jMovie

Esen Yilmaz, Matthias Schwab[1]

keywords: *java, movie, rickmovie, data cube*

ABSTRACT

jMovie is an object-oriented implementation of Rickmovie, SEP's standard data cube viewer. jMovie reads a standard SEPLib Header file and allows the user to view slice movies in all three dimensions. jMovie is implemented in Java. Consequently, jMovie ports to any Java 1.1 platform. Remote users can run jMovie from a Java-enabled web browser. The object-oriented nature of jMovie allows future modular enhancements and extensions by users.

INTRODUCTION

jMovie is based on Rickmovie, the seismic data cube viewer currently used by SEP. The code reads the SEP cube in order to convert the SEPLib Header data files into byte arrays. After conversion to byte format, each of the three dimensions of the data cube are displayed separately (Figures 1 and 2). The user can cycle through slices of a dimension manually or as an animation.

I attempted to create a simple version of Rickmovie for the display of SEP cube data sets on the web. I used the Java Advanced Windowing Toolkit (AWT) 1.1 Event Model (Laura Lemay, 1997) for the user interface and the Image class methods (Flanagan, 1997) for the graphics.

The following report will discuss how jMovie works, the results obtained from testing, followed by a discussion on future improvements on the utility.

[1]**email:** ely3@cornell.edu, matt@sep.stanford.edu

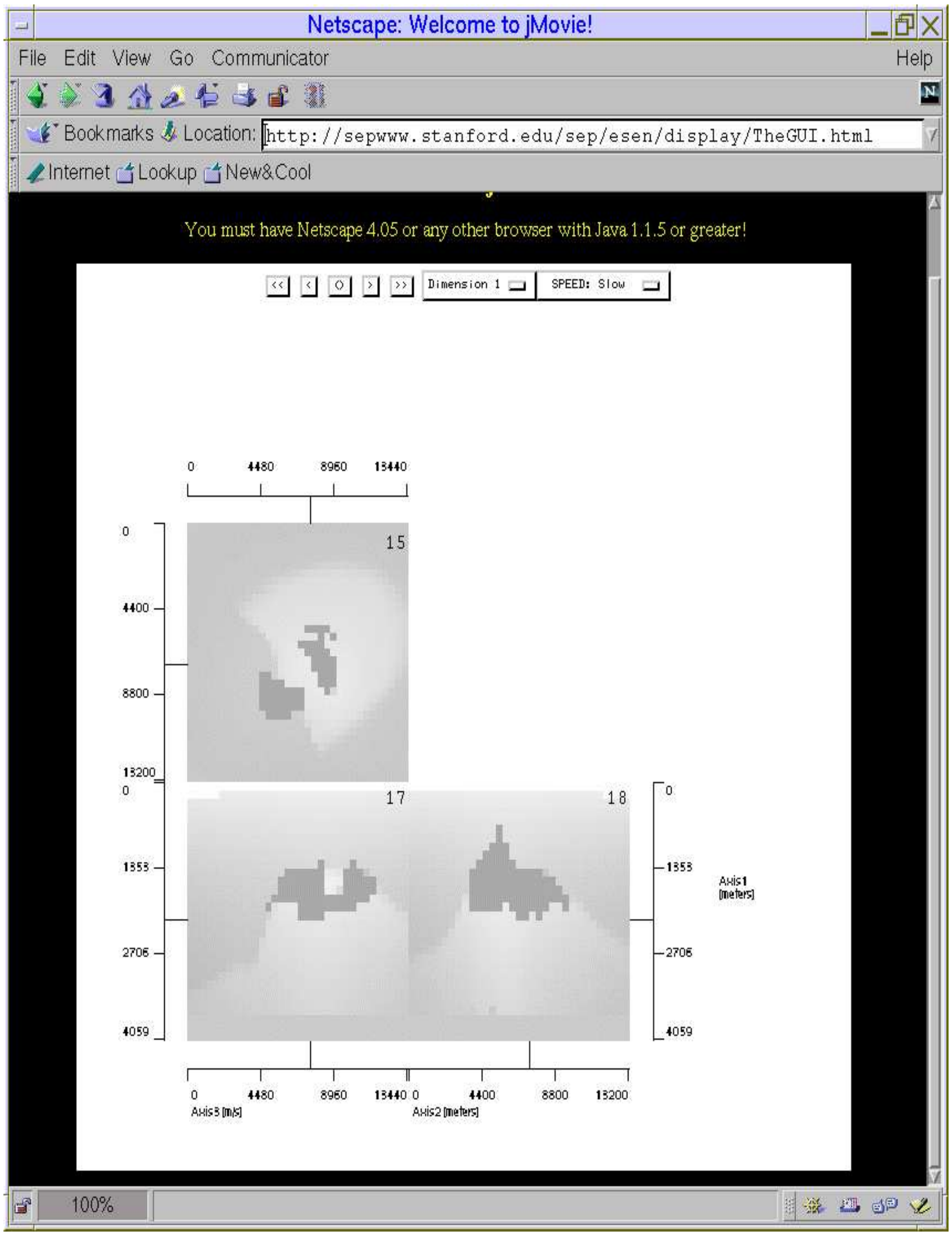

Figure 1: jMovie is a utility for displaying SEP data cubes. jMovie is shown here running as an applet over the world wide web. jMovieApplet [NR]

