PASSIVE SEISMIC AND DRILL-BIT EXPERIMENTS USING 2-D ARRAYS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF GEOPHYSICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Stephen P. Cole
August 1995
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Jon F. Claerbout (Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Francis Muir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Gregory Beroza

Approved for the University Committee on Graduate Studies:
Preface

All of the figures in this thesis are marked with one of three labels, [R] [NR] or [CR]. These define whether the figure is reproducible from the information provided on the CD-ROM version of this thesis.

If a figure is marked with an [R] it should be fully reproducible. All of the input data and source files required to create the figure are provided on the CD. The postscript file containing the figure can be removed and the cakefile rules provided on the CD should rebuild the figure on any of the supported computer architectures.

If a figure is marked with a [CR] it is conditionally reproducible. All the data and source files are provided on the CD but some particular environment may be necessary to recreate the figure. Many figures in this thesis were created using programs that run on a Thinking Machines Co. CM-5 computer. If the code that is provided will only compile on a CM-5 then the figure will be marked as conditionally reproducible.

If a figure is marked with a [NR] it is non reproducible. It cannot be recreated from the data on the CD. The figure may have been created using a commercial software product, or software written by another author that is not available on the CD. It may be a hand drawn figure or a figure for which the commands to recreate it are not available. In all of these cases, if you remove the postscript file of the figure there is no way to recreate it.

All of the reproducible and conditionally reproducible figures have a button in the caption. Clicking with the center mouse button will bring up a panel that lets you interact with the figure. You can display the figure in a separate window or destroy and rebuild the figure. If the figure is conditionally reproducible you can view the warning message that tells you the conditions under which the figure can be rebuilt.
PASSIVE SEISMIC AND DRILL-BIT EXPERIMENTS USING 2-D ARRAYS

Stephen P. Cole, Ph.D.
Stanford University, 1995

Advisor: Jon F. Claerbout

ABSTRACT

The Stanford Exploration Project (SEP) conducted a passive seismic experiment, deploying a two-dimensional array containing over 4000 geophones, in an attempt to see what such a dense array could tell us about the ambient seismic noise field and the subsurface beneath the array. I used several approaches to analyze the data. Beam steering analyzes the incident plane wavefield as a function of arrival direction. It revealed interesting, near vertically incident events that were consistent in arrival direction throughout the recording period. While the origin of these events is not clear, possible explanations such as electronic interference or nearby cultural noise sources can be ruled out. The angle of incidence constrains the source of these events to two regions — a small zone beneath the array, and a much larger zone roughly a thousand kilometers away. A followup study would be able to determine which is the correct explanation. Drill-bit source data is similar to passive data because it lacks a “time zero” at which an impulsive source was activated. I also conducted a drill-bit source experiment, using a 240 channel two-dimensional array, and applied many of the same methods used in the passive study. A scattering approach correctly located drill-bit direct arrival energy. By stacking along the correct moveout trajectory, I created an estimate of the drill-bit source signal, and crosscorrelated this signal against the data to enhance drill-bit energy. This clarified the direct arrivals from the drill bit, but revealed no reverberations of drill-bit energy, most likely because correlation times were not sufficiently long.

Approved for publication:

By Jon F. Claerbout

For Department of Geophysics
Acknowledgments

I would like to thank the sponsors, faculty, and students of the Stanford Exploration Project for making the SEP a reality. I really enjoyed being a part of it.

Jon Claerbout first described passive seismology to me and has participated in and enthusiastically supported my work at every stage. I also appreciate the advice and support I received from Francis Muir and Fabio Rocca during their time as SEP faculty members.

Dave Nichols and Lin Zhang were my partners in designing and carrying out the passive seismic experiment described in this thesis. From surveying stations to recording data at 3 AM, they were there at every stage. It was a pleasure to work with them.

I wish to thank the SEP members, Geophysics faculty and students, and others who helped deploy the 4000 geophone array used in our experiment, Willie Lee of the US Geological Survey who allowed us to record the quarry blasts, and Amoco Corporation which donated the recording equipment to Stanford.

The drillbit experiment was made possible by the kind permission of Unocal Corporation and the Japan National Oil Company (JNOC). I would like to thank Phil Johnston of Unocal for his assistance in arranging the experiment, and Leonid Vanyan of SEP for his help in planning and conducting it.

I would like to thank Sam Allen and Dan Hollis of Subsurface Exploration Company for their interest in the early stages of this work and for allowing me to conduct my first passive experiment.

I was privileged to work with and learn from many wonderful people at SEP. I’d particularly like to mention Martin Karrenbach, Matthias Schwab, Joe Dellinger, Marta Woodward and Diane Lau.

Finally I wish to thank my family, especially my parents, Florence and George, and my
friends Margaret, Scott, Alan, Jeff, and Anna Douglass, for their loving support throughout my lengthy graduate career.
# Table of Contents

Preface iv

Acknowledgments vi

1 Introduction 1
   1.1 Experiment design ........................................... 2
   1.2 Quarry blasts ............................................... 3
   1.3 Data overview ............................................... 5
   1.4 Partial stacking ............................................ 6

2 Beam steering of passive seismic data 21
   2.1 Beam steering procedure ........................................... 21
   2.2 Quarry blast recordings ......................................... 22
   2.3 Near-vertical events ............................................ 25
   2.4 Possible sources of near-vertical events ....................... 29
   2.5 Decimation tests ............................................... 35

3 Correlation and scattering analysis of passive seismic data 42
   3.1 Autocorrelation analysis ........................................... 43
   3.2 Scattering analysis ............................................. 52

4 A drill-bit source experiment 59
   4.1 Locating sources in 3-D ............................................ 62
   4.2 Velocity analysis ................................................. 62
      4.2.1 Velocity analysis at fixed surface positions ................. 66
   4.3 Source signal estimation and crosscorrelation .................. 68
A 3-D Interpolation Applied to Passive Data

A.1 Interpolation ................................................. 72
   A.1.1 Implementation ........................................... 73
   A.1.2 Synthetic example ....................................... 74
   A.1.3 Quarry blast example .................................... 76
   A.1.4 An alternate approach - plane fitting .................. 80
   A.1.5 Application: removal of cross-line smear .............. 85

B Xtpanel - an interactive panel builder 86
   B.0.6 Object values ............................................. 87
   B.0.7 Object names ............................................. 87
   B.0.8 Object actions ............................................ 87
   B.1 Examples ................................................... 88
      B.1.1 Simple buttons ......................................... 88
      B.1.2 Simple menu ............................................ 89
      B.1.3 Simple interactive parameter selection ................ 89
      B.1.4 Simple seismic processing pipeline .................... 90
      B.1.5 SEP data cube viewer .................................. 92
      B.1.6 Directory lister .......................................... 93
      B.1.7 Advanced seismic processing pipeline .................. 95
      B.1.8 ed1D ..................................................... 99
      B.1.9 Calculator ............................................... 99
      B.1.10 The xtpanel generator ................................ 100
      B.1.11 Interactive help facility .............................. 100
   B.2 Comparing xtpanel and other products ...................... 101
   B.3 Interactivity with xtpanel ................................ 102

C Bibliography 107
# List of Figures

1.1 Map of San Francisco Bay Area, showing location of experiment. .............. 3  
1.2 24 element square array proposed for the experiment. ......................... 4  
1.3 Overview of 13 by 13 geophone array. ........................................ 4  
1.4 Amplitude spectra of passive data. .............................................. 5  
1.5 Amplitude spectra after normalization. ........................................... 6  
1.6 Raw data for a typical night time record. ...................................... 7  
1.7 Continuation of previous figure. .................................................. 8  
1.8 Raw data for a day time record containing a quarry blast. .................. 9  
1.9 Continuation of previous figure. .................................................. 10  
1.10 Partial stacking geometry. ....................................................... 11  
1.11 Partial stack of night time data record 35, 0-8 seconds. ....................... 13  
1.12 Partial stack of night time data record 35, 8-16 seconds. ..................... 14  
1.13 Partial stack of night time data record 35, 16-24 seconds. ................... 15  
1.14 Partial stack of night time data record 35, 24-32 seconds. ................... 16  
1.15 Partial stack of day time data record 46, 0-8 seconds. ....................... 17  
1.16 Partial stack of day time data record 46, 8-16 seconds. ..................... 18  
1.17 Partial stack of day time data record 46, 16-24 seconds. .................... 19  
1.18 Partial stack of day time data record 46, 24-32 seconds. .................... 20  
2.1 Beam steering parameters. ....................................................... 22  
2.2 Single-trace displays of quarry blasts. ....................................... 23  
2.3 Schematic view of radial beam steering plots. .................................. 24  
2.4 Beam steering of all three quarry blasts. ..................................... 25  
2.5 Sequence of 100 msec beam steering plots for largest quarry blast. ........ 26  
2.6 Continuation of previous figure. ................................................ 27
2.7 Continuation of previous figure. ........................................... 28
2.8 Beam steering of full 32 second records, records 1-12. .................... 30
2.9 Beam steering of full 32 second records, records 13-24. .................... 31
2.10 Beam steering of full 32 second records, records 25-36. .................... 32
2.11 Beam steering of full 32 second records, records 37-48. .................... 33
2.12 Reordering data traces destroys near-vertical event coherence. .......... 34
2.13 Possible sources of near-vertical events in passive data. .................... 36
2.14 Plan view of possible source locations in near field. ....................... 36
2.15 Beam steering of records 1-12 after 2x decimation. ......................... 38
2.16 Beam steering of records 1-12 after 4x decimation. ......................... 39
2.17 Beam steering of records 1-12 after 8x decimation. ......................... 40
2.18 Beam steering of records 1-12 after 16x decimation. ...................... 41

3.1 Illustration of crosscorrelation scheme. ................................... 43
3.2 Synthetic dataset containing random plane waves. .......................... 44
3.3 Result of crosscorrelation, 1 plane wave. .................................. 45
3.4 Result of crosscorrelation, 5 plane waves. .................................. 46
3.5 Result of crosscorrelation, 100 plane waves. ................................ 47
3.6 Crosscorrelation of passive record 10. ..................................... 48
3.7 Crosscorrelation of passive record 15. ..................................... 48
3.8 Crosscorrelation of passive record 20. ..................................... 49
3.9 Crosscorrelation of passive record 25. ..................................... 49
3.10 Crosscorrelation of passive record 30. .................................... 50
3.11 Crosscorrelation of passive record 35. .................................... 50
3.12 Crosscorrelation of passive record 40. .................................... 51
3.13 Crosscorrelation of passive record 45. .................................... 51
3.14 Crosscorrelation of passive record 46. .................................... 52
3.15 3-D grid of source locations for scattering approach. ...................... 53
3.16 Moveout across the array versus distance from source. ................... 54
3.17 Scatterer semblance, depth = 0 meters. .................................... 56
3.18 Scatterer semblance, depth = 500 meters. .................................. 57
3.19 Scatterer semblance, depth = 1000 meters. .................................. 58

