
Benthic Photo Survey Documentation

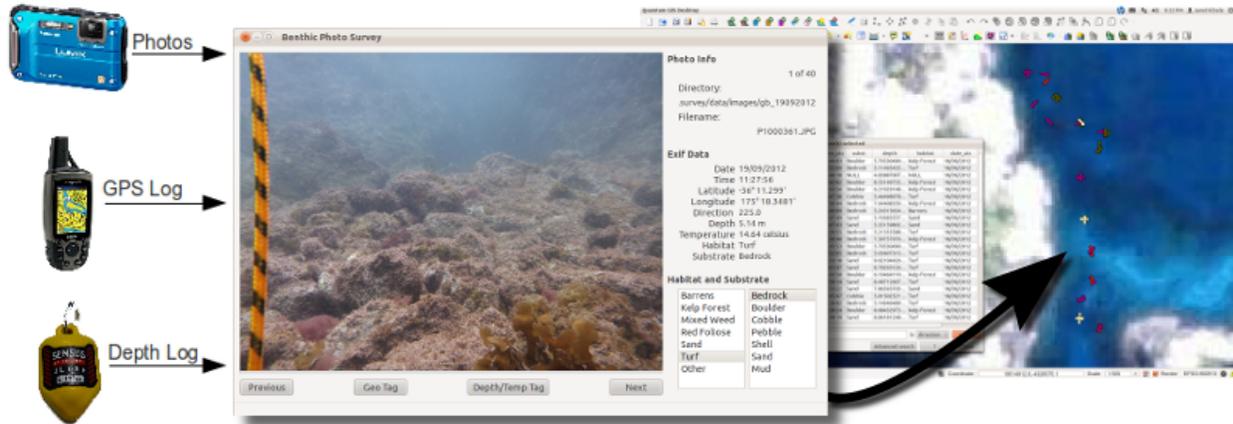
Release 0.3.18

Jared Kibele

December 16, 2014

1	Contents	3
1.1	Introduction	3
1.2	Installation	4
1.3	Tutorial	7
1.4	Code Documentation	9
1.5	Indices and tables	9
	Bibliography	11

Benthic Photo Survey (BPS) is a software tool (written in Python) that is intended to simplify the task of collecting reference data (sometimes called groundtruth data) for remote sensing of the marine environment. Using a digital camera in a water proof housing, a consumer grade handheld GPS, and (optionally) a depth logger a user can collect data in the field.



The photos and logs (GPS and Depth) are then loaded into BPS. BPS can then tag the photos with position, depth, and temperature (if using the Sensus Ultra depth/temperature logger). The users can view each photo in BPS and tag each photo with a substrate and habitat type. All data (position, depth, temperature, habitat, and substrate) are stored in the exif portion of the jpeg photos.

Once the photos are all tagged, BPS can export a GIS Shapefile with habitat, substrate, depth, and temperature attributed points for each photo's location. This shapefile can then be used to create a training set for supervised classification or as reference data for accuracy assessment.

1.1 Introduction

Quantitative methods for the derivation of habitat maps from satellite and aerial imagery require reference data (sometimes called ground truth data) to create training sets for supervised classification and for the accuracy assessment of the resulting maps. These data generally need to represent the real world habitat type at many points and, in the case of benthic habitat mapping, the depth at each of those points can also be quite useful. Collecting and processing these data can be a challenging and time consuming task and may be part of the reason that too few benthic habitat mapping studies include adequate accuracy assessment. Benthic Photo Survey (BPS) was created to make this process less painful and, hopefully, make it easier to collect training data and produce an accuracy assessment of the resulting map.

1.1.1 What You Need to use BPS

Benthic Photo Survey is designed to work with inexpensive and relatively easy to acquire equipment, partly because that's all that's really required and partly because that's all I can afford. Here's what you'll need:

1. A submersible digital camera that produces jpeg images. I use a [Panasonic Lumix DMC-TS4](#) in an [Ikelite housing](#). This camera has a built in compass that records the direction that the camera is facing when the photo is taken and BPS passes the direction into the shapefile that is created. However, this is not a requirement and any camera that produces jpeg images should be compatible with BPS.
2. A GPS that can be towed by a snorkeler or diver. If you're working with fairly high resolution imagery, you'll want a GPS receiver with WAAS capability. I am using a [Garmin GPS 60 CSX](#). I tow mine in a small waterproof plastic box on a float. See [GPS Photo Transects for Benthic Cover Manual](#) by Roelfsema and Phinn [RP2010] for details on how to build a float. BPS can import NMEA text files and GPX files.
3. A depth logger is optional but if you're going to spend the time and money to go in the field, it seems worthwhile. BPS is set up to import log files from the [Sensus Ultra](#) depth and temperature logger. It should be fairly easy to adapt it to deal with a different logger but that would require a bit of coding.
4. A computer to run BPS. BPS was developed on Ubuntu but should work on Windows and Mac as well.

