Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, July 9-12, 2018

WBT3.1

A Soft Time Synchronization Framework for Multi-Sensors in Autonomous Localization and Navigation*

Hang Hu, Jianhua Wu, *Member, IEEE* and Zhenhua Xiong, *Member, IEEE*

Abstract—**This paper proposes a soft time synchronization framework for synchronizing multiple sensors in autonomous localization and navigation. The main contribution is that we propose to synchronize all sensors through ROS nodes without hardware trigger based synchronization. Firstly, a ROS node is established for each sensor to acquire sensor messages and publish sensor messages on ROS topics. The sensors include color camera, 3D LIDAR and high-precision RTK-GPS devices in our test. Then, ROS synchronizer is used to synchronize messages on ROS topics. The time stamp of synchronized data set and generation time of the sensor messages can be used to validate the synchronization performance. From experimental results, it is shown that the proposed soft time synchronization framework can achieve low latency and low synchronization error compared to other datasets, proving that the soft synchronization framework is effective and applicable to various sensors.**

Keywords-soft synchronizaiotn framework; ROS node; camera; LIDAR; RTK-GPS;time stamp;low latency;low synchronization error

I. Introduction

In recent years, there are numerous research on autonomous localization and navigation. Especially, the advances in autonomous driving and micro aerial vehicles (MAVS) are quite significant. Nevertheless, there are still great demand on reliable localization and navigation since the actual environments are usually complex and unpredictable.

To achieve safe and reliable navigation, a comprehensive and precise perception of the environment is indispensable. In research or real applications, several types of sensors for data acquisition are usually used. Among them, cameras and LIDARS, which can provide 3D information of surrounding objects for recognition, are mostly applied. Moreover, global positioning system (GPS) and inertial measurement unit (IMU) are commonly used to provide localization information for the navigation.

Currently, all autonomous driving and flying real-time localization and navigation experiments, including some famous datasets, are collected by a navigation platform composed of multi-sensors. Most autonomous driving datasets employed cameras, LIDARs, and GPS/IMU for data acquisition, like the Malaga Urban Dataset [1, 2], the Oxford Dataset [3]. The NCLT Dataset [4] and KITTI Dataset [5] adopted real-time kinematic (RTK) GPS for better accuracy of positioning. The CCSAD Dataset [6] only used camera and GPS/IMU for autonomous driving.

The above mentioned sensors are also used in MAV navigation. The Zurich MAV Dataset [7] used camera and GPS/IMU to record the flying information. The EuRoC datasets [8] employed cameras, LIDARs and IMU, where GPS were not equipped. Camera and IMU are used for autonomous flight in [9, 10, 11]. Apart from autonomous driving and flight, in modern agriculture, autonomous navigation is also crucial. Several cameras and GPS/IMU were both equipped on the agriculture platform [12, 13].

From these real-time autonomous localization and navigation experiments, we can see that precise localization and navigation requires the utilization of several types of sensors. Since many sensors are applied for data acquisition, data synchronization and data fusion are quite important. Moreover, the time synchronization for multi sensors will influence the localization accuracy and algorithm performance directly. Low latency and high accuracy are significant when synchronizing sensor data.

Unfortunately, most sensors provide no explicit method for sensor synchronization. Some universal synchronization algorithms, like the Network Time Protocol (NTP) [14], aim to solve the synchronization problems in multi systems, and are not applicable to the data synchronization problem with many sensors equipped on a platform. If more than one camera are applied on the data acquisition platform, the cameras could possibly be internally synchronized using the method provided camera manufacturers. The Oxford RobotCar Dataset [3] used many cameras, such as Point Grey Bumblebee, Point Grey Grasshopper, and they could be internally synchronized. Edwin Olson [15] proposed a passive synchronization method to eliminate the transmission error between sensor data, but it seemed more like a method of clock correction. For multi-camera synchronization, a software-based method was proposed by Georgios Litos [16] for real-time synchronization of multi-camera. In the famous KITTI Dataset [5, 17] for autonomous driving, they used Velodyne laser scanner to trigger the cameras (PointGray CCD camera) when facing forward, which is a hardware-based synchronization method. Furthermore, the GPS/IMU data package was synchronized to the camera/Velodyne data package by comparing their timestamp, then a set of closest data will be chosen as the synchronization result. This synchronization method is common and widely applied for simultaneous localization and mapping (SLAM). As for the synchronization between camera/Lidar and GPS, there exists no common method. For some Lidar laser

*This research is supported in part by the National Natural Science Foundation of China (51675325), Program of Shanghai Subject Chief Scientist (18XD1401700).

