Short note: 3-D CRS stack software

S. Bergler and A. Müller

email: *Steffen.Bergler@gpi.uka.de*

keywords: *3-D CRS software, search algorithms, utility programs*

ABSTRACT

The aim of this paper is to outline the current state of the 3-D Common-Reflection-Surface stack software developed in Karlsruhe. This includes software and data format requirements, used algorithms, and utility programs. Theoretical details are not discussed here. Readers may refer to the references given in the paper for more theoretical insight.

INTRODUCTION

Extending the concepts of the Common-Reflection-Surface (CRS) stack for 2-D data acquisition as described in Mann (2002) to the 3-D case is theoretically straightforward (Höcht, 2002). However, five years ago (when first versions of the 2D CRS stack have been coded) it was almost impossible to run multi-parameter search algorithms even on 2-D data sets in an academic environment in a reasonable computing time. This applies even more for 3-D data sets. Furthermore, 3-D data acquisition brings the problem of data storage since several hundreds of gigabytes for industrial data sets are not uncommon.

With the strong increase in computing power and capacity of hard-disks in the recent years, 3-D seismic data processing became feasible at the University of Karlsruhe to a certain extent. Therefore, the implementation of the 3D CRS stack has been commenced. Currently, two code versions exists. The main difference is that one of the codes is command line based, whereas the other uses a QT-based graphical user interface. The latter was implemented as part of Alex Müller's diploma thesis (see Figure 1). Concerning the expected input/output data format as well as the used search algorithms, both codes are almost identical, that is, they produce the same output sections. Therefore, we will not differentiate between both codes.

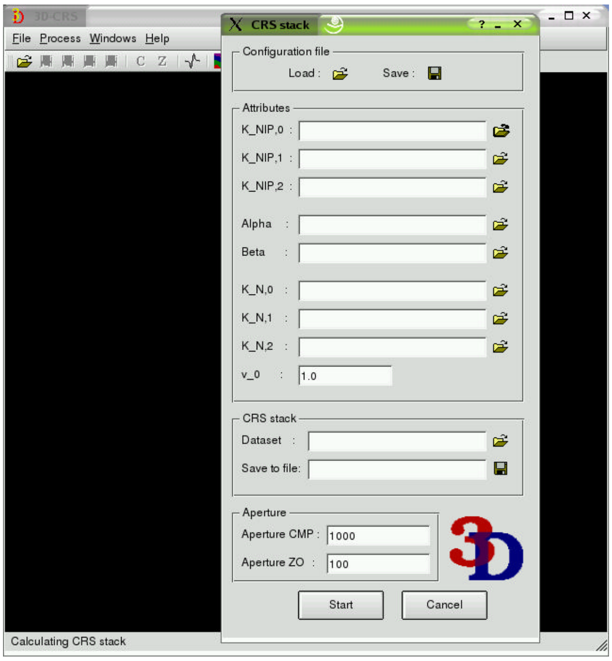
In the following, the functionality of the 3dcrs is outlined. Moreover user utility programs that can help to find the best set of parameters to run the code are introduced.

AIM OF THE 3D CRS PROCESSING

The goal of 3D CRS processing is to determine eight kinematic wavefield attributes associated with zero-offset (ZO) reflections. These can be used for several applications. One is to perform a 3D CRS stack in order to produce a ZO volume. Another very interesting application is the use of the attributes for the estimation of a smooth velocity model as described in Duveneck (2003) or a layered velocity model as explained in Höcht et al. (2003). The ZO volume as well as the coherence volume, which is a quality measure on how well the attributes are determined and which is produced during the CRS processing, are of help for the (automatic) attribute picking. As the result of the velocity estimation relies on the accuracy of the wavefield attributes, special attention must be paid to their determination.

HARDWARE AND SOFTWARE REQUIREMENTS

The 3dcrs code should comply to the current C++ standards. As far as we know, no platform specific features are used. In addition to common libraries of the C++ compiler GCC 3.x (Free Software Foundation)



v_0 : 1.0

Aperture CMP : 1000

Aperture ZO : 100

Calculating CRS stack

Figure 1: Graphical user interface of the 3D CRS stack program.

the following freely available libraries are required:

- QT 3.X (Trolltech) for the graphical user interface of 3dcrs.
- PTHREAD (Xavier Leroy) for the parallelization of 3dcrs.
- LIBDL (Free Software Foundation) for dynamic loading of shared libraries of 3dcrs.