4.1 Array used for drillbit experiment. ........................................ 60
4.2 RMS amplitude of traces, with pump locations indicated. .......... 61
4.3 Semblance as a function of x and y at the drillbit depth. .......... 63
4.4 Semblance as a function of x and z. .................................. 64
4.5 Semblance as a function of y and z. .................................. 65
4.6 Semblance as a function of depth and velocity for a fixed surface location. .... 67
4.7 Crosscorrelation with drillbit source signature. ..................... 69
4.8 Crosscorrelation with incorrect drillbit signature. .................. 70
4.9 Stack of crosscorrelations for eight records. ....................... 71
A.1 Schematic for dip estimation step of interpolation. ................... 75
A.2 Combining traces in interpolation. ................................... 75
A.3 Synthetic interpolation, crossline plots. ............................. 77
A.4 Synthetic interpolation, inline plots. ................................ 78
A.5 One crossline plot from synthetic interpolation. .................... 79
A.6 Plot of synthetic interpolation picks. ................................ 79
A.7 Quarry blast interpolation, crossline plots. .......................... 81
A.8 Single crossline plot from quarry blast interpolation. .............. 82
A.9 Quarry blast interpolation, inline plots. .............................. 83
A.10 Single inline plot from quarry blast interpolation. ................. 84
A.11 Plot of quarry blast interpolation picks. .............................. 85
B.1 A simple panel with two buttons. ..................................... 89
B.2 A simple panel with a menu and a quit button. ....................... 89
B.3 NMO velocity selection panel. ...................................... 90
B.4 Velocity selection panel. ........................................ 91
B.5 A panel to control the execution of two seplib programs. .......... 92
B.6 A viewer to examine SEP data cubes. ................................ 93
B.7 A panel to traverse a Unix filesystem. ................................ 94
B.8 A panel to control the execution of three seplib programs. ......... 96
B.9 The documentation produced by pressing the Byte doc. button. ...... 96
B.10 Interactive parameter specification. .................................. 97
B.11 The result of pressing the action button. .......................... 98
B.12 Xtpanel version of Jon Claerbout’s program ed1D. ................. 99
B.13 A calculator built using xtpanel. ................................... 100
B.14 The xtpanel generator............................................. 101
B.15 The xtpanel interactive help facility.......................... 102
B.16 Interactive Cubeplot session..................................... 105
Chapter 1

Introduction

In September of 1988, the Stanford Exploration Project (SEP) conducted a passive seismic experiment. In a remote area of the Stanford campus, 4056 geophones were set out in a two-dimensional array. Ambient seismic energy was recorded over an eighteen hour period.

Several factors motivated us to perform such an experiment. First, the problem of observing and characterizing ambient seismic noise is an interesting one, particularly in a seismically active area such as this, where the San Andreas fault is only a few kilometers away. A two-dimensional receiver array allows us to determine the arrival direction of incident waves. A large number of channels offers the possibility to detect events weaker than those seen by isolated recorders.

Second, we hypothesized that whatever energy is present in the earth, we might be able to image structures in the near surface by observing how ambient energy interacts with these structures. Ambient energy could scatter or reflect off structures in the near surface, and we might be able to detect this by observing the pattern of arrival times across the array for various incident events. Work has been done along these lines by a group of Russian authors (Troitskiy and Nikolaev, 1987), (Troitskiy et al., 1981), using ambient seismic energy as well as aftershocks of large earthquakes. They report imaging large-scale scatterers deep in the crust. Hedlin et al (1991) use an analogous approach to image secondary sources in the coda of teleseismic events.

A third motivator was our hypothesis that the crosscorrelation of two passive traces should resemble what we might record from a shot at one location and a receiver at the other. Or that the autocorrelation of recorded traces would reveal the impedance structure of subsurface layering. Examples of work along these lines include studies by Scherbaum

The final motivator was the availability of the right equipment. In 1988, Stanford received from Amoco the generous gift of approximately 200 seismic group recorders (SGR’s) and associated hardware. SGR’s are portable units that record on magnetic tape the amplitudes of ground motion for a single seismic channel. This equipment has made it possible in recent years for Stanford to conduct groundbreaking experiments in crustal and earthquake seismology. But its first use at Stanford was in our passive seismic experiment. The arrival of this equipment gave us the opportunity to experiment with some of our ideas on passive seismology.

Interest in recording ambient seismic energy in the reflection seismic community has usually been aimed at understanding noise contamination problems, though there have been occasional references to imaging with passive seismic energy, such as the work done by Baskir and Weller (1975).

1.1 Experiment design

Some overriding logistical constraints guided the design of the experiment. We had roughly 200 seismic group recorders or SGRs. Each SGR records one channel of data on its own cassette tape. A controller sends timing signals to all the SGRs via radio to synchronize recording. We could sum together the outputs of 12, 24, or 36 geophones to produce the signal recorded at an SGR. These numbers arose because there are twelve geophones per string in our equipment (with 50 feet of cable between geophones) and we could have up to three strings per SGR before the combined resistance of the cables became too large.

A second logistical constraint was our survey area. We chose to conduct this experiment on Stanford land, rather than looking for sites elsewhere, to minimize the work needed to set up the array. The most appropriate site on Stanford property was a field roughly 500 by 500 meters. We filled this area with a 2-D grid of geophones space every 25 feet in each direction. Groups of 24 geophones were combined to form a group, as shown in Figure 1.2, giving 169 recorded channels in a 13 by 13 grid. The geophone spacing of 25 feet tells us that surface waves traveling at or near air velocity, which may not be unexpected in dry soil such as at this site, will be unaliased up to a frequency of 18 Hz. The 13 by 13 grid of geophone groups is shown in Figure 1.3. With this design, the group centers are 125 feet apart.
Approximately 200 man-hours were required to survey and lay out the array. We enlisted the full manpower of SEP, about 15 people, to do this in one weekend.

### 1.2 Quarry blasts

While most of our recording was done during the night to minimize interference from cultural noise sources, we also recorded some daytime records. The U.S. Geological Survey had planned to set off several blasts in a quarry in Cupertino, CA (about 15 km away) for a research project of their own, and they kindly scheduled the blasts so that we could try to record them with our array. The location of the quarry is shown in Figure 1.1.

There were three blasts, a large “quarry blast”, with 1500 pounds of explosives in many different shot holes (this blast is used by the quarry to dislodge rock) and two smaller single charges of 300 and 100 pounds that were set off for experimental purposes. USGS seismometers in the vicinity of our array clearly see the quarry blast (indeed, the quarry blast is visible for hundreds of kilometers) but not the smaller blasts.
FIG. 1.2. 24 element square array proposed for the experiment.

FIG. 1.3. Overview of array. The array contains 169 groups, with 24 geophones per group, for a total of 4056 geophones.
1.3 Data overview

In Figure 1.4, amplitude spectra (computed by averaging over all the recorded traces) are shown for each of the 48 data records. In general daytime records (records 42 and above) are noisier, as expected. Isolated nighttime records have strong noise levels, probably due to vehicles passing on the nearby freeway. In Figure 1.5, the spectra are normalized so that each has the same RMS level. From this it can be seen that certain low frequencies are consistently present during the nighttime records. These may be due to site amplification of incident waves. The strong energy above 20Hz on record 46 is due to the largest of the three quarry blasts.

FIG. 1.4. Amplitude spectra for the 48 records taken during the survey.

Figures 1.6 through 1.9 show the data for two of the 32 second records. Figures 1.6 and 1.7 are for a record taken during the night. Figures 1.8 and 1.9 are for a record taken during the day, timed to record a blast set off by the U.S. Geological Survey in a quarry approximately 15 km from the array.
Partial stacking

Stacking is a simple and effective method to enhance the useful signal and eliminate noise. In a conventional reflection experiment we use data redundancy to enhance the reliability of the data. For our passive data we can point the array in a particular azimuth direction by stacking the data along the lines perpendicular to that azimuth direction. In these examples we choose the inline and crossline directions as well as the two diagonal directions as shown in Figure 1.10. Stacking along this set of profiles was suggested by my colleague Lin Zhang.

Let \( P(it, ix, iy) \) be the recorded 3-D data. The equations for stacking are

\[
P_1(it, ih) = \sum_{ix} P(it, ix, ih + ix)
\]

\[
P_2(it, ih) = \sum_{ix} P(it, ix, ih - ix + nx)
\]

This operation has the following advantages and disadvantages:

- Advantages
FIG. 1.6. Raw data for a typical night time record.
FIG. 1.8. Raw data for a typical day time record. The largest quarry blast is visible, beginning at around 9.5 seconds.
FIG. 1.10. Partial stacks for the two raw data records shown are above are displayed in Figures 1.11 through 1.18. Interestingly, the stack events (for example, at around 2.3 seconds in Figure 1.11) that were not apparent in the raw data. The labels A, B, C, and D on the plot refer to the profile directions shown in Figure 1.10. These four profiles are synthesized...
by partially stacking the data onto these four lines. By comparing the apparent dips of events on the four profiles, one can get a rough idea of where a particular event is coming from. A more precise tool for analyzing arrival directions will be developed in Chapter 2.

In Chapter 2, I use more sophisticated techniques to determine the arrival direction of incident waves such as the one revealed in Figure 1.11 by partial stacking.

In Chapter 3, two techniques, a scattering method and a correlation based approach, will be used to attempt to image the subsurface using passive data.

In Chapter 4, the techniques developed for passive data work will be applied to another novel seismic source, the drillbit. Passive seismic work and drillbit studies are united by the common feature that in these case, unlike conventional controlled source seismology, there is no time zero at which a source is set off. Thus the processing techniques developed for passive work will also be applicable to the drillbit case. One advantage in the drillbit case is that the source is at a known position.

In Appendix A, I develop tools for interpolation of 3-D data that have been useful in dealing with these datasets. In Appendix B, I describe xtpanel, a tool for building interactive X Window system interfaces that I developed along with Dave Nichols. It is used frequently at SEP for adding interactivity to the CD-ROM version of documents such as this thesis. Thesis figures have a button in the caption when viewed on the computer. Clicking on this button brings up an interactive panel where the user can adjust parameters and reproduce the figure.
FIG. 1.11. Partial stack of data from the left side of Figure 1.6. Note the nearly flat event at around 2.3 seconds that is not visible in the raw data of Figure 1.6.
FIG. 1.12. Partial stack of the data from the right side of Figure 1.6.
FIG. 1.13. Partial stack of the data from the left side of Figure 1.7.
FIG. 1.14. Partial stack of the data from the right side of Figure 1.7.
FIG. 1.15. Partial stack of the quarry blast data from the left side of Figure 1.8.
FIG. 1.16. Partial stack of the data from the right side of Figure 1.8. Note the various blast arrivals.
FIG. 1.17. Partial stack of the data from the left side of Figure 1.9.
FIG. 1.18. Partial stack of the data from the right side of Figure 1.9.
Chapter 2

Beam steering of passive seismic data

In this chapter, I present a technique that tries to improve upon the partial stacking results shown in the introduction, and give clearer pictures of where events present in the passive data are coming from. This technique is beam steering; semblance is computed as a function of plane wave arrival direction. The geometry of the problem is illustrated in Figure 2.1. Beam steering is applied here to the quarry blasts, and to the nighttime records where partial stacks showed evidence of near vertically incident events.

