HTML Viewer: Version 2

Robert G. Clapp

ABSTRACT

Most seismic data processing is now done remotely, rather than locally. The X11 protocol is poorly suited for remote processing. HTML5's flexibility, support for strong client-server paradigms, and low latency design allows for a platform independent solution to visualizing seismic data. Web sockets allow communication between a server at the location where the data is stored and a client written in JavaScript. JavaScript Object Notation (JSON) is used to exchange both messages and JPEG and MPEG encoded images/movies. An early prototype of the viewer shows great promise in providing a robust solution to remote viewing of seismic data.

INTRODUCTION

Viewing and interacting with multi-dimensional volumes is necessary when working with 3-D data. SEP wrote its first movie program 30 years ago and has continually expanded on this initial idea (Claerbout, 1981; Sword, 1981; Ottolini, 1982, 1983, 1988, 1990; Arroyo and Clapp, 2002; Clapp et al., 2008; Clapp, 2010). Since the early 1990s these viewers have all relied on the X11 protocol. As computers got faster, the many handshakes and bit image transferring approach of X11 led to ever more noticeable latency problems on even local networks, exacerbated when using remote connections. As a result, remote frame buffer approaches such as VNC have grown more and more popular. These approaches achieve speed by sending a compressed version of the portion of the screen that has changed. The downside of remote frame buffer approaches is that they are completely reliant on the server and suffer when significant latency exists between the client and the server.

Web development has had to deal with similar issues of how much work should be done on the client versus the server. The support of virtually every browser for JavaScript allowed a significant portion of an application to be handled on the client. In Clapp (2014) I wrote about two prototype viewers in HTML5, one that could displays a cube, and one that could display multiple slices.

In this paper I describe a new version of an HTML5 viewer that combines the features of the two prototypes described in Clapp (2014). I begin by describing how a session is initialized, then present the viewer's interface, and conclude with future plans for the viewer.

INITIALIZATION

A python script `createView` takes as input a series of files that the user wishes to view within a given directory. It creates two outputs. The first is a PHP webpage that a user can use to view the datasets. The second output is a file `.viewList` written in JSON format. An example can be seen below.

```
{
"permission":"all",
"tags":["vel","mig"],
"vel":["a.T","SEP-BYTE"],
"mig":["inv.T","SEP-BYTE"]
}
```

The configuration file specifies who can view the datasets `permission`, the name given to the datasets within the viewer `tags`, and finally the list of datasets and their format.

The PHP output file creates a login page for viewers of the datasets. This login requires both a user name and password. The password is checked by the server before a session can be initialized. A session is initialized by upgrading a standard HTML connection to a websocket connection that allows bi-directional communication between the server and the client.

Server

The server is initialized by JSON configuration files that define thevalid users, the groups they belong to and a directory where temporary files should be written. An example of the user configuration file can be seen below.

```
{
"users":["bob","jon","biondo","dave"],
"sep":["bob","jon","biondo"],
"groups":["sep"]
}
```

The server waits for websocket requests. The client and server communicate by sending JSON messages. The first message from a new client is a request for information about a given directory. The server reads the `.viewList` in the directory and checks that the user has permission to view the datasets. Assuming that the user has permission to view the datasets, it then informs the clients about the datasets. If a user chooses to view a dataset, the server checks whether if the dataset is already memory, and if not reads it into memory. The server uses a fast JPEG library to send images or FFmpeg using a RAM disk to create and send movies.

MENUS

The interface for the viewer is designed to minimize the amount of space used for controls. Figure 1 shows the interface. The icons on the left, from top to bottom, are `data, tools, sharing, controls, help`, and `connection` status. At this stage not all functions have has been implemented.