DATA FORMAT

In principle, the input data to the 3dcrs program are expected in the SEISMIC UNIX (SU) data format. This is almost equivalent to SEG-Y format, apart from the additional 3200 byte text header and a 400 byte binary header at the beginning of the SEG-Y data. Conversion from SEG-Y to SU format can be easily done by the SU utility segyread. The SU package is freely available at `http://www.cwp.mines.edu/cwpcodes/index.html`. This conversion from SEG-Y to SU as well as the conversion from foreign endian to the native endian format can also optionally be performed within the 3dcrs code.

The format of all output sections is in SU. The conversion from SU to SEG-Y can be done by the SU utility segywrite.

SEARCH ALGORITHMS

General case

Mathematically, the CRS attribute determination is a global optimization problem. This means: find the best set of eight parameters to optimize the objective function that describes the quality of the fit of the CRS traveltime surface to the reflection event in a five-dimensional data space. The object function can be any coherence measure. In the current implementation of the 3D CRS stack, the coherence criterion semblance is used.

An eight-parameter optimization is a time-consuming task. Therefore, it is mandatory to devise a strategy to reduce the time necessary for the attribute search. In the current implementation, the computation time is reduced by splitting the attribute search. This means that the eight parameters are not determined by a simultaneous search for all parameters but by searching three, two, and three parameters successively in subsets of the input data. The strategy is as follows:

- find the three parameters in the common-midpoint (CMP) volume where the traces have the same midpoint as the ZO output trace
- perform a CMP stack and construct a ZO volume
- perform in this ZO volume a two-parameter search for the linear traveltime coefficients
- perform in the same ZO volume a three-parameter search for the quadratic traveltime coefficients

The parameters can subsequently enter into a local optimizations as initial values. But even a local eight-parameter optimization is, in general, computationally costly. Hence, it should be performed improving only a subset of the parameters in the whole data volume or, as reported in Müller (2003), in subsets. First tests using a modified version of the flexible polyhedron search have shown the potential for improving the imaging quality of the ZO volume after local optimization and, therefore, in the reliability of determined parameters (Müller, 2003).

Marine case

In marine data acquisition, where single vessels tow a multiple-streamer, traces are located only in a relatively small azimuth range within one CMP bin. This renders a stable determination of three stacking parameters in a CMP volume difficult or sometimes even impossible. In the current 3D CRS implementation one can therefore optionally do the following:

- perform only a one-parameter search for an azimuth-independent stacking velocity
- perform a one-parameter search for an azimuth-independent stacking velocity, afterwards perform a three-parameter search where the result of the one-parameter search is the starting value for all three parameters and only a small variation of the starting value is permitted

Both options make the search in the CMP volume faster. The second option attempts to get all possible information regarding the azimuth-dependent stacking velocity function.

PARALLELIZATION

Parallelization allows to speed up the search routines without loss of accuracy of the determined wavefield attributes. This is done by computing independent parts of the algorithms in parallel either on different processors (symmetric multi–processing (SMP)) or on different computers (networking). The amount of acceleration that can be achieved is thereby almost only limited by two factors: the number of processors or computers available and the level of parallelism, i. e., the number of independent parts that can be found for a specific problem. Although this seems to set up a linear relationship between the number of tasks and the speedup, a subtle relationship has to be kept in mind: each additional task requires some constant overhead for its initialization and some resources for runtime administration. Thus, the gain reduces by a small amount with each task that is added.

The implementation of the 3D CRS stack currently supports SMP, only. Thereby, the parallelization is done on a per–trace basis, i. e., each process computes the attributes of a single trace in the respective subset of the data. The program is divided into two independent parts, an administrative and a processing one. The administrative part periodically checks if a process computing the attributes has been completed (so–called polling). If so, the results are stored and a new process is started by initializing it with the next stacking trace and the traces within its aperture. A network based parallelization can be implemented in the same way.

3D CRS STACK RELATED SOFTWARE

In order to simplify the choice of processing parameters needed to run the 3D CRS stack, several utility programs have been written:

