Precise Synchronization Between LiDAR and Multiple Cameras for Autonomous Driving: An Adaptive Approach

Ajay Kumar Gurumadaiah, Jaehyeong Park, Jin-Hee Lee, JeSeok Kim, and Soon Kwon, *Member, IEEE*

***Abstract*—LiDAR and camera are crucial perception sensors that provide complementary information for object detection in autonomous driving vehicles. However, the fusion of sensor data to achieve efficient object detection requires accurate calibration and precise time synchronization between the sensors. While calibration ensures the geometric relationship between the sensors, and time synchronization ensures that data from both sensors corresponds to the same moment in the real world. Even though various methods have been introduced to achieve accurate calibration between LiDAR and camera, time synchronization between sensors remains relatively unexplored. Poor synchronization between sensors caused by incorrect time stamping process significantly affects data fusion. Therefore, in this paper we present sensor time synchronization for LiDAR and camera data fusion approach, especially in autonomous driving vehicles. It also explores various techniques involved in establishing synchronization between LiDAR and camera. Subsequently, we propose a hardware level time synchronization system, including automatic hardware trigger signal delay estimation to precisely match LiDAR and camera trigger time. Furthermore, validation experiments were conducted to assess the accuracy of the proposed synchronization system in various driving scenarios accompanied by detailed experimental analysis.**

***Index Terms*—Time-synchronization, LiDAR camera fusion, LiDAR and camera calibration, object detection, autonomous vehicle.**

I. Introduction

ENVIRONMENTAL perception is one of the fundamental requirements for autonomous vehicles and involves detecting and recognizing objects in three-dimensional space. In recent research developments, object detection in autonomous driving using multiple sensor-fusion based method proved to be more promising than a single type of sensors [1]. More commonly, autonomous vehicles comprise sensors like LiDAR, camera, radar, and IMU. Among these sensors, LiDAR and camera sensors are often employed for object detection during autonomous driving [2], [3], [4]. Where depth information from the LiDAR and color information from camera are fused to realize the object class, size, and distance. Thereby, the accuracy and reliability of object detection algorithms enhanced by fusion approach [5], [6], [7].

The most crucial part of LiDAR and camera data fusion is calibration and synchronization to align LiDAR sensor measurements with the camera [8], [9], [10]. This accurate alignment ensures that the data from the different modality of sensors are registered in the same coordinate system, representing the environment captured at the same instance. Typically, the calibration process involves determining the transformation parameters between the LiDAR sensor coordinate frame and camera coordinate frame [11]. These transformation parameters include translation, rotation, and scaling factors that accurately map the LiDAR measurements to the camera coordinate frame. In the synchronization process, simultaneous data capturing is established between the LiDAR and camera [12], [13], [14]. Key aspects of the synchronization process are clock synchronization, time stamping, time alignment, and validation. However, most LiDAR and camera fusion approaches focus more on calibration, assuming that captured data were synchronized by default.

However, while it is possible to achieve accurate fusion of LiDAR and camera data in static scenarios with accurate calibration parameters, it becomes highly challenging in dynamic scenarios where the ego-vehicle or objects around it are in motion, leading to significant translational error caused by improper synchronization between sensors. Without proper synchronization, sensors capture data at different instances of the environment and thereby causes significant errors in the data fusion process [15]. Furthermore, poor synchronization accuracy directly influences other dependent processes such as environment perception and localization [16]. Therefore, sensor synchronization is one of the critical real-time requirements in autonomous driving to improve the accuracy of object detection and localization through sensor fusion [6]. To address synchronization issues, several approaches have been introduced in the literature. Some well-known approaches include hardware-triggering synchronization, software synchronization, network synchronization, and clock synchronization [16], [17], [18], [19], [20].

Manuscript received 29 May 2024; revised 24 July 2024; accepted 11 August 2024. Date of publication 21 August 2024; date of current version 15 August 2025. This work was supported in part by the Daegu Gyeongbuk Institute of Science and Technology R&D Program funded by the Korea Government (Ministry of Science and ICT) under Grant 24-IT-01 and in part by Korea Innovation Foundation (INNOPOLIS) funded by the Korea Government (Ministry of Science and ICT) under Grant 2023-TB-RD-0017 and Grant 2023-DG-RD-0041-02. *(Ajay Kumar Gurumadaiah and Jaehyeong Park are co-first authors.) (Corresponding author: Soon Kwon.)*

Ajay Kumar Gurumadaiah is with the Department of Mining and Explosive Engineering, Missouri University of Science and Technology, Rolla, MO 65401 USA.

Jaehyeong Park, Jin-Hee Lee, JeSeok Kim, and Soon Kwon are with the Division of Automotive Technology, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea (e-mail: soonyk@dgist.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3444780.

