

NAME

mbvoxelclean – 3D density filter used to flag and unflag swath soundings, particularly from lidar sensors.

VERSION

Version 5.0

SYNOPSIS

mbvoxelclean

```
[  
  --verbose  
  --help  
  --input=infile  
  --input=datalist  
  --format=value  
  --voxel-size=xysize[/zsize]  
  --occupy-threshold=value  
  --count-flagged  
  --flag-empty  
  --ignore-empty  
  --unflag-occupied  
  --ignore-occupied  
  --range-minimum=value  
  --range-maximum=value  
  --acrosstrack-minimum=value  
  --acrosstrack-maximum=value  
  --amplitude-minimum=value  
  --amplitude-maximum=value  
]
```

DESCRIPTION

mbvoxelclean identifies and flags artifacts in swath bathymetry data using a 3D density filter. The notion applied is that true targets (e.g. the seafloor) result in dense regions of soundings while sparse soundings in the water column or the subsurface are erroneous and can be flagged as bad. This technique is more appropriate for lidar data than multibeam sonar data. The resulting sounding edit events are output to edit save files which can be applied to the data by the program **mbprocess**. These are the same edit save files created and/or modified by **mbedit**, **mbeditviz**, **mbedit**, and **mbclean**. The input data are one swath file or a datalist referencing multiple swath files. Each file is read and processed separately. The rectangular prism including all of the flagged and unflagged soundings is divided into 3D voxels of the specified size. All of the soundings are read into memory and associated with one of the voxels. Once all of data are read, a density filter is applied such that containing more than a specified threshold of soundings are considered to be occupied by a valid target and voxels containing less than the threshold are considered to be empty. The user may specify one or both of the following actions:

- 1) Previously unflagged soundings in an empty voxel are flagged as bad.
- 2) Previously flagged soundings in a full voxel are unflagged.

This program will also apply specified minimum and maximum range, acrosstrack distance, and beam amplitude filters. If a sounding's flag status is changed, that flagging action is output to the edit save file of the swath file containing that sounding. This program will create edit save files if necessary, or append to those that already exist.

MB-SYSTEM AUTHORSHIP

David W. Caress
 Monterey Bay Aquarium Research Institute
 Dale N. Chayes
 Center for Coastal and Ocean Mapping
 University of New Hampshire
 Christian do Santos Ferreira
 MARUM - Center for Marine Environmental Sciences
 University of Bremen

OPTIONS

--verbose

Normally, mbvoxelclean works "silently" without outputting anything to the stderr stream. If verbosity is specified, then mbvoxelclean works in a "verbose" mode and outputs information about the files processed and the numbers of soundings flagged and unflagged.

--help

This "help" flag cause the program to print out a description of its operation and then exit immediately.

--input=*infile*

--input=*datalist*

Sets the input filename. If *format* > 0 (set with the --format option) then the swath sonar data contained in *infile* is read and processed. If *format* < 0, then *infile* is assumed to be a *datalist*, which is an ascii file containing a list of the input swath sonar data files to be processed and their formats. The program will read and process the data in each one of these files. In the *infile* file, each data file should be followed by a data format identifier, e.g.:

datafile1 11

datafile2 24 This program uses the MBIO library and will read or write any swath sonar format supported by MBIO. A list of the swath sonar data formats currently supported by MBIO and their identifier values is given in the MBIO manual page. Default: *datalist* = "datalist.mb-1".

--format=*value*

Sets the data format id of the input file specified with the --input option. If *format* < 0, then the input file specified with the -I option will actually contain a list of input swath sonar data files. This program uses the **MBIO** library and will read or write any swath sonar format supported by **MBIO**. A list of the swath sonar data formats currently supported by **MBIO** and their identifier values is given in the **MBIO** manual page.

--voxel-size=*xysize[/zsize]*

Sets the size of the voxels used for calculating the density filter, in meters. Each voxel has uniform east-west (x) and north-south (y) width specified by *xysize*. The vertical (z) voxel width *zsize* can be different, but matches *xysize* if not specified.

--occupy-threshold=*value*

Sets the count threshold used to determine if a voxel is occupied or empty. If the number of counted soundings within a voxel is \geq *threshold*, then that voxel is occupied. Otherwise the voxel is empty.

--count-flagged

If this option is specified then the soundings counted to determine if voxels are occupied or empty includes all soundings, including those previously flagged as bad. Otherwise only soundings that are unflagged before **mbvoxelclean** is run are counted.

--flag-empty

If this option is specified then any unflagged soundings in voxels considered empty are flagged as bad. This is the default behavior.

--ignore-empty

If this option is specified then any unflagged soundings in voxels considered empty are left unflagged.

