

Development of Automated Processing for High-Definition Mapping System

Prachaya Daruthep and Nikorn Sutthisangiam

Department of Computer and Information Science,
King Mongkut's University of Technology North Bangkok, Thailand

p.daruthep@gmail.com, nikorn.s@sci.kmutnb.ac.th

Abstract— High-Definition (HD) Mapping is a system using Mobile Mapping System (MMS) technique to create HD panoramic point cloud map of the georeferenced location. The process to create the HD maps consists of joint processing of large amount of data originating from various sources. These data go through various post-processing steps using associated software to create a resulting HD point cloud map. To run these processes successfully require an appropriate processing server, a correct flow of processing sequence and validation of the input/output data. These processes are error-prone and time-consuming for human users. This paper is presented to illustrate practicality of an automated processing system for the creation of HD map, which can boost the efficiency of data processing for the whole system.

Keywords— Automation, Big Data, Data Storage, High-Definition Mapping, Post Processing

I. INTRODUCTION

A High-Definition Mobile Mapping System was developed as a high-resolution surveying technology to capture geographic information at a desired location. Most MMS is a multi-sensor system that consists of mapping sensors, navigation sensors, control units [1] and vehicle-mounted cameras. Data collected from MMS includes: Package Capture (PCAP) files, processed GPS-IMU data [2] and high-definition street images. This data can amount up to a total of 2,000 data files with total files size of 7 gigabyte per 2 hours of survey period, which accumulated result up to 120,000 data files and 450 gigabyte of data size per man-month of a survey.

These big capacity data then need to be sent from a survey vehicle, through a 4G mobile network, continuously in real-time, to a remote processing server. This raise first issue that need to be addressed: the synchronization and an ability to resume transmission of big data in case of mobile network outage.

The data then have to go through post-processing steps with the following software:

- Inertial Explorer – a software for post-process GPS-IMU data.
- VP2Las – a software for converting PCAP file and Inertial Explorer output into LASer (Las) file.

- ImageOrientation – a software to organize high-definition images from vehicle-mounted cameras into format that can be used with HD maps.
- PotreeConverter – a software to convert LAS file into JSON files that can be used to render HD point cloud map.

Software involved in post-processing steps have the following characteristic:

- Each software requires a different number of inputs and generate a different number of outputs
- Some software requires output data of another software to use as an input data, while some software don't have this requirement
- Time it takes to run each software varies, from few minutes up to few hours.

These post-processing software must run in correct order and with valid input, which can prove to be a daunting and time-consuming task by users. By developing a system that can automate these steps, improvement on efficiency and correctness of the processes can be achieved. But the automation itself comes with issues: how will each software in these steps know when to initialize and start its process, and how will they validate their input and output?

And as there are large amount of data involved - the system must have a way to that organize input and output data in a sensible way, as well as generation of search-able indexes for all data - allowing the system to handle large amount of data while retain its full functionality.

II. SYSTEM HARDWARE

The proposed automated HD map processing system consists of several systems and software which needs to run on an appropriate hardware. Main hardware components used with the processing system in this paper are as follows:

- Survey vehicle – Ford Ranger double cap, standard roof installed with Velodyne LiDAR® Puck™ VLP-16, NovAtel INS SPAN-CPT GPS-IMU, Ladybug5 camera and Control PC with Windows OS (Figure 1)
- Processing Server – Intel® Xeon® processor E5-2620 v4, 64GB RAM, 3x960GB SSD, Linux OS



Figure 1. Survey vehicle

III. SYSTEM STRUCTURE AND IMPLEMENTATION

A system presented in this paper must tackle three main issues (1) Transmission and synchronization of big data over 4G mobile network (2) Execution sequence management of post-processing software (3) Data validation and indexing

Data processing diagram (Figure 2.) display how system hardware presents in the system and what software components reside within them.

A. Transmission and Synchronization of data

Integration of open-source cloud service software NextCloud [3][4] are used to handle transmission and synchronization of input data. NextCloud provides access to cloud storage server through a client application. Maintaining and synchronize remote file data using WebDAV protocol – NextCloud allow resumming of data transmission in case of internet connection outage. Automatic synchronization of file also allows continuous transmission of big data collected by survey vehicle in real time, with minimal intervention from users.

NextCloud Client application are installed on a control PC of a survey vehicle, allowing data synchronization from survey vehicle to online cloud storage on a processing server.

Data that synchronized to the processing server then can be use as an input in post-processing for generating HD map.