Digital Object Identifier 10.1109/TIV.2024.3444780

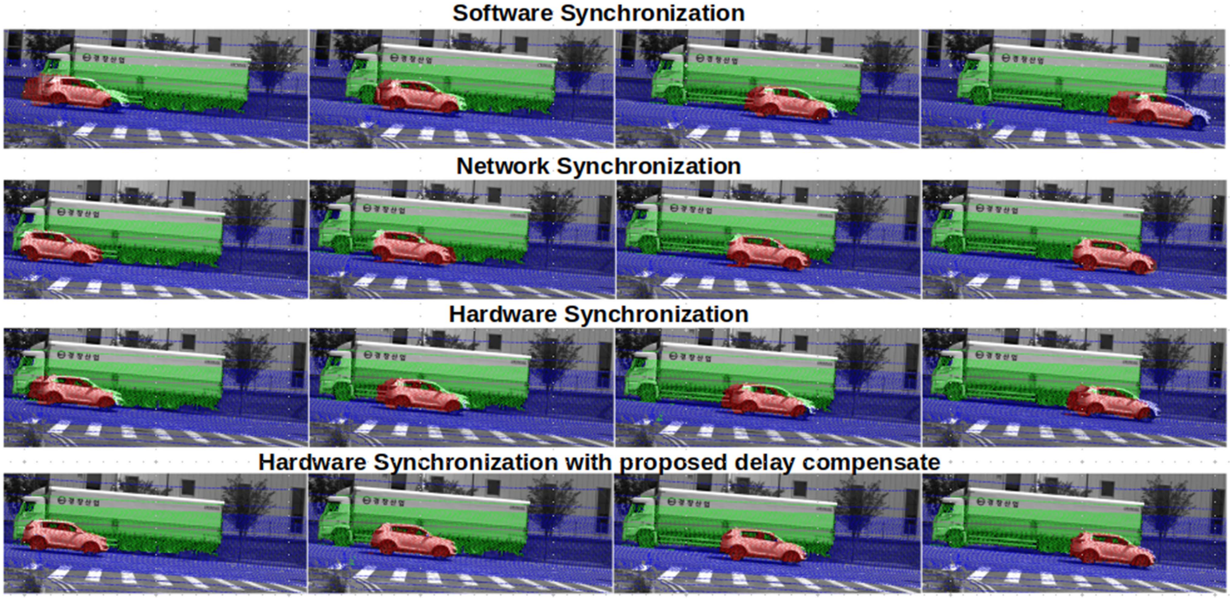

Fig. 1. LiDAR and Camera data fusion data with frames captured in different synchronization approach.

In hardware-triggering synchronization, cameras supported with hardware triggers are synchronized, which has been proven to be very promising in achieving precise synchronization. A similar approach was used in the famous benchmark KITTI dataset to establish synchronization between LiDAR and camera [21]. In this study, the authors used a mechanically rotating Velodyne scanner with reed switches attached at the bottom to trigger the camera when the scanner was facing forward. However, it should be noted that the mentioned method is limited to old-fashioned rotating scanners, and it is not feasible to employ this approach with currently available modern LiDAR as they do not support external mechanical movement. In the case of software synchronization, using the time stamp method can solve the asynchronous problem by fusing data by referring to the timestamp. However, the processing time overhead makes the solution unsatisfactory for the real-time performance requirement [22]. For network synchronization, Precision Time Protocol (PTP) and Network Time Protocol (NTP) are proposed to perform synchronization between multiple sensors [23]. Though network synchronization synchronizes the sensors to capture data with respect to common reference clock source rather than the sensor's internal clock.

Thus, there is a need for synchronization techniques that specifically address existing issues and achieve precise synchronization between LiDAR and camera for object detection in autonomous driving. Therefore, in this work, we propose a hardware level synchronization technique to facilitate object detection from LiDAR and camera data fusion data. Our significant contribution to this work can be summarized as follows:

1) We introduce hardware-level synchronization between LiDAR and camera, along with automatic time delay estimation and compensation.
2) We conducted a detailed experimental analysis to prove the significance of the proposed synchronization method compared to existing method as shown in Fig. 1. Additionally, we evaluated its real-time performance in various autonomous driving scenarios.

II. Related Work

Precisely perceiving the surrounding environment is one of the key requirements for achieving safe and reliable autonomy. Generally, perception systems need to maintain continuous awareness of the surrounding environment, either by processing 2D images acquired from cameras or by processing 3D point cloud data from LiDAR sensor. However, both sensor capabilities come with their own limitations, and it is possible to overcome limitations by fusing information from both sensors [3]. Additionally, various benchmark datasets have been released for autonomous driving, commonly adopting the LiDAR and camera fusion approach for sensing the surrounding environment. However, such methods are easily affected by sensor misalignment due to the hard association between points and pixels established by calibration matrices and precise synchronization. In the article [4], the authors mentioned that sensor synchronization is a crucial process for achieving high accuracy in sensor fusion. Since LiDAR and camera sensors have different operating frequencies, it becomes challenging to align the data from both sensors representing the environment at a specific time. Meeting precise synchronization requirements for modern autonomous systems, which utilize a wide range of different sensors with varying triggering and time-stamping mechanisms, poses a significant challenge. Existing synchronization methods may not fully address these requirements. Overall, the synchronization methods currently in existence can be broadly classified into hardware triggering synchronization, software synchronization, network synchronization and clock synchronization [6].

