MBIO
Format independent input/output library for swath mapping sonar data.

DESCRIPTION

MBIO (MultiBeam Input/Output) is a library
of functions used for reading and writing swath mapping sonar
data files. MBIO supports a large number of data
formats associated with different institutions and different sonar systems.
The purpose of MBIO is to allow users to write processing and display
programs which are independent of particular data formats and to provide
a standard approach to swath mapping sonar data i/o.

DATA TERMINOLOGY

MBIO handles three types of swath mapping data: beam bathymetry,
beam amplitude, and sidescan. Both amplitude and sidescan represent
measures of backscatter strength. Beam amplitudes are backscatter
values associated with the same preformed beams used to
obtain bathymetry; MBIO assumes that a bathymetry value
exists for each amplitude value and uses the bathymetry beam
location for the amplitude. Sidescan is generally constructed with
a higher spatial resolution than bathymetry, and carries its
own location parameters. In the context of MB-System documentation,
the discrete values of bathymetry and amplitude are referred
to as "beams", and the discrete values of sidescan are referred to
as "pixels". An additional difference between "beam" and "pixel"
data involves data flagging. An array of "beamflags" is carried by
MBIO functions which allows the bathymetry (and by extension
the amplitude) data to be flagged as bad. The details of the
beamflagging scheme are presented below.

OVERVIEW

MBIO opens and initializes sonar data files for reading
and writing using the functions mb_read_init and mb_write_init,
respectively. These functions return a pointer to a data structure
including all relevent information about the opened file, the
control parameters which determine how data is read or written,
and the arrays used for processing the data as it is read or written.
This pointer is then passed to the functions used for reading or
writing. There is no limit on the number of files which may be
opened for reading or writing at any given time in a program.

The mb_read_init and mb_write_init functions also return
initial maximum numbers of bathymetry beams, amplitude beams,
and sidescan pixels that can be used to allocate data storage
arrays of the appropriate sizes. However, for some data formats

there are no specified maximum numbers of beams and pixels, and so in general the required dimensions may increase as data are read. Applications must pass appropriately dimensioned arrays into data extraction routines such as mb_read, mb_get, and mb_get_all. In order to enable dynamic memory management of thse application arrays, the application must first register each array by passing the array pointer location to the function mb_register_array.

Data files are closed using the function mb_close. All internal and registered arrays are deallocated as part of closing the file.

When it comes to actually reading and writing swath mapping sonar data, MBIO has three levels of i/o functionality:

1: Simple reading of swath data files.
The positions of individual beams and pixels are returned in longitude and latitude by mb_read() and in acrosstrack and alongtrack distances by mb_get(). Only a limited set of navigation information is returned. Comments are also returned. These functions can be used without any special include files or any knowledge of the actual data structures used by the data formats or MBIO.

2: Complete reading and writing of data structures containing all of the available information. Data records may be read or written without extracting any of the information, or the swath data may be passed with the data structure. Several functions exist to extract information from or insert information into the data structures; otherwise, special include files are required to make sense of the sonar-specific data structures passed by level 2 i/o functions.

3: Buffered reading and writing of data structures containing all of the available information. The primary functions are:

The level 1 MBIO functions allow users to read sonar data independent of format, with the limitation that only a limited set of navigation information is passed. Thus, some of the information contained in certain data formats (e.g. the "heave" value in Hydrosweep DS data) is not passed by mb_read() or mb_get(). In general, the level 1 functions are useful for applications

such as graphics which require only the navigation and the depth and/or backscatter values.

The level 2 functions (mb_get_all() and mb_put_all()) read and write the complete data structures, translate the data to internal data structures associated with each of the supported sonar systems, and pass pointers to these internal data structures. Additional functions allow a variety of information to be extracted from or inserted into the data structures (e.g. mb_extract() and mb_insert()). Additional information may be accessed using special include files to decode the data structures. The great majority of processing programs use level 2 functions.

The level 3 functions provide buffered reading and writing which is useful for applications that generate output files and need access to multiple pings at a time. In addition to reading (mb_buffer_load()) and writing (mb_buffer_dump()), functions exist for extracting information from the buffer (mb_buffer_extract()) and inserting information into the buffer (mb_buffer_insert()).

