

The comprehensive Bathymetric Lidar Uncertainty Estimator (cBLUE)



Revision History

Revision	Date	Author(s)	Description
1.0	June 13, 2022	F.C.	First Draft.

Contents

1	cBLUE Overview	3
1.1	Statement of Purpose	3
1.2	License	3
2	Installation	3
2.1	Installing cBLUE	3
2.1.1	Download via git	3
2.1.2	Download as .zip	5
2.2	Install cBLUE Dependencies	5
2.2.1	conda	5

2.2.2	Anaconda Navigator	6
3	User Guide	6
3.1	Data Directories	7
3.1.1	Trajectory	8
3.1.2	Las Files	8
3.1.3	TPU Directory	9
3.2	Environmental Parameters	9
3.2.1	Water Surface	10
3.2.2	Turbidity	10
3.3	VDatum Region	10
3.4	Sensor Model	11
3.4.1	Riegl VQ-880-G	11
3.4.2	Leica Chiroptera 4X	11
3.4.3	Hawkeye 4X	11
3.5	TPU Metric	12
3.6	Output Options	12
3.7	Load Trajectory Files	13
3.8	Process TPU	13
3.8.1	Water Surface Ellipsoid Height	14
3.8.2	Monitoring Progress	14
4	Updates – V3.0	14
5	Open Issues	15
A	ASCII SBET Format	15

1 cBLUE Overview

1.1 Statement of Purpose

The comprehensive Bathymetric Lidar Uncertainty Estimator (cBLUE) is a software tool produced and maintained by the Parrish Lab at Oregon State University, College of Engineering, School of Civil and Construction Engineering, Geomatics Group in collaboration with NOAA's National Geodetic Survey and the University of New Hampshire Center for Coastal and Ocean Mapping.. cBLUE is designed to provide Total Vertical Uncertainty (TVU) and Total Horizontal Uncertainty (THU) for bathymetric lidar surveys.

1.2 License

cBLUE 3.0.0

Copyright (C) 2019

Oregon State University (OSU)

Center for Coastal and Ocean Mapping/Joint Hydrographic Center,

University of New Hampshire (CCOM/JHC, UNH)

NOAA Remote Sensing Division (NOAA RSD)

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

2 Installation

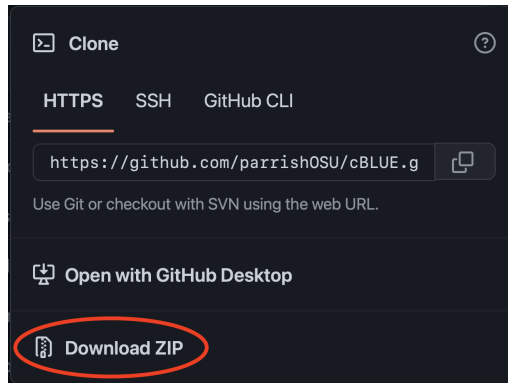
2.1 Installing cBLUE

This section provides step-by-step instructions to download cBLUE and install the various Python dependencies need to calculate TPU from .las files. While git is the suggested method for downloading cBLUE, users unfamiliar with version control software are also free to download cBLUE as a .zip file. Instructions for both are provided below:

2.1.1 Download via git

Steps:

1. In a web browser, navigate to <https://github.com/parrishOSU/cBLUE.github.io>
2. In the top right of the repository, locate the green code menu and copy the url under HTTPS



3. Clone the repository using either your computer's terminal/command prompt or the Github Desktop Application:

Terminal

- (a) Navigate to the folder where you'd like to install cBLUE

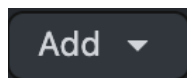
```
$ cd location/of/cBLUE
```

- (b) Clone the repository using the url from github

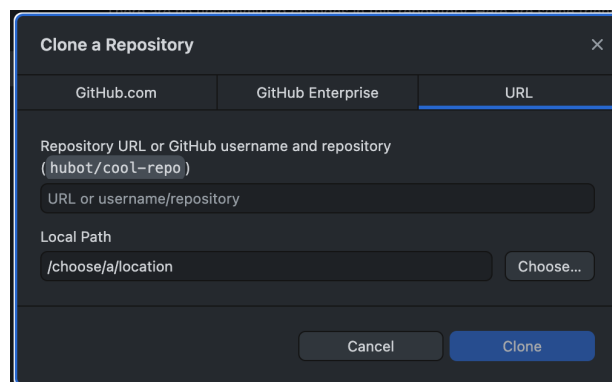
```
$ git clone https://github.com/parrish0SU/cBLUE.github.io.git
```