2.1 Beam steering procedure

Aki and Richards (1980) show that the arrival time of a plane wave moving with apparent surface velocity $c$ and arriving from a direction specified by an azimuthal angle $\phi$ at the $i$-th station of a seismic array is given by:

$$t_i = t_0 + \frac{\cos \phi}{c} (x_i - x_0) + \frac{\sin \phi}{c} (y_i - y_0) + \tau_i$$

where $(x_i, y_i)$ are the coordinates of the receiver, $t_0$ is the arrival time of the wave at a reference point $(x_0, y_0)$, and $\tau_i$ is the station residual. Thus to form a beam in a particular direction with a given velocity, we apply the time shifts prescribed by this equation, and stack the traces together. Also it is helpful to sum the resulting stack over short time windows, to reduce the effect that random noise has on the beams.
FIG. 2.1. Beam steering parameters. Incident plane waves are characterized by azimuth angle and either the dip angle or apparent velocity.

Note that the formula does not depend on the dip angle of the arriving wavefront. This is due to the ambiguity between velocity and dip angle for a two-dimensional array. A given apparent surface velocity could be due any of a number of combinations of dip angle and medium velocity.

2.2 Quarry blast recordings

Figure 2.2 shows portions of the seismograms recorded by one of our 169 groups for the three blast records. The quarry blast appears on the middle trace at around 10 seconds, but is not very obvious on a single-trace display. The smaller blasts arrive around 10.5 seconds on the other two traces, and are not at all apparent. The signal-to-noise ratio in our data, at least for daytime recording, is obviously quite small.

We were lucky to record these blasts at all. Our recording equipment consisted of 169 seismic group recorders (SGRs), each recording data on its own cassette tape and powered by a battery. The rechargeable batteries have a lifetime of about a day of normal operation. They worked fine for our nighttime recording, but had to sit using power until the middle of the next day for the blast recordings, and battery failure began to take its toll. About half the SGRs were still working for the first blast at 11 AM, and only 38 of
FIG. 2.2. Traces from the same geophone group for three different recording periods, parts of the three daytime records where blasts were set off in a nearby quarry. The largest blast is the event on the middle trace arriving at around 10 seconds. The smaller blasts are not readily distinguishable on these single-trace displays. The 169 were still working for the third blast at 12 Noon.

While the blasts are not readily apparent on individual traces, beam steering has been a very useful tool for detecting and locating them. Figure 2.3 gives a schematic view of the disc-shaped beam steering plots which will be used frequently throughout this chapter. Stacking semblance is displayed as function of arrival direction (azimuth) and apparent slowness in polar coordinates. An apparent slowness of zero, corresponding to vertically incident events, is plotted at the center. The highest slowness value, at the edge, corresponds on these plots to an apparent velocity of 2 km/sec. These plots could also be described in terms of ray parameters $p_x$ and $p_y$ as shown in the figure, illustrating the equivalence of beam steering and slant stacking.

Figure 2.4 shows the result of beam steering the data from the three blasts and summing over 100 msec windows centered around the first arrivals from the blasts. It can be seen that the three blasts arrive from the same direction, all with an apparent velocity of around 4.5 km/sec. I gained the three panels independently so that the three blasts would have the same relative strength. In fact the blast from the middle panel, the quarry blast, gives a much higher semblance value than the other two. If we were to perform the
FIG. 2.3. Beam steering parameters. Different azimuths are displayed around the circle, and different apparent velocities in the radial direction. A vertically incident event (infinite apparent velocity) would be plotted at the center.

same summation over time as shown in Figure 2.4, but using the entire 32 second records instead of a small window, the quarry blast would still dominate the middle plot, while the two smaller blasts would no longer stand out. Restricting the summation to a small window located at the right time has enabled us to see all three blasts.

While the blasts arrive in a consistent direction, that direction surprisingly differs considerably from the direction of the quarry, as can be seen by comparing these plots and the quarry direction indicated on the map shown earlier. The difference is on the order of 45 degrees. One possibility is that these first arrivals have traveled along a path that does not follow a straight line from the quarry to the array. The local geology gives some support for this theory. A direct path from the quarry to our survey would pass through large amounts of sandstone and conglomerate, while a path that initially turned further to the west and then came back to our array would remain for the most part in a faster, more well-consolidated basalt formation that parallels the San Andreas fault, which passes about 5 km to the southwest of our survey area. Another possibility is that the energy follows a more direct path and is then rerouted by the near surface, for instance by a tilted weathered layer beneath the survey.

Another interesting fact to note about the blasts is that there is strong evidence of
FIG. 2.4. Beam steering result for 100 msec windows centered on the first arrival of energy from three blasts in a quarry 15 km distant from the array. Refer to figure 2.3 for an explanation of the individual plots, and compare to the map in Chapter 1 to relate these directions to the survey geometry. All three blasts arrive in the same direction with apparent velocities around 4-5 km/sec.

energy from the blast following different paths to reach our array. Figures 2.5 though 2.7 shows many 100 msec frames from the large quarry blast following the first arrival shown in Figure 2.4. All these frames have been gained identically, so the later arrivals do give semblance values that are quite high. Before the blast arrives, semblances are all much smaller. Thus we can assume that all the strong events shown in Figures 2.6 and 2.7 are due to the blast. Note that this late-arriving energy comes in a variety of directions, some of it arriving in a direction much closer to the direction of the quarry. One possibility is that the earliest arrivals traveled through the faster basalts to the west, and then later energy traveling in the straight-line direction arrived. The frame at time 14.1 seconds has a fairly high semblance value right at the center, indicating vertical incidence. This is possibly a reflection off the moho.

2.3 Near-vertical events

One of the most interesting features of the dataset, discovered earlier by partial stacking, was the presence of near-vertically incident events in the nighttime records. These events are weak and not noticeable on individual seismograms. But partial stacking displayed them quite clearly.

Our first thought was that since the events are almost vertically incident, they must be due to some electrical interference. More careful beam steering now suggests that this
FIG. 2.5. Frames showing the result of beam steering for 100 msec windows following the first arrival shown in the middle frame of Figure 2.4. This is the largest of the three quarry blasts. The energy in these frames is much stronger than that in the rest of the 32 second record, so all this energy must be due to the quarry blast, suggesting multiple travel paths or scattering.
FIG. 2.6. The continuation of Figure 2.5. Slower arrival from quarry direction
FIG. 2.7. The continuation of Figure 2.6.
is not so; these events seem to have a consistent direction of propagation that is close to vertical but not quite vertical. If that is so, they cannot be due to electrical noise.

Figures 2.8 through 2.11 show the result of beam steering the 48 different records in our survey. The entire 32 second records were used to produce each of these plots. The dominant feature of most of them, near the center, is the near-vertically incident events. While these events are close to the center, they are consistently just off-center. Their azimuth and apparent slowness indicate a fairly consistent arrival direction, from the east, with an apparent velocity of around 12 km/sec.

To illustrate why I think the events are not due to electrical interference, I took one data record and randomly re-ordered the positions of the 169 traces in our array. If the events are real, and not perfectly flat, then re-ordering the traces in this way should effectively eliminate them. Figure 2.12 shows the result of beam steering the re-ordered traces, along with the original beam steering result for that record. There are some strong semblance values near the center, but they are not as strong as the values in the original beam steering result. More importantly, they are no longer in the same place, but are mostly clustered around the center. This is precisely what we would expect to happen; the semblances are not going to drop to zero because there was strong, near-vertically incident energy present. After re-ordering it will still stack in to some extent. But the directional consistency should be, and is, lost. Thus it is clear that these events are not due to electrical interference.

### 2.4 Possible sources of near-vertical events

Given that we have ruled out electrical interference, what could be the cause of such events? One possibility is that these are not seismic waves travelling in the Earth but sound waves travelling in the atmosphere, incident on the array from above. Two factors suggest that this is unlikely. First, the steep angle of incidence implies that such a source would be located within 2 degrees of the zenith, almost directly overhead:

\[
\arcsin \frac{0.3 \text{ km/sec}}{10 \text{ km/sec}} = 1.7^\circ
\]  

Second, the source would have to be stationary throughout the experiment, which spanned an 18 hour period, in order to mimic this seismic response.

Having ruled out electrical interference and atmospheric disturbances leads us to conclude that these events are seismic in origin. We would like to know where the source of
FIG. 2.8. Result of beam steering for 12 different nighttime records (records 1-12). The feature consistent from plot to plot represents near-vertically incident events, arriving from the east with an apparent velocity of around 12 km/sec.
FIG. 2.10. Result of beam steering for records 25-36.
FIG. 2.11. Result of beam steering for records 37-48. Records 42 and above were recorded during the day.
FIG. 2.12. One of the records from Figure 2.8 (left) and the result of applying the same processing to data where the trace positions have been randomly re-ordered (right). The re-ordering has removed the high-ssemblance values seen on the left, verifying that the high ssemblance values near the center of the left plot were due to actual events and not some sort of interference.

these events is located. There are two possibilities, which were illustrated in Figure 2.13. One is directly beneath the array, in the upper 15 km or so of the crust where earthquakes occur. This is the cone-shaped region in Figure 2.13. The orientation of the cone is indicative of the observed arrival direction of the incident waves. The size of the cone is due to the fact that our resolution in apparent velocity space was somewhat limited; the near-vertical events arrived at between 8 and 12 kilometers per second.

A problem with the near source hypothesis is that the steep angle of incidence restricts the location of such sources to a relatively small zone located beneath the array. This would mean either that we were very lucky in choosing the site for our array, or that such small-scale seismic activity occurs in many places.

The second possibility is that this energy is from a more distant source, and that the energy has followed a raypath similar to that shown in Figure 2.13, travelling down to some depth, then turning toward the surface. A large distance is required to explain the very steep angle of incidence. While quarry blast energy from 15 kilometers away reached the array at an apparent velocity of 4-5 kilometers/second, energy arriving at the much steeper angle implied by 8-12 kilometers/second must have come from a much more distant source. An apparent velocity of 8 km/sec corresponds to a source at a distance of approximately
1000 kilometers for P waves. An apparent velocity of 12 km/sec corresponds to a distance of 3000 kilometers.