B. Monitoring system for sequence management of post-processing software

As there are various of software involve in the post-process of HD map data, and each software can have different number of input and output while output of one software can be an input of another software. In this system design, we create a process monitoring system to control execution sequence of all software.

The processing monitoring system is developed in PHP scripting language deployed on an Apache web server located on the processing server. The system also make use of MySQL database to store metadata needed in an execution of the system. PHP, Apache and MySQL are chosen due to their ease of deployment and availability under GNU General Public License (GPL).

This overarching monitoring system will monitor the whole system for valid input data of each software. When the monitoring system detect a valid input data of a corresponding software, the monitoring system will invoke a command to run that software. The monitoring system run a software in a fire-and-forget manner, with no need to check for validity of the resulting output data, this is due to the fact that - the output data of a preceding software will be used as an input data of the following software – thus, the monitoring system only needs to check validity of the input data of each software without worrying about their output.

System data flow are shown in data processing diagram (Figure 2.). Prerequisite input data and output data of each post-processing software and their output data are shown in Table 1.

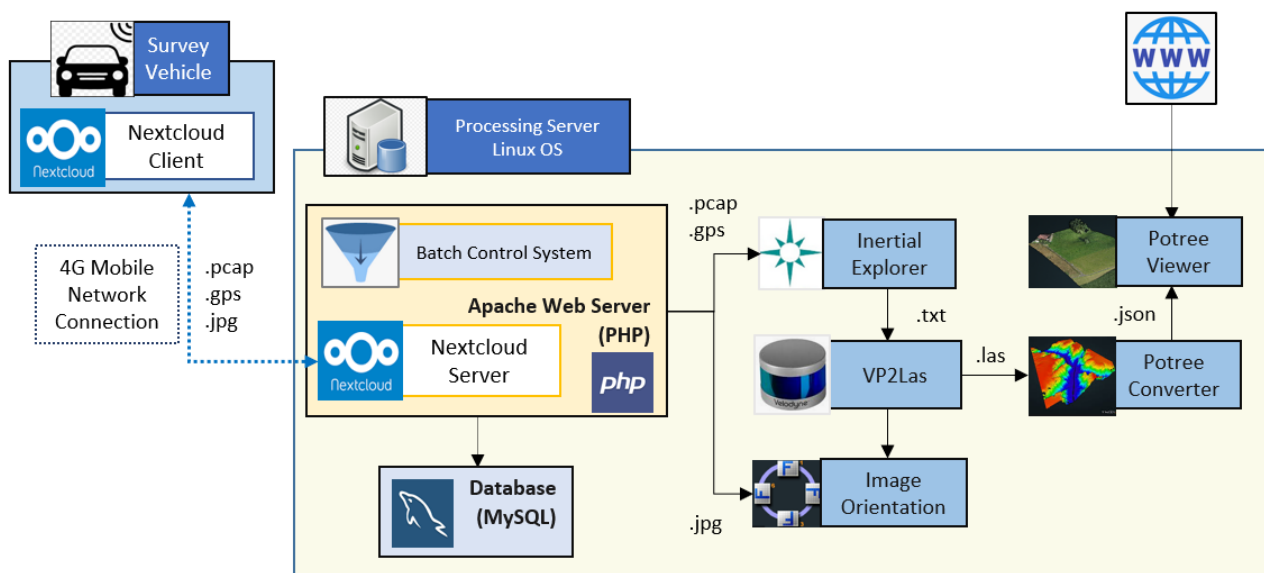
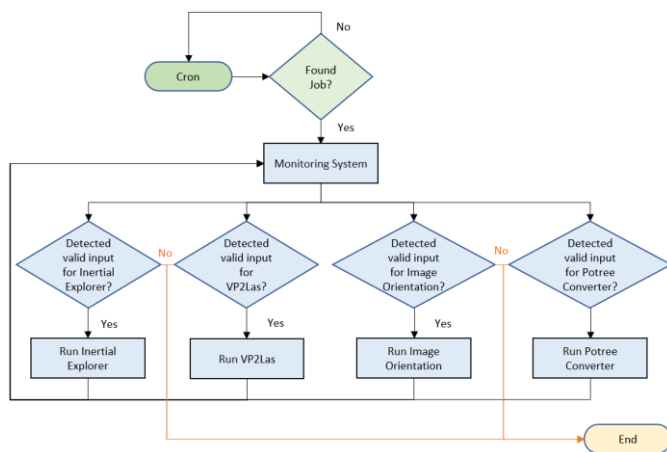