1.1.2 What BPS Actually Does

There are 4 steps to what BPS does.

1. It imports your GPS and/or depth and temperature logs into a database.
2. It writes data from those logs to the exif portion of each of your jpeg photos based on the time stamp on each photo and the time stamps in the respective logs.

3. You use BPS to tag each photo with a habitat and/or depth and temperature.
4. BPS can export a point shapefile with a point at each location where a photo was taken. Each point is attributed with the habitat, substrate, depth, temperature, direction, and file path to the photo.

Citations

1.2 Installation

Benthic Photo Survey (BPS) runs on Linux, Mac, and Windows operating systems. There are a few different ways to install BPS. You can [download the code](#) from the [bitbucket](#), you can install with the python [pip](#) installer, or you can [git](#) to clone the code from the [bitbucket](#) repository. More details are provided below along with information about software requirements and details specific to installing on different platforms.

1.2.1 Requirements

The following items need to be installed in order for BPS to work. Make sure you read at least to the end of this section before you start installing this stuff. There may be an easier way than installing all these things individually. Then check the section below for your operating system. There may be separate quirks to the installation process for Windows, Linux, and Mac. I will try to give step-by-step instructions for each operating system below.

1. [Python 2.7](#) BPS may work on older versions of Python but I wouldn't bet on it
2. [GDAL 1.8](#) or greater
3. [Matplotlib 1.1.1](#) or greater, can be installed as part of [SciPy](#) or with [PythonXY](#) (see end of this section).
4. [PyQt4](#)
5. [pynmea](#) can be installed with [pip](#) by typing `pip install pynmea` at the command prompt (or `sudo pip install pynmea` if you're on Linux or Mac and need root privileges to install).
6. [pyexiv2](#) has actually been deprecated now. I will eventually rewrite the code to use [GExiv2](#) but, for now, [pyexiv2](#) is required.

That's kind of a lot of stuff to have to install. Fortunately, items 1 - 4 can be taken care of with the [PythonXY](#) installer. [PythonXY](#) is a set of free scientific software tools all bundled together for easy installation. You can download the [PythonXY](#) installer, run it, make sure that items 1 - 4 are selected, and it will take care of all those steps for you. If you decide later that you want some of the other items to be installed as well, you just run the installer again and check more boxes. It makes things much easier, especially on Windows.

1.2.2 Ways to Get BPS

There are two different ways to get release versions of BPS and one way to get the latest code from the repository. These are outlines of the steps needed that should be applicable to all operating systems. For a more step-by-step approach, check the OS specific sections further down the page.

Installing the Latest Release with pip

Go to the command line and type `pip install BenthicPhotoSurvey`. Depending on your OS, you may need to type `sudo pip install BenthicPhotoSurvey` instead and enter your root password.

In order to run BPS, you have to figure out where `pip` installed the code and that can vary according to OS and configuration. If you have version of `pip` $\geq 1.2.1.post1$, you can figure it out by typing `pip show BenthicPhotoSurvey` on your command line. The response should contain the location of the `benthic_photo_survey` directory. To run BPS, navigate to that directory and start BPS by typing `python benthic_photo_survey.py` (on Windows, you'll leave off the `python` and just type `benthic_photo_survey.py`).

On my computer it looks something like this:

```
blah@blahbidyblah:~$ pip show BenthicPhotoSurvey
---
Name: BenthicPhotoSurvey
Version: 0.1
Location: /usr/local/lib/python2.7/dist-packages
Requires:
```

So, to launch, I type:

```
blah@blahbidyblah:~$ cd /usr/local/lib/python2.7/dist-packages/benthic_photo_survey
blah@blahbidyblah:/usr/local/lib/python2.7/dist-packages/benthic_photo_survey$ python benthic_photo_
```

...and the program should run and you can move on to the *Tutorial*. I will, hopefully, get around to packaging BPS a little better so it's easier to launch after installing but this will have to do for now.