A. Hardware Triggering Synchronization

Hardware synchronization is a method used to achieve precise synchronization between multiple sensors by sending external triggering signals from primary devices to secondary devices [24]. In some cases, an external synchronization control unit is used to send the triggering signal to multiple devices to initiate data acquisition from the sensors. More commonly, cameras equipped with hardware triggering use this approach to establish synchronization between the sensors. However, despite the promising performance of the hardware level synchronization approach in the case of multiple cameras, its applicability is rarely demonstrated in synchronizing LiDAR and cameras due to the lack of a standard approach to enable hardware triggering protocols and interfaces. Even though the hardware level synchronization approach introduced by the authors in the KITTI benchmark dataset [21] proves to be promising in the specific setup, its applicability is limited to externally rotating mechanical LiDAR and is impossible with modern commercially available LiDAR's. Moreover, externally mounted reed switches are mechanical devices that can be susceptible to damage, leading to potential failure or inaccuracies in the synchronization process. Also, external reed switches for synchronization limit the flexibility to adjust the synchronization parameters dynamically. This can be problematic in scenarios where real-time adjustments are necessary to improve synchronization accuracy.

B. Software Synchronization

Even though software-based synchronization approach offers several advantages over hardware synchronization approach in terms of flexibility and applicability, there are certain limitations that make it challenging to meet the real-time requirement in autonomous driving. The synchronization framework implemented in the ROS system is well-known software-based synchronizer and is commonly used in real-time sensing applications [14]. Additionally, several methods have been introduced to improve the efficiency of ROS software-based synchronization [1]. Despite these enhancements, ROS software-based synchronization is limited by poor latency in data transmission and processing. As ROS operates on distributed system architecture, the communication between different nodes and the processing of data can introduce latency, which may affect the real-time synchronization of LiDAR and camera data. Additionally, the performance of ROS-based synchronization relies heavily on the computational capabilities of the system and network conditions, which can further impact accuracy.

C. Network Synchronization

Network synchronization methods typically rely on a centralized clock or timestamping mechanisms to establish a common time reference across all devices. These methods then synchronize sensor data frames based on the associated timestamps and ensure that the sensor data from different devices accurately correlated [16]. More commonly, Precision Time Protocol (PTP) and Network Time Protocol (NTP) are used to establish network synchronization among the sensing devices. PTP is a network protocol specifically designed for clock synchronization in distributed systems [25]. It uses hardware timestamps and the exchange of synchronization messages to achieve sub-microsecond accuracy. And it operates in a primary-secondary architecture, where the primary clock distributes information to secondary clocks. NTP, on the other hand, is a widely used protocol for synchronizing clocks over the internet. It operates in a hierarchical architecture, where a set of time series provides time information for clients. NTP can achieve synchronization within milliseconds or even microseconds, depending on the network conditions [22]. However, network-based synchronization relies on the availability and stability of the network infrastructure. Any network disruptions or fluctuations can affect the synchronization accuracy and introduce inconsistencies in the sensor data. Additionally, while network-based synchronization methods can achieve sub-millisecond to microsecond accuracy, they may not offer the same level of precision as hardware-based synchronization methods.

D. Clock Synchronization

Clock synchronization between multiple sensing devices refers to the process of aligning the timestamps of multiple sensor data to enable accurate sensor fusion. This synchronization can be archived through various methods including hardware synchronization, software synchronization, and a combination of both. In this synchronization method, synchronization among the clocks enables one clock to correct its time to match with another clock [22]. However, this synchronization may not be applicable to LiDAR and camera clock synchronization.

III. Proposed Approach

The main objective of the proposed sensor synchronization is to ensure that the data acquired from different sensors corresponds to the same event and has the same timestamp. While there are well-known benchmark datasets [21] that provide synchronized data, they involve a series of post-processing steps and do not guarantee that the sensor-sampled data has the timestamp corresponding to the same event. However, real-time perception tasks in autonomous robots require highly precise synchronization between LiDAR and camera. Mis-synchronized sensor data contributes to temporal noises in the sensor fusion algorithm and leads to estimation errors in both translational and rotational parameters. Motivated by the requirements of existing autonomous vehicle perception tasks and the limitations of the existing system, we proposed a synchronization architecture for LiDAR and camera, along with automatic synchronization delay correction. This section provides a more detailed description of the proposed approach, which is divided into the following subsections: Data collection, Synchronization, Time delay error modelling, and Multiple camera synchronization.

A. Data Collection and Calibration

1) Data Collection: In this work, the Hesai Pandar 64 channels LiDAR [26] and Flir 2D camera [27] are deployed in the vehicle sensor kit, as shown in Fig. 2. The sensor kit is attached to the rooftop of the vehicle to gain the optimal field of view. It includes three cameras and a 360° LiDAR. The internal clocks of all cameras and the LiDAR sensor are synchronized to the PC primary clock using PTP. This clock synchronization ensures