GitHub Desktop

- (a) Open GitHub Desktop and locate the "Add" button



- (b) Under "Add" choose "Clone Repository" and select "URL"



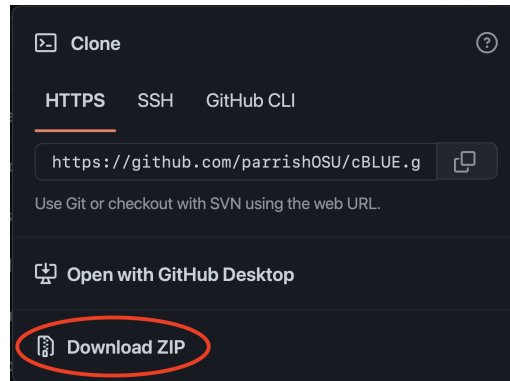
- (c) Enter the path to the location you'd like to install cBLUE and paste the url found in Step 2 in the appropriate boxes
- (d) Press the "Clone" button to begin download



4. Once the download is complete, check that the folder `cBLUE.github.io` is in the desired location.

2.1.2 Download as .zip

1. In a web browser, navigate to <https://github.com/parrishOSU/cBLUE.github.io>
2. In the top right of the repository, locate the green code menu and click "Download Zip"



3. Once the download is complete, move the `.zip` to your desired location and extract to `cBLUE.github.io`.

2.2 Install cBLUE Dependencies

cBLUE is designed to be a cross-platform software, therefore this installation guide should be valid for all Windows, MacOS, and Linux users. However, cBLUE does require Python 3 to be installed on the user's machine, as well as all the Python library dependencies needed to run cBLUE. Additionally, while it is possible to install cBLUE using any Python package manager, we strongly suggest using `conda`, which can be downloaded for free [here](#). In this installation guide we provide instructions to install and run cBLUE using both the Anaconda Navigator GUI as well as the `conda` CLI.

2.2.1 conda

1. In your computer's terminal/command prompt, navigate to `cBLUE.github.io`

```
$ cd location/of/cBLUE/cBLUE.github.io
```
2. Locate the file `cblue.yml`
3. Create a new virtual environment using the command

```
$ conda env create -f cblue.yml
```
4. cBLUE dependencies should begin downloading shortly.
5. Once all dependencies have successfully downloaded, activate the new environment using the command

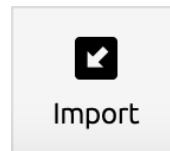
```
$ conda activate cblue
```

6. To check that the dependencies are installed properly, use the command

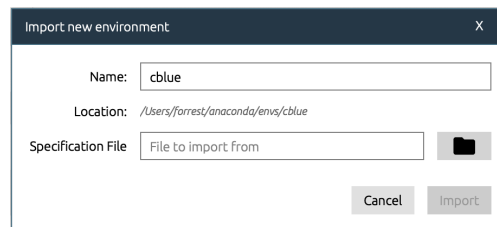
```
$ python CBlueApp.py
```

2.2.2 Anaconda Navigator

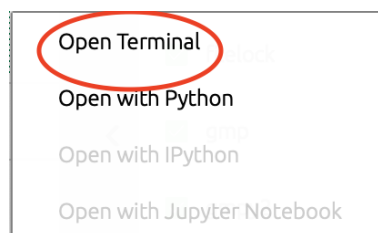
1. Open "Anaconda Navigator" and navigate to the "Environments" menu on the left hand side.
2. Locate the "Import" button on the bottom left



3. In the "Import New Environment" dialog box, enter "cblue" for "Name" and navigate to the file `cblue.yml` for the "Specification File"



4. Click "Import"
5. Once the downloads are complete, locate "cblue" in the environments list and click on it to activate. A green triangle should appear next to the environment name once it is activated.
6. To check that the dependencies are installed correctly, click the green triangle next "cblue" and select "Open Terminal"



7. Navigate to the location of `cBLUE.github.io`

```
$ cd location/of/cBLUE/cBLUE.github.io
```

8. Open cBLUE using the command

```
$ python CBlueApp.py
```

3 User Guide

This section provides step-by-step instructions to perform a Total Propagated Uncertainty (TPU) estimation from bathymetric lidar data. In order to perform this

estimation, it is assumed that the user has properly install cBLUE and its dependencies. If cBLUE or its dependencies are not installed, please refer to Section 2 of this manual.