Downloading and Installing a Release from bitbucket

Use your web browser to [download the code](#) from bitbucket. Unzip that file somewhere. Navigate into the `benthic_photo_survey` directory that contains `benthic_photo_survey.py` and type `python benthic_photo_survey.py` (on Windows, you'll leave off the `python` and just type `benthic_photo_survey.py`).

Getting the Latest Code from bitbucket

First, make sure you have `git` installed. Then open the command line where you'd like to put the code and type `git clone https://jkibeles@bitbucket.org/jkibeles/benthic_photo_survey.git`. That will download the latest code **which may not be stable**. To run it just follow the instructions from the previous subsection.

1.2.3 Windows

The following steps will demonstrate how to install the software required to run BPS, use the `git` version control system to download BPS, and then run BPS on Windows.

- **Satisfy Install Requirements 1 - 4 with PythonXY**

1. Download the latest [PythonXY](#) installer from the [PythonXY Windows Download Page](#). Just click on one of the `.exe` files listed under 'Current Release'. The multiple links all lead to the same file, just on different servers.
2. Run the installer, and make sure that Python2.7.x (currently 2.7.3 but that may change, you just don't want 3.x), GDAL, PyQt4.x, `pip`, and `Matplotlib` are selected. These items are listed under the expandable python menu. You can check whatever additional items you'd like installed as well.
3. Click through the dialogs to finish the installation process.

- **Install `pynmea` with `pip`**

1. Go to the [command prompt](#) and type `pip install pynmea`. If you installed pip in the previous step, that command should work and you should be able to move on to the ‘installing pyxiv2’ step below. If there was a problem with pip, you can use the sub-steps below to install it.
2. If you get a message that says something about the ‘command pip not found’, then try typing `easy_install pip` at the [command prompt](#). If that doesn’t work, you’ll need to install [python-distribute](#) and [pip](#) as described in the next sub-step. If it does work (and it installs pip), just go back to the previous sub-step and it should work.
3. Download the correct installer for your system architecture (32 or 64 bit) from the list of [python-distribute installers](#). We installed python2.7.x so you want the one that matches your system architecture and ends with `py2.7.exe`. After running that installer, you should be able to type `easy_install pip` followed by `pip install pynmea`. The good news is that python-distribute and pip can be used to install all sorts of useful python stuff and, now that you have them installed, that’ll be easy.

- **Install pyxiv2**

1. Go to the [pyxiv2 download page](#), scroll down to the Windows section appropriate to your system (32 or 64 bit) and download the latest installer for Python 2.7. Note that the latest versions are at the bottom of the list rather than the top.
2. Run the installer

- **Install git if you don’t already have it**

1. Go to the [git download page](#), download the installer for your system, and run it.
2. The default list of components on the ‘Select Components’ screen is fine.
3. The choice of ‘Start Menu Folder’ doesn’t really matter. I used the default.
4. On the next screen, choose “Run Git from the Windows command prompt”.
5. On the ‘Configure Line Ending’ screen, select the default of ‘Checkout Windows-style, commit, unix-style’.

- **Use git to get BPS**

1. Open a [command prompt](#) and navigate to a directory where you would like to install BPS
2. Type `git clone https://jkibele@bitbucket.org/jkibele/benthic_photo_survey.git`. This will clone the contents of BPS code repository to your computer.
3. If, after changes have been made to BPS, you want the latest version you can return to a command prompt in this directory and type `git pull`.

That should do it. Make sure you make a note of where you installed BPS. Then take a look at the [Tutorial](#). I intend to add a section there specifically about using the test data for a dry run. For now just check out the “Using BPS” section and look for the `test_data` directory at the same level as the `data` directory. There are some images, a gps log, and a sensus depth log in there that you can use to get the hang of it.

1.2.4 Mac

Need to write up these steps too.

1.2.5 Ubuntu

Need to write up these steps too.

1.2.6 Testing

I will describe how to run the automated tests and how to use the test data that's installed with BPS to make sure everything is working.

1.3 Tutorial

This document may need to be fleshed out a bit better but this should serve as an outline.