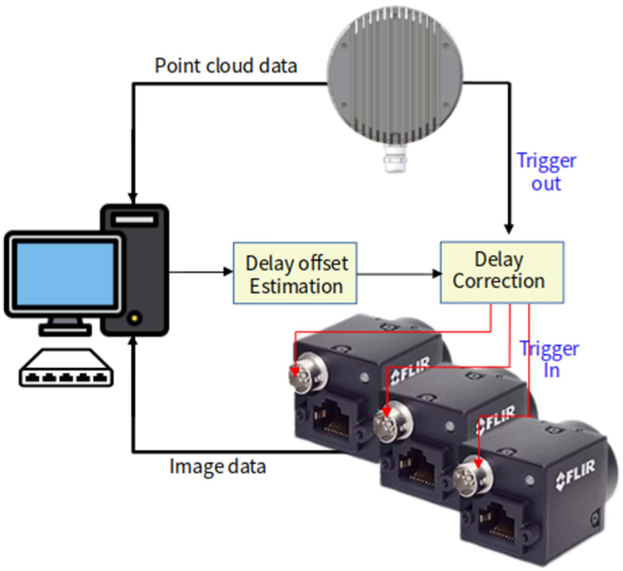

Fig. 2. Overall architecture design of proposed synchronization system.

that all sensors capture data with respect to a common time reference.

2) Calibration: The coordinate frame of LiDAR is denoted as l, and camera frame is denoted as c. We have utilized our previous work for LiDAR to camera calibration [28] to estimate the extrinsic parameters between LiDAR and camera. After estimating the camera intrinsic and LiDAR to camera extrinsic parameters, 3D points are projected onto the image using a distortion free projective transformation given by the pinhole camera model, as shown in the (1).

$$P_i = C_{\text{intrinsic}}[R/t]P_l \tag{1}$$

Where, P_l is a 3D point from the LiDAR point cloud, and P_i is the projected 2D pixel in the image plane. In (2), $C_{\text{intrinsic}}$ is the camera intrinsic matrix obtained using the checkerboard board calibration method. *[R/t]* represents the LiDAR to camera extrinsic matrix, which includes rotation and translation matrices describing the change of coordinates from the LiDAR to the camera coordinate system based on the calibration approach discussed in the [28].

$$C_{\text{intrinsic}} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The camera intrinsic matrix $C_{\text{intrinsic}}$ is composed of focal lengths f_x and f_y in pixel units, and the principal point (c_x, c_y), which is close to the image center. Meanwhile, $[R/t]$ represents the joint rotation and translation matrix product of a projective transformation and a homogeneous transformation. The 3-by-4 projective transformation maps 3D points represented in camera coordinates to 2D pixels in the image plane and is represented in normalized camera coordinates as shown in (5).

The homogeneous transformation, encoded by extrinsic parameters *R* and *t*, represents the change of basis from the LiDAR coordinate system *l* to the camera coordinate system *c*. Therefore given the representation of the point *P* in LiDAR coordinates, P_l we obtain P_c in the camera coordinate system as follows,

$$P_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} P_l,$$

Where the homogeneous transformation is composed of *R*, a 3-by-3 rotation matrix, and *t*, a 3-by-1 translation vector.

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Therefore,

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \\ Z_l \\ 1 \end{bmatrix} \tag{4}$$

Combining the projective transformation and the homogeneous transformation in (4), we obtain the projective transformation that maps 3D points in the LiDAR coordinates into 2D points in the image plane and normalized camera coordinates:

$$Z_c \begin{bmatrix} x^1 \\ y^1 \\ 1 \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ Z_c/Z_c \end{bmatrix}$$

Putting the intrinsic and extrinsic components together, we can expand the projection equation $P_i = C_{\text{intrinsic}}[R/t]P_l$ as shown in (5).

$$\begin{bmatrix} u^1 \\ v^1 \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_l \\ Y_l \\ Z_l \\ 1 \end{bmatrix} \tag{5}$$

Where, (u^1, v^1) represents the projected 3D points in the image coordinates.

B. Synchronization

In the proposed approach, synchronization between the LiDAR and camera was established using the hardware triggering approach. Due to the operating frequency of the LiDAR being much slower than that of the camera, we chose the LiDAR as the reference for time synchronization. The LiDAR sends a trigger signal to the camera to enable capturing within a specific LiDAR FOV. The basic structure of the proposed synchronization method is mainly consists of three modes: Data acquisition, Data alignment, and Time delay correction.

1) Frame Capturing: In the data acquisition mode, the internal clocks of the LiDAR and camera are synchronized with IEEE PTP [26] to maintain a common time source among the multiple sensors. By default, the LiDAR and camera stamp the captured data with their respective internal clock. However, it is not possible to align data from multiple sensors by referring to the timestamp from their internal clock sources. Therefore, establishing a common clock source among the sensors using PTP facilitates the alignment of frames based on timestamps during the fusion process. With Precision Time Protocol (PTP) network, clock synchronization of LiDAR and camera systems is attainable through a shared timebase.

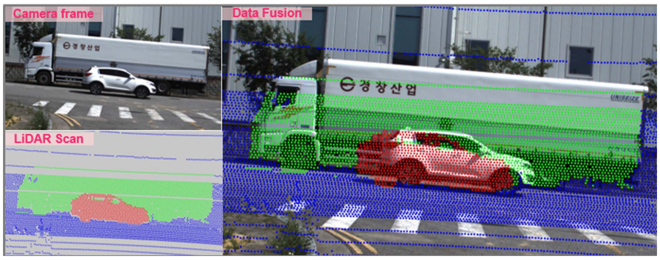

Fig. 3. Projection of LiDAR points on image plane using transformation matrix.