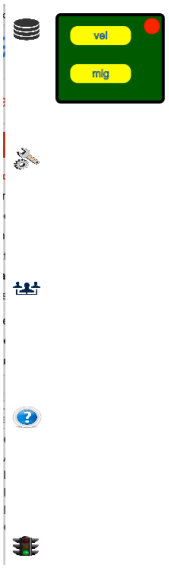

Figure 1: The initial view presented to the user. Note the menu icons on the left for `data, tools, sharing, controls, help`, and `connection`. The `data` icon has be selected showing the available datasets. [**NR**]

The `data` menu contains the list of tags the user defined for the webpage. To open a dataset you click (touch on touch display) the data icon. A menu opens up listing all of the datasets. You then drag them into the main viewing area. When dragging successive datasets to the viewing area you can choose to either combine them in the same panel (see the multi-panel section below) or in separate displays. See figure 2 as an example of displaying multiple datasets. Each dataset must exist in the same coordinate system and each view is linked. Moving or zooming in one display will affect all of the displays.

Currently the edit menu only has one option, clipping. This menu allows the user to interactively change the clip of datasets that are stored as floats. Figure 3 shows an example of clip menu. The user selects which datasets he wishes to clip and what type of clipping to use. Currently both a standard pclip and a soft clip Claerbout (2014) are supported. When multiple datasets are selected these datasets are clipped together allowing for accurate amplitude comparison.

The sharing menu allows a user to broadcast their current display by selecting the `share` option. A user is also provided with a list of all the users displaying the current page and can choose to sync their view with another user.

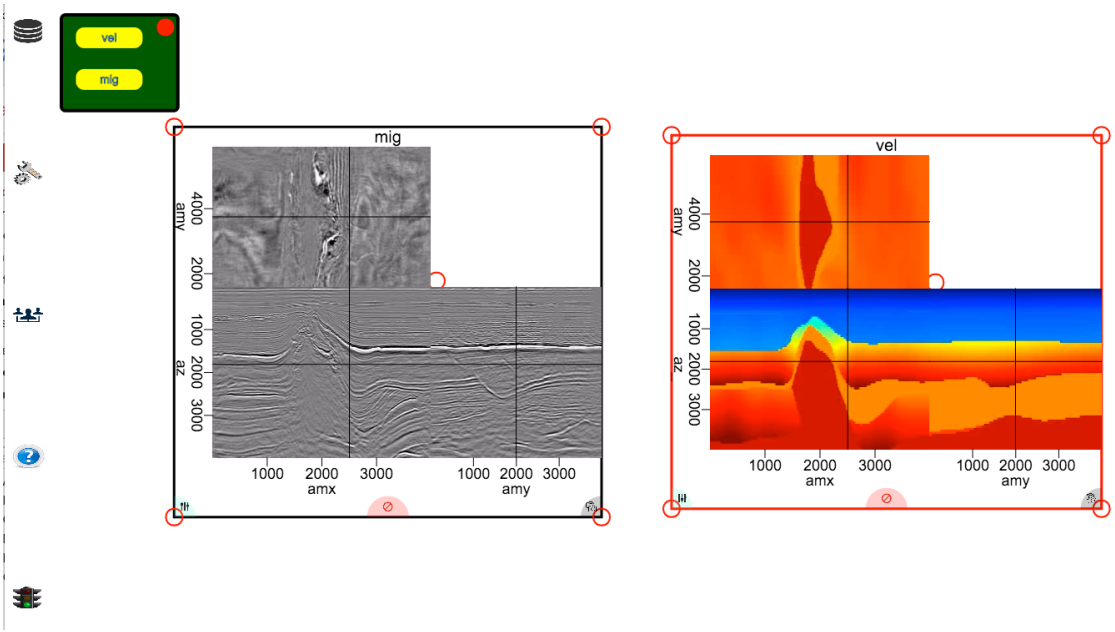

Figure 2: Viewing two datasets simultaneously. In this case velocity and a migrated image existing in the same coordinate system. [**NR**]

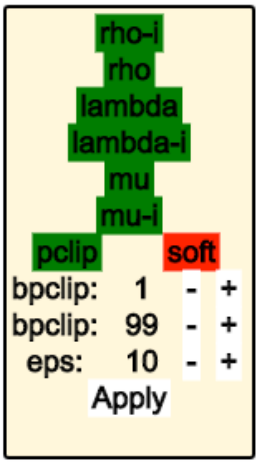

Figure 3: The `clip` menu gives the user the ability to interactively change the clip of a dataset, clip multiple datasets to the same level, and use a traditional, linear, clipping approach or a soft clip approach. [**NR**]

Currently the control menu is inactive. In the future it will allow the user to specify what various keyboard, mouse, and touch controls do. The help menu displays the current controls available to the user given the type of device they are using. Finally the status menu displays either a green or red light depending on whether a connection is active.

