The Balloon program

Jon F. Claerbout

ABSTRACT

The "Balloon" program allows the operator to bring up interactively SEP's vector plots (vplot) and movie cubes, to move these plots around, to juxtapose them, and to zoom, rotate, and reflect them, and to overlay "comic strip" text balloons. Additionally a simple paint-like utility allows manually drawing lines, polygons, boxes, arrows, and rectangular erasures. Any plane may be selected from a movie cube by means of rotating, slicing, and zooming, and optional axis labels are automatically supplied. Raster can be displayed as variable brightness or wiggle traces. A file is saved that can be interpreted on our Imagen and Apple laser printers.

INTRODUCTION

Plotting things is like raising children, always a nuisance, but when done successfully, a source of much satisfaction.

Pointing and annotating

When we find something of interest in our data, we need to point it out to people. When we are speaking, the "pointing out" is easy, but when we are preparing reports, theses, and books, the pointing gets awkward. It means a trip to a draftsman, or attempting to do a neat job ourselves, or writing a special purpose program to attach a pointer to a plot. Usually we do not take the trouble to do any of these, and we try to point out the event by using words. This fails us in many ways. First, the axes of the data, both space and time, must be adequately numbered. Second, seismic events occupy areas, not points or lines. And sometimes the area is nonrectangular. Words make a clumsy pointer.

There is little doubt that we need an interactive system of pointing at seismic information and annotating it with information. For starters, we need something like the text balloons found in comic strips. And we also need the kind of capability seen in personal computer software such as MacDraw and MacPaint, but we need to have those capabilities on top of our seismic data and line drawings, not just on a blank sheet of paper.

Cutting and pasting

After revising the first five chapters of FGDP and preparing new examples with my new *Zplane* and *ed1D* programs, I was disappointed that the lecture notes I sent back to Stanford did not include figures. After completing my last book I resolved that I would never again prepare a book where figures need to be manually pasted in each time text is revised. The obvious obstacle is that the mathematical composition software we use (a version of LaTeX) doesn't incorporate figures. Someone will solve this problem quicker and better than I do, meanwhile, I decided to be ready for that time. So I began thinking about collecting figures in a computer directory. The first question is what form figures should take.

Juxtaposing and rezooming

My survey of SEP report figures and those in my books showed another need: We often juxtapose two plots for comparison. And we often find that plots we have made are neither the right size nor the best vertical exaggeration for publication. So we need to an interactive screen to bring up several plots and to move them around and stretch them.

Variety of types—vplot a good compromise

Scanning FGDP, IEI, and recent SEP reports, I was impressed by the diversity of figure types, which speaks against any uniform solution to the problem of computerized figure inclusion, but I was not discouraged because I observed that the large majority of figures were of two types, seismic-section style plots and line drawings. This is fortunate first since almost all our local line drawings are done thru the local vector plotting software "vplot" and second, thanks to Joe Dellinger, raster data is now incorporated in the vplot system although most of our seismic raster plotting is not yet done from within vplot.

Over the long term, we might like to transform our plots to some standardized plot language or page description language, but over the shorter term, our local system works better with our programs and our devices. When a good standard comes available we can write a filter to convert our plots to it.

Postscript

A commercial system called "Postscript" might in the future be more suitable than vplot. Here are some points in Postscript's favor:

- It is built into many laser printers including the Apple laser printer this paper is produced on.
- It has the high quality text you are reading.
- It allows windowing through an arbitrary curve.
- It has high quality line joins
- Since it is the output language of TeX, our mathematics typesetter, figure insertion may turn out to be easy.
- The figures at the end of this paper were done with Postscript on an Apple laser printer (and then manually cut and taped in)

On the other hand, we have no interactive Postscript device. Further, although Sun Microsystems is believed to be producing one, our experience with one commercial version of GKS leads us to wonder whether Postscript will handle raster with reasonable speed. Time will tell.

RUNNING THE BALLOON PROGRAM

You don't need any written document to tell you how to run the program. It is self explanatory. There are 76 possible situations where the program tells you what it is doing or what you should do. As a programmer I found the 76 prompts little more troublesome to include than a sprinkling of 76 judiciously placed comment statements as the prompts play much the same role. Additionally, there are five built-in tutorial scripts.

The Balloon program is not a general purpose figure editor. It works sequentially. You bring in a movie cube or a vector plot, and you tumble it or move it where you want, and then you press a "hardcopy" button which puts a copy in an output file and leaves another indelible copy on the screen. Then you can bring in another vector plot or another cube (or the same one again) and continue. Alternatively, you can choose the "paint" activity or the "balloon" activity.

