GPGN438 Senior Design:
Exploratory Seismic Data Analysis in the Field

Advisor/Client: Dave Hale
Team Members: Colton Kohnke

December 9th, 2013

Contents

List of Figures

List of Tables

1 Executive Summary

Seismic surveys take a significant amount of time and money to complete correctly. This project aims to develop software for the Colorado School of Mines (CSM) to aid with viewing seismic survey data and geometry in the field. The software will also be able to help catch errors in the field. It will also serve as a teaching tool for the students of the Colorado School of Mines Geophysics Field Camp to help them better understand seismic surveys.

The software will accomplish this by reading the SEGD files in real time from the recording truck along with the observation notes and the flag locations from a GPS unit. This data will then be plotted in map and section view in order to give a snapshot of the seismic survey. Basic processing can then be done quickly including gain, amplitude balancing, lowpass filtering, normal-moveout correction and surface-wave attenuation to create a dynamic stacked section.

The project will be completed by May 2014 and will be used as a replacement for ProMAX in the Summer 2014 Geophysics Field Camp.

2 Problem Statement

This project seeks to develop software in JAVA that aids with the exploration of seismic data in the field. The survey exploration includes the interactive display of survey geometry and seismographs. This software will also be able to perform simple processing tasks in the field. The program will serve as a bridge from the seismic crew to the students of Field Camp and serve as a teaching tool for the student's understanding of the seismic survey.

3 Introduction & Background

Geophysical data collection is an expensive operation that costs both time and money. In the field, it is in the data collector's best interest to make sure that the data collected is good quality.

ProMAX is the current system used in the field to view the seismic data coming in from the recording station. This system, while good for in depth processing, is not robust enough to quickly display survey data and interactively display the survey geometry. The CSM Field Camp is unique in that they surveys are not conventional, that is, the surveyors are trying parameters or setups that aren't typical of a corporate survey.

There can also be a lack of understanding between the students and the seismic survey with regards to how the data is collected in the field. This software will seek to help the students at Field Camp better understand the seismic survey while they are in the field. The software can also be used as a teaching tool to the students in Field Camp to help them better understand the aspects of a seismic surveys.

4 Deliverables to Client

The deliverables of this project include two main items. The first is a program written in JAVA that accomplishes the design objectives outlined in the next section. The second deliverable will be a documentation of the code in both standard notation and as a PDF "how-to" style guide.

5 Design Objectives

5.1 Interactive Display of Survey Geometry

The first objective is to be able to import station (flag) locations. This needs to be accomplished from a variety of GPS sources including CSV, Excel, GPX and tab-delimited text files. The GPS coordinates need to then be converted to UTM for easier processing. Once these stations have been imported and converted, they need to have the option of being exported so the conversion does not need to be applied again. The suggested columns are listed in Table 1. Once the conversion is applied, the next step is to plot the locations in map view.

Table 1: GPS Export Spreadsheet Fields

StationID	UTM Easting	UTM Northing	Elevation

The second objective is to be able to compute source and receiver locations. This is done for each valid shot FFID (Field File ID) by using the observer files to determine the source station number, live recording channels and receiver station numbers. The station locations are then used to find the location of the source and the elevation is read from a USGS Digital Elevation Map. All of this data is then stored into a spreadsheet file with columns denoted in Table 2. Some FFIDs correspond to bad shots and will need to be ignored dynamically by the program routines.

Table 2: Source-Receiver Locations Spreadsheet Fields

FFID	SEGD Filename	Source	StationID	Channel Number	SourceXYZ	ReceiverXYZ

Once the data has been extracted from the observation files, the program needs to be able to plot source, receiver and midpoint locations in map view. In map view, there needs to be able to plot a piecewise curve that represents the seismic line. The program also needs to be able to plot elevation profiles along the source, receiver and midpoint lines. Finally, there needs to be a slider to select and show points corresponding to each FFID that will provide a graphical history of the seismic survey and help to catch mistakes.

5.2 Interactive Display of Seismograms

The first step to displaying the seismograms is to convert the SEGD files (bytes) to an array of an array of floats (2D flaot array). Once this is accomplished, the seismograms need to be interactively displayed by using the FFID slider. This will display all seismograms for a particular FFID.

Another method of display will be to display all seismograms within a circle that the user draws in map view.

Two more sliders, called the min-max sliders, that are populated with the minimum and maximum offsets will be implemented. As these sliders move, the program will plot all seismograms with offsets within the range of the min-max sliders.