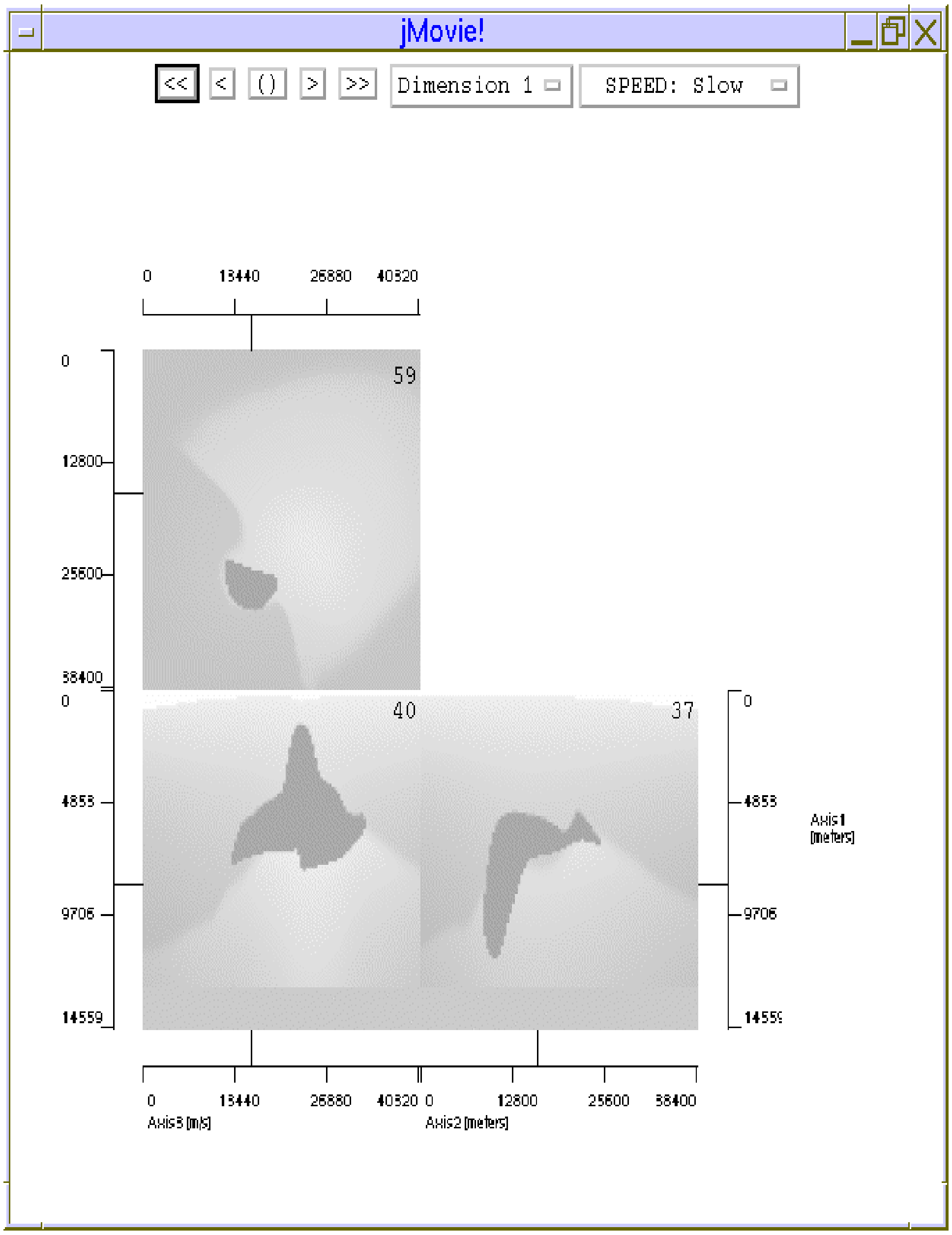

Figure 2: jMovie is shown here running as an application. jMovieApp [NR]

HOW IT WORKS

Applet vs. Standalone Application

When run as an applet, jMovie takes its file location parameter from the `PARAM` field of the `APPLET` tag of its HTML file.

```
<APPLET CODE = "TheGUI.class" WIDTH = 700 HEIGHT = 700>
<PARAM NAME=file VALUE="http://sepwww/pub/sep/esen/display/smaller.H">
</APPLET>
```

When run as a standalone application, jMovie requires the user to input the parameter in the command line. For example:

```
java TheGUI file:/homes/them/esen/movie/display/smaller.H
```

Reading the Data

The program calls the byte converter, which in turn calls the SEP reader class. This class creates an object of type SepCubeHandler. The SepCubeHandler is responsible for reading and parsing the header file, as well as storing the float data into a float array. The SepCubeHandler returns a Java hashtable with the SEP header information and a Java float array with the SEP data. TheGUI hands the header information to the AxisMaker and the data float array to the byte converter. We define valid data files to be SEPLib Header files with the suffix ".h" or ".H," since SEP data is not yet defined as a MIME type. The passing of the header information enables the creation of axes. [2].

Creating the Images

Once the integer array of data is passed to the main program after being read from the file, it is given to the program's Imager object for processing. The Imager object manipulates the three main image variables of jMovie - CurrentImg, CurrentImg2, and CurrentImg3 - representing the image being displayed for each dimension at a particular time.