In order to perform a TPU estimation using cBLUE, it is necessary to have two sets of files: at least one .las file containing information on the lidar return signal and at least one corresponding SBET (.txt) file containing information on the sensor trajectory at the time of each lidar pulse. These files are (usually) obtained directly from the sensor platform. Further information on these file types is outside the scope of this manual.

To open cBLUE, in the terminal of your choosing, execute the following command from inside the cBLUE.github.io folder.

```
$ python CBlueApp.py
```

The following message should appear in your terminal:

```
=====
NOAA REMOTE SENSING DIVISION'S
comprehensive Bathymetric Lidar Uncertainty Estimator

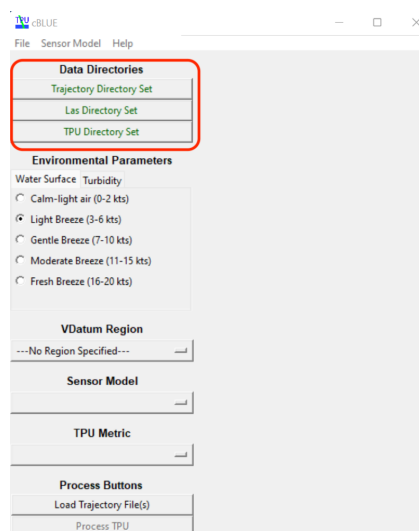
          @@@@,      @@@,      ,@@@ ,@@@@@@@@@@@
          @@@###%@@@@, @@@,      ,@@@ 5@@@@ 5@@@@#####
          */ ,      *@@@ (@@@) 5@@@@ 5@@@@ @@@, @@@/
          @@@@@@@@. @@@@ #@@@@ @@@, @@@, @@@, .@@@
          /@@@ .@@@ @@@@@@@@@@@@. .@@@ /@@@ 5@@@@ %@@@@@@@@@@@.
          @@@ *@@@@@@@@@@@@@, 5@@@@ @@@# @@@# @@@, @@@#*****
          *@@@ @@@@ *@@@ @@@ @@@, ,@@@ @@@
          ,@@@ @@@ @@@. 5@@@@ /@@@ @@@. @@@/ %@@@
          @@@, %@@@ ( ,@@@@@@@@@@@@@ @@@@@@@@@@@@@@ %@@@@@@@@@@@/ @@@@@@@@@@@@@@
          /@@@@# @@@@@@@@@@@@ @@@@@@@@@@@@@@% @@@@@@# @@@@@@@@@@@@#

          (c) 2019 CCOM/JHC, OSU, NOAA RSD
          License: GNU Lesser General Public License v2.1
=====
```

(Note: the cblue conda environmental must be installed and activated in order to run cBLUE. If cBLUE fails to open, it is likely that your dependencies are not installed or the environment is deactivated. Refer to Section 2 of this manual for information on resolving these issues.)

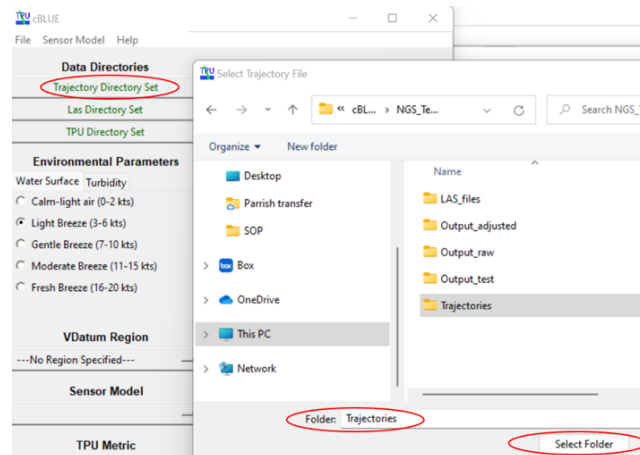
3.1 Data Directories

The first step in performing a TPU estimation using cBLUE is to point the software to the .las and SBET (.txt) files on your system. This is done using the Data Directories selection menus.