All of this is useless unless the data can actually be seen. Therefore, interactive controls for gain, amplitude balancing and lowpass filtering (Butterworth) need to be implemented and the results plotted. This will be done by using sliders, or interactive graphical menus.

5.3 Simple Processing

In the field, there is often the need to do basic processing in order to see a better picture of what is happening in the subsurface. Tools for basic processing of the data will be included in the developed software. The guaranteed processing tools will be for surface-wave attenuation and normal-moveout correction, but more are being investigated.

5.4 Other Requirements

Speed is a top priority when working in the field. Decisions are typically made quickly in order to move the survey along at a reasonable rate. If this software is fast enough, it can be used as a supplement to those decisions.

The user-interface must also be easy to use. The layout of the controls must be logical and follow a sequential order. These different features must be documented in the end-user documentation and be consistent in their implementation.

6 Decision-making & Assessment of Alternative Approaches

The decisions during the main software development section of this project deal with efficiency. It is possible that choices in the field need to be made very quickly and as a result, the software needs to be able to perform operations efficiently to keep pace. If the software is called to display a seismic trace, it needs to be able to find the data and display it in the least amount of operations. Similarly, if an operation is applied to a set of data, it should be applied in the least amount of compute time.

The software is not currently optimized for speed because that would slow down development. However, specific places in the software have been marked for efficiency improvements during the testing phase. One area that has been marked for optimization is the storage of the data from the SEGD files. Currently, shots are stored as a JAVA ArrayList and is traversed linearly. This causes an $O(n^2)$ search time where n is the length of the array. A much faster implementation would be a Binary Search Tree which has $O(n)$ traversal time. Another option is implementing a hast table, which returns a value in near constant time, $O(1)$.

The decisions during the later stages will revolve around end user usability. These choices will include aesthetic placement of tools in the software to create a logical progression of tasks for the end user. The decisions in this section will also revolve around how to lay out the documentation for the end user. These decisions will be made during the final stages of the project and will likely involve bringing in fresh eyes to test the documented work flows.

7 Design Solution

The tool that is being used in this design project is simulation. Each method that is written is tested by calling the method on the 2013 Field Camp data. These simulations provide the developer with real data to test the software against and with speed/efficiency benchmarks to try to improve.

In addition, before a feature is integrated into the larger code, it is tested against itself for errors and correct implementation. This speeds up the development process because it limits the area where bugs can occur at any given time.

8 Implementation Plan

8.1 Safety

The main safety concern of this project is the ergonomics of the developer. This project does not require field work or manual labor in the traditional sense. However, it does require a significant amount of time to be spent coding at a computer which may cause irritation of the wrist or forearm. The developer will mitigate the risk of encountering this problem by taking breaks, stretching and using correct posture while at the workstation.

8.2 Timeline

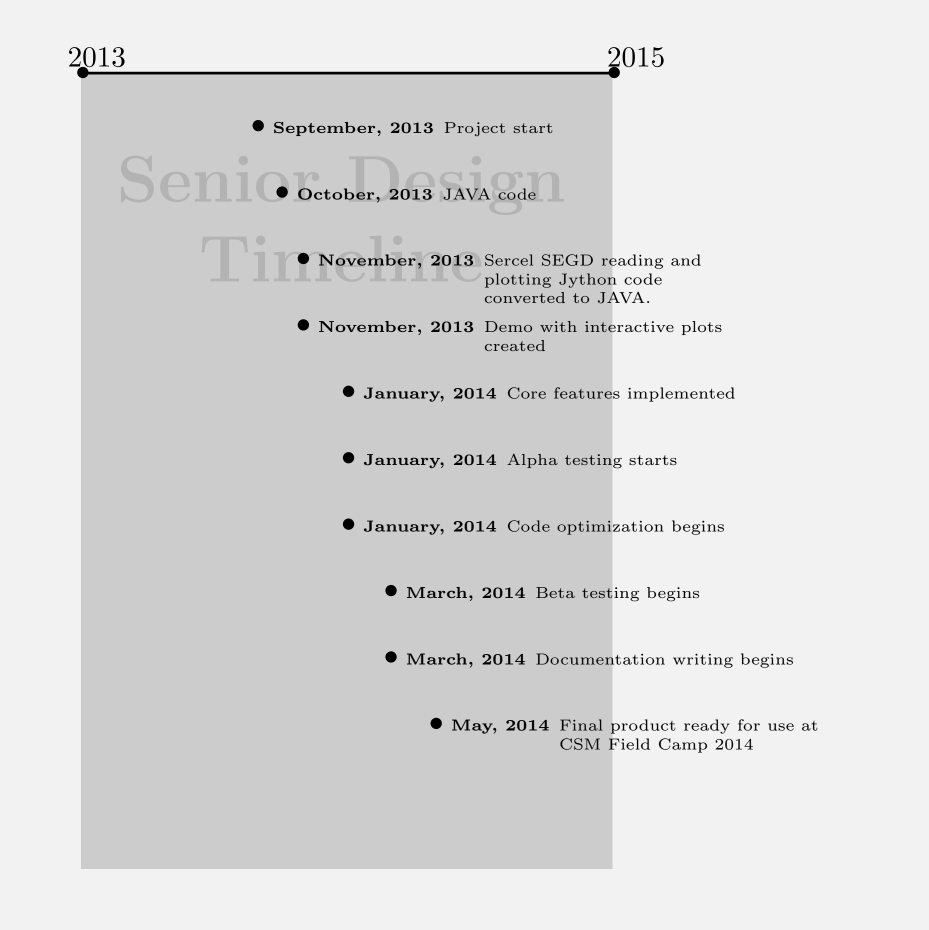
2013
2015
Senior Design Timeline
September, 2013 Project start
October, 2013 JAVA code
November, 2013 Sercel SEGD reading and plotting Jython code converted to JAVA.
November, 2013 Demo with interactive plots created
January, 2014 Core features implemented
January, 2014 Alpha testing starts
January, 2014 Code optimization begins
March, 2014 Beta testing begins
March, 2014 Documentation writing begins
May, 2014 Final product ready for use at CSM Field Camp 2014