Assuming t^L represents the timestamp for a LiDAR data point, and t^C represents the timestamp for a corresponding camera frame referenced from the PTP common clock source. Also the association of timestamps along with the sensor data can be written in vector form as (6) and (7).

$$L_s = \left[(X_i \quad Y_i \quad Z_i), \quad t_i^L \right] \tag{6}$$

Where L_s represents the single LiDAR scan, with i points, and each point carries a PTP time stamp t_i^L.

$$C_f = \left[(u \quad v), \quad t_i^C \right] \tag{7}$$

Where C_f is the camera frame with PTP timestamp t_i^C for the i frame. Unlike LiDAR, the timestamp is included for each frame, not for each pixel. After capturing the data from the sensors synchronized using the PTP network, the LiDAR point cloud is aligned onto the corresponding image frame using the projection matrix described below as shown in Fig. 3.

$$p_d, t_d = \sum_{n=i}^{n} (C_f - \text{project}(L_s)) \tag{8}$$

Where *project* maps Lidar points on to image plane using the transformation matrix described in the (5).

$$p_d, t_d = \sum_{i=1}^{n} \left(\left[(u \quad v), \quad t_i^C \right] - \left[(u^1 \quad v^1), \quad t_i^L \right] \right)$$

$$p_d = \sum_{i=1}^{n} \frac{\sqrt{(u - u^1)^2 + (v - v^1)^2}}{n}$$

$$t_d = \sum_{i=1}^{n} \frac{t_i^L - t_i^C}{n}$$

p_d is the average projection error and t_d is the observed average time delay between the LiDAR point timestamp and camera frame timestamp over n frames. Where n represents the number of frames acquired during specific interval, in our case it is set every 2 minutes with 1200 frames approximately.

Algorithm 1: Adaptive Dynamic Time Delay Estimation and Trigger Offset Correction.

```
Input: Image frame $C_f = [(u, v), t_i^C]$ and LiDAR scan
       $L_s = [(X_i, Y_i, Z_i), t_i^L]$
Output: Trigger delay offset $\Delta t_d$ Initialisation:
        $\Delta t_d$=initial offset delay, $p_{thr}$=projection error
        threshold, $t_{thr}$=time gap threshold
for each LiDAR scan $L_s$ do
    Trigger camera with $\Delta t_d$
    Project $L_s$ onto $C_f$ & Compute $p_d, t_d$
    Case: Static object
    if $p_d \geq p_{thr}$ then
        Re do calibration...
    end if
    Case: Dynamic object
    if $t_d \geq t_{thr}$ then
        Adjust the camera trigger offset
        $\Delta t_d = \pm t_d$
    end if
end for
return $\Delta t_d$
```

2) Reference Time Measurement: Without a specific reference point, accurately estimating p_d and t_d may not always be possible. This challenge arises from the difference in the nature of data acquisition between LiDAR and camera sensors. While LiDAR captures a comprehensive 360° view of the environment in a single scan, the camera can only capture limited portions of it. Moreover, the timestamps associated with LiDAR scans and camera images represent different instances: the camera timestamp marks when the image frame was captured, whereas the LiDAR timestamp indicates when the LiDAR completed one scan. As a result, finding a precise LiDAR point timestamp that aligns with the camera frame timestamp for comparison becomes difficult, affecting the accuracy of p_d and t_d.

Hence, in the proposed method, we identify the specific instantaneous LiDAR reference point with respect to the center of the camera image by projecting LiDAR points onto the image, as illustrated in Fig. 4. After projection, the vertical(ϕ) and horizontal (θ) LiDAR angles of the points projects onto or near the center of the image are calculated as follows:

$$\phi = atan2(Y, X)$$

$$\theta = atan2(Z, \sqrt{(X_2, Y_2)})$$

Further, p_d and t_d corresponding points are estimated for points closer to the image center. With our experimental vehicle, the LiDAR 18th vertical channel aligns with the camera image center at horizontal sweeping angles $\theta_{\text{middle}} = 180.0^0$, $\theta_{\text{left}}! = 358.4^0$, and $\theta_{\text{right}} = 276.8^0$ for the middle, left, and right cameras, respectively. These specific horizontal sweeping angles are then used as a reference point to initiate the trigger signal from the LiDAR to the camera.

3) Time Delay Correction: Initially, the camera is set with an initial triggering offset Δt_d to commence the camera exposure upon receiving the LiDAR trigger signal specific to the field of view that aligns with the respective camera. Upon receiving the camera frame C_f, the LiDAR scan L_s is projected onto the camera using the projection matrix. Even though the LiDAR being able to send a trigger signal directly to the camera, there will always be a significant offset between the camera and LiDAR capture times. This offset is caused by the FOV mismatch between the camera and LiDAR, as well as internal processes within the camera and signal transmission delays. In the proposed method, the LiDAR reference time and camera reference time are continuously matched and compensation parameter are calculated in real time to adjust the camera capturing time, as indicated in the algorithm's sequence of processes.