3.1.1 Trajectory

The "Trajectory Directory Set" menu allows the user to select the location of the SBET files containing the sensor trajectory parameters at the time of each laser pulse.



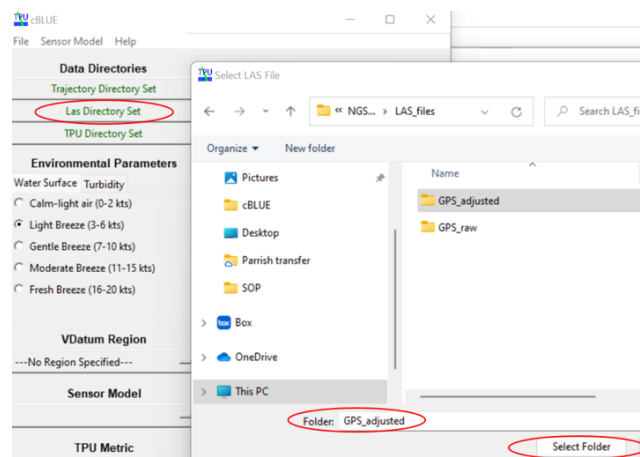
These files are usually generated by the lidar system and therefore their generation and file standards are outside the scope of this manual, however a brief overview of ASCII SBET formatting is available in [Appendix A](#). If you run into issue regarding the SBET files, please consult the lidar system's user manual or contact the manufacturer before raising an issue with the cBLUE maintenance team, as these issues are often the result of erroneous data and are therefore difficult to reproduce/troubleshoot. An example of SBET files is shown in the figure below.

_files > NGS_Test_Data_raw_and_adjusted_times > Trajectories

Name	Date modified	Type	Size
20160515.1_880_p_sbet_lidar_tpu.log	9/21/2016 8:47 AM	Text Document	1 KB
20160515.1_880_p_sbet_lidar_tpu.txt	9/21/2016 8:47 AM	Text Document	946,699 KB
20160515.2_880_p_sbet_lidar_tpu.log	9/21/2016 9:21 AM	Text Document	1 KB
20160515.2_880_p_sbet_lidar_tpu.txt	9/21/2016 9:21 AM	Text Document	728,789 KB
20160517_880_p_sbet_lidar_tpu.log	9/21/2016 9:24 AM	Text Document	1 KB
20160517_880_p_sbet_lidar_tpu.txt	9/21/2016 9:24 AM	Text Document	602,521 KB

3.1.2 Las Files

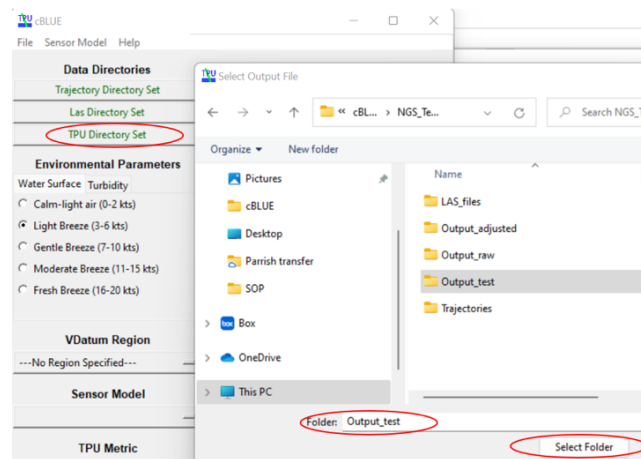
The "Las Directory Set" menu allows the user to select the location of the .las files containing the point cloud coordinates from each laser pulse.



Currently, given the popularity of .las, especially in the bathymetric lidar community, these are the only files types supported. There are no plans to add further file types*. Additionally, note that the GPS times in the LAS file should be Adjusted Standard GPS Time. This is important because the GPS times are how cBLUE matches up points in the LAS file with corresponding records from the trajectory. For more information on LAS file specifications, please refer to the ASPRS LASER (LAS) FILE FORMAT EXCHANGE ACTIVITIES, found [here](#).

3.1.3 TPU Directory

The "TPU Directory Set" menu allows the user to select the location of the output files.

