used in Frost_Byte are located.

RUNNING Frost_Byte

To compile Frost_Byte under some flavor of Unix, cd to the directory containing Frost_Byte, and run GNU make (probably by typing "gmake"). The proper environment variables for VR Juggler must already be set up.

After compilation, run Frost_Byte by executing the run_frostByte script (probably by typing "./run_frostByte").

Compiling and running under other OS's will be very different, of course.