H. Hu and J.H. Wu are with the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China (e-mail: kemain@sjtu.edu.cn, wujh@sjtu.edu.cn).

Z.H. Xiong is with the State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China (021-34206547; fax: 021-34206086; e-mail: mexiong@sjtu.edu.cn).

scanners, they can receive standard PPS (Pulse per Second) pulse and NMEA sentences with positioning data through network connection, then hardware synchronization can be achieved by GPS clock. However, the synchronization frequency is only 1 HZ, which is usually much lower than the actual requirements. The synchronization methods introduced above are based on hardware platform, usually depend on complex electrical circuit and only apply to some specific sensors.

We listed the sensor equipment and synchronization method of some famous autonomous datasets in Table 1 to show the current research. There is currently no universal synchronization framework of multi-sensor, no matter in autonomous driving research, or autonomous flying study. From the table, we can find that CCD cameras are widely used and only support hardware based synchronization. Moreover, hardware based triggering and synchronization method is not applicable for USB cameras, such as Intel RealSense, Xsus Xtion, Microsoft Kinect, ZED camera and so on, which can only be connected to the computer through a USB link, not the network or other common links. However, these cameras are widely applied in robotics scenarios and navigation.

TABLE I. COMPARISON TO EXISTING DATASETS

Dataset	Cameras	Lidar	GPS	IMU	Synchronization
Malaga [1]	CCD	√	√	√	Hardware-based
Oxford [3]	CCD	√	√	×	Hardware-based
NCLT [4]	CCD	√	√	√	Hardware-based
KITTI [5]	CCD	√	√	√	Hardware trigger
CCSAD [6]	CCD	×	√	√	Hardware trigger
Zurich [7]	CCD	×	√	√	Hardware-based
EuRoC [8]	CCD	√	×	√	Hardware-based
FieldSafe [12]	CCD USB	√	√	√	Hardware Software

In this paper, we propose a soft synchronization framework of multi-sensor based on ROS platform, which is applicable for most sensors used in autonomous localization and navigation, include the sensor that does not meet the hardware based triggering and synchronization requirements.

We validate our framework using three different sensors, and then collect a set of data in real time to prove the synchronization result. We show that our method can achieve time synchronization, and compare the result with some SLAM datasets. The comparison shows that the method meets the synchronization error requirements.

II. SYNCHRONIZATION FRAMEWORK

The soft synchronization framework for multi-sensor data acquisition and time synchronization is developed under the Robot Operating System (ROS), which is a widely used flexible framework, and supports many common sensors. It contains many powerful tools and libraries. Therefore, we consider designing a synchronization framework within ROS.

The schematic of the synchronization process is shown in Fig. 1. We employ ROS nodes to organize the whole framework and divide the synchronization framework into three parts: data acquisition, data synchronization and data output. In our framework, no more hardwares are required for synchronization, and messages of all sensors are transmitted in the software.