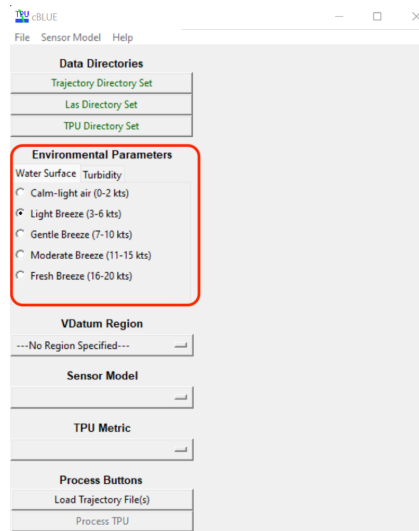


The files output by cBLUE will match the .las files in the "Las Directory Set" location, with _TPU appended to the file name. These output files will contain the TPU as ExtraBytes fields. More information on LAS ExtraBytes can be found [here](#).

3.2 Environmental Parameters

cBLUE operates by using a combined subaerial and subaqueous model. The subaqueous model is designed to simulate the range of possible values for each laser shot, given a set of environmental parameters. Therefore, it is important that the user tune these parameters to reflect the best approximation of the water conditions at the time the bathymetric survey was conducted.

*As of cBLUE V3.0, compressed LAS files (.laz) are also accepted by cBLUE.



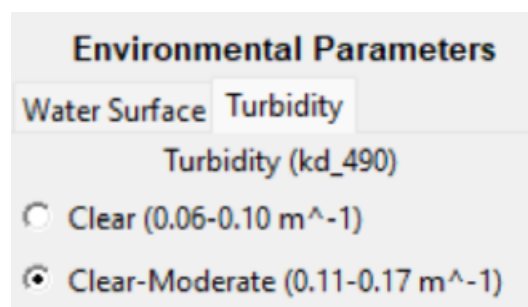
3.2.1 Water Surface

The "Water Surface" menu allows the user to select the range of water surface parameters that best approximate the water surface conditions at the time of the bathymetric survey.

(Note: cBLUE V2.X included an option to model the water surface from using the pointcloud data. This option has been removed as of cBLUE V3.0. For more information on why this option was removed, please refer to Section 4 of this manual).

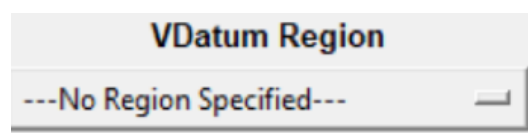
3.2.2 Turbidity

The "Turbidity" menu allows the user to select the range of turbidity parameters that best approximate the water clarity conditions at the time of the bathymetric survey.



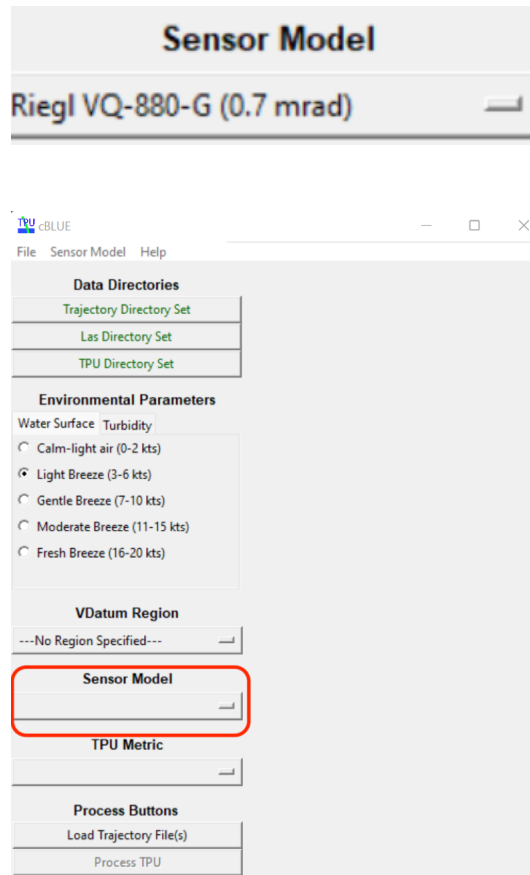
3.3 VDatum Region

Vertical datum uncertainty is a component uncertainty often included in bathymetric TPU modeling. The "VDatum Region" menu allows the user to select a VDatum region to use in the modeling process. The purpose of this setting is to enable users to include VDatum datum transformation uncertainty (https://vdatum.noaa.gov/docs/est_uncertainties.html) as a component uncertainty in the TPU computation. However, this parameter is not required to perform the uncertainty estimation and may be left as "-No Region Specified-" by the user.