A problem with the distant source hypothesis is that a source hundreds of kilometers away would have to be quite strong, or a distributed source that encompasses a wide area, to be detected at this distance.

Lateral velocity variation can invalidate almost any seismological analysis including those above. A strong $v(x, z)$ velocity gradient could mean this energy came from somewhere other than the two regions shown in Figure 2.13. The quarry blasts did reveal evidence of lateral velocity gradients. These gradients affected the azimuth angle of the incident waves, changing some by as much as 45 degrees, but had little effect on their apparent velocities. Following the same logic, the near-vertical events could have been precursors to the October 1989 Loma Prieta earthquake, whose epicenter was approximately 60 kilometers to the southeast of this array, whose arrival direction has been changed by strong lateral velocity gradients near the array. In addition to changing the azimuth, however, the gradient would need to change the apparent velocity from 8 or more km/sec to near 6.5 km/sec, which is the apparent velocity for a P wave source at a distance of 100 kilometers. There is no evidence that velocity gradients at this site are that strong.

Having two or more of these arrays separated by some distance would allow us to triangulate and reduce the uncertainty in determining where these events are coming from.

### 2.5 Decimation tests

Given that the 169 channel, 4056 geophone array is able to analyze the arrival direction of incident waves, an important question is, how much of this array was required in order to do this work? Can the same results be obtained with half as many geophones and channels, or even fewer?

To answer this question, I performed a series of tests where I decimated the data records, keeping one out of every 2, 4, 8, or 16 channels. Figures 2.15 through 2.18 show the beam steering results for a set of 12 records (the same records shown in Figure 2.8) at these various decimation levels.

The results do not seem to be severely affected when only half the available channels
FIG. 2.13. Illustration of possible sources of near-vertical events seen in passive data. Events could originate in a relatively narrow zone beneath the array, or from a larger, more distant zone. [passive-ellipses] [NR]

FIG. 2.14. Close-up of map of survey area showing approximate boundaries of the near-field zone of possible source locations that could have generated the events seen in the data. [passive-mapzoom] [NR]
are used, but further decimation has an adverse effect. Using only every four channels, the near-vertically incident feature is not present on a large number of channels. From this result, we can conclude that for an array of this spatial extent, the minimum number of channels needed is on the order of half the number we deployed, which would be 85 channels compared to 169 in our array. Below that number, a significant decline in directional resolving power occurs.
FIG. 2.15. Beam steering results for records 1-12, with only half of the channels used.

[passive-decim.2.1] [R]
FIG. 2.16. Beam steering results for records 1-12, with only one out of every four channels used. [passive-decim.4.1] [R]
FIG. 2.17. Beam steering results for records 1-12, with only one out of every eight channels used. [passive-decim.8.1] [R]
FIG. 2.18. Beam steering results for records 1-12, with only one out of every sixteen channels used. [passive-decim.16.1] [R]
Chapter 3

Correlation and scattering analysis of passive seismic data

In this chapter, I use two methods to attempt to image subsurface layers from passive data. The first is a correlation-based approach. My advisor Jon Claerbout hypothesized that the crosscorrelation of two seismic noise data traces should resemble what we would record from a shot at one trace location and a receiver at the other. This idea is illustrated in Figure 3.1. For each source-receiver offset, there is a certain arrival direction for incident plane waves that will follow the source-receiver raypath. If that incident plane wave is present, then that segment of the traveltime hyperbola will be present.

A synthetic data example illustrates how this method works. In Figure 3.2, three synthetic datasets are shown. One contains a single dipping plane wave, another five plane waves, and the third contains one hundred plane waves. The dips of the plane waves have been chosen randomly. A different arbitrary source signature has been convolved onto each event, by choosing a random source amplitude for each frequency component. This simulates the random dips and source signatures that we might expect to encounter in the ambient noise field.

To process the data, we crosscorrelate each trace against all other traces. This gives, in effect, the traveltime as a function of trace-to-trace offset. We sort the traces according to this offset, and perform a partial stack into a number of offset bins. This gives a profile, on which we should see the normal moveout hyperbola. In Figure 3.3, with a single plane wave, only a single offset of that hyperbola would be present. But as we add more plane waves, we illuminate more source-receiver offsets. In Figure 3.4, five different incident
angles are present, and an event at the right zero-offset time having moveout that appears to be hyperbolic is visible. This event is more sharply defined in Figure 3.5, with one hundred different plane waves present.

Real data results for a number of records are shown in Figures 3.6 through 3.14. In most cases there is little evidence of the sort of hyperbolic events we would expect, and those features that might be these events do not seem consistent between records as we would expect.

The lack of such hyperbolic features is disappointing. However, the success in the synthetic case depended partly on the presence of a flat subsurface layer. We know that the geology of our site, which is in close proximity to the San Andreas fault, is quite complex. This method should be tested in an area with known flat layers, to see if it can be beneficial.

### 3.1 Autocorrelation analysis

Another correlation-based approach would be to simply autocorrelate the traces. For a plane-layered earth, autocorrelation of the traces should reveal the subsurface impedance structure, as given by the Kunetz-Claerbout relation.

This analysis was also performed on the passive data. The results are not shown here because while there was some coherency in the autocorrelations from trace to trace in a
FIG. 3.2. Synthetic datasets contain 1, 5, and 100 plane waves with randomly selected dips and source signatures.\[\text{corr-pmod}\] [R]
FIG. 3.3. Result of crosscorrelation processing for a single dipping plane wave. Each trace is correlated against every other trace, and then the traces are binned and partially stacked according to the inter-trace distance.
FIG. 3.4. Result of crosscorrelation processing for the data containing five dipping plane waves. [corr-pst.5][R]
FIG. 3.5. Result of crosscorrelation processing for the data containing one hundred dipping plane waves. An event can be seen at the proper zero-offset time (0.15 seconds) having roughly hyperbolic moveout. correlation, 100 plane waves
FIG. 3.6. Result of crosscorrelation processing for data record 10.

FIG. 3.7. Result of crosscorrelation processing for data record 15.
FIG. 3.8. Result of crosscorrelation processing for data record 20.

FIG. 3.9. Result of crosscorrelation processing for data record 25.
FIG. 3.10. Result of crosscorrelation processing for data record 30.

FIG. 3.11. Result of crosscorrelation processing for data record 35.
FIG. 3.12. Result of crosscorrelation processing for data record 40. [corr-pstack.40] [R]

FIG. 3.13. Result of crosscorrelation processing for data record 45. [corr-pstack.45] [R]

given data record, there was little agreement between records. This suggests that none of the coherent events observed in the autocorrelations are related to the subsurface geology.

3.2 Scattering analysis

The second technique is a scattering-based approach. Here we picture the subsurface near the array as a three dimensional grid of possible source locations, as shown in Figure 3.15. At each of these locations, given an estimate of the necessary stacking velocity, we stack or compute semblance along the moveout trajectory.

Scatterers near the array should act as secondary sources. While we can’t image these sources with conventional migration since we don’t know the time at which they “explode”, we should be able to see in the data the moveout patterns that identify the source location.

A similar scheme was used by Troitskiy and Nikolaev (1987) to image scatterers using teleseismic data and ambient noise data with the NORSAR array.

We create a 3-D grid of possible scatterer locations in the subsurface. For each location, we find the moveout trajectory for energy coming from that point, and compute semblance
over time along this trajectory in our data. We can then look at the average semblance over time, or the maximum semblance observed over all times, to get an idea of the scatterers that may lie in the subsurface.

An important first question is, given the size of our array, how far away can scatterers be identified as such? The limited size of the array will make it impossible to see any moveout for events coming from scatterers beyond a certain distance. Beyond this distance, scattered energy will be indistinguishable from plane waves. Troitskiy and Nikolaev were using the NORSAR array, 110 km across, and was therefore theoretically able to image very deep structures. Our array is only 500 meters across, so our scattering analysis will be more localized. Figure 3.16 shows the moveout across the array for a scattered event versus distance of the scatterer from the array. This figure assumes a constant velocity of 2000 meters/sec, a very favorable choice. For higher velocities the dropoff will be sharper. The time difference drops to less than four milliseconds at around 5 kilometers from the array, so our scattering analysis definitely cannot go beyond that point. I have chosen one kilometer as an upper bound in the analysis that follows, to avoid coming too close to the limit.

Figure 3.17 shows the image that is obtained for the \( z = 0 \) depth level when this
FIG. 3.16. Moveout observable across our array for a scattering event as a function of scatterer distance from the array. Beyond about 5 kilometers the observable moveout drops to less than the time sampling interval, so scattered events will be indistinguishable from plane waves.

The scattering algorithm is applied to portions of several different nighttime and daytime records. The x and y dimensions of the grid of scattering locations shown are three times the size of the survey. Thus points up to 750 meters from the array center are shown in this plot. The dominant features in this plot are the radial streaks. These are due to the fact that as we get further from the array center, our summation path becomes flatter and flatter. Eventually we reach the point where we are summing along a plane wave path, and from there out we would just get a radial streak.

Figures 3.18 and 3.19 show the results for depths of 500 and 1000 meters. A feature worth noting in all of these is the general agreement among the daytime blast records, and among the nighttime records, but not between the two groups. The reason for this is that what we are seeing when we sum along these hyperbolic paths is, predominantly, plane wave energy, which was quite different during day and night recording periods. In Figure 3.19, I've performed the same computation as for Figure 3.19, but I've summed along plane wave paths rather than hyperbolic paths. The fact that these two plots are quite similar is disappointing; it means that the algorithm is seeing mainly plane wave energy imperfectly stacked along hyperbolic paths, rather than scattered energy.
If we are to see scattered energy, probably it is necessary to filter out the strong plane waves present throughout our dataset.

For the most part, I believe that what we are seeing with this analysis is plane wave energy stacking is somewhat because its trajectory is close enough to that being used for scatterers. I base this conclusion on the fact that most features track outward in radially from the center of the zero depth plot. The passive data seem to be dominated by events coming from the far field. Some technique that removes plane waves might allow us to do a better job of scattering analysis. But this method is very useful in the drill bit case described in the following chapter.
FIG. 3.17. Semblance measure of scatterer strength as a function of position for a depth of 0 meters. The different frames correspond to different recording periods. The top four panels are nighttime records; the bottom four are daytime records. Of these, all but record 45 contain blasts. These plots cover a 1500 meter square area centered around the array.
FIG. 3.18. Same as Figure 3.17 but for a depth of 500 meters.
FIG. 3.19. Same as Figure 3.17 but for a depth of 1000 meters.
Several authors have used the drill bit as a seismic source. In one type of study (Rector et al., 1988), an accelerometer is placed on the drill string, giving an estimate of the signature of the drill bit source. Crosscorrelating this estimate with the data recorded by one or more geophones enhances drill-bit energy in the recorded traces and gives results whose quality is comparable to VSP data.