The balloon activity allows you to outline a box and type text into it. As you type text into the box, the letters get smaller and smaller so as to continue fitting in the box. (Retrospectively, perhaps you would prefer to select a letter size, and then have the box expand in some way as you type.) When you finish with the typing, you can resize and move the box, and draw out a triangular pointer from any of the boxes sides. Then you press the hardcopy button and can select another activity.

The paint option allows you to a number of shapes based on two points, a line (which compounds into a polygon) an arrow, a rectangle, or erasure inside a box, or windowing (erasure outside a box). As before, you make objects sequentially—after you finish an arrow, you can't erase it except by covering it with an opaque rectangle.

After generating a couple of dozen plots to select for this report, I can attest that the biggest aggravation associated with the Balloon program is that once you press the hardcopy button, the figure part you were dealing with becomes indelible. Perhaps with some kind of object-oriented language it would not be too great a burden to extend the Balloon program to be able to keep "alive" all the parts of the figure and allow them all to be randomly accessed and edited.

PROJECT STATUS

The project has now reached a plateau where it has not been touched in several months, never-the-less, if all goes well, more work will be done on it in the future.

Missing objects

There is no text, except for that in the balloons, and it is a rather poor grade of graphics text.

Movie cubes are accepted only in byte format, so there is no interactive scaling data of the data, which limits the program's usefulness in browsing through data.

There is no way to include SEG-Y format data except by other SEG-Y to SEP cube format software that I have not tested.

How it works

The basic sharable strategies used by the Balloon program are described in a separate accompanying document on "Svplot." It describes plotting strategies shared by most of my other interactive programs.

Activity modules

There are five activity modules or in the language of the program itself, five *fig_parts*, balloons, vplot, cubes, painting, and a generic activity. Each package resides in a file: *ball.c vplot.c raster.c paint.c,* and *custom.c.* These five are similar to one another and they are also similar to a master box control file *fig.c.* The two-page file *custom.c.* is a good place for a learner to begin.

Level of effort

I am somewhat chagrined by the amount of time I have spent on plot software, perhaps three months on *Balloon.* But now the work is done and I hope the benefits are obvious and will remain with us for a long time. Also, I learned a lot about writing interactive code, and how to try to make parts of it portable and reusable. I used to write giant programs whereas now I have learned to write in modules each formed around a data structure.

Browsing the code

The code is split over several files. The size of the file in lines is the number in the left column. The first number given is the number of lines in the file.

UTILITY		
130	gapaxis.c	cube axis rotation and subwindowing tool.
APPLICATION		
432	fig.c	main control panel.
465	box.c	movable resizable rotatable box.
478	ball.c	text balloons
173	vplot.c	vplot in box
359	raster.c	SEP cubes.
342	paint.c	you draw arrows, vectors, erase blocks, etc.
109	custom.c	generic application
APPLICATION SUPPORT		
179	map.c	screen layout (mostly Sunview)
381	pldbs.c	read vplot (essentially pldb)
167	cube.c	read sepcubes

MINOR PROBLEMS

Studying the code you will see that *pldbs.c* reads vplot. I split the vplot utility *pldb* into a main and a subroutine so I could use the subroutine in both pldb and in balloon. I am catching only vector commands, so you see there is a need to pass plots thru generic pen first to convert everything to moves and draws. Joe wrote 'vppen dumb=y' for this purpose. Maybe I should should pipe input thru it. Raster input is still a hooker. For the present limited functionality *pldbs.c* could be changed from a 400 line file to a 50 line file. Joe proposes instead to replace *pldb.c* by *dovplot,* thousands of lines, to get full functionality on input vplot files.