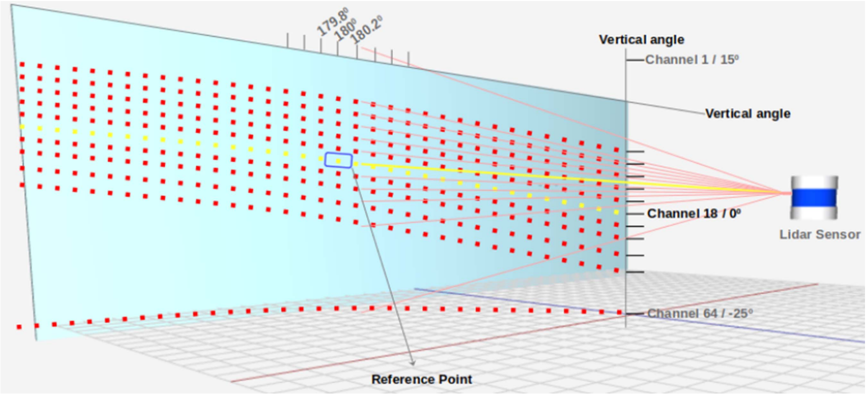

Fig. 4. Estimating the reference point in the LiDAR scan for each associated camera.

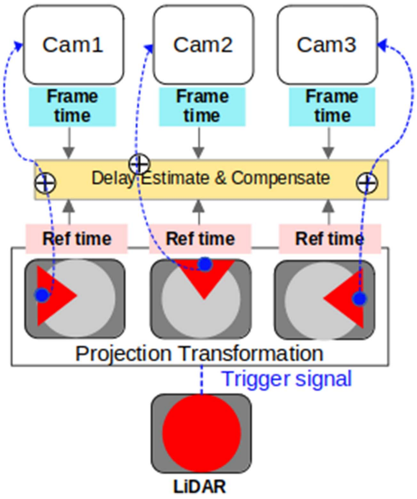

Fig. 5. Synchronization architecture for multiple camera.

In the case of multiple cameras, the trigger signal from the LiDAR is generated for each camera at different LiDAR FOVs, and the delay is computed based on the three LiDAR reference points with respect to each camera. As shown in Fig. 5, trigger signals from the different LiDAR FOVs are generated and sent to the camera. The correction module modulates the trigger signal, adding the required delay compensation estimated from the proposed approach as flow depicted in Algorithm 1.

IV. Experimental Evaluation

To verify the effectiveness of the proposed algorithm, extensive experimentation was performed with different scenarios on real road in Technopolis town, located Daegu city, South Korea. The sensor installation setup on the vehicle is shown in Fig. 6. The sensor kit installed on the top of the vehicle includes a Pandar 64 3D LiDAR (Hesai Technology, Shanghai, China) with 360° horizontal field of view and 40° vertical field, placed in the middle of the sensor kit. Three Blackfly S Gig E machine vision cameras (Teledyne Flir, Wilsonville, Oregon, U.S) are positioned in the middle, left, and right of the sensor kit. Throughout the experiments, the LiDAR sensor operated at 10Hz frequency, while the camera operated at different frequencies, up to a maximum of 30Hz.For each synchronization method, LiDAR and camera data were captured for 45 minutes, comprising 27,000 frames across 4 trials, where “trials” refers to instances of the data collection process. The system and sensors were restarted between each trial by turning off the input power.

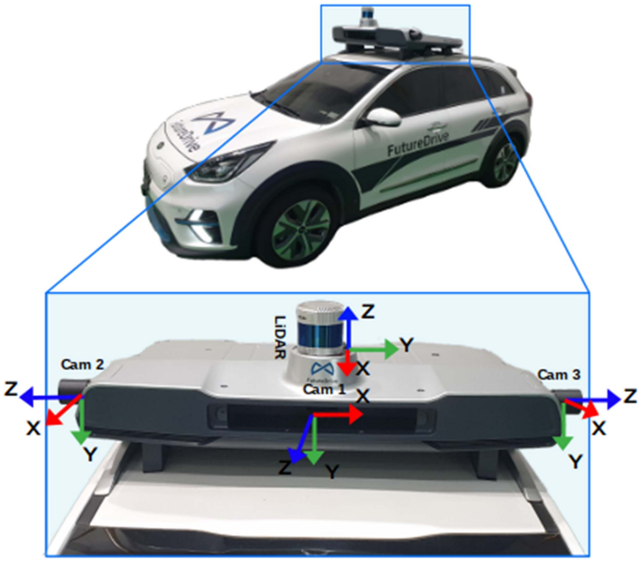

Fig. 6. Vehicle sensor kit installed with LiDAR and camera.

The efficiency of synchronization between the LiDAR and camera was observed during object movement and at different distances. Hence, we experimented with multiple scenarios involving dynamic objects with varying speeds and distances. Initially, stopped the ego-vehicle and recorded moving vehicle at multiple constant speeds from a consistent distance to understand the effect of synchronization with respect to moving vehicle. Furthermore, ego-vehicles drove at a constant speed and recorded other vehicles moving at various speeds and distances. Additionally, we conducted a comparison analysis

