GPR_XSU

Version: 1.01.10.01
Last revision date: 1-10-2001
Description: GPR_XSU transforms GPR data from traces collected evenly in time to evenly spaced, spatially located traces. Use GPR_PROC to process the data before transforming. GPR_XSU differs from GPR_XFRM in the following ways. 1) Only DZT files can be input; other formats are not accepted. To convert from one GPR storage format to another, use GPR_CONV.EXE. Either or both DZT and SU files can be output. 2) The antenna separation must be input. 3) The offset to the marking point on the antenna array must be input. 4) The output XYZ file also contains the XYZ locations of the leading and trailing antennas after the "mark" location.

The transformation is performed as follows. The user specifies start and stop locations and a step size, and locations are calculated for the new traces. The user also specifies a bin size (which defaults to the step size but may be smaller or greater than the step size to accommodate dense or sparse data sets). Each bin is centered about a new trace location. A location (user-defined X, Y, and Z values) is assigned each old trace by splining, using the data from the MRK and XYZ files. All traces located within the bin are averaged to create the new trace for each bin. Empty bins are set to the mean value of the input data type (e.g., 32768 for unsigned short integers). Empty bins can be avoided by enlarging the step size or the bin size. X, Y, and Z values are calculated for each bin and saved to disk as an "XYZ" file. A "MRK" file is also saved.

The input to this program is a "CMD" file, an ASCII text file containing keywords (or parameters) describing how to process the radar data. The CMD file specifies the data file name (and optionally the storage format). Inspect the example file GPR_XSU.CMD for usage.

NOTES:
Only 1 to 4 headers are supported in DZT files.
There is no graphic display of the data.
To display the processed data, use programs such as GPR_DISP.EXE or FIELDVEW.EXE.

THE KEYWORDS

Following is the list of keywords and their default values. The documentation format is:
"KEYWORD: **keyword** = default value".
Look at GPR_XSU.CMD as an example command file with correct usage and default keyword values. The file GPR_XSU.CMD has most comments stripped out, and GPR_XSU.CM_ has all comments removed.

********** PROGRAM CONTROL **********

KEYWORD: **batch** = "FALSE"
Place the program in batch mode (no pauses) if "TRUE". The program will normally pause at times before ending.

KEYWORD: **debug** = "FALSE"
Place the program in debug mode if "TRUE" (for developers)

********** SPECIFICATION OF INPUT DATA **********
The storage format is determined by inspecting the file. If the program cannot recognize a flavor of the DZT format below then an error message may be issued.
Recognized storage formats are:
DZT GSSI SIR-10A file with embedded (512- or 1024-byte) info header

KEYWORD: **dzt_infilename** = ""
This is the input GPR binary data file name.

KEYWORD: **channel** = 0
This keyword is for use with multiple channel DZT files only. It is the channel number in multi-channel data sets and is indexed from 0. GSSI data can have up to 4 channels (channel = 0, 1, 2, or 3).

KEYWORD: **mrk_infilename** = ""

KEYWORD: **xyz_infilename** = ""

MRK and XYZ files are used to determine which traces are "marked" and what the coordinates are of the marked traces. They contain the number of sets stated on the first file record with the sets listed on following records.

Example MRK file containing marked trace locations:
```
3
104
256
897
```

Example XYZ file containing X, Y, and Z locations of the marked traces:
```
3
10.0  10.0  293.456
20.0  10.0  294.567
30.0  10.0  295.678
```

KEYWORD: **mrk_offset** = 0
This is the distance from the "marked" point on the antenna array to the center of the antenna array. Greater than 0 is to forward (that is, in the direction of tow).

KEYWORD: **ant_sep** = 0
This is the distance in meters between the centers of the antennas.

********** SPECIFICATION OF OUTPUT FILES **********
The binned GPR data are stored in either or both the DZT or SU storage formats. At least one output filename must be given. Use GPR_CONV.EXE to convert between storage formats. Two more files are automatically written: an MRK file and an XYZ file. The MRK file contains every trace number. The XYZ file contains the X, Y, and Z locations for the "mark point" (the center of the antenna array) and the leading and trailing antennas.

KEYWORD: **dzt_outfilename** = ""
This is the name of the DZT GPR data file that is written to disk. Note that output files are single-channel only.

KEYWORD: **su_outfilename** = ""
This is the name of the GPR data file that is written to disk using the SU storage format.

********** SELECTING TRANSFORMING OPTIONS **********
This group determines how the GPR data are transformed. Each trace in the input data file is assigned an X, Y, and Z value based on the XYZ and MRK files that were supplied. You must select either the "X" or "Y" direction to use as the reference axis for the new, binned, output data.

KEYWORD: **spatial_dir** = 0
This is the reference axis to use for the binning operation and for the "spatial_....." keywords below. Select 0 for the X-axis, or 1 for the Y-axis.

KEYWORD: **spatial_start** = 0.0
This is the starting coordinate on the reference axis.

KEYWORD: **spatial_stop** = 0.0
This is the ending coordinate on the reference axis.

KEYWORD: **spatial_step** = 0.0
This is the uniform distance between binned, output traces.

############ NOTE #############
"spatial_start" and "spatial_stop" must have the same directional sense as the coordinates in the XYZ file. For example, if "spatial_dir" is 1 and the Y values in the XYZ file are increasing as trace numbers are increasing, then "spatial_start" must be less than "spatial_stop" and "spatial_step" must be positive. On the other hand, if the Y values decrease as the trace numbers increase, then "spatial_start" must be greater than "spatial_stop" and "spatial_step" must be negative.
##############################

KEYWORD: **bin_size** = 0.0
This is the size of the bin used to calculate new traces. If it is negative or equal to 0.0, then it is the same as "spatial_step" size. If it is greater than 0.0, then it can be less than or greater than "spatial_step". The bins are centered about the new trace locations (traces are "placed" at the center of a bin) determined from the above 4 values. All traces that are located in a bin are averaged to create the new trace (this is called stacking).

Usage: `GPR_XSU cmd_filename`
Required command line arguments:
cmd_filename - The name of the keyword file.
Optional command line arguments (do not include brackets): none
Examples:

```
gpr_xsu xfile1.cmd
```