MBIO supports swath data in a number of different formats, each specified by a unique id number. The function mb_format() determines if a format id is valid. A set of similar functions returns information about the specified format (e.g. mb_format_description(), mb_format_system(), mb_format_description(), mb_format_dimensions(), mb_format_flags(), mb_format_source(), mb_format_beamwidth()).

Some MB-System programs can process multiple data files specified in "datalist" files. Each line of a datalist file contains a file path and the corresponding MBIO format id. Datalist files can be recursive and can contain comments. The functions used to extract input swath data file paths from datalist files includes mb_datalist_open(), mb_datalist_read(), and mb_datalist_close().

A number of other MBIO functions dealing with default values for important parameters, error messages, memory management, and time conversions also exist and are discussed below.

For more see complete UNIX user manual http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbio.html#lbAU

Mbmakeplatform
Creates or modifies an MB-System platform file, which
defines the sensors on a survey platform, along with the positional and angular
offsets for each sensor.

DESCRIPTION

MBmakeplatform creates or modifies an MB-System platform file.

MB-System programs such as mbpreprocess and mbprocess use
a platform model to calculate the trajectory of mapping
sensors from position and attitude data. In general, position and attitude data
derive from sensors located separately from mapping sensor on any survey platform,
and so the position and attitude of mapping sensors must be calculated using
the relative positional and angular offsets between the mapping sensors and
those ancilliary sensors. These three-dimensional rotations and translations
are called lever arm calculations.

A platform file defines the sensors on a survey platform, along with the positional and angular
offsets for each sensor. A number of data types can be defined (e.g. position,
sensor depth, roll and pitch, heading, bathymetry, backscatter, photographic
imagery), and the source sensor for each data type can be set separately.
The specifics of survey platforms can vary greatly, and consequently the format
used to describe an integrated survey system must have considerable flexibility.

The supported types of survey platforms include surface vessels, ROVs,
AUVs, manned submersibles, and tow bodies. Sensors may include survey sensors
such as multibeam sonars, sidescan sonars, lidars, and cameras. Sensors may also
include equipment that generate position and/or attitude data, such as GPS's,
VRUs, pressure gauges, and INS's. All sensors produce at least one kind of data
(e.g. position), and most produce multiple kinds of data. For instance, modern
multibeam sonars produce both bathymetry and backscatter, and inertial navigation
systems used on autonomous platforms produce both position and attitude data.
Through this program, users can construct platform models that represent complex
survey systems with multiple mapping sensors and redundant position and attitude
sensors.

For complete info
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbmakeplatform.html

MBhysweeppreprocess

Performs preprocessing of multibeam data in the Hysweep HSX format (MBIO format 201).

DESCRIPTION

mbhysweeppreprocess reads a Hysweep HSX (format 201) file, interpolates the asynchronous navigation and attitude onto the multibeam data, and writes a new HSX file with that information correctly embedded in the multibeam data. The user must specify a projection for the easting-northing navigation used in HSX files. This program can also fix various problems with multibeam data, and allows for lever arm correction of offsets between the sonar, the motion sensor, and the positioning sensor (provided the navigation and attitude data included in the HSX file are uncorrected for sensor offsets).

For complete info:
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbhysweeppreprocess.html

mbsvpselect
Mbsvpselect chooses and implements the best available sound speed model for each swath file in a survey

DESCRIPTION

Mbsvpselect chooses and implements the best available sound speed model for each swath file in a survey. The user provides a list of the available sound speed models and specifies the criteria used for model selection. The program uses mbset to turn on bathymetry recalculation by raytracing through the sound speed model selected for each swath file.

Description:

The tool aims to help users to automatically apply the sound velocity correction to the survey files. since most surveys involve several SVPs, the seletion of the appropriate SVP for each survey profile is still missing in MB-System.

After finding the appropriate svp for each profile based on the choosed method, the results are copied to a txt file that shows each survey profile with the corresponding SVP. the tool also calls mbset automatically so no need to assign SVP to the data. it is done automatically.