3.4 Sensor Model

As of cBLUE V3.0, there are 3 lidar sensor platforms available in cBLUE. These include the Riegl VQ-880-G, Leica Chiroptera 4X, and Leica Hawkeye 4X. The Parrish Group at OSU is also actively working to develop subaqueous models for more sensors to be included in future versions of cBLUE. To request a sensor model, please post an issue on the cBLUE GitHub page [here](#).



3.4.1 Riegl VQ-880-G

The Riegl VQ-880-G sensor includes a programmable beam with a selectable beam divergence. Subaqueous models are available for each beam divergence in cBLUE. The following values are the beam divergences modeled by cBLUE in milliradians.

- 0.7 mrad
- 1.0 mrad
- 1.5 mrad
- 2.0 mrad

3.4.2 Leica Chiroptera 4X

As of V3.0, a subaqueous model is available to estimate TPU from LAS data collected by Leica Chiroptera 4x sensors.

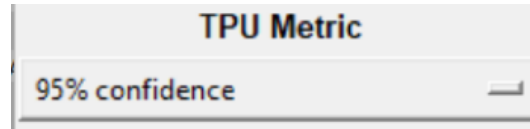
3.4.3 Hawkeye 4X

As of V3.0, a subaqueous model is available to estimate TPU from LAS data collected by Leica Hawkeye 4x sensors. (Note: the "shallow" channel of the Hawkeye 4x sensor is identical to the Chiroptera 4x. Therefore, similarity between estimates calculated

using either of these two models on the same dataset may be nearly identical in shallow waters.)

3.5 TPU Metric

As of cBLUE V3.0, users may select to output TPU values at either 95% Confidence or $1-\sigma$.



Previous versions of cBLUE defaulted to $1-\sigma$. Conversion between $1-\sigma$ and 95% Confidence for Total Vertical Uncertainty (TVU) and Total Horizontal Uncertainty (THU) can also be performed after estimation using the relationships below:

TVU:

$$95\% \text{ Confidence} = 1.96 \times (1-\sigma)$$

THU:

$$95\% \text{ Confidence} = 1.7308 \times (1-\sigma)$$

3.6 Output Options

cBLUE V3.0 allows the user to choose between exporting the estimated TVU/THU solely as ExtraBytes in the new LAS file, or to also include a comma separated values (CSV) table with the TVU/THU. This option may be suitable for users with small datasets who wish to work with the resulting uncertainties in software that does not handle LAS files, or those who are unfamiliar with the LAS format. However, the process of building and saving a CSV file with information for each point can be slow and is therefore not recommended for users with large files, or a large number of files.

The information included in the optional output CSV is detailed in Table [2](#).

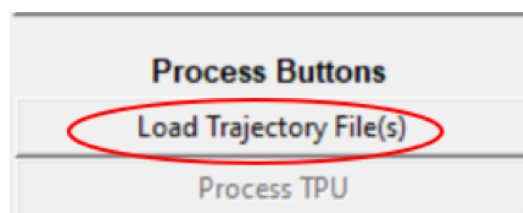
Field	Description
GPS Time	The GPS Time of each point in the format of the corresponding input LAS/LAZ file
X	The X coordinate of each point in the format of the corresponding input LAS/LAZ file
Y	The Y coordinate of each point in the format of the corresponding input LAS/LAZ file
Z	The Z coordinate of each point in the format of the corresponding input LAS/LAZ file
THU	The Total Horizontal Uncertainty of each point as estimated by cBLUE
TVU	The Total Vertical Uncertainty of each point as estimated by cBLUE
Classification	The classification of each point as specified in the input LAS/LAS file

Table 2: Output CSV fields

While other fields can be added to the CSV by modifying cBLUE's code, it is strongly recommended that users wishing to access additional fields simply export as ExtraBytes and use the las2txt tool from LASTools, which can be downloaded [here](#).