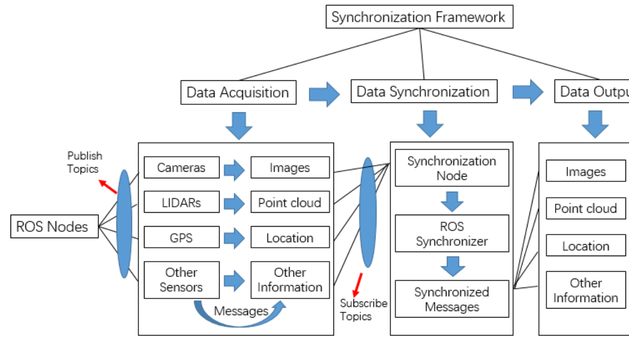


Figure 1. Schematic of the synchronization process

A. Data Acquisition

In order to synchronize the sensor data by ROS, we need to acquire the sensor data first. Since ROS provides interfaces for many types of sensors. As shown in Fig. 2, most known robotic sensors are supported by official ROS packages, like ZED stereo camera and Velodyne VLP-16 3D LIDAR that adopted in our experiment. The three figures below just show part sensors supported by ROS. More information can be viewed on the website: http://wiki.ros.org/Sensors. ROS provides interfaces and drivers for many types of sensors. Especially, for those sensors which ROS does not provide interfaces, through the driver provided by the manufacturers, these sensors can also be connected to ROS and provide time stamp.

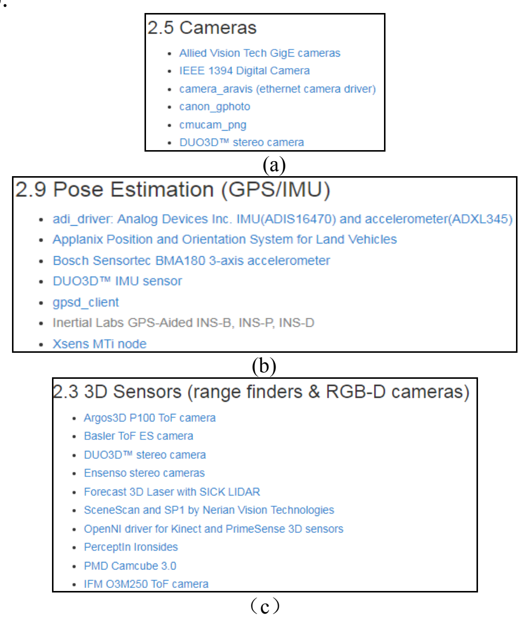


Figure 2. Sensors supported by ROS: (a) Common Cameras. (b) Common 3D sensors. (c) Common GPS/IMU devices

Through the provided interfaces, we can acquire the sensor data by subscribing ROS topics. The ROS topics can provide real-time sensor data and its time stamp, which can be used for synchronizing. When sending ROS messages between ROS nodes, communication on topics happens. We can use the publisher and subscriber to send and receive the message or sensor data at a fixed frequency (determined by sensors).

In our experiment, images and point cloud, provided by the ZED camera and Velodyne LIDAR, were acquired by ROS interface and published on ROS topics. However, for some sensors, ROS does not provide interfaces to acquire data, like RTK-GPS devices. In this case, we can first acquire the sensor data through its own code, and then publish the data and its time stamp on a ROS topic. Fig. 3 shows the publishing process of the GPS data.

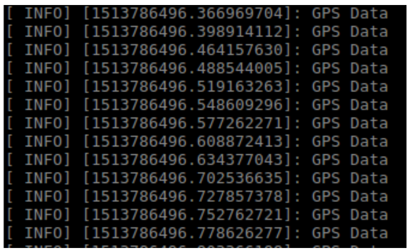


Figure 3. Published GPS data information

A respective ROS node for each sensor can be established to assure that the messages between the sensors are not interfered mutually, and then sensor data can be published on ROS topics. When we need to use the sensor data, we can subscribe the corresponding topic of the sensor.

B. Data Synchronization

As mentioned above, many cameras do not support hardware based triggering and synchronization method, which also happens in some sensors used in robotics. Under this condition, the sensor data cannot be synchronized as they are generated, and thus need to be synchronized in software.

Moreover, the frequencies of different sensor data are usually different in practical localization and navigation. For example, the sensors used in our experiment have different frequencies as 30 Hz (images), 10 Hz (point cloud), and 10 Hz (GPS data), respectively.

Assuming that we have published sensor data on ROS topics as introduced above, we can use the TimeSynchronizer filter to synchronize messages from multiple sources. We subscribe the published topics as coming channels of the synchronizer. Fig.4 illustrates the synchronization policy.

