Synchronized Data Acquisition System (SDAS) – a Software Approach for Synchronizing Data Recording from Multiple Sensors

Jacek Piątkowski
Czestochowa University of Technology
Czestochowa, Poland *jacek.piatkowski@icis.pcz.pl*

Łukasz Karbowiak
Czestochowa University of Technology
Czestochowa, Poland *lukasz.karbowiak@icis.pcz.pl*

Filip Depta
Czestochowa University of Technology
Czestochowa , Poland *filip.depta@icis.pcz.pl*

Abstract

The use of many different types of data sensors makes it possible to better represent and understand a given phenomenon. However, the problem becomes the synchronization and fusion of this data. Our goal was to develop a lightweight and flexible system for synchronized data acquisition from various sensors. We designed the Synchronized Data Acquisition System (SDAS), which uses a self-designed Edge Control Protocol (ECP) and Temporal Sample Alignment (TSA) algorithm to synchronize the acquired samples across all the sensors connected to the SDAS. As samples are synchronized during writing data files, we can call that a software-based synchronization. We also conducted tests to validate our solution.

Keywords: Multi-sensor data fusion, Data synchronization, Synchronized data acquisition.

1. Introduction

The employment of many types of sensors to acquire information on phenomena or processes is a frequent and commonly utilized practice and multi-sensor data fusion is an extensively discussed issue in numerous works [1, 2, 3, 4]. Using different sensors usually provides a more accurate depiction and understanding of the process or phenomenon under study. However, this requires the use of well-thought-out solutions for combining and synchronizing various types of data, which are often delivered at not exactly the same moments in time. Inadequate cross-matching of such data can lead to a situation in which the data no longer represent the same event. An example of this would be the Flight Six anomaly of the Ingenuity Mars Helicopter, caused by the loss of a single image from a navigation camera [5]. Luckily, the accident did not happen because the flight control system of the helicopter was built to handle synchronization issues among other things.

In this work, we would like to present a system for Synchronized Data Acquisition System (SDAS) that allows data acquisition from different types of sensors.

2. Related Work

Many multi-sensor data fusion techniques can be found in the literature. Most often, however, these techniques are classified into one of three main categories: (a) *data-level fusion* – combining raw sensor data directly [2, 6], (b) *feature-level fusion* – extracting relevant features from sensor data and merging them [1, 2, 3], and (c) *decision-level fusion* – combining decisions or outputs of individual sensors to make a final decision or inference [2, 6]. Researchers also often turn to less commonly used strategies and create solutions that combine several techniques. Other data fusion strategies also used are: (d) *sensor-level fusion* – a low-level fusion of output data or measurements [7], (e) *model-based fusion* – use of various models (statistical, probabilistic, regression, machine learning) to integrate information from multiple sensors [2, 8], and (f) *context-based fusion* – combines sensor data with additional contextual information, such as scene meaning or surrounding information, to improve the interpretation and reliability of the fused data [9].

3. Synchronized Data Acquisition System (SDAS)

The SDAS was developed to synchronize measurement data acquired from different types of transducers or sensors. The system uses software-implemented synchronization using a data-level fusion strategy. It is a distributed system (see Fig. 1) with one Main Controller (MC) and numerous Sensor Controllers (SCs). Each SC is a separate process that (i) communicates with the device/sensor it controls, (ii) maintains a connection to the MC via Edge Control Protocol (ECP) and (iii) performs its own tasks. The Master Controller (MC) is responsible for synchronizing the operation of SCs and communicating with SCs in the overall system. Each SC can operate independently. Under a running MC, each SC is represented by a unique thread with its own connection between the MC and SC.

The ECP is our lightweight protocol, using TCP as a backbone. Its purpose is to connect the SCs with MC. Then the MC can send commands to the SCs about the times when the recording session should be started, split, or stopped. However, the current time is not sent within the ECP, as each SC runs on the same host as MC. Hence, SCs have access to an accurate OS's clock (in our implementation it is Linux's monotonic clock) and can retrieve precise time from it. Breaking communication with one SC does not stop the operations of other controllers or the entire system. Instead, each SC continues its work until it reaches the timeout common for all SCs. Hence, even if MC crashes, the session always finishes simultaneously on all SCs. When the recording session is started or split, each SC creates a new data file and records samples from its sensor. Each data file name begins with the same timestamp, then its name diverges. Hence, it is easy to group corresponding parts of the session by using these timestamps. Such a naming convention is also convenient for further data sorting or filtering.

