Interactive slant stacking

Rick Ottolini

ABSTRACT

Limited aperture slant stacks extract time dip information from the data. The slant stack aperture width and slope are specified by drawing a dip line anywhere along the data. The lates possiblet generation of high speed graphics workstations makes interactive two-dimensional signal processing.

INTRODUCTION

Much of seismic data processing is a search for the best processing parameters. In the past, constrained by computer resources, we would apply some heuristic, often automated, find the best processing parameter with one or a few guesses. Supercomputing changed this strategy by letting one try many parameters—either recomputing the process for a suite of parameter values or iteratively converging upon some criterion. Now interactive processing adds a third strategy to automated guessing and exhaustive search—user-guided search. Interactive processing was not feasible for multi-dimensional seismic signal processing without the immediate feedback of graphical supercomputing provided by the latest workstations.

Slant stacking has been applied to most seismic data processing stages (Ottolini, 1987a). Slant stacking extracts time dip information from data. Time dips are either used in a geometric sense to discriminate signal or for their wave propagation information for seismic imaging.

Slant stacking has two important parameters—stack angle and stack aperture. The stack angle has been generally selected by exhaustive search, several dozen at time. This works well for gathers, though is expensive for sections. Also, one may have to adjust the range and bounds of stack angles to fit interesting features on the data.

The stack aperture for slant stacking profiles or gathers has generally defaulted to the entire cable length. However, limited apertures, sometimes called local slant stacks (McMechan, 1983), controlled directional receptivity (Sword, 1987) and beam stacks (Kostov and Biondi, 1987) have been increasingly used.

The search solution presented in this paper is to select these two slant stacking parameters interactively and then immediately see the slant stacked data. The visual representation of slant stacking parameters is very important. I find a direct representation—a line across the data showing slope and aperture—to be more effective than indirect controls off the data.

INTERACTIVE SLANT STACK PROGRAM

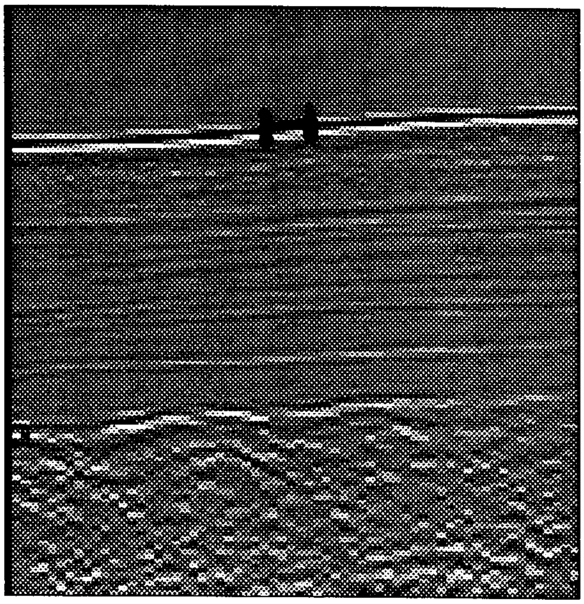
The program xss—x-windows slant stack—starts by reading the SEPLIB dataset specified on the command line. Xss has two modes—(1) local slant stack and (2) linear moveout. Both modes are initiated by drawing a dip segment with the mouse button (Figure 1a). The slant stack mode will return the data slant stacked at the aperture width and time slope of the dip line (Figure 1b). The result is moved-out to undo the slant stack time shift. Subsequent drawings of the dip are not cumulative, but reprocess the original data. This slant stacking algorithm was described by Ottolini (1983). Variations of slant stacking mode are to (a) show the slant stacked data, (b) show the power of the slant stack data, (c) show the data weighted by semblance along the slant stack (Kong, et al, 1985), (d) show the the signcount along the slant stack (Hansen, et al., 1988), and (e) show the data weighted by the signcount along the slant stack.

The linear moveout mode applies linear moveout as to flatten any data feature the dip line is drawn parallel to (Figure 1c). (This requires linear moveout *opposite* to the drawn slope and subsequent dip slopes sum into the previous slope.)

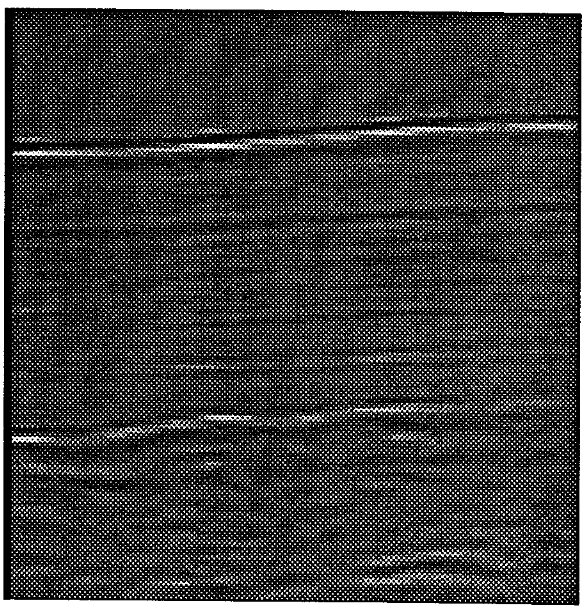
Table 1 summarizes the functionality of the xss program. These functions are accessed by clicking a control menu or typing the first letter of function.