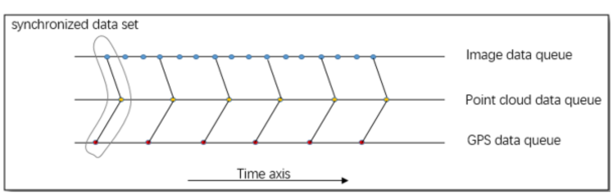


Figure 4. Synchronization policy

Each line represents a data queue of each sensor with time stamp. Through the synchronizer, we can obtain sets of synchronized data. The synchronizer takes the data queue of least frequency as the baseline. For every data on the baseline queue, take the nearest data on other queues to join the set. When all queues have been picked, these data compose a synchronized data set.

C. Data Output

We can output the synchronized data sets in the form of a callback that takes the identical number of channels, and then synchronized data set of each sensor can be called. The timestamp of the sensor data documented their specific generation time. We take the current ROS time as the unified time of the data set when calling synchronized data.

III. Experimental Setup

In order to evaluate our synchronization framework, we assembled a mobile platform for data acquisition, which employs several sensors: one color camera, a laser scanner sensor, GPS receivers (both differential GPS and GPS-RTK). In the following, we describe our sensor specifically.

A. Color Camera

Image grabbing was performed by a ZED color camera, which can capture up to 3840x1080-size images at a maximum frame speed of 30 fps and transfer them to the computer via a USB link.

In our experiment and datasets, the images were captured at 30 fps with a dimension of 1280x720 pixels. The camera is shown in Fig. 5 below.

Figure 5. ZED camera

B. Laser Scanner

We used a Velodyne VLP-16 sensor, shown in Fig. 6, to capture the point cloud data of the surround-ings. Velodyne's new Puck, VLP-16 sensor is the smallest, and most advanced product in Velodyne's 3D Lidar product range. It supports 16 channels, ~300,000 points/second, 360 degree horizontal field of view and a 30 degree vertical field of view, with ± 15 degree up and down. In our experiment, the VLP-16 was configured to measure up to a maximum range of 100 meters at 10 fps.

Figure 6. Velodyne puck VLP-16 sensor

C. GPS Devices

Apart from images and point cloud data, the vehicle position is also critical. Global Positioning System (GPS) has become the most reliable system for land surveying and vehicle positioning [2]. GPS can provide high accuracy and spread widely around the worldwide.

A standard GPS system consists of a mobile receiver and a satellite constellation. However, GPS devices are divided into several kinds due to their operation principle and positioning accuracy, including normal GPS, differential GPS (DGPS) and GPS-RTK systems. Naturally, each GPS system has their own applications and limits.

TABLE II. COMPARISON TO EXISTING DATASETS

Dataset	GPS type	GPS Frequency
MALAGA [2]	RTK GPS	4 HZ
Oxford [3]	Normal GPS	50 HZ
NCLT [4]	Normal GPS RTK GPS	5 HZ 1 HZ
FieldSafe [12]	RTK GPS	20 HZ
[19]	RTK GPS	10 HZ

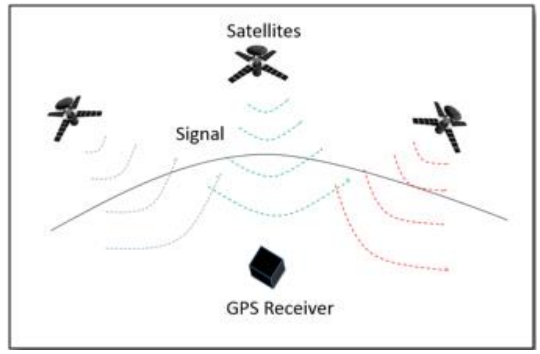
For different GPS systems, the positioning accuracy varies from centimeters to meters, and their data transmission frequency is also varied. Fig.7 (a) illustrates the operation principle of normal GPS systems. Most autonomous datasets used normal GPS system for mobile robot localization. The localization accuracy is about 3 meters, which is only applicable for rough localization. Generally, the data transmission frequency of normal GPS systems can be up to 100 Hz, much higher than other GPS system. The output frequency of different GPS systems is shown in Table 2.