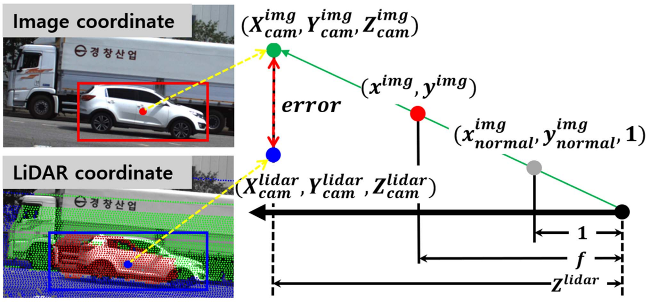
Image coordinate
$(X^{img}_{cam}, Y^{img}_{cam}, Z^{img}_{cam})$
error
(x^{img}, y^{img})
$(x^{img}_{normal}, y^{img}_{normal}, 1)$
LiDAR coordinate
$(X^{lidar}_{cam}, Y^{lidar}_{cam}, Z^{lidar}_{cam})$
1
f
Z^{lidar}

Fig. 7. Projection of point in image coordinate to world coordinate.

between various time synchronization approaches, including software synchronization, network synchronization, and hardware level synchronization, to verify the significance of the proposed method.

A. Evaluation Metrics

In the sensor fusion approach, synchronization between LiDAR and camera sensors is crucial for efficient environmental perception in many applications. Assessing the synchronization system is essential to ensure accuracy and reliability within the context of the application domain. In the context of autonomous driving, achieving efficient synchronization between the sensors is particularly challenging due to numerous factors such as capturing time, frame drop resulting from network overload, and projection errors.

1) Time Gap: The time gap t_d represents the difference in time between the LiDAR reference timestamp and the corresponding camera frame timestamp.

$$t_d = t_i^L - t_i^C \tag{9}$$

Where, t_i^L and t_i^C represents the LiDAR point and camera frame timestamps, respectively.

2) Frame Loss Percentage: Frame loss within the LiDAR and cameras synchronization system is one of the significant factors directly affecting the system's accuracy. These instances of frame loss often occur due to factors such as network overload, timing mismatches, temperature effects, power fluctuations, and processing limitations. The percentage of frame loss is estimated for every 2 minutes using the following equation:

$$\text{Frame}_{\text{loss}} = \frac{\text{Desired}_{\text{frames}} - \text{Recieved}_{\text{frames}}}{\text{Desired}_{\text{frames}}} \times 100 \tag{10}$$

3) Projection Error in World Coordinate: Projection error between the LiDAR and camera data involves estimating the Euclidean distance between the 3D points in the world coordinate system and converted 3D coordinates of 2D pixels in image coordinates. Converting camera pixels to world coordinates involves utilizing the camera calibration parameters and depth information provided by the LiDAR sensor as shown in the Fig. 7.

$$\begin{bmatrix} x^{\text{img}}_{\text{normal}} \\ y^{\text{img}}_{\text{normal}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x^{\text{img}} \\ y^{\text{img}} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X^{\text{img}}_{\text{cam}} \\ Y^{\text{img}}_{\text{cam}} \\ Z^{\text{img}}_{\text{cam}} \end{bmatrix} = \begin{bmatrix} x^{\text{img}}_{\text{normal}} * Z^{\text{lidar}}_{\text{cam}} \\ y^{\text{img}}_{\text{normal}} * Z^{\text{lidar}}_{\text{cam}} \\ 1 * Z^{\text{lidar}}_{\text{cam}} \end{bmatrix}$$

Where $Z^{\text{lidar}}_{\text{cam}}$ is obtained from the LiDAR point data and the extrinsic rotation matrix R and translation vector t, as shown below:

$$\begin{bmatrix} X^{\text{lidar}}_{\text{cam}} \\ Y^{\text{lidar}}_{\text{cam}} \\ Z^{\text{lidar}}_{\text{cam}} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X^{\text{lidar}} \\ Y^{\text{lidar}} \\ Z^{\text{lidar}} \\ 1 \end{bmatrix}$$

$$p_{d3D} = \frac{1}{N} \sqrt{\sum\nolimits_{i=1}^{N} (P^{\text{img}}_{\text{cam}} - P^{\text{lidar}}_{\text{cam}})^2} \tag{11}$$

As illustrated in Fig. 7, the projection error between the LiDAR and camera frames is estimated in the 3D coordinate frame. First, the selected LiDAR point $P^{\text{lidar}}(X^{\text{lidar}}, Y^{\text{lidar}}, Z^{\text{lidar}})$ is transformed to the camera coordinate $P^{\text{lidar}}_{\text{cam}}(X^{\text{lidar}}_{\text{cam}}, Y^{\text{lidar}}_{\text{cam}}, Z^{\text{lidar}}_{\text{cam}})$ and the corresponding pixel in the camera image is restored in 3D. The 3D projection error is calculated using (11).