1.3.1 Initial Setup

1. You should make sure you have BPS installed, running, and tested as described in *Installation*.
2. As mentioned in the *Introduction*, you will need a float on which you can tow your GPS.
3. Set your GPS tracking on and make sure the interval is short. I set mine to log position once every 4 seconds.
4. Make sure your GPS datum is set to WGS84. That tends to be the default setting but you should make sure. You can use a different datum but you should make sure that the configuration file is set to match. See [Configuring BPS](#).
4. Set the logging interval on your depth sensor. It needs to be particularly short if you are free diving. I set mine to log every 2 seconds.
5. Attach depth sensor to something. I generally attach it to the camera housing. I may need to compensate for the fact that the camera is usually a meter or so above the bottom when I take the photos but that seems preferable to getting the logger full of sediment.

1.3.2 Clock Synchronization

Since BPS will use time codes to find positions and/or depths for each photo it is essential that relevant clocks are all in sync. Before getting in the water, make sure your camera's clock is set to the correct local time. It's easiest to use the time display on your GPS for this purpose. The Panasonic Lumix DMC-TS4 that I use has a built in GPS that is supposed to be able to set the camera's clock to GPS time. If you use this feature make sure to double check it. There camera clock should match the GPS clock within about a second. The depth logger actually sets the time codes when the log is downloaded from the logger. Therefore, the computer that you use to download the depth log needs to have its clock synced to GPS time when the file is downloaded. If your computer's clock is set by the internet, this is usually good enough (in my experience) but you should double check.

1.3.3 While in the Water

Regardless of whether you are on SCUBA or free diving, you should pause for a few seconds before taking a photo and pull down lightly on the rope you have attached to your GPS float. To get the best possible accuracy, you need the float to be directly over you. Pausing at the bottom will also minimize the effect of any clock synchronization problems for both depth and position. Also, don't drown. That's bad for data collection.

1.3.4 Back on the Surface

If you're using a Garmin GPS60 like I am, **do not** save the active track. Saving the track and then downloading the saved track gets rid of the timestamp on each position rendering it useless as far as BPS is concerned. Instead, you want to download the active track. I use [GPSbabel](#) to download the track and convert it to a GPX file. Many newer

GPS models output GPX files without the need for conversion. You just need to make sure you're getting a GPX file that has a timestamp for each point within the track.

You should make backups of your photos before letting BPS operate on them. I've never had a problem but, since BPS is writing to part of the jpeg file, there is a possibility that it could corrupt the photo so you should have a backup.

For convenience, I keep my BPS GPS logs in `benthic_photo_survey/data/GPS` and my depth logs in `benthic_photo_survey/data/sensus`. For photos, I create a directory for each set of photos within `benthic_photo_survey/data/images`. BPS is set up to operate on a directory of photos as a unit. For instance, when creating a shapefile all photos within the currently loaded directory will become a point in the shapefile.