Differential GPS (DGPS) technique utilize the reference station to improve the accuracy. Through comparing the receiver data with the reference station data and computing, the receiver data can be corrected. The DGPS corrections increase the GPS positioning accuracy to tens of centimeters. The operation principle of DGPS systems are shown in Fig.7 (b). Xiu-zhen Wu [18] used differential GPS for mobile vehicle localization, where the GPS devices are Flexpak6 with 5cm positioning accuracy, produced by Novatel Company.

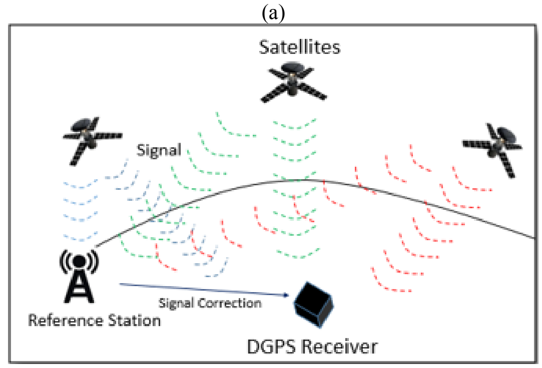
Real-Time-Kinematics (RTK) technique employs the carrier phase of the GPS satellite to improve the accuracy. Different from DGPS systems, the RTK reference station is close to the GPS receiver, usually tens of meters as shown in Fig.7 (c). In the technique, a fine delay estimation is got and localization accuracy can be improved up to 1 cm, which is fairly high and can be applied in most occasions and projects, like object recognition and grabbing. Yukung Choi et al [19] set a Trimble GPS with RTK in the vehicle, and the data transmission frequency can be up to 10 Hz, much lower than the normal GPS systems. However, the localization error would be less than 5cm, better than both the normal GPS systems and DGPS systems.

We placed 2 RTK-GPS devices (Fig.8) in our experimental platform, which produced by a local company with 1cm localization accuracy and 10 Hz data transmission frequency. We collect and process the sensor data under the Ubuntu Linux (64 bit) operation system on Jetson TX 2. This embedded AI supercomputer is portable and powerful, which can meet image processing and data computing requirements.

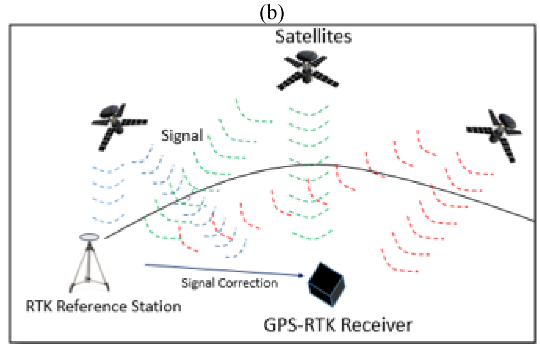
We designed an interface to display all the data in real time. The interface is designed by QT applying RVIZ modules. We could display image, point cloud, and location information in real time as shown in Fig. 9. In Fig. 9, three pictures are used to describe the same scene. The left two pictures respectively represent the original image and point cloud. The right picture represent the object location provided by RTK-GPS devices.



(a)



(b)



(c)

Figure 7. GPS operation principle: (a) normal GPS system. (b) DGPS system. (c) RTK-GPS system

(a) (b)