There are 5 methods for choosing the appropriate SVP for each survey profile. These methods are:

1. Nearest SVP in position: the middle position of each survey profile is calculated and the geodesics (shortest distance on the ellipsoid) to all SVPs are calcualted. and the SVP with the shortest distance is chosen. when the middle position of the survey profile is calculated there is an option to check for 0 lat 0 long wrong values. if it is

found at the starting the geodesic will be calculated to the end of the profile.

2. Nearest in time: the time interveal between the starting time of the profile and the time of the SVP, and the SVP with the shortest interval will be chosen.

3. Nearest in position within time: a default time radius from the profile is set as 10 hours, and within this period the nearest SVP in position is chosen. if none of the SVPs are within this period the nearest in position will be taken despit of the period threshold. The period threshold can be set by the user.

4. Nearest in time within range: similar to the previous option but this time a default range of 10000 meters is set and within this range the svp nearest in time is chosen. also this 10000 meter value could be set by the user.

5. Nearest in season within range: similar to the previous option the selected SVP could be chosen based on the month only not on the year. it means within the specified range the user could chose either the svp nearest in time or the svp nearest in month (this could be interpreted as the svp that falls in the same seasonal period despite of the year when it was taken).

Mbsvpselect reads the .inf file of each swath file referenced in a recursive datalist structure to determine the location and collection time of the relevant data. The ancilliary *.inf, *.fbt, and *.fnv files must be created first. The water sound speed models (called SVPs by convention as an acronym for Sound Velocity Profiles) to be used must include one of three supported file headers specifying the time and location of the model.

For complete info:
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbsvpselect.html

mbset
This is a utility for creating and
modifying mbprocess parameter files.

The actions of mbprocess are controlled by text
parameter files. Each mbprocess parameter file
contains single line commands that set
processing modes and parameters. The -P option of

mbset is used to modify a single mbprocess parameter
command. This option can be invoked as many times as desired
on the command line, allowing mbset to set multiple
mbprocess processing parameters and modes.
If the swath data file specified by the -Iinfile option of
mbset has an existing mbprocess parameter file,
then that parameter file will be read and the existing
parameter values will be modified. If no mbprocess
parameter file exists, then mbset starts with
default processing parameters, modifies those, and then
generates a new parameter file.

For complete info:
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbset.html

Mbvelocitytool
Interactive water sound velocity profile editor.

DESCRIPTION

MBvelocitytool is an interactive water sound velocity profile
(SVP) editor used to examine multiple SVPs, to create
new SVPs, and to model the impact of SVP modification on swath bathymetry
data. SVPs created using MBvelocitytool can be used by the program
mbprocess to recalculate swath bathymetry from raw travel time
and angle data.

In general, MBvelocitytool is used to examine SVPs obtained from
swath data files (see mbsvplist manual page), XBTs, CTDs, or
databases, and to construct new profiles consistent with these various
sources of information. The SVPs are represented by a set of paired
depth and velocity values which are connected by linear interpolation.
Users may load a number of SVPs for display. Users may load or create
a single editable SVP and then interactively modify this profile.

When users load swath bathymetry data containing raw travel time and
angle data (many but not all swath data format include this information),
MBvelocitytool recalculates the bathymetry by raytracing through
the current SVP model. If the bathymetry in the input swath file has
been edited (e.g. with mbedit), then the associated edit save
file (*.esf file) will be loaded as well, and the beams flagged as
bad will not be used by MBvelocitytool.
The bathymetry of each ping is fit with a line,
and bathymetry residuals are calculated for each good beam relative
to the linear fit. The average of the bathymetry residuals is displayed
along with "error bars" indicating the standard deviations of the
residuals. Anomalously shallow bathymetry maps into negative residuals

and deep bathymetry into positive residuals; the residuals are displayed so that shallow is up and deep is down. If the seafloor is reasonably smooth so that a linear fit is appropriate, then the residuals will accurately reflect any problems with the water velocity profile. If the water velocity profile is correct, then the residual plot will be roughly flat. If the water velocity profile is significantly in error, then the outer beam depths may anomalously shallow (edge curl up) or deep (edge curl down). In practice, the editable velocity profile is altered interactively until a reasonably residual pattern is achieved.