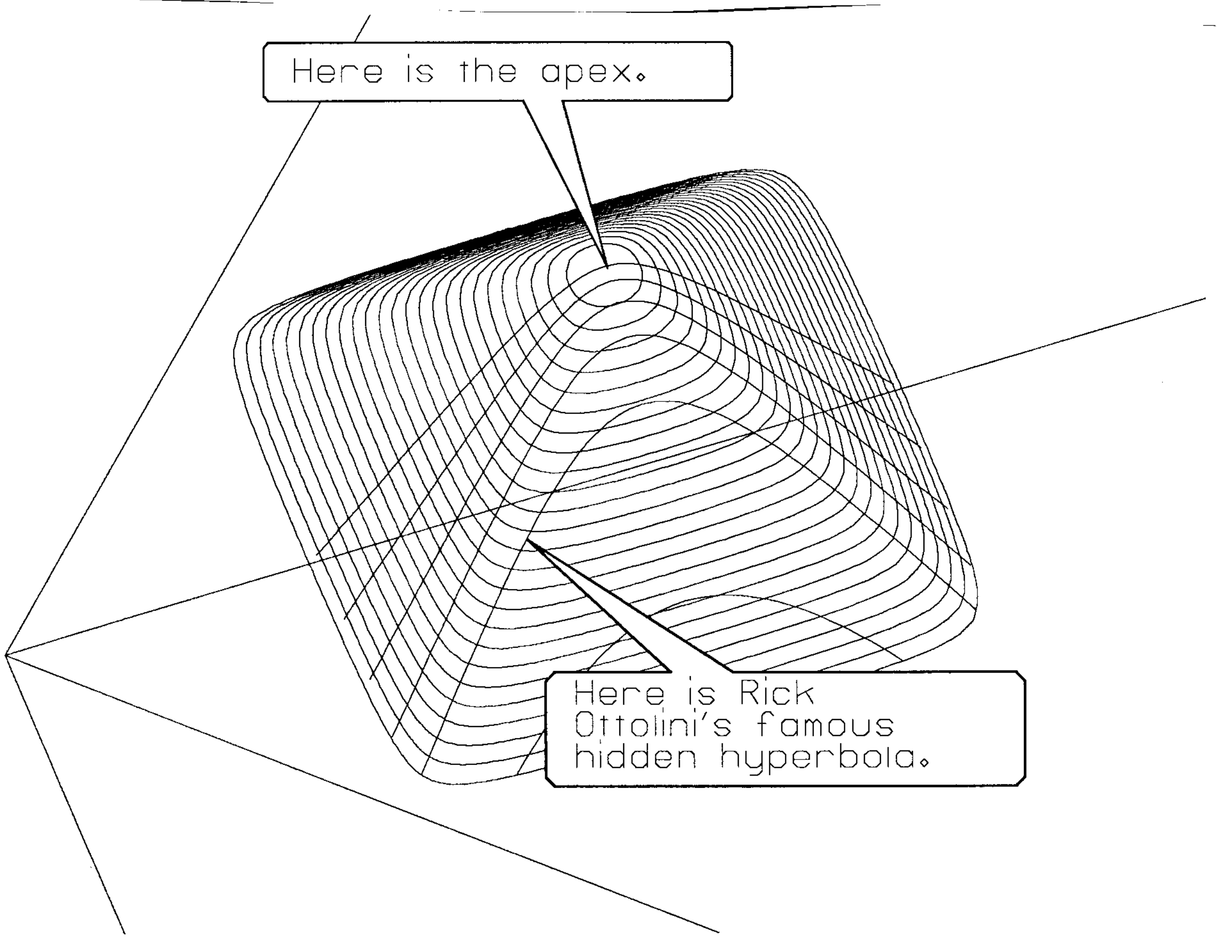
Spacing in balloon is not good. There are two problems: One is excessive granularity of letter sizes. Often you think it should fill the balloon differently, but it really can't. Another problem is that font width tables are not available thru vplot calls. If anyone wants to mess with this, everything you need to know is on the last page of *ball.c.* I think people might actually use all vplot's 16 fancy fonts, if they could scan them interactively.

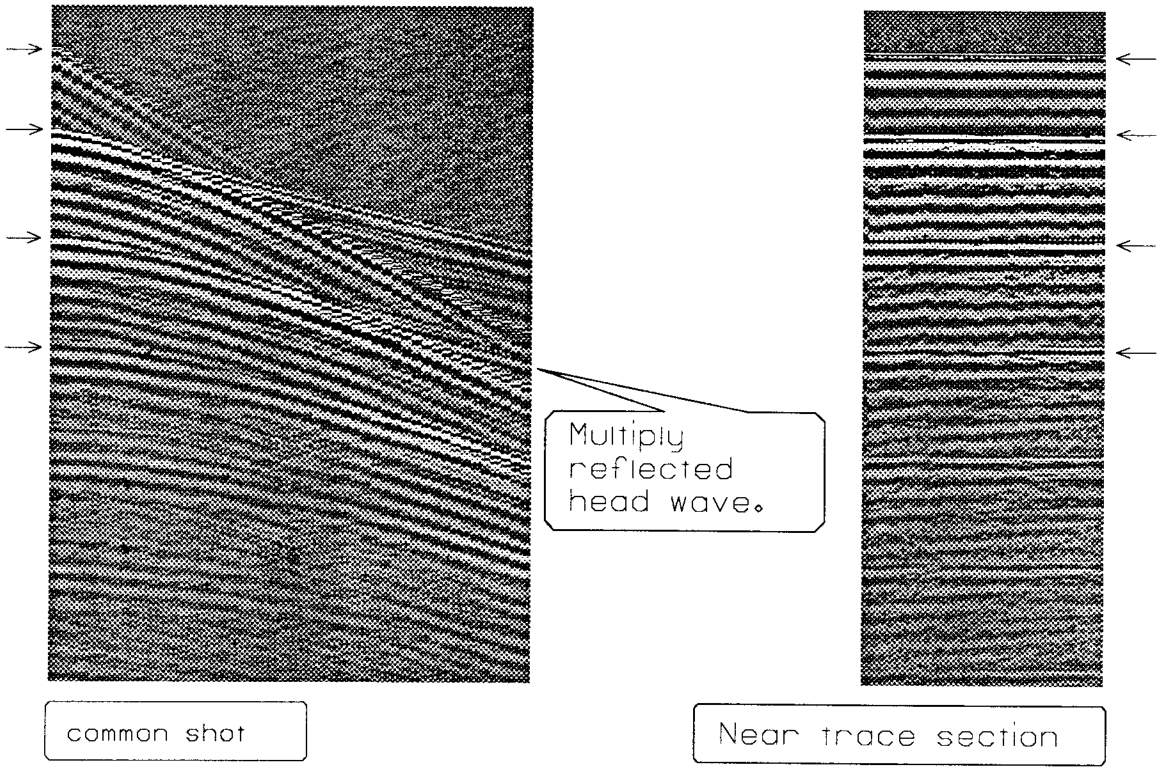
Raster data brightness is rendered both interactively and on the final copy. One polarity of clipped data is rendered interactively as red. For the vector plots, color is not rendered on the interactive screen but it is rendered in the plot file. SEP software has some capabilities to handle both forms of color at once, but this has not yet been evaluated or attempted.