Figure 8. RTK-GPS devices: (a) GPS receiver. (b) GPS antenna

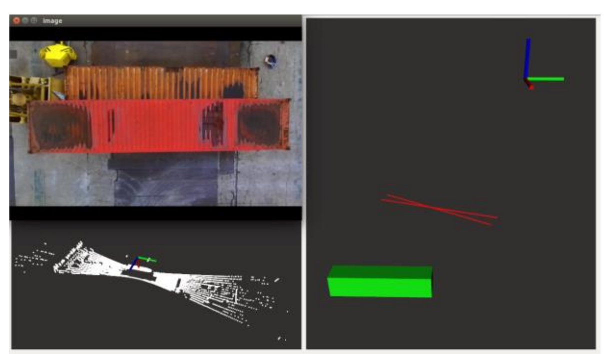
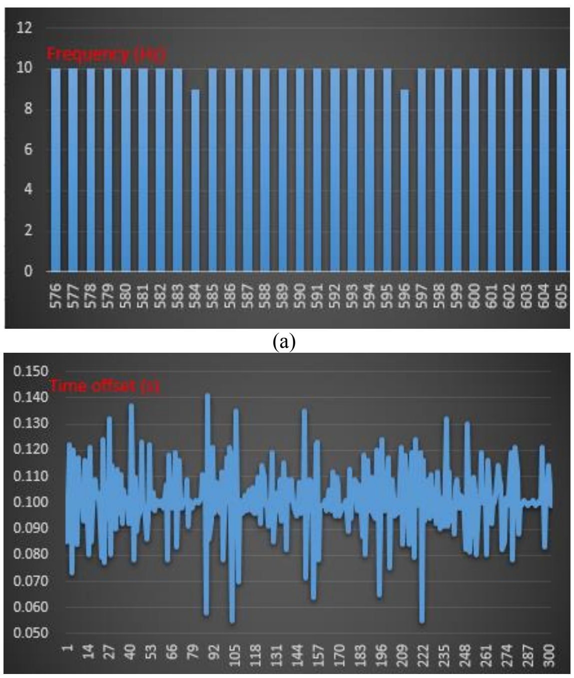


Figure 9. Display interface

IV. Experimental Results

In order to validate our synchronization framework, we used the proposed synchronization method with prepared sensors to collect the sensor data. There are two indicators to prove the synchronization performance: synchronization frequency and synchronization error between sensors. The synchronization of the sensor data requires high output frequency, low latency of sensor data, and low synchronization error, which indicates the data accuracy. Low latency and high accuracy are what we pursue. In order to validate the latency and accuracy, we documented the timestamp of synchronized data sets, and respective time stamp of the sensor data. The documents were collected by moving the vehicle equipped with sensors and TX2 computer around the campus randomly about half a minute when the synchronization framework had performed.

Firstly, we use the collected data to check the sensor frequency so that we can conclude the synchronization effects. We list the frequency and the time offset of GPS messages in seconds in Fig. 10.



(b)

Figure 10. Original data from GPS: (a) original frequency (b) original time offset of sequent GPS information time stamp

From Fig. 10, we can see that GPS data frequency is not so stable at 10 Hz, and the time interval between sequent messages are not always 0.1 S, which is fluctuating between 0.05 and 0.14 S. The same situation also occurs for LIDARs and cameras.

We analyzed the datasets and plotted the results into several figures, and then explained our results from two aspects: output frequency and synchronization error.

A. Output frequency

Fig. 11 shows statistic result of the collected datasets in the form of frequency histogram. The horizontal axis represents the timestamp (in seconds) of each synchronized set of sensor data, and the vertical axis represents the corresponding frequency (output times in a second) in the entire datasets.