PLOTTING OPTIONS

At the bottom of each panel the user can select to open up a viewing options menu (left), close the panel (center), or a multi-plot menu.

The viewing options panel allows the user to change how the current panel displays its data. Figure 4 shows an example of this panel. The user can change which face of the hypercube is displayed and may change the order of the various axes and whether or not to reverse them. The user can also choose whether to display and in what color the axis labels, tics, grid-lines, and axis values.

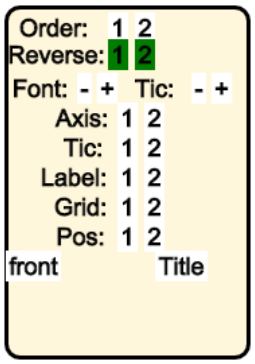

Figure 4: The view menu allows the user to change how the data is displayed. The user can change which face of the hypercube is displayed and may change the order of the various axes and whether or not to reverse them. The user can also choose whether to display and in what color the axis labels, tics, grid-lines, and values. [**NR**]

The multi-plot menu is only valid when multiple datasets have been dragged to the same panel. Figure 5 shows an example of the multi-panel. The user can display multiple datasets in one of three ways. In the first option, the datasets can flip between the different displays. The user controls the speed of these flips. In the second option, the user can overlay the two datasets controlling the opacity. For the final option, the user views two superimposed datasets with a sliding curtain two flip between the two dataasets. Figure 6 shows the last option, the migrated image model on the left side of the display and the velocity on the right.

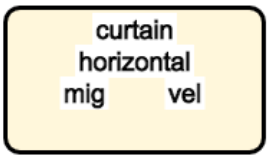

Figure 5: The multi menu allows the user to specify how to display multiple datasets in the same panel. The user can select from flipping datasets, opacity, or a curtain approach. [**NR**]

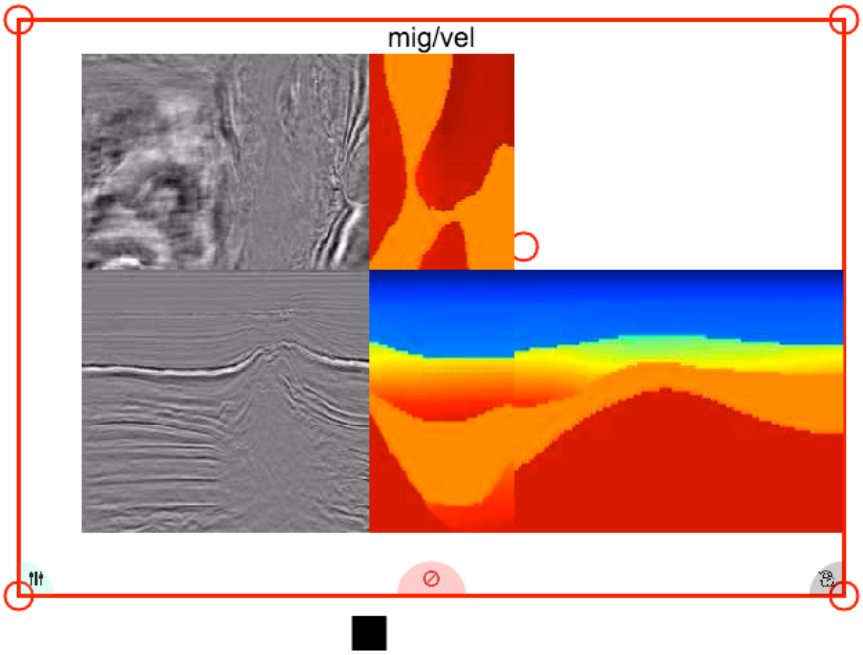

Figure 6: An example of the displaying multiple datasets using the curtain rod to flip between two images. [**NR**]

CONTROLS

As stated in the previous section, all of the available controls can be seen by accessing the help menu. The controls are different based on whether the user is using a touch screen or a mouse. In the following section I describe the currently available controls.