Table 1. xss functions	
mouse draw	set stack slope and stack aperture
stack	slant stack mode (default)
moveout	linear moveout mode
power	show slant stack semblance
count	show slant stack sign count
weight	semblance filter the data
Filter	signcount weight the data
reset	restore original data
window resize	resize the data
save	save processed data
print	save printable image
quit	exit

(a) Input Data Section



(b) Slant–stacked Data



(c) Linear–moved Data

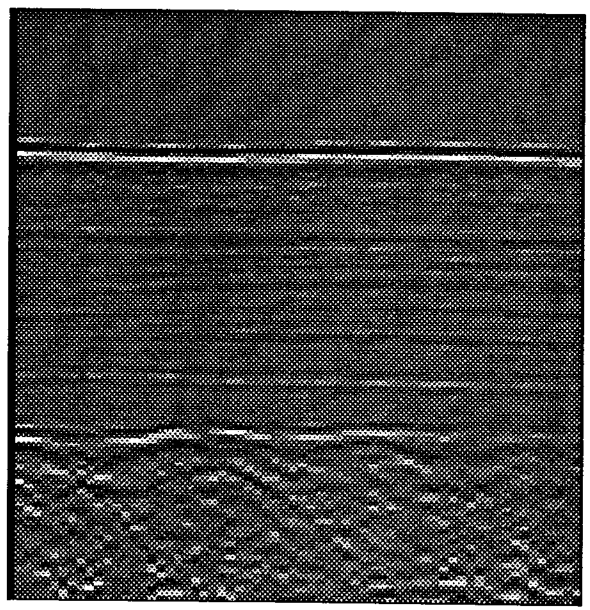


FIG. 1. Snapshots from the interactive slant stacking program: (a) input dataset, the five-trace wide dip line used to slant stack and moveout is highlited as an 'H'. It is five traces wide and parallel to the top reflector; (b) less than a second later on a DEC 3100 this local slant stack section appears; (c) linear moveout mode flattens the seafloor horizon.

IMPLEMENTATION NOTES

Xss was written in C and X-Windows (version 11 release 3). It was developed on a Sun Microsystems 3-110 and tested on a Convex, DEC 3100, SONY NEWS 1730, and Ardent Titan. The code is organized as objects in straight C (Ottolini, 1987c). Table 2 lists the objects used. One subroutine does most of the number-crunching and can be converted into FORTRAN for computers that compile FORTRAN more efficiently than C.

Table 2. **xss objects**	
data	seplib dataset
trace	trace manipulation
slant	slant stack parameters and routines
xw	X-Windows framework
main	program organization

Important implementation issues are (1) graphical supercomputing, (2) the right user "feel", and (3) portability. The DEC 3100, by far, gave the fastest interactive slant stacking response. This is attributed not so much to the floating point speed that is equivalent to a Convex or an Ardent TITAN, but the efficient coupling of the graphics interface, X-Windows, to the CPU. Previous work (Ottolini, 1987b) speculated that coupling specialist number-crunching and graphics computers together would give the best results, but this has not worked in practice (Ottolini, 1988).

This program was partially motivated as purchasing benchmark for the 1989-class of workstations—15+ MIPS, 3+ MFLOPS. The raw speed (MIPS) accelerates graphics and dataflow, while the floating point speed (MFLOPS) helps number-crunching. There are many two-dimensional seismic processes that are a couple of MFLOPS. These were too costly to contemplate on the previous generation of workstations that were an order-of-magnitude slower, but ripe for implementation now.

The right "feel" is best provided by directly interacting with the image. I spent some time tuning the behavior of dip lines. Off-image controls don't cut it, except for secondary functions. The interactive interface to many of our multi-dimensional seismic processes will have to be explored.

Graphics portability is finally a reality with X-Windows, C, and FORTRAN on about every workstation. The program compiled and ran on the above five platforms with almost no alterations, but wide performance variations. Portability will simplify the integration of new workstations and software development at SEP.

CONCLUSIONS

Interactive processing of multidimensional seismic data is now feasible. It is an alternative method of processing parameter search to automated guessing and exhaustive search. It will increase the quanity and quality of educational software exercises. The important implementation issues are high speed graphics supercomputing, the right user interface and portability.

Two dimensional transform workbench

Plans are to create additional interactiver multi-dimensional seismic software. This includes normal-moveout as fast as the cursor movement and various 2-D time migration algorithms. These may be considered as seismic applications of two-dimensional transforms—the linear generalized Radon transform, the hyperbolic generalized Radon transform, and the Fourier transform.

REFERENCES

Hansen, K.M., Roy-Chowdhury, K., and Phinney, R.A., 1988, The sign filter for seismic event detection: Geophysics, **53**, 1024-1033.

Kong, S.M., Phinney, R.A., and Roy-Chowdhury, K., 1985, A non-linear signal detector for enhancement of noisy seismic record sections: Geophysics, **50**, 539-550.

Kostov, C., and Biondi, B., 1987, Improved resolution of slant stacks using beam stacks: SEP-**51**, 343-350.

McMechan, G., 1983, P-x imaging by localized slant stacks of t-x data: Geophy. J. Roy Astr. Soc., **78**, 723-734.

Ottolini, R., 1983, Signal-noise separation in dip space: SEP-**37**, 143-150.

Ottolini, R., 1987a, A classification of slant stack imaging methods: SEP-**51**, 363-366.

Ottolini, R., 1987b, Multi-computer seismic programs: SEP-**51**, 409-414.

Ottolini, R., 1987c, Techniques for organizing interactive graphics programs: SEP-**51**, 421-432.

Ottolini, R., 1988, Interactive residual synthetic aperture radar imaging: SEP-**57**, 457-468.

Sword, C., 1987, The CDR method: SEP-**51**, 317-342.

SEP COMPUTERS 4/89