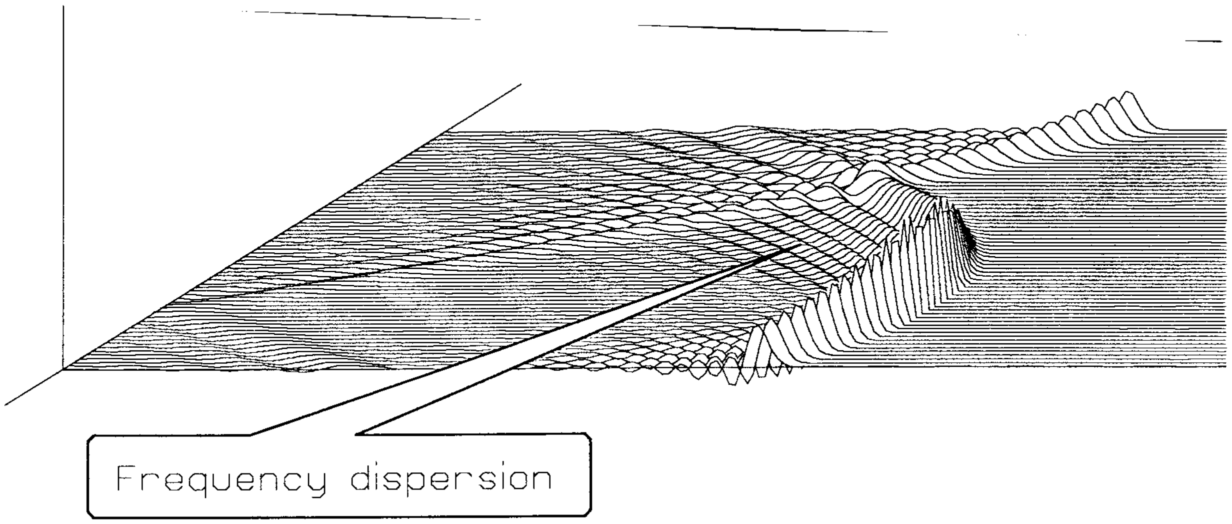
ACKNOWLEDGEMENT

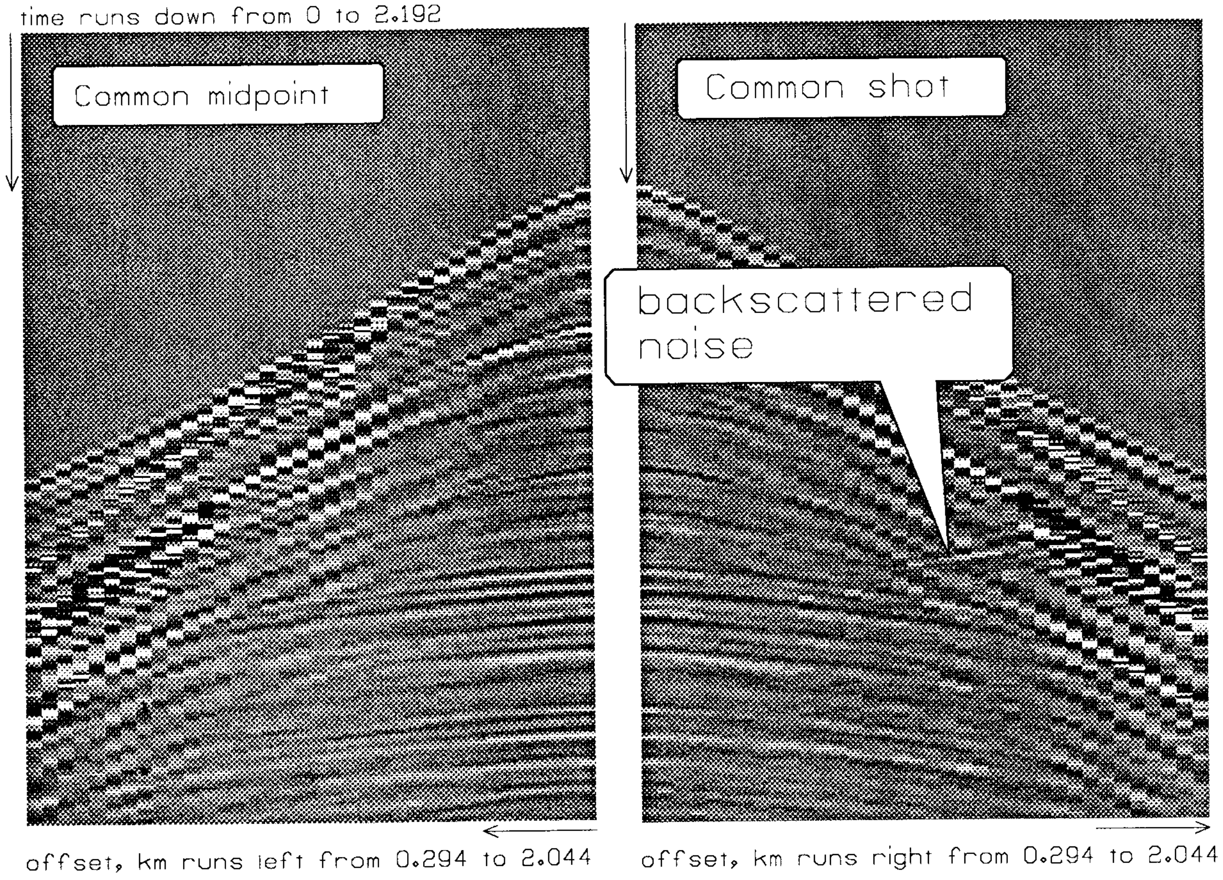
I am thankful to Joe Dellinger for installing raster in SEP's intermediate plot language, vplot, and to Steve Cole for a rewrite of pspen, SEP's filter for converting vplot into Postscript, that allowed