A second study (Katz, 1980) located the drill-bit signal using an eight channel array, with no accelerometer on the drill string, for the purpose of guiding the drill bit during drilling.

In a third study, Kostov (1990) obtained an estimate of the drill-bit source signature from 2-D seismic data. A 1-D array containing up to 90 geophones was used, and the drill-bit signal was separated from other noise sources by virtue of its spatial coherency. While Kostov was able to detect direct arrivals from the drill bit, the strongest noise sources seen were surface sources located off the line, which could not be suppressed with the 1-D array. These noise sources complicated processing considerably.

A 2-D array with a large number of channels, such as that used in the passive seismic experiment of the previous chapters, offers improved ability to determine the arrival direction of incident energy and to suppress unwanted noise sources. In November 1990, I used a 240 channel 2-D array to record data during the drilling of a well in a producing oilfield in Wyoming. A diagram of the array is shown in Figure 4.1. The 240 channels were selected from a larger array of 480 channels that was previously used to study ambient noise in the same oilfield (Cole and Vanyan, 1990). The array consisted of 11 lines of geophones, with an average inline spacing of 100 feet and an average crossline spacing of
150 feet. On one line in the middle of the array a smaller spacing of 50 feet was used.

FIG. 4.1. 240 channel array used for drill-bit study. Dots are geophone locations, asterisk at top is location of drilling rig. Geophones were cemented in place at a depth of 30 feet.

Data were recorded over a period of three days, during which the drill bit covered a range of depths from approximately 800 to 1400 feet. Operational constraints limited us to this fairly narrow range. A total of 64 records, each 99 seconds long and with a sampling interval of two milliseconds, were taken during that time. During most of the recording time, many strong surface noise sources were active, including drilling-related equipment and 10 to 12 producing pumps located throughout the array. We were fortunate to have been able to turn off these pumps at one point during the recording. Eight records were taken during this quiet period. As will be seen later in the paper, first results from this quiet period are much superior to the results obtained from other records.

In this paper, I begin the study of the data by searching for sources of energy in 3-D. I construct a cube of possible source locations and then, given an estimate of the velocity model, stack the data (or compute semblance) along the moveout trajectory corresponding to each source location. To begin, I use a constant velocity. When applied to data from the “quiet” period where pumps located within the array were turned off, this technique reveals what appear to be direct arrivals from the drill bit. Two different records from that time period give a consistent picture. In contrast, data from outside the quiet period
FIG. 4.2. RMS amplitude for a portion of one data record as a function of receiver location. Asterisks mark the locations of pumps, a number of which were active during this record.

Present a noisy background against which it is not possible, at least with this simple method, to detect the drill bit signal.

While stacking with a constant velocity appears to do a good job, it would be useful to obtain velocity information directly from the data, rather than arriving at a good stacking velocity by trial and error. A velocity analysis can be performed at the drill rig location; I scan for hyperbolic events whose tops lie beneath the surface location of the drill rig. The parameters are stacking velocity and the depth of the source. This analysis also reveals the drill bit signal, when applied to data from the quiet recording period, and concludes that the constant velocity value that seemed to position the drill bit energy well in 3-D was in fact a good choice.

Finally, having found via constant velocity stacks what I believe to be energy from the drill bit, I stack the data along the correct moveout trajectory to come up with an estimate of the drill bit source signature, analogous to the source signal estimate obtained by Rector et al. from an accelerometer placed on the drill string. Crosscorrelation of the data with this estimate enhances drill bit energy, preserves the moveout present in the data, and compresses the continuous-time drill-bit signal to a narrow window centered around the zero lag of crosscorrelation.
4.1 Locating sources in 3-D

To search for sources (such as the drill bit) in 3-D, I first construct a cube of possible source locations. Then, given an estimate of the velocity structure, I compute the moveout trajectory for energy arriving at the array from each possible source. I compute semblance over this moveout trajectory and examine the resulting 3-D semblance cube.

Figure 4.3 shows constant-depth slices from such a cube, taken at the known depth of the drill bit. A constant velocity of 10000 ft/sec has been used to compute moveout trajectories. The four frames correspond to four different data records; the two top records are from the “quiet” period where pumps within the array were turned off. In the case of the bottom two records the pumps were operating. The asterisk denotes the surface position of the drilling rig. When the pumps are turned off, the largest semblance values (shown in white) occur at the drill bit location. These semblance values are likely due to direct arrivals from the drill bit.

Figures 4.4 and 4.5 show constant y-coordinate and constant x-coordinate slices through the same cube, for the same four data records. The agreement with the drill bit location in all three dimensions strengthens the claim that these large semblance values are due to direct arrivals from the drill bit. The resolution in space is not very good because only frequencies up to 40 Hertz have been used. Higher frequencies degrade the pictures—probably not because the drill bit signal does not contain higher frequencies but rather because the moveout of the signal departs from a constant-velocity hyperbola, and the higher frequencies are more sensitive to mis-stacking.

4.2 Velocity analysis

Rather than selecting a good stacking velocity by trial and error, it would be better to perform a velocity analysis. Velocity analysis proceeds much like the search for sources in 3-D, except now we scan over velocity in addition to scanning over the three spatial dimensions and over time.

The result of velocity analysis is a five-dimensional data volume. This means that the technique is expensive. As an example, using 20 points in x and y, a single depth, 30 velocities, and records containing 4000 time samples and 240 input channels requires roughly 80 minutes of cpu on a Convex C-1. More importantly, the five-dimensional volume means that the result will be difficult to display and interpret. We can easily
FIG. 4.3. Stacking semblance as a function of x and y coordinates at the drill-bit depth for four different data records. The largest semblance values are shaded in white. For the top two records, taken while pumps within the array were turned off, the largest semblance values are at the drill bit location, suggesting that we are able to see direct arrivals from the drill bit. For the bottom two records, where the pumps were operating, the results are much poorer. Note that the top and bottom plots are scaled differently for comparison: the largest semblances in the top plots are nearly three times those on the bottom.
FIG. 4.4. Stacking semblance as a function of x and z coordinates for a plane passing through the drill bit location. The asterisk denotes the location of the drill bit; note the change in depth between the top and bottom pictures. The largest semblance values, shaded in white, coincide with the drill bit location for the two records from the recording period where pumps within the array were turned off.
FIG. 4.5. Stacking semblance as a function of $y$ and $z$ coordinates for a plane passing through the drill bit location. The asterisk denotes the location of the drill bit. As in the previous two plots, the largest semblance values correspond to the drill bit location, further suggesting that the semblance values are due to direct arrivals from the drill bit.
remove one dimension from the result by performing semblance computations over a single window whose length is that of the entire trace. For sources that should be operating continuously over time, this is a sensible thing to do.

To further simplify interpretation of the results, we can hold one or more parameters fixed while the others are varied. This approach is taken in the following section. I hold \( x \) and \( y \) fixed to the known location of the drill bit (and other sources) and compute semblance as a function of stacking velocity and source depth.

### 4.2.1 Velocity analysis at fixed surface positions

We know not only the drill-bit location, but also the locations of other sources. Sixteen pumps (some not operating during this experiment, others operating intermittently) and one steam injection well are also located within the bounds of the array. We can simplify velocity analysis by limiting ourselves to hyperbolic trajectories whose tops are below these points.

In this section, I perform velocity analyses at the surface locations of the drill rig, the steam injection well, and three pumps. This is done for two different data records – one where the pumps were operating, and one where they have been temporarily turned off.

Figure 4.6 shows the result of velocity analysis at the drill rig location. Again the largest semblance values are shown in white. The top 1% of semblance values are clipped by the plot program and shown in black. As in the case where a constant velocity was used to compute semblance in 3-D, here we get a clear picture of what I believe to be direct arrivals from the drill bit. The horizontal line on the plots indicate the depth of the drill bit. This line intersects the main feature of the plots (for the records from the “quiet period”) near the constant velocity of 10000 feet/second that we found in the previous section to do a good job of stacking. As in the previous section, when the pumps are operating there is no good evidence of drill bit signal.

The dominant features in most plots dip toward lower velocities at greater depths. This feature can easily be explained. Energy from a source at great depth will exhibit very little moveout across the array. The same moveout can be expected from shallower sources only if the medium velocity is correspondingly higher. There is a tradeoff between depth and stacking velocity, and this tradeoff explains the dipping features. This tradeoff is a necessary consequence of the fact that our source is operating continuously over time rather than being impulsive.
FIG. 4.6. Semblance as a function of velocity and depth for a fixed surface location, the point above the drill bit. The depth of the drill bit is shown on each plot by the horizontal line. Larger semblance values are shaded white, with the largest 1% clipped and shown in black. The top plots are for records where pumps within the array have been turned off. In this case the direct arrivals from the drill bit are easily seen. Pumps were still operating during the recording of the two records shown at bottom. The result is that there is no clear drill-bit signal.  

[drillbit-vzd] [CR]
4.3 Source signal estimation and crosscorrelation

Once we have found the position of the drill bit and the velocity that best stacks its energy, we can stack the data along the appropriate moveout trajectory to obtain an estimate of the drill bit source signal. This is analogous to the signal from the accelerometer placed on the drill string by Rector et al. Crosscorrelating the data with this source signal estimate enhances drill bit energy in the data.

A plot of data after such a crosscorrelation is shown in Figure 4.7. Here the traces have been sorted by their offset from the well. There is an event at zero lag on the near offsets that can be tracked across nearly all offsets. It is important to note that crosscorrelation preserves the moveout of the drill bit signal. We would just need to back out the moveout correction applied to align drillbit energy, then we could perform the same search for sources described above after crosscorrelation.

In correlating the stacked trace against a given input trace, that trace is omitted from the stack, to avoid the obviously large correlation that would result. To show that this correlation is not an artifact of the processing, consider Figure 4.8, where the same analysis was performed, but an incorrect moveout correction was used. With the wrong moveout applied, the drillbit energy does not stack properly and the correlation reveals nothing.