Keyboard controls

There are number of keyboard shortcuts available to the user. The following summarizes current shortcuts.

Key	Action
Ctrl-l	Lock the displays. Any changes made to a given panel will apply to all panels.
Ctrl-u	Unlock the displays.
c	Rotate the color table
f	Go forward along the front panel direction
b	Go backward along the front panel direction
F	Start a movie in the front panel direction going forward
B	Start a movie in the front panel direction going backward
Arrow keys	Move left, right, up, or down.
Shift arrow	Start a movie in the left, right, up, or down
h	Cycle through hiding the menus
p	Save the current display

To navigate with the mouse, the user can click to move to a new location. Holding down the mouse button will draw a window that will zoom into a region. Double clicking will return the display to its original form (zooming out).

Currently the viewer supports one and two finger controls for touch devices. The user can use one finger to touch the location they wish to move to. They can also move around a dataset by continuing to press on the display. They can flick their finger to quickly move through several frames (start a movie). With two fingers the user can pinch to zoom in and zoom out. In the future, further touch controls will be added.

FUTURE WORK

There are several improvements that I want to make to the viewer. These changes can be generally broken into three categories: controls, tools, and reproducibility.

I want to expand the capabilities of the touch controls, adding multi-touch capabilities beyond zooming. Control handling is written in an object-oriented fashion that will allow the controls to be changed on the fly using the `control` menu. I want to add the ability to pick with the viewer and to annotate figures.

Currently all datasets must be present at the initialization of the viewer and exist in the same multi-dimensional space. In the future, I want to support the creation of datasets. An example of creating a dataset might be using a velocity picking tool that then creates an Root Mean Square (RMS) velocity file and potentially the result of Normal MoveOut (NMO).

Currently the viewer records all user actions. We need to add the ability to save and then rerun these actions. In addition, rather than using JavaScript support for creating a PDF from the current view, I would like to translate the given view using vplot which supports vector graphics.

CONCLUSION

In this paper we described the current state of a HTML viewer for seismic data.The viewer solves the problem of X11 latency by using a client-server model without installing any software. It allows the user to interact with dataset using keyboard, mouse, and touch interfaces. In addition, multiple people can simultaneously interact a dataset and share their perspective views.

REFERENCES

Arroyo, E. R. and R. G. Clapp, 2002, Displaying seismic data with vtk: SEP-Report, **111**, 395–403.

Claerbout, J., 2014, Automatic default for hyperbolic softclip : SEP-Report, **152**, 177–186.

Claerbout, J. F., 1981, On-line movies: SEP-Report, **28**, 1–4.

Clapp, R. G., 2010, Hypercube viewer update: SEP-Report, **140**, 229–232.

———, 2014, SEPHTML5: HTML5 and interactive visualization : SEP-Report, **152**, 351–358.

Clapp, R. G., D. M. Chen, and S. Luo, 2008, Hypercube viewer: SEP-Report, **134**, 179–192.

Ottolini, R., 1982, Interactive movie machine user's documentation: SEP-Report, **32**, 183–195.

———, 1983, Movie cubes: SEP-Report, **35**, 235–240.

———, 1988, Movies on the Macintosh II: SEP-Report, **59**, 255–268.

———, 1990, Seismic movies on the XView graphics system: SEP-Report, **65**, 315.

Sword, C. H., 1981, SEP goes to the movies: SEP-Report, **28**, 11–20.