--unflag-occupied

If this option is specified then any flagged soundings in voxels considered occupied are unflagged. This is the default behavior.

--ignore-occupied

If this option is specified then any flagged soundings in voxels considered occupied are left flagged. This is the default behavior.

--range-minimum=*min-range*

If a *min-range* value is specified, then any unflagged soundings that are closer to the sensor than *min-range* are flagged.

--range-maximum=*max-range"];*

If a *max-range* value is specified, then any unflagged soundings that are farther from the sensor than *max-range* are flagged.

--acrosstrack-minimum=*min-acrosstrack*

If a *min-acrosstrack* value is specified, then any unflagged soundings that are more to port of the sensor than *min-acrosstrack* are flagged. Because this filter is applied as the data are read, before density filtering, use of the **--unflag-occupied** option could result in soundings flagged by this minimum acrosstrack filter being unflagged.

--acrosstrack-maximum=*max-acrosstrack"];*

If a *max-acrosstrack* value is specified, then any unflagged soundings that are more to starboard of the sensor than *max-acrosstrack* are flagged. Because this filter is applied as the data are read, before density filtering, use of the **--unflag-occupied** option could result in soundings flagged by this maximum acrosstrack filter being unflagged.

--amplitude-minimum=*min-amplitude*

If a *min-amplitude* value is specified, then any unflagged soundings that have beam amplitudes less than *min-amplitude* are flagged. Because this filter is applied as the data are read, before density filtering, use of the **--unflag-occupied** option could result in soundings flagged by this minimum amplitude filter being unflagged.

--amplitude-maximum=*max-amplitude"];*

If a *max-amplitude* value is specified, then any unflagged soundings that have beam amplitudes greater than *max-amplitude* are flagged. Because this filter is applied as the data are read, before density filtering, use of the **--unflag-occupied** option could result in soundings flagged by this maximum amplitude filter being unflagged.

EXAMPLES

Suppose one wishes to filter the soundings in three lidar files in the format produced by the 3D at Depth WiSSL sensor (WiSSL = Wide Swath Subsea Lidar). A datalist referencing these six files exists as the file *datalist.mb-1* and has the contents:

```
20180925_145000_NorthernSteep.mb233 233
20180925_145100_NorthernSteep.mb233 233
20180925_145200_NorthernSteep.mb233 233
```

Use the following to apply a density filter to each of these files using voxels that are 0.04 m X 0.04 m X 0.04 m and an occupy threshold of 5. This command causes **mbvoxelclean** to flag any soundings found in voxels containing less than 5 valid soundings.

```
mbvoxelclean --input=datalist.mb-1 \
--voxel-size=0.04/0.04 \
--occupy-threshold=5 \
```

```
--flag-empty \
--verbose
```

The verbose output looks like:

```
Program mbvoxelclean
Version $Id: $
MB-system Version 5.6.002
-----
Processing 20180925_145000_NorthernSteep.mb233...
    Actually reading 20180925_145000_NorthernSteep.mb233.fbt...
        Opening edit save file...
5454 survey data records processed
4362859 beams good originally
25495 beams flagged originally
8701246 beams null originally
2615 beams flagged by density filter
0 beams unflagged by density filter
0 beams flagged by minimum range filter
0 beams unflagged by maximum range filter
-----
Processing 20180925_145100_NorthernSteep.mb233...
    Actually reading 20180925_145100_NorthernSteep.mb233.fbt...
        Opening edit save file...
5452 survey data records processed
4361125 beams good originally
24250 beams flagged originally
8699425 beams null originally
3051 beams flagged by density filter
0 beams unflagged by density filter
0 beams flagged by minimum range filter
0 beams unflagged by maximum range filter
-----
Processing 20180925_145200_NorthernSteep.mb233...
    Actually reading 20180925_145200_NorthernSteep.mb233.fbt...
        Opening edit save file...
5453 survey data records processed
4361528 beams good originally
30589 beams flagged originally
8695083 beams null originally
17709 beams flagged by density filter
0 beams unflagged by density filter
0 beams flagged by minimum range filter
0 beams unflagged by maximum range filter
-----
MBvoxelclean Processing Totals:
-----
3 total swath data files processed
16359 total survey data records processed
13085512 total beams good originally
80334 total beams flagged originally
26095754 total beams null originally
0 total beams flagged in old esf file
```

0 total beams unflagged in old esf file
23375 total beams flagged by density filter
0 total beams unflagged by density filter
0 total beams flagged by minimum range filter
0 total beams unflagged by maximum range filter

SEE ALSO

mbsystem(1), mbedit(1), mbinfo(1) mbprocess(1),

BUGS

Hay errores aqui.