me to use raster on the Apple laser printer.

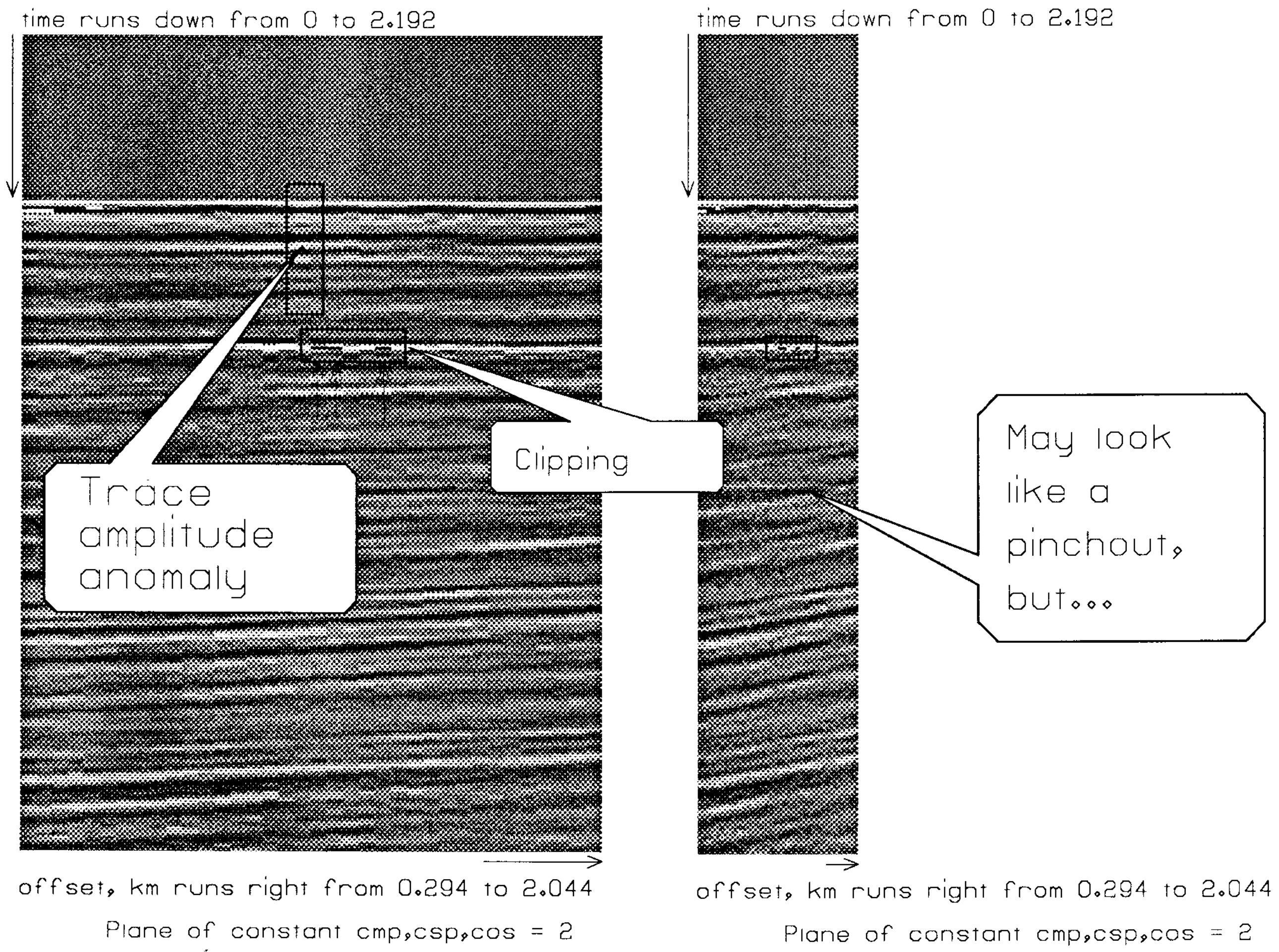
I am also thankful to Joe Dellinger for preparing for me a program *vppen* that among other things, converts the general vplot language to the restricted language that svplot supports, namely moves and draws.