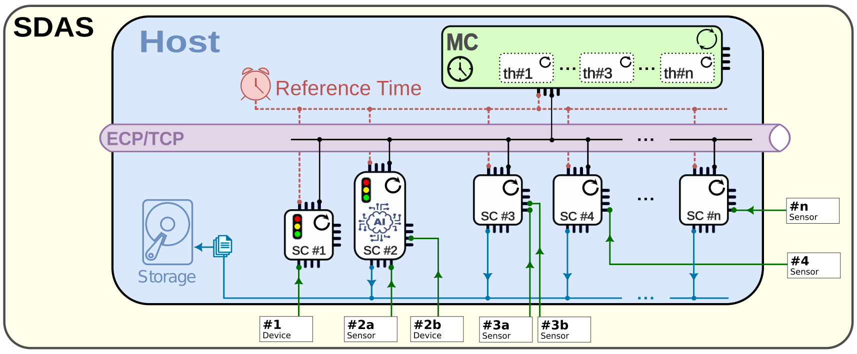

Fig. 1. The scheme of the SDAS

A key component of SDAS is the Temporal Sample Alignment (TSA) algorithm we developed. The primary role of the TSA algorithm is to minimize the accumulated latencies in the recorded data – corresponding samples from multiple sensors should be as close in acquisition time as possible. The TSA algorithm tracks the timestamps of individual samples and adjusts these samples to the expected moments in time. The samples can be adjusted by removing excessive samples or by imputation of samples if there are too few of them. The imputation can be triggered if there are too few samples for too long or if there is a significant number of lacking samples (*significant* here means, beyond the explicitly defined limit). On the other hand, if the number of acquired samples is higher than expected, the sample removal mechanism is triggered. Sample adjustment is performed only when the new sample is acquired.

To facilitate data management and storage the data is sliced into chunks (smaller files). Every SC saves its data files separately from the others, but the beginning timestamps of all files from the same time period are the same.

4. Tests and results

4.1. Testbed

We decided to conduct two experiments to test our TSA algorithm. In the first experiment, we record frames from the two USB cameras (the same models) in two ways. The first way is to start the recording session and not align the frames. In the second way, we let the algorithm perform the frame alignment for the recording session. Then, we evaluate the algorithm's effectiveness for each case. In the second experiment, we align samples from the video camera and LIDAR.

Computer running *Ubuntu 22.04.1*, equipped with *Intel Core i7-9800X* processor, 32 GiB RAM, and *Intel 512 GB SSD 660p Series* disk as a storage device. Two *Microsoft LifeCam Studio* USB video cameras directly connected to the computer. Lidar Velodyne Puck Hi-Res is connected to the computer using an Ethernet interface and communicates via the UDP protocol.

4.2. Methodology

We have modified our camera's and lidar's SC to save each frame's and UDP package timestamp for further analysis. The SC generates the timestamps after the sample-grabbing phase. If a sample is imputed, the timestamp is repeated.

Then, we perform recording sessions with and without the sample alignment algorithm. Each session lasts **at least two hours**.

Based on the session starting timestamp, expected sampling period, frame timestamp, and its number, we calculate the series of *time differences* (t_d) between the expected acquisition time and generated timestamp for each device separately. These series are helpful for visual analysis of the TSA algorithm's mechanisms. If absolute values of the series are used to obtain statistics, it is explicitly declared.

4.3. Results

Despite using a high-quality LIDAR in the study, the developed system makes a significant improvement (Fig. 2.a). The red color indicates the session with the TSA algorithm inactive.

