1.Instruction

This software is for analyzing offline sidescan sonar data files in xtf format. Four files could be acquired after processing per xtf file: basic information(in json format), trajectory information with timestamp(in mat format), port sides data(in mat format), starboard sides data(in mat format). Meanwhile, there are some simple python script for further processing the output files, such as drawing the trajectory image, drawing the sidescan image and so on. The software is working on Linux.

2.Installation

(1)Dependencies

This software depends on the MATIO library, namely MATLAB MAT file I/O library(MATIO github). If you are working on ubuntu, you can install the library with the following commands:

```
sudo apt get install libmatio-dev
```

Additionally, the library path is need in the CMakeLists.txt while building the program. We can use dpkg command to get the path:

```
dpkg -L libmatio-dev
```

The path of libmatio.so is the parameter we need in the CMakeLists.txt.

On the other hand, RapidJSON, a JSON parser and generator for C++, is used for json processing. For more information please refer to: http://rapidjson.org/.

(2)Building

```
cd xtf_data_analyze
mkdir build
cd build
cmake ..
make
```

Attention, the path of libmatio.so in the CMakeLists.txt must be check, or the building will fail.

The execute programs can be found in the build folder: singleFileProcess, multiFileProcess. The singleFileProcess is for single file processing, with the path of an xtf file as input. The multiFileProcess is for multi files processing, with the path of an xtf folder as input.

(3)Test

There are three xtf files in the data folder for testing.

Single file test:

```
cd build
./sigleFileProcess ../data/Demowreck.xtf
```

We can get the output files in the folder containing the xtf file. So four output files can be found in the data folder:

The basic information data: 000_1236_info.json,

The trajectory with timestamp data: 000_1236_TRAJ.mat,
The port side data: 000_1236_PIM.mat,
The starboard side data: 000_1236_SIM.mat.
Multi files test:

```
cd build
./sigleFileProcess ../data/
```

We can arrange the trajectory files into a folder:

```
mkdir trajectory
cp *_TRAJ.mat ./trajectory
sudo rm -rf *_TRAJ.mat
```

3.The output files format

(1)The file of basic information data:

Further information of most of the parameters can be found in the *eXtended Triton Format(XTF) Rev.41*. Please refer to XTFFILEHEADER Structure and CHANINFO structure. The struct of the data file is as the following:

```
{
   “BaseInfo”:
    {
       “FileName” : xxx;
       “SonarName” : xxx,
       “TotalPingNumber” : xxx,
       “NavUnits” : xxx,
       “NumberOfSonarChannels” : xxx,
       “NumberOfBathymetryChannels” : xxx
    },
    “FirstPing”:
    {
       “Time” : xxx,
       “Longitude” : xxx,
       “Latitude” : xxx
    },
    “LastPing”:
    {
       “Time” : xxx,
       “Longitude” : xxx,
       “Latitude” : xxx
    },
    “chan_x”:
    {
       “TypeOfChannel” : xxx,
       “SubChannelNumber” : xxx,
       “CorrectionFlags” : xxx,
       “BytesPerSample” : xxx,
       “VoltScale” : xxx,
       “Frequency” : xxx,
       “HorizBeamAngle” : xxx,
       “TiltAngle” : xxx,
       “BeamWidth” : xxx
    }
}
```

(2)The file of trajectory with timestamp data:

In the trajectory mat file, the key is “traj_matrix”, the value is a matrix (TotalPingNumber x 12). Each row is composed of a ping data, and the rows are arranged in time order with the first ping data as the first row. The column parameters are as the following:

```
[timestamp, longitude, latitude, altitude, slantRangePort, slantRangeStbd, numSamplesPort, numSamplesStbd, sensorSpeed, sensorPitch, sensorRoll, sensorHeading]
```

The 5th and 6th column data is the slant range of the port side and the starboard side. The 7th and 8th column data is the number of samples of the port side and the starboard side. We can use these data for trajectory image, slant range correction, geometric correction and so on.

(3)The file of port side data

In the port side mat file, the key is "port_intensity_matrix", the value is a matrix (TotalPingNumber x numSamplesPort). We can use the matrix data to draw the port side of sidescan image in grayscale or other format.

(4)The file of starboard side data

Nearly the same as the file of port side data, except the key is "stbd_intensity_matrix".

4.Some simple function in python

There are some python scripts to process the output files. Some libraries or packages may need for the scripts, please install them if you want to use the scripts.

(1)draw the sidescan image

We can draw the sidescan image with the draw_intensity_image.py, the port side data and the starboard data files are needed. An example is as the following:

```
python3 draw_intensity_image.py xxx_PIM.mat xxx_SIM.mat
```

(2)draw the trajectory image

We can draw the trajectory for a single file, an example is as the following:

```
python3 draw_trajectory_image.py xxx_TRAJ.mat
```

Or, we can arrange some trajectory data files into a folder, then the folder path will be the input for multi trajectories drawing. All the trajectories are in the same coordinate of an image. An example is as the following:

```
python3 draw_multi_trajectory_image.py FOLDER_PATH
```