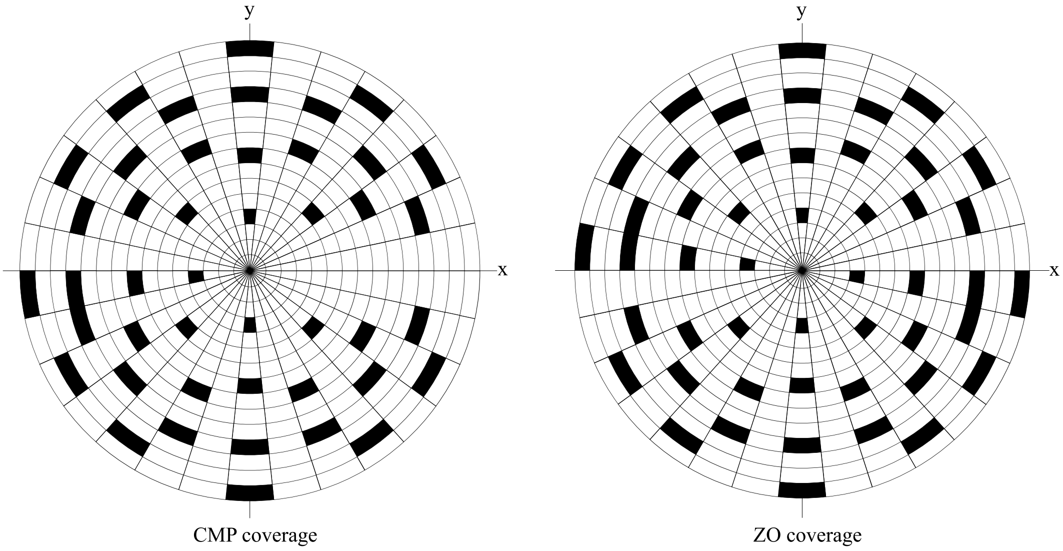


Figure 2: Example of rosette plots. For the displayed case each black segment indicates a trace location.

- With rosie one can display the azimuthal distribution of traces for a CMP or a ZO volume as a rosette plot (see Figure 2). This may help to chose the best search strategy as well as to check the reliability determined attributes that have been determined.
- With 3dcrsmo it is possible to produce moveout-corrected CMP, ZO, and CRS volumes in order to have control over the determined parameters. In principle, there are two reasons when the events are not flattened in moveout-corrected volumes: (a) the reflection is non-hyperbolic and the search apertures have to be adjusted or (b) the parameters are not well determined and the search strategy has to be changed. An example of a moveout corrected CRS volume is shown in Figure 3.
- The program 3dbuilder has been produced to visualize 3-D data volumes, e. g., results of the 3-D CRS stack. The front panel is shown in Figure 4.

CONCLUSIONS AND OUTLOOK

The 3dcrs code as well as the utility programs for processing have been shortly introduced. The 3-D CRS stack produces ZO volumes and attribute volumes which are of interest to produce a detailed depth image. Therefore, the 3-D CRS stack is, of course, not to be seen as a stand-alone imaging tool but has to be integrated into an entire workflow (as it is currently done in the 2-D case). Advantages of this are:

- CRS stack volumes and attribute volumes are provided for interpretation
- kinematic wavefield attributes can be used to construct a velocity model in a data-driven way
- picking of events can be done in high-quality ZO volumes
- post-stack depth migration can be performed with high-quality ZO volumes to produce competitive results when compared to pre-stack depth migration

PUBLICATIONS

A detailed description of a 3D CRS stack implementation is shown in Müller (2003).

ACKNOWLEDGMENTS

This work was kindly supported by the sponsors of the *Wave Inversion Technology (WIT) Consortium*, Karlsruhe, Germany.