Figure 1: Senior Design Project Timeline

8.3 Division of Responsibility & Effort

The project team is composed of a single developer, so the whole responsibility is on the only member of the team.

8.4 Budget 1: Actual

No budget is required for the actual project as the project does not require any traveling or other expenditures outside the realm of a standard class at CSM.

8.5 Budget 2: Professional

Ideally to get a working product to the client, the following budget would be used in a professional setting.

Table 3: The Professional Budget

Item	Base Cost	Quantity	Total	Notes
Developers	$20.00/hr	360	$7,200	2 Devs for 4 work weeks.
Continued Support	$15.00/hr	260	$3,900	1 Dev at 5 hours/week for 1 year.
Base Total			$11,100	
Overshoot			$1,110	10% addition
Total			$12,210	

9 Implementation

9.1 Data Acquisition

The data used to help develop the software is the 2012 CSM Geophysics Field Camp seismic data provided by Dr. Dave Hale. No further data acquisition is planned for this project.

9.2 Data Reduction, Analysis, Interpretation, Integration

This project revolves more around the display of field data than the analysis and active interpretation of the data. Certain reductions will be implemented to the data in an interactive way. These reductions include gain, lowpass filtering and amplitude balancing. The software will also serve as a hub for a place to view all the seismic data for a survey and integrate it together to make plots of the seismic line, such as a brute stack.

9.3 Error Analysis

The error analysis of this project will be related to the processing and display of imported data. These processes include making sure tools like the gain and filtering are correct, and the plotting of the data is accurate.

Currently the code for the gaining and filtering the data has been written, but it has not been tested against a known working algorithm. The shot plotting function has been tested against the known working Jython code and has been confirmed to work as expected.

9.4 Results

The results so far of the project have been the implementation of software that imports GPS points from Lat/Lon and transforms them into UTM coordinates before plotting them in map view. The software also imports SEGD files from the Sercel format and plots the nearest shot interactively as the user moves a mouse around the map view screen. The GPS and SEGD data can be imported and exported to files for easier imports later. Basic tools for manipulating the data have also been implemented, including gaining the data and passing the data through a lowpass filter.

9.5 Discussion and Conclusions

So far, everything has been running well and the setbacks have been mitigated by meetings with the advising professor.

The main pitfall the team has faced on this project has been time. As a single developer, I have not been able to dedicate enough time to this project and I should be much farther along at this point in time. The second half of the project will be more strenuous, but the results will come faster because my own coding ability has increased throughout this project.

10 Recommendations for Future Work

The next step for the project is to read the observation reports and finish implementing the main core features to create an alpha product. From there, code optimization can begin to occur and more efficient testing can begin.

After the software has been optimized for speed, it then needs to be optimized for usability. This will allow for the end-user to efficiently use the software.

11 References

12 Resumes

13 Appendices