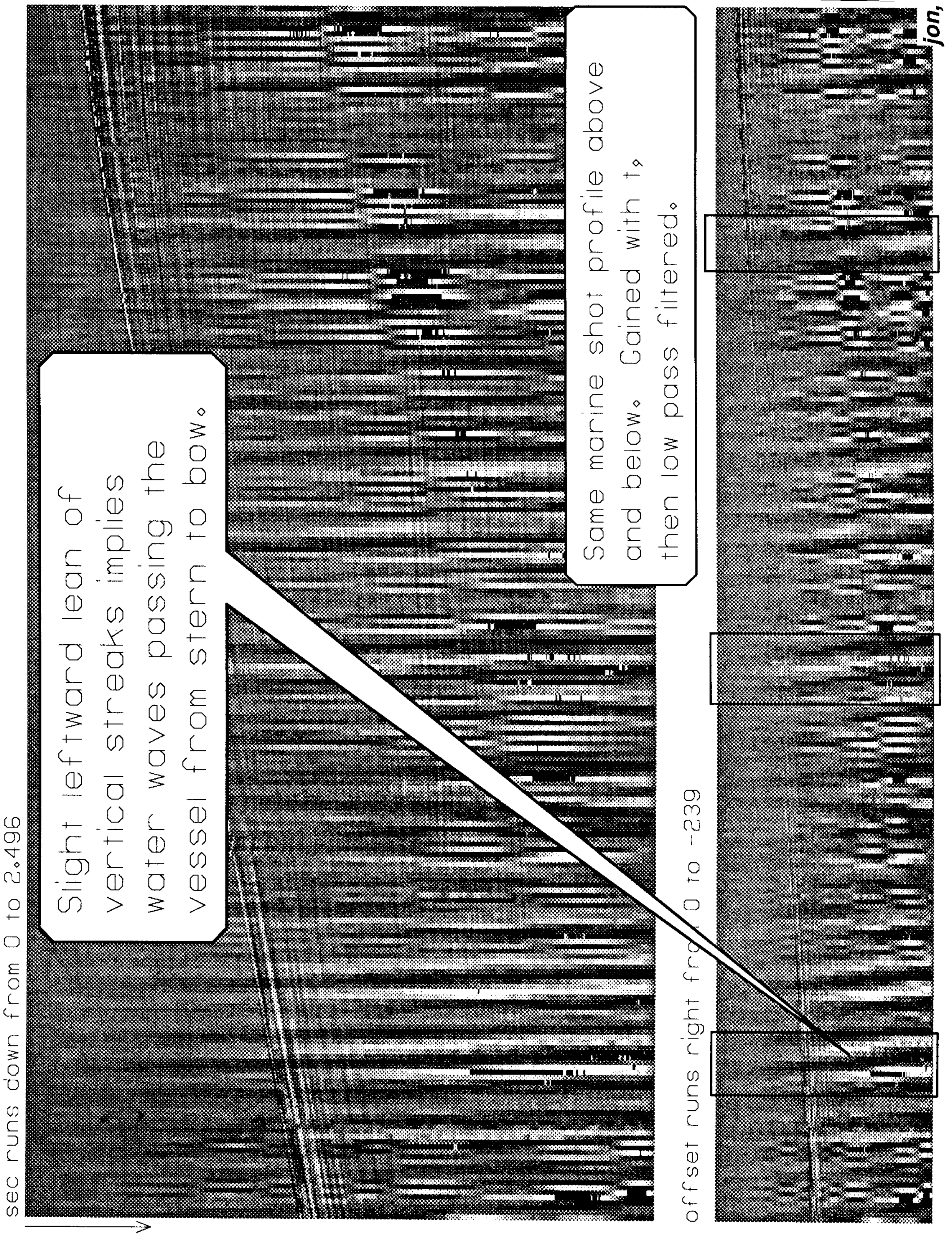

Here is the apex.
Here is Rick Ottolini's famous hidden hyperbola.

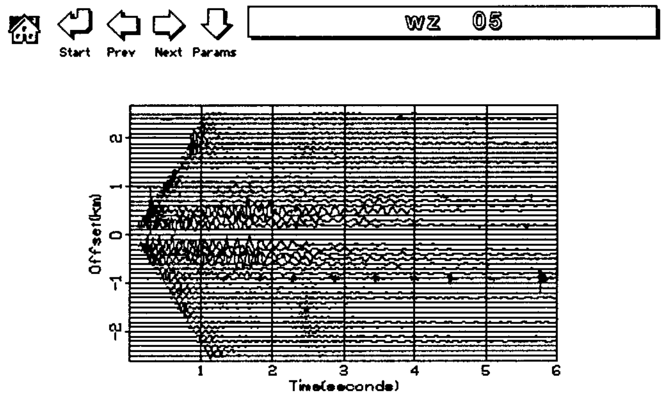

Multiply reflected head wave.
common shot
Near trace section

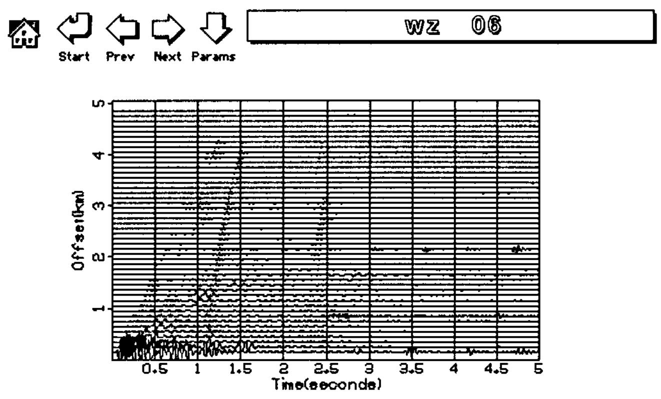

Frequency dispersion

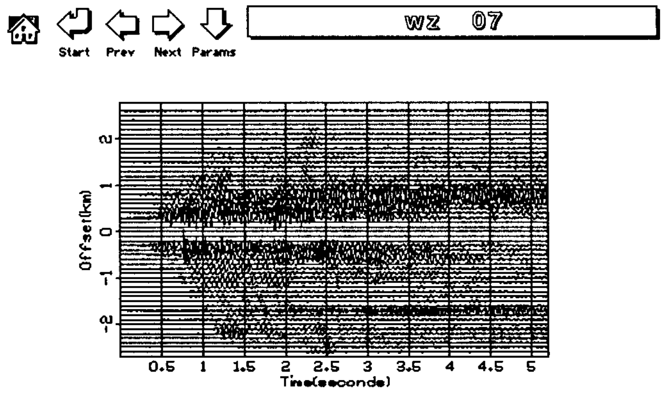

time runs down from 0 to 2.192
Common midpoint
Common shot
backscattered noise