Figure 4.9 shows the result of summing after correlating the eight records taken while the pumps within the array boundaries were turned off. This summation improves the picture of the direct arrivals somewhat, but unfortunately it does not bring out any reverberations of drillbit energy as we might have hoped. Probably we need longer correlation times that are available with conventional reflection seismic recording hardware.
FIG. 4.7. Data after crosscorrelation with an estimate of the drill bit source signal. The event at zero lag is the direct arrival from the drill bit. [drillbit-xcor.21] [R]
FIG. 4.8. An incorrect moveout correction has been made prior to stacking and cross-correlation, verifying that the energy seen at zero lag in the other crosscorrelations is not an artifact of the processing sequence, and is therefore most likely direct arrival drill bit energy.
FIG. 4.9. Sum of crosscorrelations for eight records where the drillbit was at a constant depth and all pumps within the survey area were turned off. Stacking records in this way gives a longer effective crosscorrelation length, but still no arrivals are visible other than the direct arrival. [drillbit-xcor.sum] [CR]
Appendix A

3-D Interpolation Applied to Passive Data

A.1 Interpolation

Jon Claerbout (1992) developed a 2-D interpolation scheme based on a mono-planewave assumption. To fill in a gap between two traces, a spatial prediction method is used to determine the dip, and the missing traces are constructed by combining the two known traces from either end of the gap, with time shifts and weights appropriate to the estimated dip and the various distances.

In three dimensions, the location of a trace to be filled in by interpolation will not always lie on a line drawn between some pair of input traces. Thus while it was sufficient in 2-D interpolation to find the apparent dip between two traces, in 3-D interpolation we need to find the true dip. A pair of traces is insufficient for finding the true 3-D dip. At least three traces must be used, since three points are required to define a plane.

Given this realization, a simple 3-D analog to Claerbout’s 2-D mono-planewave scheme can be proposed. To construct a trace at a location, examine the three nearest traces. Use the spatial predictor described in Claerbout’s paper to find the dip between one pair of traces, then another pair (with a different source-receiver azimuth). From these two apparent dip measurements, the true dip can be recovered and used for interpolation, following Claerbout.

I develop a slightly more general alternative to this simple approach, one that uses an arbitrary number of neighboring traces and is less susceptible to problems that might be
caused by a single bad trace.

A.1.1 Implementation

The steps in Claerbout’s 2-D method can be summarized as follows:

- Identify a gap of missing or dead traces.
- Estimate coherence between the two traces at the ends of the gap, for all possible dips, using small overlapping time windows.
- For each time window, pick the best dip.
- Fill in each dead trace with a sum of the two end traces, each time shifted according to the dip, and weighted by distance.

In 3-D there are some additional complications. The location at which we wish to construct a trace by interpolation may not lie on a line connecting two live traces, particularly if the spatial sampling is irregular. When there is such a line, we could follow the 2-D logic and apply the prediction scheme to the two traces. This would give an estimate of the dip along that profile — the apparent dip, not the true 3-D dip. The true dip and apparent dip are related by (Slotnick, 1959):

\[ \sin \phi' = \cos(\theta - \theta') \sin \phi, \]  

(A.1)

where \( \phi \) is the true dip, \( \phi' \) is the apparent dip, and \( \theta \) and \( \theta' \) are the azimuth angles of the down-dip direction and the direction along which the apparent dip is measured, respectively.

We could shift the two traces using this apparent dip and sum. A better idea, however, and a necessity for the case of irregular spatial sampling, is to estimate instead the true dip, using a number of nearby traces. We need at least three; from two we can only find the apparent dip in one direction.

We take the three or more nearest traces, two at a time, and apply the spatial prediction operator, as in the 2-D case. This gives us the coherence as a function of the apparent dip along the direction joining the two traces. Having done these computations for each trace pair, we then loop over all possible 3-D dips (a two-dimensional space, \( p_x \) vs. \( p_y \)). For each dip, given the orientation of each trace pair, we compute what apparent dip we
would see given a particular true dip. The coherence for this apparent dip is extracted from the table constructed earlier, and the coherencies are summed for all trace pairs. The result is a 2-D image of what we will call "generalized coherence" as a function of $p_x$ and $p_y$. From this, we pick the best dip.

Then, for each of the neighboring traces, given the true dip, we compute the apparent dip along the line joining the trace and the location of the trace to be interpolated. This gives for each trace a time shift. The three or more neighboring traces are time shifted, weighted based on their relative distances, and summed to produce the interpolated trace.

The steps in the 3-D scheme can be summarized as follows:

- For a given output point, find the N nearest live traces.
- Estimate coherence between each pair of traces in the group, for all possible apparent dips, using small overlapping time windows. Tabulate the results.
- Loop over all 3-D dips (true dips), parameterized by $p_x$ and $p_y$. For each $(p_x, p_y)$ pair:
  - Given the true dip, compute the apparent dip between each pair of traces.
  - Extract the coherence for that apparent dip from the tables computed earlier.
  - Add coherencies for all trace pairs to get a generalized coherence for this dip.
- Scan over all $(p_x, p_y)$ a second time to see which dip has the best coherence. Pick the best dip.
- Construct the output trace as a sum of the N neighbors, each time shifted according to the true dip, and weighted by distance.

These steps are illustrated graphically in Figures A.1 and A.2.

### A.1.2 Synthetic example

In this section, we test the algorithm on a simple synthetic dataset to illustrate its operation. Figures A.3 and A.4 show a number of slices taken through a synthetic 3-D cube, before and after interpolation. The cube contains 13x13 traces and 128 time samples. The geometry mimics the SEP passive experiment (Nichols et al., 1989) with a 38.1 meter spacing between traces in both directions.
FIG. A.1. Given the N nearest neighbors (here three), we first compute coherence as a function of the apparent dip between each trace pair. [interp-idraw-step1] [NR]

FIG. A.2. After deciding on the best true dip, we compute the apparent dips between each neighbor and the output, then time shift, weight, and sum. [interp-idraw-step2] [NR]
The dataset contains four dipping events. One is near-vertically incident (apparent velocity 8 kilometers/second). A second has a low apparent velocity of 2 km/sec, and the third and fourth are close in both apparent velocity and azimuth angle - one has an apparent velocity of 3 km/sec and a down dip azimuth of 30 degrees (relative to the x axis) while the other has an apparent velocity of 4 km/sec at an azimuth of 60 degrees. Roughly half of the 169 traces were randomly removed prior to the interpolation, which was used to restore the missing data.

In this example, and in the real data example to follow, the five nearest traces were used to construct each interpolated trace. Since the contributing traces are weighted in inverse proportion to their distance from the output location, if the output location is the same as the location of one of the contributors, that trace gets a weight of one while all others get zero weight. In other words, the interpolated result honors the input data.

The interpolation has done a good job for the most part. Problems can be seen in a few places, such as in the crossline slice displayed at a larger scale in Figure A.5. There the top two events, the 8 km/sec and 2 km/sec events, cross. At the crossing point there is no problem, as whichever dip is picked works fairly well for both events. But there is a problem when the two events are near one another. Then they are close enough for both dips to occur in the same small computation window used by the program. But since only one dip is picked, the treatment of the second event can be poor.

The dips picked by the algorithm are tabulated and output by the program as an optional diagnostic. A contour plot of the dips picked for the synthetic is shown in Figure A.6. The 8 km/sec and 2 km/sec events can be seen at the center and bottom of the plot, respectively. The two events with quite similar dips give two maxima close together, in the upper right quadrant.

**A.1.3 Quarry blast example**

We have applied the method to a real 3-D dataset, the record from the SEP passive experiment containing the largest of the three quarry blasts.

The blasts were recorded late in the experiment, with many of the recording instruments beginning to fail because of dead batteries. At the time of the record used here, 87 of the 169 instruments had failed. The interpolation scheme gives us the opportunity to estimate the missing data. Actually the interpolations shown here, in Figures A.7 through A.10, are done onto a 25x25 grid, with half the original spacing, to make the data cube a
FIG. A.3. Four crossline slices from synthetic dataset, before (left) and after interpolation.
FIG. A.4. Four inline slices from synthetic dataset, before (left) and after interpolation.
FIG. A.5. A single crossline slice shown at a larger scale. The only significant error occurs just to the left of the point where the top two events cross. There the two dips occur in the same computation window, but only one is picked, and the treatment of the second one (the flatter event) is poor.

FIG. A.6. Contour plot of dips picked by algorithm as a function of $p_x$ and $p_y$. The 2 km/sec and 8 km/sec events are shown at the bottom and near the center. Although not well separated in apparent velocity or azimuth, the other two events are clearly distinguishable.
little larger and easier to view.

The dips picked by the algorithm are shown in Figure A.11. Most of the picks are off to one side of the center of the plot — this indicates a dominant arrival direction, not surprisingly in the direction of the quarry. The range of dips for these arrivals is consistent with other analyses that have been done on the quarry blast data.

Note the significant number of picks around the edges, at very low apparent velocities. The prediction filtering used to estimate coherence as a function of dip is similar to a crosscorrelation of the two traces. Given a fixed data length, correlation uses more data for smaller lags than for larger lags (lower apparent velocities) where it must avoid going off the end of the trace. Coherence values for the lower apparent velocities, then, are based on less data; this introduces a bias that makes the extreme points more likely to be picked, we believe. This problem can easily be circumvented by padding the input data to allow for a uniform correlation length.

A.1.4 An alternate approach - plane fitting

Spatial prediction gives us the coherence as a function of apparent dip, measured along a number of azimuths. As an alternative to the dip scanning method used above, we could pick the best apparent dip for each trace pair, and then from these many apparent dips, select one (or more) true dips for interpolation. For example, we could fit a single plane to all the apparent dip measurements using least-squares. The overdetermined problem looks like this:

\[
\begin{pmatrix}
\cos \theta'_1 & \sin \theta'_1 \\
\cos \theta'_2 & \sin \theta'_2 \\
\vdots & \vdots \\
\cos \theta'_N & \sin \theta'_N
\end{pmatrix}
\begin{pmatrix}
\sin \phi \cos \theta \\
\sin \phi \sin \theta
\end{pmatrix} =
\begin{pmatrix}
\sin \phi'_1 \\
\sin \phi'_2 \\
\vdots \\
\sin \phi'_N
\end{pmatrix}.
\]

(A.2)

Here the unknowns are \( \phi \) and \( \theta \), the dip and azimuth angles of the true dip, the plane we are fitting to the data. The known \( \phi'_i \) and \( \theta'_i \) are the measured apparent dips and the azimuth angles along which they are measured.

In the presence of a single dip, this method works well, and is significantly less costly than a global search of the dip space. In the presence of multiple dips, however, the least-squares technique will choose an intermediate dip to minimize the error, a dip which may not interpolate any of the dips particularly well. The dip-scanning method presented
FIG. A.7. Crosslines from quarry blast, before (top) and after interpolation.
FIG. A.8. Single crossline from quarry blast, before (left) and after interpolation.
FIG. A.9. Inline profiles from quarry blast, before (top) and after interpolation.
FIG. A.10. Single inline profile from quarry blast, before (left) and after interpolation.
FIG. A.11. Contour plot of dips picked by algorithm as a function of $p_x$ and $p_y$. Most picks lie below the origin, indicating energy incident on the array from the south, the direction of the quarry. \[ \text{interp-blast-picks} \] [NR]

above will at least interpolate one dip, that which gives the best coherence, well.