In order to calculate bathymetry values from travel time observations, geometrical raypaths are traced through the SVP for each beam. Because the sound velocity gradients are uniform between the depth-velocity nodes (linear interpolation), the raypaths are be calculated analytically as pieces of circular arcs. This raytracing algorithm is the same used in the program mbprocess.

One important aspect of the raytracing is the handling of the initial takeoff angles associated with each beam or sounding. In general, the raytracing will begin at a point in the sound speed model that has a sound velocity different than the surface sound velocity (SSV) used by the mapping sonar for the original beamforming. The usual approach is to use Snell's law to adjust the starting angle for this change in sound velocity. This amounts to an assumption that the original SSV was correct and that the rays pass through an insignificantly thick layer in which the sound speed equals the SSV before transitioning to the sound speed implied by the SVP. This is the default setting for raytracing in MB-System. Alternatively, one can proceed with raytracing using the original angle but this is rarely useful or correct. Finally, if the SSV used by the sonar is judged to have been incorrect, then the takeoff angle must be corrected for the erroneous beamforming as well as for the difference between the SSV and the initial raytracing sound velocity. This correction must take the sonar geometry into account because the impact of changing the SSV on a beam angle from a flat receive array is very different from a V-shaped or curved array. All three of these angle correction modes are available in mbvelocitytool.

MBvelocitytool can be used in conjunction with mbprocess. If the user uses the Save swath svp file option to save an SVP model developed through the analysis of a particular swath data file, MBvelocitytool also sets the associated mbprocess parameter file so that mbprocess recalculates the bathymetry using the new SVP model. The program mbset may be used to set the SVP file in the parameter file for any swath data file. Users may also save SVP models without setting any mbprocess parameters by using the Save editable profile option.

Sometimes the bathymetry residuals show structure indicative of persistent artifacts in the bathymetry (e.g. certain parts of the swath may be persistently shallower or deeper than the rest of the swath). In this situation, it is possible to export the residuals and to then apply them in mbprocess as static corrections to the bathymetry. This is accomplished by using the Save residuals as offsets option under the File menu.

If a user attempts to read in swath bathymetry that does not contain the travel time and beam angle data required for bathymetry recalculation, MBvelocitytool will estimate the travel times and angles from the bathymetry by assuming a 1500 m/s half-space (and then post a warning dialog). Although the user can proceed to model bathymetry recalculation by modifying the active SVP just as with proper data, the travel times and angles are not in general correct and so the modeling and any results it gives are, well, bogus. A more useful approach is to leave the SVP alone and simply export the residuals to be applied as static corrections in mbprocess. This approach allows users a practical means of correcting older multibeam bathymetry that was originally calculated with an incorrect SVP but which contains no travel time or angle data.

For compete info:
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbvelocitytool.html

Mbprocess
This program performs a variety of swath data processing functions in a single step (producing a single output swath data file), including merging navigation, recalculating bathymetry from travel time and angle data by raytracing through a layered water sound velocity model, applying changes to ship draft, roll bias and pitch bias, applying tides, and applying bathymetry edits from edit save files.

DESCRIPTION

The program mbprocess is a tool for processing swath sonar bathymetry data. This program can perform a variety of swath data processing functions in a single step (producing a single output swath data file), including:

- Merge edited navigation generated by mbnavedit.
- Apply bathymetry edit flags from mbedit and mbclean
- Recalculate bathymetry from raw travel time and angle data by raytracing through water sound speed

models from mbvelocitytool or mbsvplist.
- Apply changes to roll bias, pitch bias, heading bias, and draft values.
- Recalculate sidescan from raw backscatter samples (Simrad multibeam data only).
- Apply corrections to sidescan based on amplitude vs grazing angle tables obtained with mbbackangle.
- Apply tides to bathymetry.
- Insert metadata.

The actions of mbprocess are controlled by text parameter files. Each mbprocess parameter file contains single line commands that set processing modes and parameters. The program mbset can be used to create and modify mbprocess parameter files. Other programs such as mbedit, mbnavedit, mbvelocitytool, mbnavadjust, and mbclean modify or create (if needed) mbprocess parameter files.