Figure 2. Data processing diagram

TABLE 1. INPUT AND OUTPUT OF EACH POST-PROCESSING SOFTWARE

Software	Input Data	Source of Data	Output
Inertial Explorer	.pcap .gps	Survey Vehicle	.txt
VP2Las	.txt	Inertial Explorer	.las
ImageOrientation	.jpg	Survey Vehicle	.jpg
Potree Convertor	.las	VP2Las	.json

The automation of the monitoring system itself is achieved by using Linux's cron jobs of the processing server. The use of cron allows OS to run commands or scripts at a given date and time as setup by server administrator. Flowchart on how the monitoring system interact with post-processing software via cron jobs are shown in Figure 3.

**Figure 3.** Monitoring system flowchart

From the chart, it is important to note that each post-processing software may run parallel with each other as long as a prerequisite input data from the preceding software are present in the system. Setting up the system to run in this way simplifies the need of error-handling such that, if run-time error occurs on any software step, then the monitoring system will only need to re-run that software step only - with no need to re-run all software steps altogether again.

The result running status of each software step (completed/failed/retry) along with timestamp of the run status are store as metadata in MySQL database for further reporting uses.

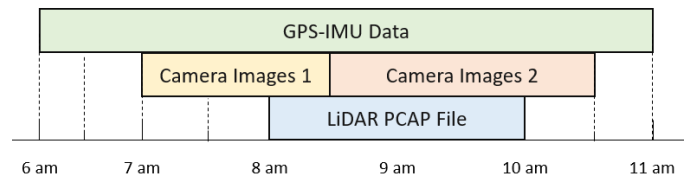
C. Data validation and indexing

The system must have a capability to handle validation and indexing of data due to (1) non-uniformity of input data from survey vehicle and (2) needs of data searchability

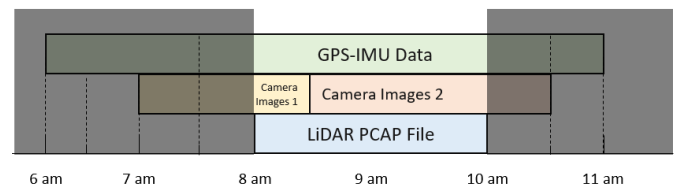
1) The handling of non-uniformity of data from survey vehicle: as survey vehicle collect input data using its LiDAR sensor, GPS-IMU device and cameras. The common issue is that, in any one day of surveying – the survey vehicle may keep its GPS-IMU device running for the whole day, and only switching its LiDAR sensor on when passing through a

designated area. This result in smaller number of PCAP file compared to number of GPS-IMU data.

In this case, the monitoring system must be able to tell the corresponding post-processing software on which part of the input data to be used in the process. The monitoring system achieved this by looking at timestamp of a smaller part of data, then finding a corresponding larger part of data using an overlapped timestamp. For example, in Figure 4, GPS-IMU device is running non-stop from 6am to 11am. While camera is running in two interval, from 7am to 8.30am and from 8.30am to 10am. The LiDAR sensor are running from 8am to 10am.

**Figure 4.** Result input data with timestamp from survey vehicle

In this case, the system take a timestamp from smaller data part, which is a PCAP file, and got a timestamp value between 8am to 10am, and from this timestamp – the monitoring system will now know that the corresponding images and GPS-IMU data to use as the input will also have a timestamp between 8am to 10am. In Figure 5, the non-greyed-out area illustrate part of input data that will be used in post-processing software based on PCAP timestamp

**Figure 5.** Input data that will be used in post-processing

2) Enable searchability of data involved in the system: to allow all data involved within the system to be search-able. The monitoring system add metadata regarding input and output data into a MySQL database every time it detects a new set of data in the system. The data are stored using Entity–Attribute–Value (EAV) data model, which enable the system to store large amount of data of various categories into a single centralized data table. The using of EAV result in simple implementation of metadata storage and add scalability to the system when more data categories are needed to be add into the system.

Table 2 show an example of metadata stored using EAV data model.