The best solution to the case of conflicting dips may be a plane-fitting approach that fits multiple planes to the set of apparent dip measurements. Fitting multiple planes is significantly more difficult than fitting a single plane. Possibly an L1-norm scheme could help.

A.1.5 Application: removal of cross-line smear

The theory behind the interpolation scheme presented here could also be used to address the problem of cross-line smear in 3-D marine surveys, discussed by Yilmaz (1987). Cable feathering causes midpoint locations to be distributed over the cells of a 3-D marine survey. If there is a significant amount of cross-dip in the subsurface, this midpoint scatter brings with it time shifts that cause a departure from hyperbolic moveout, and a smearing of the stacked amplitudes. Removal of the midpoint smear requires an estimate of the dip. The algorithm described here could be used to automatically determine the dip. Time shifts could be computed and applied to map all midpoints to the cell center.
Appendix B

Xtpanel - an interactive panel builder

This appendix describes a program called xtpanel that I wrote, along with Dave Nichols, to facilitate user interaction with existing non-interactive software. It is used extensively at SEP to allow readers to interact with documents such as this thesis, recreating the figures with the ability to try out new parameters or look at alternative pieces of data. I have included a description of xtpanel here because I feel it is a useful component in moving toward the goal of reproducible research that may be of interest to readers of this thesis.

The xtpanel program manages a set of interactive objects on the screen. These can either be specified on the command line or in a script file. The appendix contains the manual page for xtpanel, which describes the command line and script file syntax in more detail.

The following types of objects are supported:

- messages
- text fields
- dialog boxes
- sliders
- buttons
menus
lists
toggle buttons
variables (objects with no screen representation)

**B.0.6 Object values**

In xtpanel, each interactive object has a string value associated with it. Interacting with the screen representation of an object will change its string value (e.g., moving a slider sets the value to the value of the slider, typing in a dialog sets the value to what you type, choosing an item in a menu sets the value to the string associated with the menu selection.) The primary task of each object is to maintain its string value.

**B.0.7 Object names**

Every object may have a name. This name is used to refer to the value of the object in actions.

**B.0.8 Object actions**

Every object can also have a set of actions associated with it. These actions are performed whenever the object's string value is updated. If more than one action is specified for an object, they are performed in the order specified. The action can be one of five types:

QUIT  Exit the xtpanel program.

PRINT  Print a string to the standard output.

STRING  Set the object's string value to a new value.

ASSIGN  Set another object's string value to a new value.

SYSTEM  Execute a command.

Enormous flexibility is gained by letting any object execute a command as the result of interaction with it. Xtpanel can be used as a way of interactively connecting existing Unix programs that do not have interactivity built into them. An impressive example
of this is an 80 line xtpanel script that acts as an interactive front end to the Unix calculator program, bc. This 80 line program (see the Calculator section below) has similar functionality to the 2070 line C program xcalc distributed with X windows.

An action may contain two types of special characters.

1. The string value of any object can be used in an action string by referring to its name, preceded by a dollar sign. The following action string sets the value of its object to the current string value of another object whose name is \texttt{otherobj}:

   \begin{verbatim}
   action="STRING $otherobj"
   \end{verbatim}

2. The string resulting from executing a command can be used in an action string by enclosing it in backquotes. For example, the following action string sets the value of an object to the current directory using the UNIX command \texttt{pwd}:

   \begin{verbatim}
   action="PRINT The current directory is 'pwd'"
   \end{verbatim}

The aim of this design is to allow the user to tie together existing Unix programs while leaving as much flexibility as possible.

B.1 Examples

The following pages contain example xtpanel script files and screen dumps of the corresponding interactive program. If you are reading this document on a CD all the figures are interactive, if you click on the button associated with a figure the program will be executed.

B.1.1 Simple buttons

The first script puts two buttons on the screen. Pressing one button quits xtpanel; hitting the second button prints a message. Figure B.1 shows the panel produced.

\begin{verbatim}
button={ label="QUIT" action="QUIT" }
button={ label="hit me" action="PRINT AARGH!\n"
\end{verbatim}
B.1.2 Simple menu

The second script implements a menu. The resulting panel is shown in figure B.2.

```plaintext
button={ label=quit action=QUIT }
menubutton={ label=numbers action="PRINT choice is $val \n"
    item={ label=one value=1 }
    item={ label=two value=2 }
    item={ label=three value=3 }
}
```

B.1.3 Simple interactive parameter selection

This example shows how xtpanel can be used in conjunction with the cake rules for an interactive document (Claerbout and Nichols, 1990) to make a dull figure come alive. The default cake rules for an interactive document check for the existence of a file called `name.panel`, where `name` is the name of a figure. If this file exists the rule will run xtpanel when the button in the caption is pressed. Figure B.3 shows a single synthetic event. When
the user presses the button in the caption they are asked to choose an NMO velocity to flatten the event. Here is the xtpanel file that specifies the interactive program.

```
button={ label=QUIT action=QUIT }

slider={
  label="Select a velocity for NMO"
  min=1 max=5 value=2 format="%.2f" width=300
  action="NMO <Dat/nmoin.H val=$\{val\} | \n    Wiggle title="$velocity=$\{val\}" pcclip=100 | Tube numcol=16 &"
}

message={ value=" hit ""ok"" to run the program " }
```

![Input hyperbola](image)

**FIG. B.3.** A synthetic dataset, click on the button to choose the correct NMO velocity.

When the user clicks on the OK button the data has NMO applied at the chosen velocity and the result is displayed on the screen. Fig B.4 shows the interactive panel in use.

### B.1.4 Simple seismic processing pipeline

The next example controls a simple processing job using SEP software. A dialog box is used to input a file name. This file is read into a bandpass filter program, then converted to one byte per sample (for display) by the program Byte, then plotted. Sliders are used
FIG. B.4. The simple velocity selection panel in use.
to specify the cutoff frequencies for the filter, and for the percentile of the filtered data that is clipped to the maximum intensity in the float to byte conversion.

```plaintext
button={ name=QUIT action=QUIT }
button={ label=GO
    action="Bandpass flo=$flo fhi=$fhi phase=$phase < $file \\
    > /tmp/band.h out=/tmp/_band.h8; \\
    Byte pclip=$pclip < /tmp/band.h | Ta2vplot | Xtpen; \\
    /bin/rm /tmp/band.h &"
}
dialog={ name=file label="input file" value="Dat/wz.25.H" }
hbox={
    vbox={
        message={ value=Bandpass }
        slider={ name=flo label=flo min=0 max=125 value=0 format="%2f" }
        slider={ name=fhi label=fhi min=0 max=125 value=125 format="%2f" }
        choice={ name=phase label=phase value=0
            item={ label="zero" value="0" }
            item={ label="minimum" value="1" }
        }
    }
    vbox={
        message={ value=Byte }
        slider={ name=pclip label=pclip min=0 max=100 value=99 format="%2f" }
    }
}
```

FIG. B.5. A panel to control the execution of two seplib programs.  

B.1.5 SEP data cube viewer

Here we use xtpanel not to process data, but to examine, sample by sample, the values contained in a dataset. From an input three-dimensional dataset, a single plane is selected
(by a slider) and then the floating point values from that plane are displayed in a two-dimensional scrollable text field.

The first script gets the file name from the user:

```plaintext
button={ name=QUIT action=QUIT }
dialog={ name=file value="Dat/cube.H" }
button={ label=GO 
    action="xtpanel -cpp -DFILE=${file} -file viewer2.panel &" }
```

and then calls a second script, which displays the data:

```plaintext
button={ name=QUIT action=QUIT }
dialog={ name=file value="Dat/cube.H" }
button={ label=GO 
    action="xtpanel -cpp -DFILE=${file} -file viewer2.panel &" }
```

![FIG. B.6. A viewer to examine SEP data cubes.](xtpanel-viewer) [NR]

Note that when the first script invokes the second, it uses the `-cpp` flag to pass the second script through the C preprocessor. This is one way to pass variables from one script to the next.

**B.1.6 Directory lister**

Now a more complicated script. This script can be used to navigate a directory hierarchy. It puts all the files in the current directory into a list. When you choose a file it does one
of two things. If the file is a directory, it goes to that directory and reruns the program to produce a new listing panel. If the file is a regular file it performs the Unix command specified in the dialog. Note that the rules governing newlines within an action allow us to embed a complete multiline shell script within an action. The resulting panel is shown in figure B.7.

    button={ label=DONE action=QUIT }
    hbox={
        message={ value="Directory:" }
        message={ value='pwd' }
    }
    dialog={ name=command label="Command for files" value="xterm -e vi" }
    list={
        name=name label="NAMES"
        action=" if test -d $val
        then
            cd $val
            xtpanel -file examples/script/lister &
        else
            $command $val &
        fi"
        itemlist={ list='echo -n . * ' separator=" " }
    }

    xtpanel-list

FIG. B.7. A panel to traverse a Unix filesystem.
B.1.7 Advanced seismic processing pipeline

Next is a multi script example. It is designed to allow the user to choose the parameters for a pipeline of three seplib processes. The main script is used to edit dialog fields containing parameters that specify the input file and the parameters for each of the three programs (Byte,Ta2vplot and Tube). Once the user is satisfied with the parameters he can press the button at the bottom of the panel to run the command. Figure B.8 shows the main panel. Here is the corresponding script file. Notice that the layout for each program is the same, it is defined in a macro and the macro is used three times, once for each program.

```plaintext
button={ label=QUIT action=QUIT }
dialog={ name=input label="Input filename" value="Dat/in.H" }

! macro for a dialog for a program called "NAME", a popup panel and self doc
#define PROGRAM(NAME) 
 hbox={ name=noborder 
 vbox={ name=noborder 
 button={ label="NAME panel" 
 action="ASSIGN NAME 'xtpanel -file pars/NAME'"} 
 button={ label="NAME doc." 
 action="xtpanel -cpp -file pars/doc -DPROG=NAME -font fixed &"} 
 }

dialog={ name=NAME value="" }
}

PROGRAM(Byte)
PROGRAM(Ta2vplot)
PROGRAM(Tube)

! perform the action or popup an error dialog if the input doesn’t exist
button={ label="PRESS FOR: Byte | Ta2vplot | Tube" action="if test -f $input ; then
        <$input Byte $Byte | Ta2vplot $Ta2vplot | Tube $Tube &
      else
        xtpanel -message 'File $input does not exist' -quit
      fi"
}

Each parameter file dialog has a two buttons next to it. The lower button invokes a script that causes the program to self document. This documentation is shown in a text object. The script to do this is very short, note that the preprocessor is used to replace "PROG" by the appropriate program name. (The connection of stdin to /dev/tty is to force the program to self doc.)