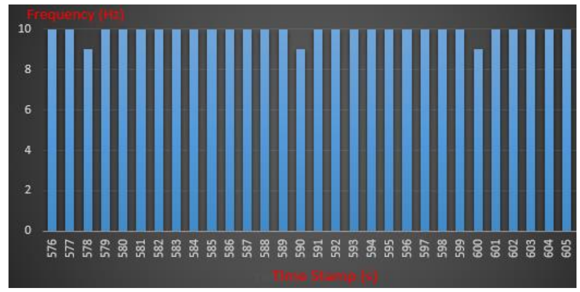


Figure 11. Output frequency of collected data

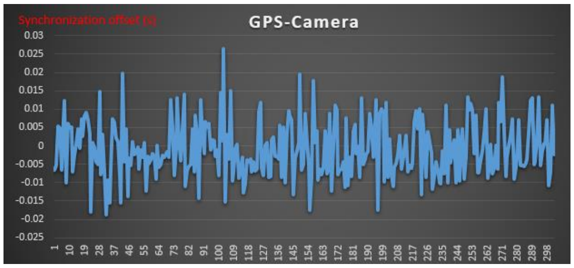
From Fig. 11, we can find that the output frequency of sensor data is stable under our soft synchronization framework. Specifically, we can output the synchronized data at 10 (sometimes 9) times in a second, while frequencies of three sensors are respectively: 30 Hz (ZED camera), 10 Hz (Velodyne LIDAR), and 10 Hz (RTK-GPS devices). Therefore, our synchronization framework guarantees relatively high frequency and low latency.

B. Synchronization Error

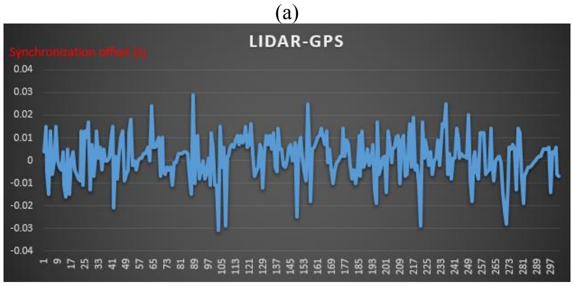
Apart from analyzing the output frequency, we show the synchronization offsets between every two sensors for each synchronized data set, which is illustrated in Fig. 12.

Fig. 12 shows the variation of synchronization error between every two sensors of about 300 sets of synchronized data. The synchronization error was obtained by subtracting the generation time (time stamp) of two sensors. The horizontal axis represents the set number. From these figures, we can find all synchronization offsets between sensors of the entire datasets are less than 30 ms (0.03 s), usually within 10 ms. Since the frequency of the synchronized data is 10 Hz, the interval of two contiguous sets is about 0.1 s, far more than the biggest synchronization offset.

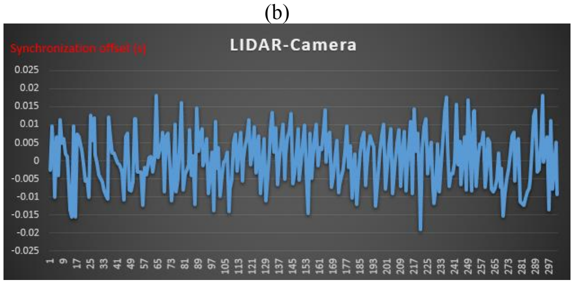
As mentioned above, there are other hardware based synchronization approaches in autonomous localization and navigation applications. We list the comparison of sensor configuration and synchronization performance between the famous datasets and the proposed soft synchronization framework in Table 3. It can be seen that the sensors cannot be fully synchronized in the famous datasets, since the GPS data frequency is even lower than the output frequency, which would cause uncertain synchronization error. However, the proposed framework can synchronize all the sensors, no matter there are applicable for hardware-based triggering or not. Thus, compared to hardware synchronization techniques, proposed synchronization framework try to establish a balance between output frequency and synchronization error.



(a)



(b)



(c)

Figure 12. Synchronization errors between sensors: (a) synchronization error between GPS and camera. (b) Synchronization error between LIDAR and camera. (c) Synchronization error between LIDAR and GPS.

TABLE III. FREQUENCY COMPARISION TO SOME DATASETS

Dataset	Camera	LIDAR	GPS	Output Frequency	Synchronization
Malaga	25 Hz	40 Hz 75 Hz	4 Hz	30 Hz	Hardware
CCSAD	20 Hz	×	1 Hz	3 Hz	Hardware
[19]	30 Hz	×	10 Hz	30 Hz	Hardware
Ours	30 Hz	10 Hz	10 Hz	10 Hz	Software