3.7 Load Trajectory Files

After indicating the appropriate data directories and selecting the appropriate parameters for the lidar survey, the user should press the "Load Trajectory Files" button to process the SBET data.



The trajectories may take a few moments to load, during which time you can track the progress in your terminal/command window.

```

water_surface_ellipsoid_height : -38.09
}
Loading trajectory files...
N/A% (0 of 3) | Elapsed Time: 0:00:00

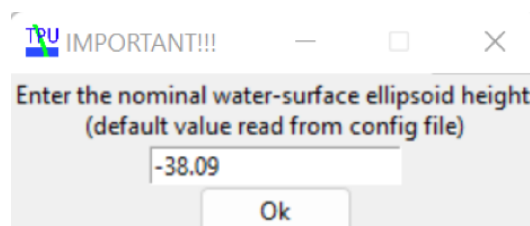
```

3.8 Process TPU

Once the trajectory files are successfully loaded, the "Process TPU" button will become clickable. Clicking this button will initiate the TPU estimation model.

3.8.1 Water Surface Ellipsoid Height

The "Process TPU" will bring up a window asking the user to enter the ellipsoid height of the water surface. The purpose of this parameter is to enable a slant range correction. This value should be close, but it does not need to be perfect, as changes in this parameter generally only impact the computed TVU on the order of millimeters. The preferred method of obtaining this value is to use an average ellipsoid height of water surface (Class 41) points in the LAS file. Alternately, method of obtaining an approximate value for this parameter is to use [VDatum online](#) to compute the separation between the NAD83 ellipsoid and one of the following tidal datums, depending on which was closest to the water level at the time the bathymetric lidar dataset was acquired: mean lower low water (MLLW), mean low water (MLW), mean tide level (MTL), mean high water (MHW), or mean higher high water (MHHW). If the stage of tide at the time the bathymetric lidar was acquired is unknown, it is usually reasonably safe to select local mean sea level (LMSL) as the "to" datum; typically, this will only impact the computed TPU values on the order of millimeters.



3.8.2 Monitoring Progress

After clicking "Ok" on the Water Surface Ellipsoid Height window, cBLUE will begin to estimate TPU. To monitor the progress of cBLUE, please refer to your terminal/command window. When the model has completed successfully, you will see the following:

```
C:\WINDOWS\system32\cmd.exe - python.exe cBlueApp.py
}
Loading trajectory files...
100% (3 of 3) |#####| Elapsed Time: 0:01:22 Time: 0:01:22
Calculating TPU (single-processing)...
N/A% (0 of 1) | Elapsed Time: 0:00:00 ETA: ---
---extra_byte_df
-----
total_thu total_tvu
0.0 0.335649 0.342423
1.0 0.337368 0.348967
2.0 0.338642 0.353749
3.0 0.335358 0.341267
4.0 0.338545 0.353353
...
17329189.0 0.347385 0.355460
17329190.0 0.345779 0.349225
17329191.0 0.349832 0.364672
17329192.0 0.348531 0.359749
17329193.0 0.346334 0.351290
[17329194 rows x 2 columns]
100% (1 of 1) |#####| Elapsed Time: 0:01:02 Time: 0:01:02
DONE!! (close cBLUE before running again)
```

4 Updates – V3.0

The following list details the major updates associated with cBLUE V3.0. This is not a comprehensive list of updates to the code, which can be found under the github commit history associated with cBLUE.

- Added Sensor Models:
 - Riegl VQ 880-G :
 - * 0.7 mrad
 - * 1.0 mrad
 - * 1.5 mrad
 - * 2.0 mrad
 - Leica Chiroptera 4X
 - Hawkeye 4X
- Removed "Direct from Point Cloud" Option
- Added TPU Metric User Selection
- Updated Dependency (laspy 2.0 → laspy 3.0)
- Updated File Types (.laz files accepted)
- Added a "minimum" TPU value of 3.0 cm. Values below this threshold were determined to be erroneously small.
- Optional CSV output now available.