offset, km runs left from 0.294 to 2.044
offset, km runs right from 0.294 to 2.044

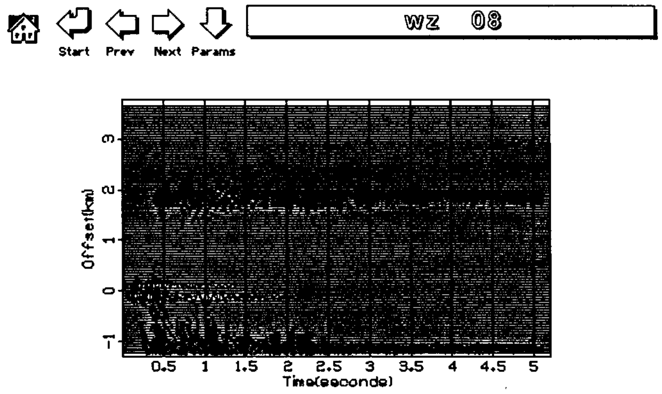

time runs down from 0 to 2.192
time runs down from 0 to 2.192
Trace amplitude anomaly
Clipping
May look like a pinchout, but...
offset, km runs right from 0.294 to 2.044
Plane of constant cmp,csp,cos = 2
offset, km runs right from 0.294 to 2.044
Plane of constant cmp,csp,cos = 2