V. CONCLUSION

We present a soft synchronization framework of multi sensors by ROS and evaluate its synchronization performance by real-time experiments. Our proposed soft synchronization framework can synchronize all kind of sensors used in autonomous localization and navigation, like USB cameras. As for the synchronization performance, our output frequency can be up to 10 Hz that is limited by sensor frequencies. Under our framework and sensor configuration, we can provide 1 cm localization accuracy (RTK-GPS) and a worst-case synchronization error of 30 ms (absolute offset) among the sensors. From our experimental results, we synchronized high-accuracy RTK-GPS devices, and achieved low latency and low synchronization error while guaranteeing high localization accuracy. Based on the proposed soft synchronization framework, sensor synchronization will not depend on complex hardware system and synchronization performance can be guaranteed.

REFERENCES

[1]. Blanco Claraco, et al. "The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario." *International Journal of Robotics Research*, 2014, 33(2): 207-214.

[2]. Blanco, Jose-Luis, Francisco-Angel Moreno, and Javier Gonzalez. "A collection of outdoor robotic datasets with centimeter-accuracy ground truth." *Autonomous Robots*, 2009, 27(4): 327-351.

[3]. Maddern Will, et al. "1 year, 1000 km: The Oxford RobotCar dataset." *International Journal of Robotics Research*, 2017, 36(1): 3-15.

[4]. Carlevaris-Bianco Nicholas et al. "University of Michigan North Campus long-term vision and lidar dataset." *International Journal of Robotics Research*, 2016, 35(9): 1023-1035.

[5]. Geiger Andreas, et al. "Vision meets robotics: The KITTI dataset." *International Journal of Robotics Research*, 2013, 32(11): 1231-1237.

[6]. Guzmán Roberto, Jean-Bernard Hayet, and Reinhard Klette. "Towards Ubiquitous Autonomous Driving: The CCSAD Dataset." *International Conference on Computer Analysis of Images and Patterns*. Springer, Cham, 2015.

[7]. András L. Majdik, Charles Till, and Davide Scaramuzza. "The Zurich urban micro aerial vehicle dataset." *International Journal of Robotics Research*, 2017, 36(3): 269-273.

[8]. Michael Burri, et al. "The EuRoC micro aerial vehicle datasets." *International Journal of Robotics Research*, 2016, 35(10): 1157-1163.

[9]. Ke Sun, et al. "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight." arXiv preprint arXiv:1712.00036 (2017).

[10]. Inkyu Sa, et al. "Build your own visual-inertial odometry aided cost-effective and open-source autonomous drone." arXiv preprint arXiv:1708.06652 (2017).

[11]. Gianpaolo Conte and Patrick Doherty. "An integrated UAV navigation system based on aerial image matching." *Aerospace Conference, IEEE* 2008.

[12]. Mikkel Fly Kragh, et al. "FieldSAFE: Dataset for Obstacle Detection in Agriculture." *Sensors*, 2017, 17(11): 2579.

[13]. Ball, David, et al. "Vision - based Obstacle Detection and Navigation for an Agricultural Robot." *Journal of Field Robotics*, 2016, 33(8): 1107-1130.

[14]. David L Mills, "Internet time synchronization: the network time protocol." *IEEE Transactions on communications*, 1991, 39(10): 1482-1493.

[15]. Edwin Olson, "A passive solution to the sensor synchronization problem." *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), 2010.

[16]. Georgios Litos, et al. "Synchronous image acquisition based on network synchronization." *Computer Vision and Pattern Recognition Workshop*, CVPRW'06., 2006.

[17]. Balazadegan Sarvrood, Yashar, Siavash Hosseinyalamdary, and Yang Gao. "Visual-LiDAR Odometry Aided by Reduced IMU." *ISPRS International Journal of Geo-Information*, 2016.

[18]. Xiu-zhen Wu, et al. "An Improved Monocular ORB-SLAM Method." *2016 International Conference on Artificial Intelligence and Computer Science*, 2016.

[19]. Choi, Yukyung, et al. "All-day visual place recognition: Benchmark dataset and baseline." *IEEE International Conference on Computer Vision and Pattern Recognition Workshops* (CVPRWVPRICE). 2015.