The input file "infile" must be specified with the -I option. If "infile" is a datalist, then mbprocess will attempt to process each swath data file identified by recursively reading the datalist. Otherwise, mbprocess will attempt to process "infile" directly.

For any swath data file "datafile", the program will look for and use a parameter file with the name "datafile.par". If no parameter file exists, mbprocess will infer a reasonable processing path by looking for navigation and mbedit edit save files.The data format can also be specified, though the program can infer the format if the standard MB-System suffix convention is used (*.mbXX where XX is the MB-System format id number).

The processed output swath files produced by mbprocess are named using a convention based on the data format id. MB-System data formats are specified using two-digit or three-digit numbers (see the MBIO manual page). If an input swath data file is named "root.mbXX", where XX is the format id, then the default processed output file will be "rootp.mbXX" (e.g. mydata.mb71 -> mydatap.mb71). The "p" inserted before the ".mbXX" suffix indicates the output file has been created by mbprocess. If the input file does not follow the *.mbXX naming convention, then the output filename will just consist of the input name with "p.mbXX" added as a suffix (e.g. mydata -> mydatap.mb71)

By default, mbprocess will only process a swath data
file if the processed output file is either missing or out
of date relative to the input swath data file, the parameter
file, or any of the ancillary data files referred to in
the parameter file (e.g. navigation files, edit save files,
svp files). If the -P option is specified, mbprocess
will process every file, whether it needs it or not.
ANCILLARY DATA FILES

MB-System also uses a number of ancillary data files, most
of which relate to mbprocess in some way. By default,
these ancillary data files are named by adding a short suffix
to the primary data file name (e.g. ".par", ".svp", ".esf", ".nve")
The common ancillary files are listed below. The example names
given here follow from an input swath data file name of mydata.mb71.

The processing parameter file used by mbprocess has
an ".par" suffix. These files are generated
or modified by mbset, mbedit, mbnavedit,
mbvelocitytool, mbnavadjust, and mbclean.

The most prominent ancillary files are metadata or
"inf" files (created from the output of mbinfo).
Programs such as mbgrid and mbm_plot try to check "inf"
files to see if the corresponding data files include data within
desired areas. The program mbprocess automatically generates
an "inf" file for any processed output swath file.
Also, the program mbdatalist is often used to
create or update "inf" files for large groups of swath data files.

The "fast bath" or "fbt" files are generated by copying the swath bathymetry to a sparse,
quickly read format (format 71). Programs such as mbgrid,
mbswath, and mbcontour will try to read "fbt" files
instead of the full data files whenever only bathymetry
information are required. The program mbprocess
automatically generates an "fbt" file for any processed output swath file.
Also, the program mbdatalist is often used to
create or update "fbt" files for large groups of swath data files.
These files are not generated or used
when the original swath data is already
in a compact bathymetry-only data format.

The "fast nav" or "fnv" files
are just ASCII lists of navigation generated using mblist
with a -OtMXYHSc option. Programs such as mbgrid,
mbswath, and mbcontour will try to read "fnv" files
instead of the full data files whenever only
navigation information are required. These files are not generated or used
when the original data is already

in a single-beam or navigation data format.

The bathymetry edit save file generated by mbedit and
mbclean has an ".esf" suffix.

A water sound velocity profile (SVP) file generated
by mbvelocitytool has an ".svp" suffix unless
the user specifies otherwise.

Water sound velocity profile (SVP) files generated
by mbsvplist also use the ".svp" suffix.
However, multiple SVP files may be
extracted from each input swath file, so the files are
numbered using a "_YYY.svp" suffix, where YYY increments
from 001.

These navigation files can be read independently using format 166.
Adjusted navigation files generated by mbnavadjust
have an ".naY" suffix, where "Y" is a number between 0-9.
The mbnavadjust package may be used multiple times
for a survey; the adjustments are numbered sequentially from
"0":

    mydata.mb71.na0

    mydata.mb71.na1

    mydata.mb71.na2

and so on. These navigation files can be read independently using format 166.

For complete info:
http://www.ldeo.columbia.edu/res/pi/MB-System/html/mbprocess.html