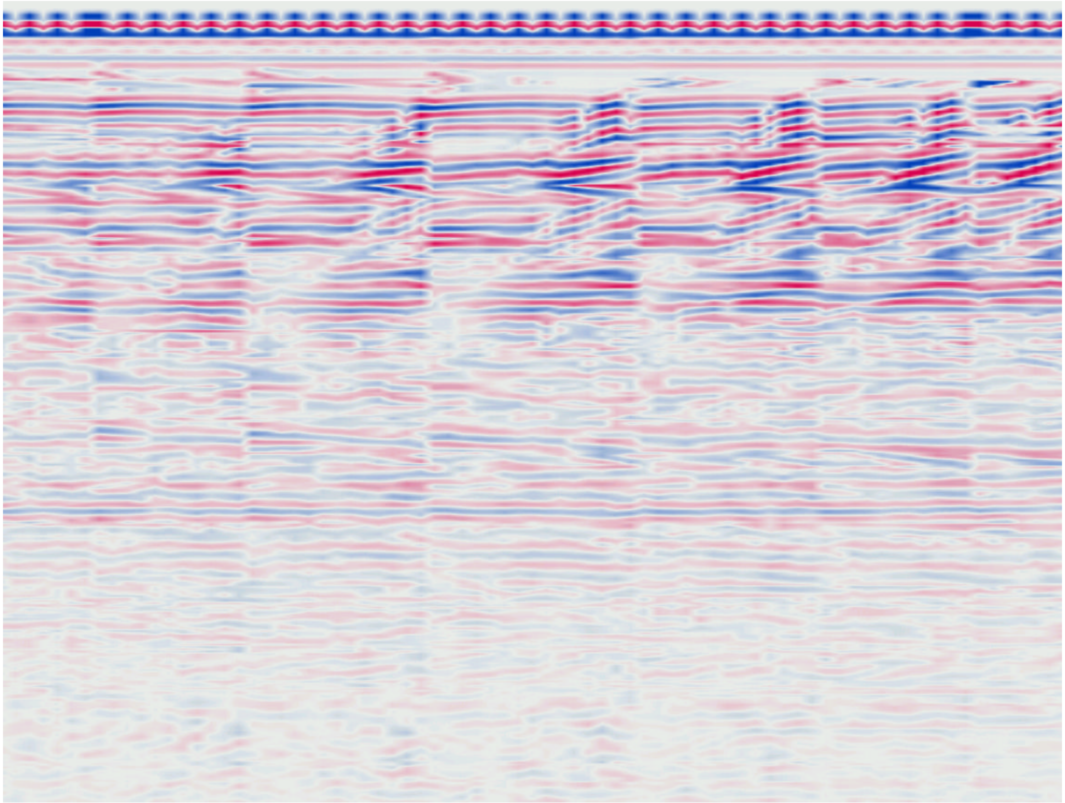
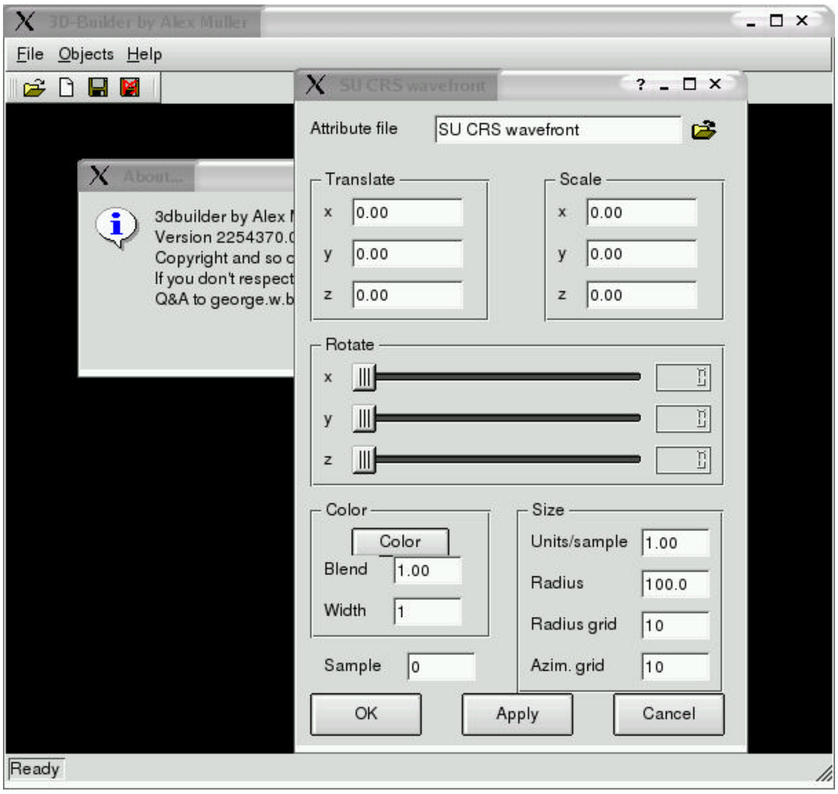


Figure 3: Example of a moveout-corrected CRS volume. If a reflection event can be approximated by the hyperbolic CRS traveltime surface, the event is flat.



3dbuilder by Alex M
Version 2254370.0
Copyright and so c
If you don't respect
Q&A to george.w.b

Attribute file: SU CRS wavefront

Translate
x 0.00
y 0.00
z 0.00

Scale
x 0.00
y 0.00
z 0.00

Rotate
x
y
z

Color
Blend 1.00
Width 1

Sample 0

Size
Units/sample 1.00
Radius 100.0
Radius grid 10
Azim. grid 10

Figure 4: Front panel of the program 3dbuilder.

REFERENCES

Duveneck, E. (2003). 3D tomographic velocity model estimation with CRS attributes. *Annual Report, Wave Inversion Technology Consortium*.

Höcht, G. (2002). *Traveltime approximation for 2D and 3D media and kinematic wavefield attributes*. PhD thesis, Universität Karlsruhe, Germany.

Höcht, G., Bergler, S., Perroud, H., and de Bazelaire, E. (2003). 3D macro-velocity invserion by means of kinematic wavefield attributes. In *Extended Abstracts*. 65th Mtg., Eur. Assn. Geosci. Eng. Session D-31.

Mann, J. (2002). *Extensions and applications of the Common-Reflection-Surface Stack Method*. Logos Verlag, Berlin.

Müller, N.-A. (2003). The 3D Common-Reflection-Surface Stack - Theory and Application. Master's thesis, Universität Karlsruhe.