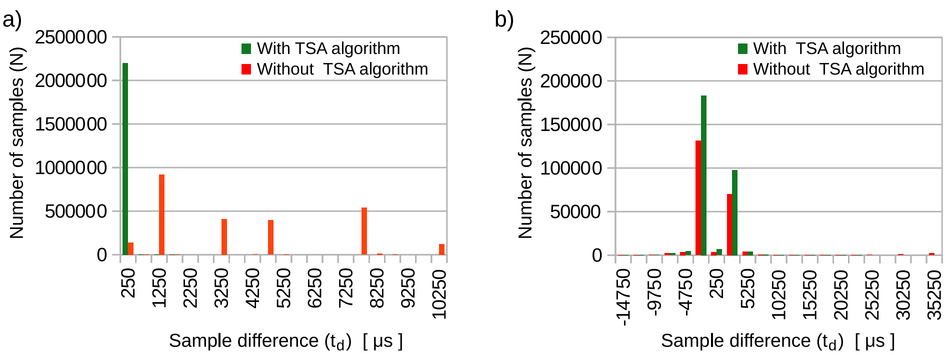

Fig. 2. The histograms of time differences t_d between expected and real time of sample reception: a) LiDAR samples in the LiDAR and camera experiment; b) two cameras experiment

It is noticeable that the values are spread widely in the sample difference domain. This means large differences between the expected times of receiving samples and their actual times. The standard deviation value calculated on the set is 2876.19 μs. The green bars represent the results with the algorithm enabled. There is a visible improvement in sample synchronization. The green bars represent the results with the algorithm enabled. There is a visible improvement in sample synchronization. The largest portion of the samples (99.8%) fall within the narrow range of time inaccuracy from 0 to 500 μs. The value of t_d standard deviation reaches 49.91 μs, which is over 50 times lower than in the case of the inactive TSA algorithm.

When recording camera data, the use of the TSA algorithm also achieves better data synchronization. The spread of time deviations is smaller (see Fig. 2.b) than in the case of data recording without using the TSA algorithm. In this case, we can see a t_d spread ranging between $27000 - 33000$ μs. The standard deviation is 2259.76 μs, and the average value is 5.76559 μs. We can observe that there is less variation in the timestamps of the video frames. The standard deviation slightly decreases in this case to a value of 2249.84 μs, but the average changes to a value of 0.400489 μs.

5. Conclusion

This paper presents a Synchronized Data Acquisition System, designed to synchronize data acquired from different types of sensors. Experimental results show the effectiveness of the author's TSA algorithm in improving sample synchronization. The presented system allows for an increase in the precision of data synchronization in comparison with their recording without the use of the algorithm. Overall, SDAS with the TSA algorithm offers a flexible solution for synchronizing data from different sensors, providing more accurate observations in multi-sensor systems.

References

1. Hyder, A., Shahbazian, E., Waltz, E.: Multisensor Fusion. NATO Science Series II: Mathematics, Physics and Chemistry - Vol. 70 (2002)
2. Chen, C., Jafari, R., Kehtarnavaz, N.: A survey of depth and inertial sensor fusion for human action recognition. Multimed Tools Appl, 3, 4405-4425 (2017)
3. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-Task Multi-Sensor Fusion for 3D Object Detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7337-7345 (2019)
4. Wang, Z., Wu, Y., Niu, Q.: Multi-sensor fusion in automated driving: A survey. IEEE Access 8, 2847–2868 (2019)
5. Grip, H.: Surviving an In-Flight Anomaly: What Happened on Ingenuity's Sixth Flight. https://mars.nasa.gov/technology/helicopter/status/305/surviving-an-in-flight-anomaly-what-happened-on-ingenuitys-sixth-flight/ (2021)
6. Jardine, A, Lin, D., Banjevic, D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing 20 (7), 1483-1510 (2006)
7. Piątkowski, J.: The visualization of shaft vibration using the algorithm of phase-amplitude data interpolation. Journal of Applied Mathematics and Computational Mechanics 15, 137–148 (2016)
8. Kulawik, J., Kubanek, M.: Detection of false synchronization of stereo image transmission using a convolutional neural network. Symmetry 13 (1), 78 (2021)
9. Mansour, M., Davidson, P., Stepanov, O., Piché, R.: Towards semantic slam: 3d position and velocity estimation by fusing image semantic information with camera motion parameters for traffic scene analysis. Remote Sensing 13 (3), 388 (2021)