The images for each dimension are created by methods createSliceData and createPixelData in Imager. The three dimensional integer array is used to create a two dimensional float array depending on which dimension is being sliced. This two dimensional array is then passed to createPixelData, where it is made into a one dimensional array of grayscale values. The resulting array is then used to create the current image variable for the selected dimension.

[2] To be implemented. Currently the entire SepCubeHandler object is passed back to the main program.

Graphical User Interface

The graphical user interface (GUI) was created using the Java 1.1 AWT (Flanagan, 1997). The user can choose from the following options:

- select a dimension to loop over. [choice menu]
- select a speed for the animation. [choice menu]
- step foward or backward through the selected dimension. [button]
- animate the selected dimension forwards or in reverse. [button]

In addition to the visible choice menus and buttons, the user can click within the axes to jump to a specific slice.

I implemented the event handling for the buttons and choice menus using the Java 1.1 Event Model (Laura Lemay, 1997). Action listener and item listener classes are added to the buttons and choice menus respectively. The buttons and choice menus call their respective action or item listeners when activated. The mouseDown method of the main class was then overridden to enable the user to click within the axes.

The Graphics

All images in jMovie are double buffered. The images are first written to offscreen buffers, and then dumped on to the screen from the buffers. Double buffering eliminates flickering and the partial drawing of images during animation.