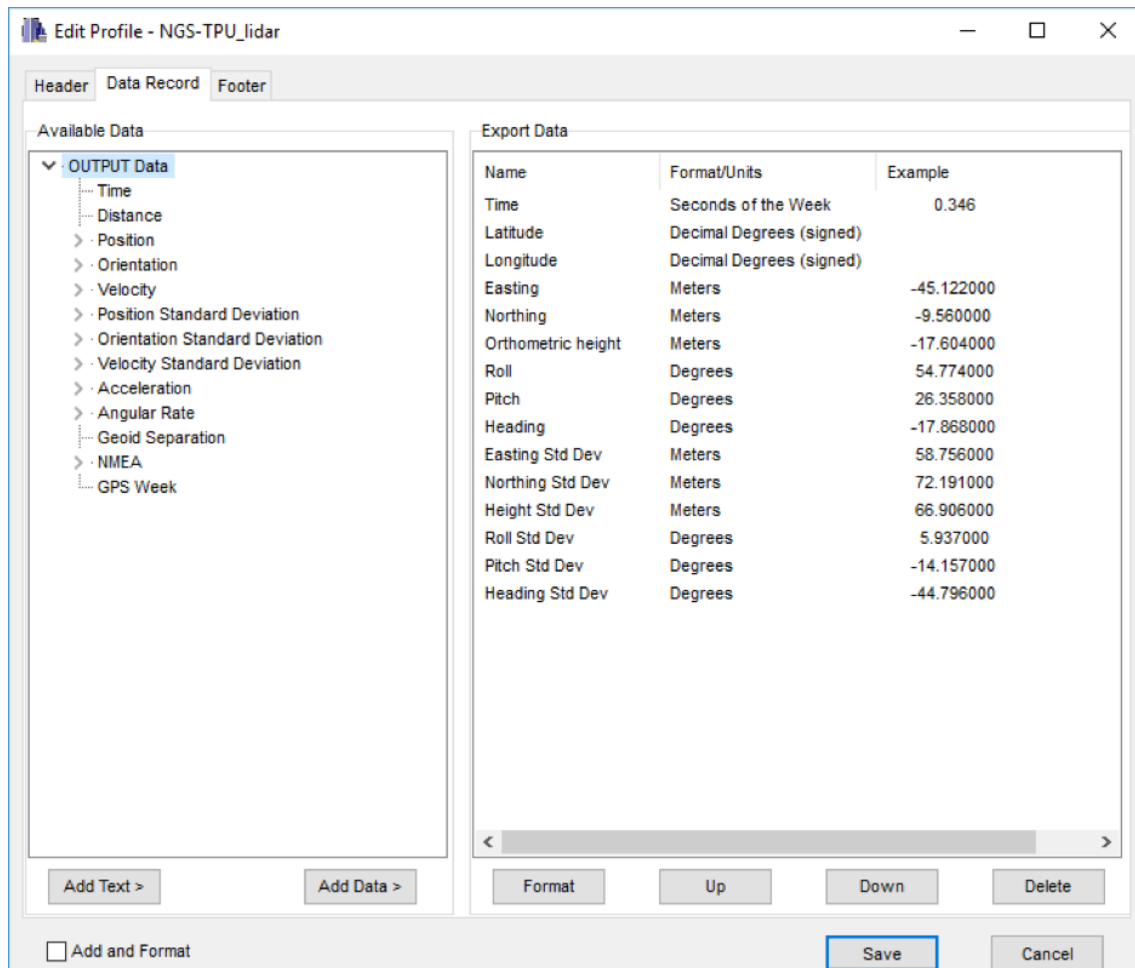
5 Open Issues

This section details the open "issues" for cBLUE. This is not a list of known bugs, but a list of possible additions and modifications that might be made to cBLUE. These additions may be tackled by members of the Parrish Group Cblue maintenance team at Oregon State University, or may be contributed by members of the cBLUE community pending review from the Parrish Group.

- Additional Sensors
 - Any additional sensor models generated by the cBLUE community are welcome and encouraged. If you've generated a subaqueous look up table using the Monte Carlo process detailed [link needed](#), we would be happy to help you implement it in cBLUE.
-

A ASCII SBET Format

The format of the trajectory files needed for cBLUE is a custom ASCII SBET with the following format:



References

- [1] Firat Eren, Jaehoon Jung, Christopher E. Parrish, Nicholas Sarkozi-Forfinski, and Brian R. Calder. Total Vertical Uncertainty (TVU) Modeling for Topo-Bathymetric LIDAR Systems. *Photogrammetric Engineering & Remote Sensing*, 85(8):585–596, August 2019.