TABLE 2. EXAMPLE OF METADATA STORED USING EAV DATA MODEL

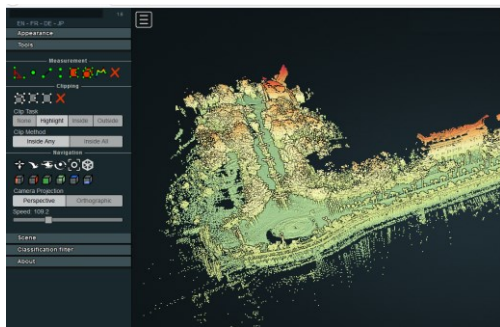
Meta ID	Meta For	Meta Value
01	survey_round	20191009-01
20191009-01	pcap_files	Data_01.pcap
20191009-01	image_path	./image_01/ ./image_02/
20191009-01	json_path	./result_json/
....		

An example metadata can be read as follow: for survey number 01, performed on date 09 October 2019, it consists of one PCAP file named Data_01.pcap. Camera images associated with this PCAP file can be located in processing server folder path of ./image_01/ and ./image_02/. The resulting JSON data that can be used to generate a HD point cloud map can be found in processing server folder path ./result_json/. and so on.

The result metadata from indexing process can then be used to create graphical user interface (GUI) that allow users to access data conveniently. The development of the GUI, however, is beyond of scope of this paper.

IV. THE RESULTING HD MAP

After post-processing steps are done, JSON results can then be used to render a HD point cloud map as shown in Figure 6.

**Figure 3.** Rendering of high-definition point cloud map

The point cloud map is rendered using Potree Point Cloud Viewer with JSON source located on the processing server. This allow users to access the point cloud map remotely using any web browser that supports rendering of 3D graphic via WebGL - such as Google Chrome, Firefox or Microsoft Edge. HD maps from this system can then be used to display geometry model of the surveyed area and to measure distance between each point cloud. This allow users to make remote viewing and measurement of, for example, street width, building height and street level of the surveyed area.

V. FUTURE WORK AND CONCLUSION

For further development, the system presented can be improved upon by integrating additional post-processing software or algorithms, such as an elevation smoothing algorithm of GPS trajectory data [5], to the system. Another important direction of future development is the integration of

HD panoramic street view (e.g. Google Street View) as the proposed system already contains PCAP, GPS-IMU and image data required for the creation of such map.

With further adjustment on monitoring process, the system can be set up as a distributed system, as the monitoring system presented can handle a decentralized deployment of post-processing software. From the data modelling point of view, the system can be modified to use Relational Model (RM) instead of EAV model – this, however, require an evaluation of system requirements, such as volume and number of data categories or ease of development for the system GUI.

The result automated system from this paper can be used as a stand-alone system for creating HD maps, or as an automatic system framework to use with other HD mapping tools or related applications in the future.

ACKNOWLEDGMENT

This paper has been research under Development of High Performance Automate Computing System for MMS Project, Technology and Digital Division, Hydro Informatics Institute (HII).

REFERENCES

- [1] A. Yaemjaem, N. Sutthisangiam, A. Sompan, N. Sa-ngiam and P. Jindasee, "Development of High-Definition 3D Mapping System for Water Resources Management," 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, 2018, pp. 219-223.
- [2] N. Sutthisangiam and S. Bamrungruk, "Software-Based Timing Synchronization for Point Cloud Reconstruction," 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, 2018, pp. 37-41.
- [3] S. Alexander Pavlovich, P. Anton Yurievich and B. Vladimir Alexandrovich, "Integration of Open-Source Software for Automation of Electronic Document Flow in the Structural Unit of an Educational Organization," 2018 IV International Conference on Information Technologies in Engineering Education (Inforino), Moscow, 2018, pp. 1-4.
- [4] Nextcloud official website of the project. [Electronix resource]. – available at <https://nextcloud.com>.
- [5] Phuengsaeng, Premanan and Nikorn Sutthisangiam. "Automatic Corrective Elevation of GPS Trajectory in Urban Area." 2018 18th International Symposium on Communications and Information Technologies (ISCIT) (2018): 458-462.



Prachaya Daruthep received Bachelor degree in Computer Science from School of Information Technology, King Mongkut's University of Technology Thonburi, Thailand, in 2006. Currently studying Master degree in Department of Computer and Information Science, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand, with expected graduation year to be 2020.



Nikorn Sutthisangiam received Bachelor degree in Electrical Engineering from Faculty of Engineering, King Mongkut's Institute of Technology North-Bangkok, Thailand in 2003. Master of Science in Communications Engineering and Doctoral degree in Electrical Engineering from The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), King Mongkut's Institute of Technology North-Bangkok, Thailand, in 2006 and 2014, respectively.