1.3.5 Configuring BPS

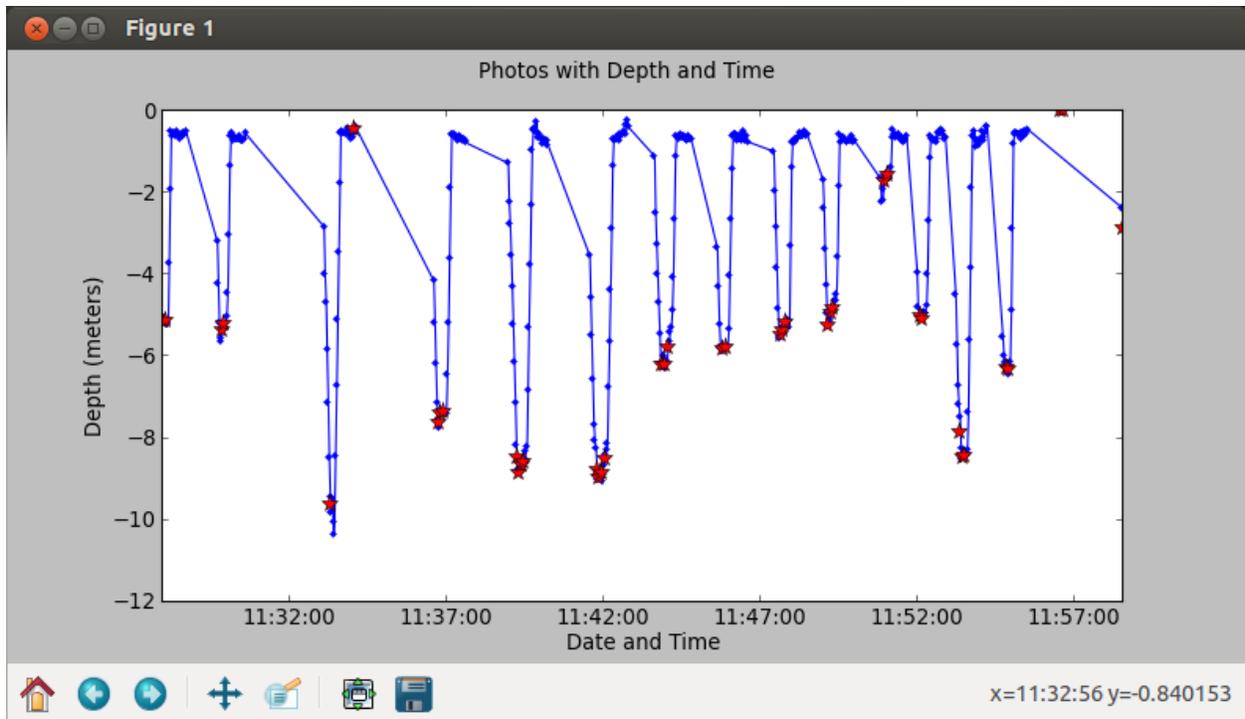
At some point I'd like to add a graphical interface for configuring BPS but, for now, you'll have to edit `configuration.py` in a text editor. There are explanatory comments in the file and it should be fairly straight forward.

You'll want to set `LOCAL_TIME_ZONE` to the time zone you're collecting data in. You'll probably also want to change the list of habitats and the list of substrates. Just read the comments in the code.

If you are deploying more than one camera / gps / depth logger set up at a time, you will need to take precautions to avoid confusion. You will need to change the `db_path` in `configuration.py` and keep the logs from the two setups separate. For the sake of this tutorial, let's assume you're only using one setup at a time.

1.3.6 Using BPS

1. Launch BPS by navigating to the `bps_package` directory and typing: `python bps_gui.py` in the command line. Some windows installations will require that you just type `bps_gui.py`. If you installed bps with git, `bps_package` directory will be `benthic_photo_survey/bps_package` relative to wherever you put it.
2. Load your GPS file. Select 'Load GPS Log' from the file menu, select your GPX file and click 'Open'. This will load the positions and time stamps into a database specified in `configuration.py`.
3. Load your depth log. Select 'Load Depth Log' from the file menu. This step is optional but it will load depths and temperature readings from the depth log into the database specified in `configuration.py`.
4. Load your photos. Select 'Load Photos' from the file menu and chose the directory that contains the photos you would like to tag and use to create a shapefile.
5. You can now use the 'Next' and 'Previous' buttons (or the arrow keys on your keyboard) to scroll through your photos. You can tag the photos one at a time with Depth and Temperature or location using the buttons at the bottom of the application or you can use the items in the 'Actions' menu to tag all the loaded photos with location or depth and temperature. Either way, you'll see the assigned attributes in the 'Exif Data' area on the right side of the application window.
6. You can choose 'Depth Plot' from the 'Output' menu. This will use Matplotlib to render a graph of the depth log for the loaded photos with stars that represent where each photo was taken. This plot is particularly useful for data that has been collected while free diving. You can verify that the clocks were adequately synced by making sure that the photos were taken at the deepest parts of the dives.



7. Export a shapefile using the 'Output' menu. This shapefile can be viewed using most GIS software. I generally use QGIS.

1.4 Code Documentation

The following is autogenerated from the doc strings in the Python code. This section is **not** going to be useful if you are simply working with Benthic Photo Survey as a user. This documentation will be of use to those who want understand how the code works. ...or it might be of use if I get around to finishing it. Much of the code is poorly documented at the moment.

1.5 Indices and tables

- *genindex*
- *modindex*
- *search*

- [RP2010] Roelfsema, C., Phinn, S., 2010. Integrating field data with high spatial resolution multispectral satellite imagery for calibration and validation of coral reef benthic community maps. *J. Appl. Remote Sens* 4, 043527–043527.