```plaintext
button={ label=DONE action=QUIT }
text={ value='( <dev/tty PROG 2>&1)' height=400 width=500 }
```

Pressing the upper button brings up a subsidiary panel that contains interactive objects that can be used to specify the parameters. When the subsidiary panel is closed the dialog
FIG. B.8. A panel to control the execution of three seplib programs.

FIG. B.9. The documentation produced by pressing the Byte doc. button.
is set to contain the parameters chosen on the subsidiary panel. Figure B.10 shows the panel for Byte. The script file for this panel follows.

```plaintext
slider={
    name=pclip min=50 max=100 value=98 format="%.1f"
}
slider={
    name=gpow min=0. max=4. value=1. format="%.2f"
}
menubutton={
    name=gainpanel label="gainpanel=>" value=1
    itemlist={ list="1 every all" }
}
choice={
    name=transp value=n
    item={ label=yes value=y }
    item={ label=no value=n }
}
button={
    label=Done
    action="PRINT pclip=$pclip gpow=$gpow gainpanel=$gainpanel transp=$transp "
    action=QUIT
}
```

![Diagram of the panel for Byte parameters](image)

**FIG. B.10.** The panel for interactively setting Byte parameters is obtained by pressing the “Byte panel” button. [xtpanel-bytepar][NR]

Finally Figure B.11 shows the result of pressing the bottom button to run the command. Notice that the self doc is still available, as the xtpanel process to perform the self doc is run in the background. The final command is also run in the background so that you can have multiple results visible on the screen at the same time so that they can be compared.
FIG. B.11. The result of pressing the action button. [xtpanel-result] [NR]
B.1.8 ed1D

Jon Claerbout (Claerbout, 1991) developed the interactive program ed1D to allow users to experiment with various aspects of one-dimensional seismology. Among other things, ed1D lets the user edit a function in the space or spatial frequency domain, and see the result in the other domain. In Figure B.12 we implement this particular aspect of ed1D in an xtpanel. The script is 240 lines long, so we have not included it here. It is available on the CD-ROM version of this report.

![xtpanel ed1D](image)

FIG. B.12. An xtpanel analog to Jon Claerbout's ed1D program, for experimenting with Fourier transforms. While this version is crude compared to the actual ed1D program, it took only about one hour to write. [xtpanel-ed1D][NR]

B.1.9 Calculator

Figure B.13 is an example that emphasizes xtpanel's ability to take advantage of the many powerful but non-interactive features built into UNIX. The bc calculator built into UNIX has most of the features of a standard scientific calculator. But it lacks a nice interactive interface. The X windows source distributed by MIT contains an interactive calculator xcalc. Written in C, xcalc is over 2000 lines long, and doesn't take advantage of the calculator already available in UNIX. The xtpanel script uses the bc calculator and provides an interface similar to that of xcalc. The script is 80 lines long, and available on
the CD-ROM version of the report.

![Diagram of xtpanel calculator](image)

**FIG. B.13.** A calculator built using xtpanel. The 80 line script gives a calculator with features comparable to the 2000 line C program xcalc.  

It is likely that this 80 line script required much less time to write than the xcalc application. And because the script syntax is very simple, adding features to this calculator would be a much easier task than adding features to xcalc.

### B.1.10 The xtpanel generator

While the xtpanel script language is intended to be easy to read and use, it would be nice if it weren’t necessary to learn it in order to use xtpanel. For this reason, we created the xtpanel generator. This is a series of xtpanels that let the user build a panel interactively, object by object. Figure B.14 shows the top-level generator panel. Buttons are provided to add the various objects to the panel. At any point, the resulting script file can be examined, or previewed by running xtpanel on it.

### B.1.11 Interactive help facility

Scrolling through a lengthy manual page to find a particular topic can be tedious. We created an interactive help facility for xtpanel, which is a series of panels that present, in a menu-driven form, the various parts of the manual page, along with some other pertinent
FIG. B.14. The xtpanel generator is a series of xtpanel scripts that let users build new xtpanels interactively. [xtpanel-generator] [NR]

information. The top level panel of the help facility is shown in Figure B.15. General information is shown in the scrollable text field on the main panel; the menu lets the user choose from a list of additional topics.

B.2 Comparing xtpanel and other products

Many software packages address the general problem of building a graphical interface to simplify life for the end user. Xtpanel is worthwhile only if it offers something that these other packages (many of which are public domain) do not.

Some packages are complete graphical user interface builders. These programs let the user create an object, move it around on the screen or change its attributes – all interactively. While these programs are very powerful, they also tend to be somewhat difficult to use. Typically the work must be done interactively; the various configuration files are verbose and not easily edited by the non-expert. Such packages also are usually more self-contained; it is not as easy to integrate existing non-interactive UNIX software as it is in xtpanel.

Another set of tools takes a simpler approach, with all the configuration information specified on the command line. Two examples, distributed with the X windows source
code from MIT, are \texttt{xmessage} and \texttt{xmenu}. These construct a panel with a set of messages, or a menu with several items. While such tools are very easy to use, they are typically quite limited in what they can do.

Xtpanel was designed to fill in the gap between these two classes of tools. It is meant to be very easy to use, but capable of generating panels that are complex and powerful. Also xtpanel takes more advantage than many other products of the power of UNIX, making it easy to run system commands and to incorporate their results into the action of the panel.

### B.3 Interactivity with xtpanel

In Figure B.5, xtpanel was used to create an interactive frontend to an existing seismic processing program, the bandpass filter program \texttt{Bandpass}. While building frontends in this way can be useful, often the overhead of having to re-run the entire job makes this level of interactivity unsatisfying. Here we describe a way to use the interactivity of xtpanel from within programs.

Using xtpanel from within a seismic processing program offers some advantages. In
the filtering example above, because the entire process is re-run each time a parameter is changed, a lot of extra work is done. Each time the RUN button is pressed, the input dataset (containing a single impulse) is recreated. Then the filter program must read in the data, as well as the parameters necessary to run the job. In such a trivial example, this extra work does not take much time. But in larger tasks, the extra overhead required to completely re-run a job may make the interactive performance poor.

To solve this problem, we devised a way to use xtpanel from within a processing program. After reading in its data and getting set up to run, a program brings up an xtpanel containing objects that specify some of the program parameters. Then the program waits for input from the xtpanel. When input is received, the program executes, and then waits for additional panel input. Additional panel input causes the program to be re-run, but without having to do all the overhead that was done at the start of the program.

The first program to use this interaction is a seismic data cube viewing program called Cubeplot. Cubeplot displays a perspective view of a seismic data cube. Displayed on the three faces of the cube are slices taken from within the data volume. In its standard batch mode of execution, the user specifies on the command line the three slices to be shown. If, on the command line, the user specifies popup=y, then a panel containing three sliders appears. As the user changes a slider, a different data slice is displayed on one of the three cube faces.

The xtpanel script is built into the C language source code for Cubeplot. Here is the script:

```c
button={ label=Quit action="PRINTQUIT" action=QUIT }
 vbox={
  hbox={ name=noborder width=400
    message={ value=axis-1 }
    scrollbar={ label="axis 1" name=pan1 min=0 max=N1 format="%0f"
      width=300 value=FRAME1
      action="ASSIGN val1 $val"
      action="PRINT frame1=$(pan1) frame2=$(pan2) frame3=$(pan3)\n"
    }
    message={ name=val1 value=FRAME1 }
  }
  hbox={ name=noborder width=400
    message={ value=axis-2 }
    scrollbar={ label="axis 2" name=pan2 min=0 max=N2 format="%0f"
      width=300 value=FRAME2
      action="ASSIGN val2 $val"
      action="PRINT frame1=$(pan1) frame2=$(pan2) frame3=$(pan3)\n"
    }
    message={ name=val2 value=FRAME2 }
  }
  hbox={ name=noborder width=400
    message={ value=axis-3 }
    scrollbar={ label="axis 3" name=pan3 min=0 max=N3 format="%0f"
      width=300 value=FRAME3
    }
}```
However, the user can substitute a different script, containing whatever program parameters are of interest. Here are the instructions that appear in the on-line program documentation:

```
popup Specifying popup=y brings up an xtpanel (if you have xtpanel installed) with three sliders. Moving these sliders changes the frames plotted on the three cube faces. If you pipe the output of Cubeplot to "Xtpen cachepipe=n" you will see the display update as the sliders are moved. You can specify your own xtpanel script file by doing popup_file=filename.
```

The program then calls two subroutines: `popup_start` brings up the xtpanel. This is called after the program’s preliminary work (reading in the input data, getting parameter values, etc.) has been done. A second routine, `popup_check` is called after each pass through processing the data. This routine waits for output from the xtpanel. It then adds the new output to the table of parsed command line arguments. I.e. if the user moves a slider and the xtpanel prints “framel=100” then after `popup_check` it is as though “framel=100” had been specified on the command line. When this routine returns the program reprocesses the data (in the case of Cubeplot case plotting a new figure) using the new parameters.

Users interested in the details of these routines are referred to the CD-ROM version of this report, where the source code for Cubeplot and the popup routines is contained. The most important point to make about the source is that the modifications to the Cubeplot program were quite small — just two subroutine calls, and a few lines to interpret the new parameter values that come back from the panel. Interaction can easily be added to any program in this way.

Figure 8.16 displays a typical interactive session using the built-in xtpanel interaction of Cubeplot. If you are reading the report on CD-ROM, pressing the button at the end of the figure caption will bring up this example. Moving the sliders will change the display.

It is worth noting that although Cubeplot is a C language program, the xtpanel popup facility can just as easily be used from within Fortran and Ratfor programs.
FIG. B.16. Interactive Cubeplot session using xtpanel. The xtpanel is brought up from within the program. Each time a slider is moved, a subroutine is run to re-draw the image. CD-ROM readers can click on the button at the end of this caption to run Cubeplot.
Xtpanel future possibilities

We wrote xtpanel to fill a need, to add interactivity to the large library of non-interactive seismic processing programs available at SEP. In addition to making day-to-day data processing chores easier, this interactivity is also useful in building tutorials and interactive documents, such as this report. While other public-domain interface-building software exists, we believe that no other simple package is flexible enough to meet these needs.

Because it does not depend on other SEP software or geophysical software in general, xtpanel has been distributed through several public-domain source channels. Feedback from interested users has helped xtpanel to evolve to a reliable state in a short period of time.

There are several minor ways in which xtpanel could be improved in the future. Support of Motif in addition to the MIT Athena widget set would make xtpanel compatible with the software environments of more potential users. Because of the modular way in which xtpanel has been written, and because the Athena widgets and Motif have many features in common, adding Motif support would not be very difficult.

Another promising possibility is to be able to put images (pixmaps) in the background of xtpanel objects. If this were possible, one could, for example, very quickly build seismic tools that incorporate data picking, by having an image of the data in the background of certain interactive objects.
Appendix C

Bibliography

REFERENCES