RESULTS

Performance

Overall jMovie is significantly slower than Rickmovie. The extent to which jMovie's performace could be improved is unclear.

The lack of speed of the IO leads to delay in the startup. The lack of speed of the Imager object leads to skipping of frames. In general, jMovie is faster when run locally as an application than when run remotely as an applet. Currently, the application uses a better image scaling algorithm for smoother images than the applet. In a future versioin, we may enable the user to select the scaling algorithm.

When using a larger test data cube (Figure 1) of dimensions 105x97x97, both versions of jMovie were slowed down considerably. jMovie Application skipped every other frame during animation at the Slowest speed setting, while jMovie Applet skipped two out of every three frames during animation at the slowest speed setting. A future improvement would be to add an option to display only the dimension being animated, as well as the option of not displaying axes.

A Bug

jMovie is able to dynamically switch between operations. When animating one dimension, the user can click on the axes of another dimension and immediately begin animating the new dimension. However, this results in a misleading Dimension choice menu. If the user were to click on the third dimension's axes, that dimension would now be selected while the choice menu may still show some other dimension as being selected. This was seen to be the only operational problem.

Applet vs. Standalone Application

All of the code for jMovie is able to run as an applet or a standalone application. When trying to get the main class, TheGUI, to run however, the jMovie applet was not able to run due to the presence of a constructor needed for the jMovie application. As a solution, the main class was compiled twice (one with and one without the constructor), while the remaining classes remained unchanged. The jMovie applet can be found at `http://sepwww.stanford.edu/sep/esen/display/TheGUI.html`. The jMovie standalone application is at `/homes/them/esen/movie/display`.

FUTURE WORK

The following is a list of possible enhancements for future implementation. The most important enhancement would be the addition of Java programs, dynamically loaded by jMovie, that act as filters on the data cube before it is displayed. The development of image manipulation plug-ins would take full advantage of jMovie's object-orientation and allow developers to expand the program's capabilities.

- Scalable interface: to match the user's viewing window and resolution. This would allow the use of data cubes with more than 200 samples per dimension without losing resolution - currently the slice area is set to 200x200 pixels, consequently a slice greater than 200x200 pixels would be reduced in size.
- Optional display of axes: for optimal performance on slower computers or over remote connections when using jMovie Applet.
- Displaying only the dimension being dealt with.
- Dealing with less or more than three dimensions: Expanding jMovie to display SEP data files with one, two, four, and five dimensions would make it more of an equal to rickmovie.
- Displaying the data in different ways: such as in the form of a rotatable cube.
- Zooming of data.
- Data annotation: Text bubbles that indicate special features.
- Storing state on disk.

CONCLUSIONS

The main advantage of jMovie over Rickmovie is its capability to interactively explore SEP data across the web. In general, jMovie's Java implementation should make it extremely portable to any platform: PCs, UNIX, and soon your toaster. jMovie's object-oriented design invites future enhancements, such as plug-ins and faster display. jMovie should benefit greatly from the release of the Java 1.2 2-D API and its future implementation in browsers.

ACKNOWLEDGEMENTS

Esen wants to thank Biondo Biondi for suggesting a Java version of Rickmovie. James Rickett installed and updated the Java interpreters and browsers on our development platform, Linux. Bob Clapp provided the data set we used as the example above. Morgan Brown helped editing the TeX document and gave feedback on the design. Matthias wants to thank Mr Rasband at the National Institute of Health for all kind of questions about his Java graphics package ImageJ. Matthias suggested the overall jMovie class hierarchy. He also provided the code for the SEP data I/O and the conversion and display of floats to bytes. Esen built the user interface, programmed the event handling and image processing, and debugged most of the code.

REFERENCES

Flanagan, D., 1997, Java in a nutshell: O'Reilly, Cambridge.

Laura Lemay, C. L. P., 1997, Teach yourself java 1.1 in 21 days: Sams Net, Indianapolis.