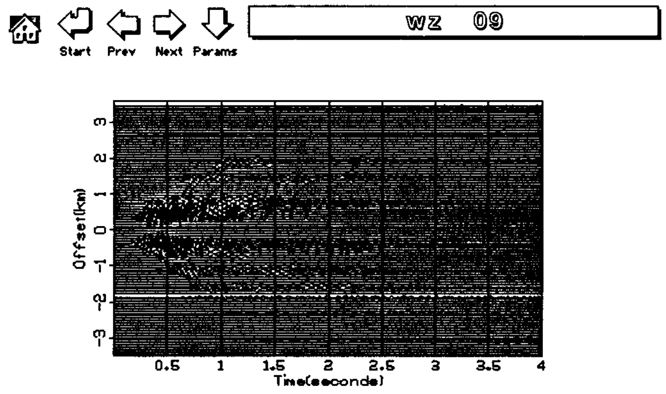

sec runs down from 0 to 2.496
offset runs right from 0 to -239
Slight leftward lean of vertical streaks implies water waves passing the vessel from stern to bow.
Same marine shot profile above and below. Gained with t, then low pass filtered.
jon,

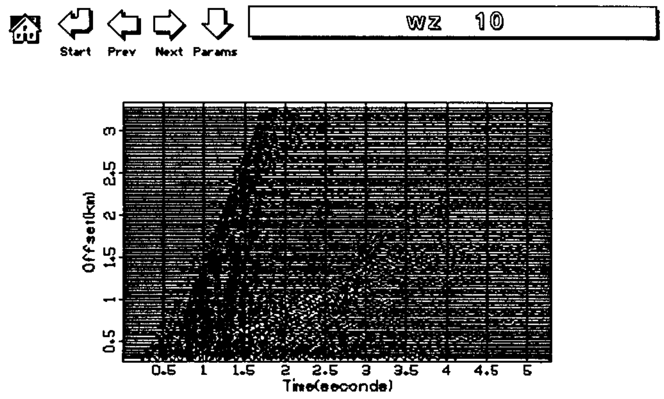

Start
Prev
Next
Params
wz 05
Offset(km)
Time(seconds)

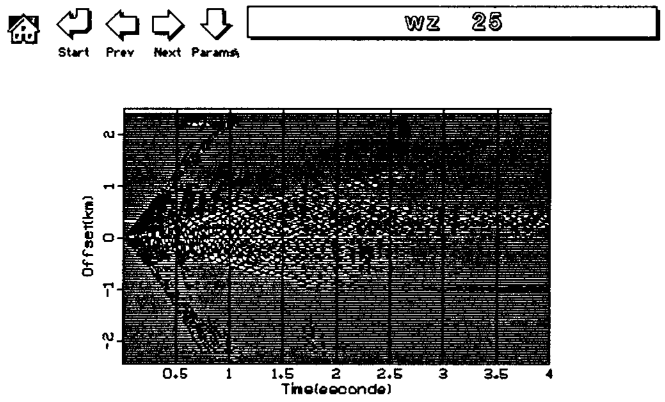

Start
Prev
Next
Params
wz 06
Offset(km)
Time(seconds)

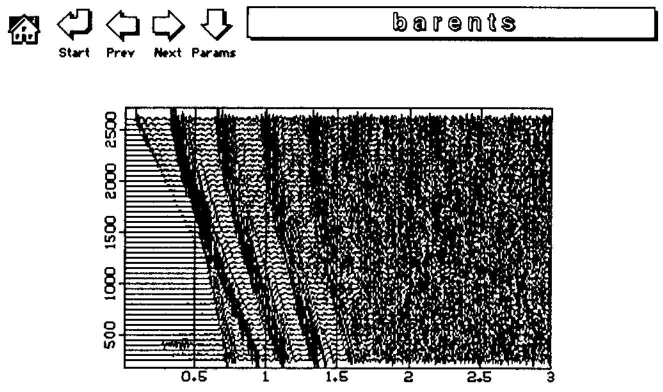

Start
Prev
Next
Params
wz 07
Offset(km)
Time(seconds)


Start
Prev
Next
Params
wz 08
Offset(km)
Time(seconds)


Start
Prev
Next
Params
wz 09
Offset(km)
Time(seconds)


Start
Prev
Next
Params
wz 10
Offset(km)
Time(seconds)


Start
Prev
Next
Params
wz 25
Offset(km)
Time(seconds)


Start
Prev
Next
Params
barents