# SSSv vegetation height processor

Daniel Rodriguez Pérez & Noela Sánchez-Carnero

2012

**Abstract**

The purpose of this Octave program is to load a Sidescan Sonar (preferably with vertically orientated), to perform a basic bottom detection and, from it, to detect the height and scattered acoustic power from any objects protruding above the detected bottom. All the data can be visualized from the program or exported to ASCII text files and images in PGM format.

The program offers a menu of options that follow the logical order the user has to follow, from acoustic data file loading, algorithm parameter selection, bottom detection, data processing and result export. A brief description of these steps follows.

## [ 1] Load SSSv RAW files

The program will prompt the user for the filename pattern of the RAW files (as saved by the echosounder) to be imported into the program.

`Input filename pattern (e.g.  L0001-D20000101-T*-EA400.raw):`

Notice that the filename pattern must include the ".raw" extension (lowercase, in those operating systems that distinguish between lower and uppercase in filenames), because the program will generate a ".mat" file with the same name the first time any RAW file is read into memory in order to accelerate repeated data loading.

This option allows to process several ".raw" files with the sane file name pattern. To avoid memory exhaustion only file names are stored. The corresponding data files are loaded into memory, one file at a time, during the processing step (below).

## [ 2] Input algorithm parameters

There are several parameters used by the detection algorithm that can be modified by the user. The program will ask for them and offer sensible defaults. Notice that, on entering this menu, all the results of a previous processing (using possibly different parameter values) will be cleared out by default, and then replaced by the new entered ones.

**Coordinates.** The result obtained for every ping is georeferenced using the GPS coordinates stored in the RAW files (or NaN in case no GPS records were available for a ping); these are (lat, lon) WGS84 coordinates. Alternatively, the program can convert these coordinates to UTM (X,Y) coordinates in the same datum.

[ 1] lat, lon

[ 2] UTM X-Y

When option [ 2] is selected, the program will prompt the user for the UTM zone to be used in the transformation.

The zone number will not be included in the output.

**Near field aprox. distance.** Every ping has some initial bins (corresponding to the upper water column) with reverberation noise from the acoustic wave. The bottom detection algorithm is based on the maximum bounce intensity detection, which is usually lower than reverberation noise. For this reason, these bins have to be removed before applying the algoritm.

The "near field" distance ($d_{nf}$) can be computed from the side-scan sonar frequency ($f$) and larger dimension ($D$), and the sound speed in the water ($v_s$) as

$$d_{nf} = \frac{D^2}{\lambda}$$

However, some other reasonable lengths can be assigned for this distance (measured from the transduced to the end of the near field), such as the default suggested by the program, without recoursing to the previous computation. Notice that a very large value for $d_{nf}$ will make the algorithm fail in shallow regions (where it will detect the bottom at twice its depth, due to the second echo); on the other hand, a very small $d_{nf}$ value will detect a false bottom close to the surface, in the reverberation zone. The program suggests a safe value for $d_{nf}$.

**Maximum depth.** In some circumstances, the echosounder records "lost pings" (pings with missing information). Those pings the bottom detection algorithm can be deceived giving unrealistic large depths. As far as the algorithm applies to individual pings, one by one, these pings must be filtered out to avoid errors. Note that a very large value will render the filter useless, but a very small value will lead to unrealistic bottom detection within the water column.

**Bottom detection threshold.** Bottom position is determined finding the bin where the maximum intensity value (outside the near field region) and mooving upwards (along the bins before this one in the ping) untin the difference of the bin intensity relative the the maximum one becomes larger than a given threshold. This threshold is the value the user has to input (in dB). Notice that a very small threshold will locate the bottom very close to the maximum intensity of the ping which, may be due to signal noise or due to slant angle incidence, may be very different from the true bottom position. A very large value can take as sea bottom the upper parts of the vegetation growing on it.

**Water column averaging window radius.** To differentiate bottom vegetation from the water column surrounding it, a noise level has to be defined (that caused by the signal backscattering in the water column). This base level (noise level) is computed, for every ping, as the average intensity within a moving window centered in that ping. around every ping using a moving window average. The height of the window is set to twice the maximum expected vegetation height (see below) and it is localized above that height of the highest vegetation to be found. The width of the window (the number of pings before and after a given one used for that average) is set in by the user in this option. A very large number of pings in a region close to a noisy water column can shift the water column level upwards preventing the detection of true vegetation; it can also shift it downwards in the noisy area, thus providing unrealistic large vegetation heights there.

**Vegetation detection threshold above the water column.** The algorithm will consider as vegetation any signal above the sea bottom with an appreciable intensity some threshold above the base noise level (defined above). This threshold determines how appreciable that signal is required to be with respect to the water column level. A very small value for this threshold will give too large vegetation heigts (then the water column noise is misidentified as vegetation). A very large number will underestimate the presence or height of the vegetation.

**Maximum vegetation admitted height.** To avoid further effects of lost or noisy pings or overestimation of vegetation heights due to the water column noise, any height larger than this maximum admitted height will be discarded, and set to NaN in the output.

## [ 3] Select output directory

The results exported will be written to a directory selected here.

`Output directory (default .):`

By default, the data will be exported in ASCII format, but in the next step the user is also offered the possibility of writing images of the detected vegetation from the echograms to images in PGM format.

`Output PGM images of the selected echogram? (y/N)`

## [ 4] Run processor

This menu option starts the process of data loading and processing according to the file selection done in menu [2].

The results computed in the previous step are the vegetation height for every ping and the depth corrected power scattered by the vegetation identified in that ping. These results are stored in memory to allow their export or visualizsation in the following menus.

## [ 5] Display results

The user can select any of the two following options to visualize the computed data.

`Select information to display:`

`[ 1] Vegetation height`

`[ 2] Scattered Power`

The vegetation height will be shown in a 3-D representation. A height threshold will be asked for, so that only heights above that threshold will be represented.

`Small height crop (default 0.1 m):`

The scattered energy will be shown in a 3-D representation. A power threshold (in dB) will be asked for, so that only vegetation scattering more than that power will be represented.

`Scattered power threshold (default -50 dB):`

## [ 6] Export results

The program allow the export of the data jointly, in one single large file, or separately, in one file per ".raw":

[ 1] One file per RAW

[ 2] Single file with all data

## [ 7] Quit