4) Projection Error in Pixel Coordinate: The projection error is computed between the center of the bounding box coordinates of the detected object from the projected LiDAR points $P^{\text{lidar}}_{\text{img}}(X^{\text{lidar}}_{\text{img}}, Y^{\text{lidar}}_{\text{img}})$ and the point $P^{\text{img}}(X^{\text{img}}, Y^{\text{img}})$ in the camera image coordinate using the (12). The equation calculates the root mean square of the Euclidean distances between the corresponding LiDAR and camera points, providing a measure of the accuracy of the projection alignment. A lower projection error indicates a more accurate alignment between the LiDAR and camera frame.

$$p_{d2D} = \frac{1}{N} \sqrt{\sum\nolimits_{i=1}^{N} (P^{\text{lidar}}_{\text{img}} - P^{\text{img}})^2} \tag{12}$$

B. Synchronization Approach

1) Software Synchronization: For software synchronization, a well-known and commonly used ROS-based time synchronization method was adopted and tested with recorded camera and LiDAR data. The ROS system synchronizes the data according to the order received in the node. The callback functions registered in the ROS system trigger whenever there is LiDAR and camera data. However, this method only considers the data arrival time at the computer, instead of capture time; hence, it does not guarantee that the received sensor data represents the same state of the world. Also, as the number of sensors increases, the system introduces critical time deviations due to poor transmission latency. Fig. 8 shows the fusion data of LiDAR and camera captured using the ROS-based software-based time synchronization method. The left side of the images presents the static and dynamic scenarios, while the right side of the images demonstrates the synchronization effect on moving vehicle in different directions, with different speeds.

Table I presents the projection error and time gap error estimated by projecting LiDAR points onto the camera image using the projection matrix obtained through the calibration process. The mean and variance of the distance error linearly increase with respect to the speed of the moving object, while the time gap between the frames remains the same.

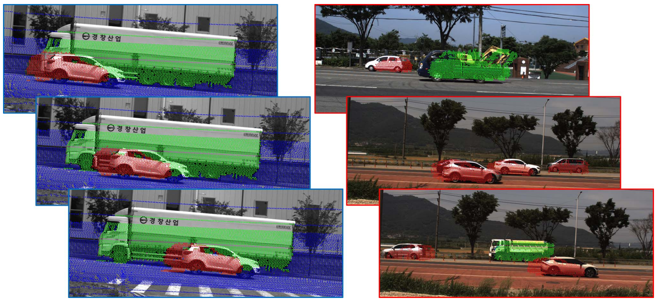

Fig. 8. Fusion of LiDAR and camera frames captured in **software** synchronization mode.

TABLE I
EXPERIMENTAL EVALUATION OF A **SOFTWARE** SYNCHRONIZATION APPROACH

No. of Trials	Projection Error	40 kmh	60 kmh	80 kmh	Time Gap	Frame Loss(%)
Trial 1	p_{d3D}	**50.36**	25.36	36.68	33.49	17.0
	p_{d2D}	**50.04**	19.08	40.89		
Trial 2	p_{d3D}	42.08	48.32	**52.05**	51.09	16.3
	p_{d2D}	29.12	26.35	**52.03**		
Trial 3	p_{d3D}	28.91	39.25	**59.02**	55.32	15.5
	p_{d2D}	35.39	22.10	**61.02**		
Trail 4	p_{d3D}	45.33	48.11	42.01	48.07	17.2
	p_{d2D}	40.02	20.11	36.99		

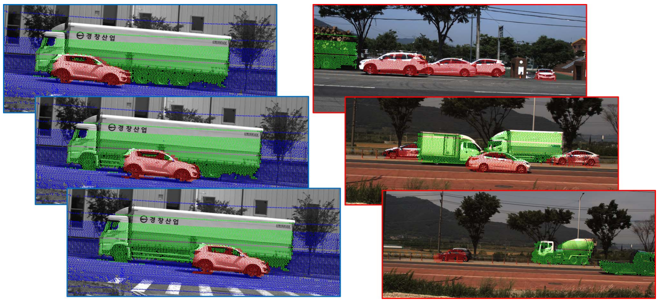

Fig. 9. Fusion of LiDAR and camera frames captured in **network** synchronization mode.

2) Network Synchronization: In case of network synchronization, the LiDAR and camera are synchronized using IEEE 1588 PTP, with the processor set as the primary and the camera and LiDAR set as secondary. Despite adjusting the camera's internal clock according to the LiDAR primary clock, the capturing time of each sensor is not synchronized. However, it is possible to align the closest camera frame with the LiDAR frame by checking the timestamp associated with each LiDAR camera frame. Although increasing the frame rate of the camera would be beneficial for capturing much closer frame, a random time gap error still affects the synchronization accuracy. Moreover, in case of multiple camera synchronization, higher bandwidth increases the likelihood of data transmission delays, which further amplifies the synchronization errors. Fig. 9 presents the projection of LiDAR points onto camera images, and Table II presents the statistical analysis of projection error, time gap, and frame loss observed in network synchronization mode.

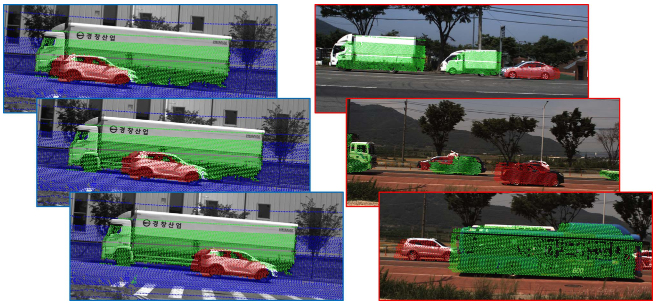

Fig. 10. Fusion of LiDAR and camera frames captured in **hardware triggering** synchronization mode.

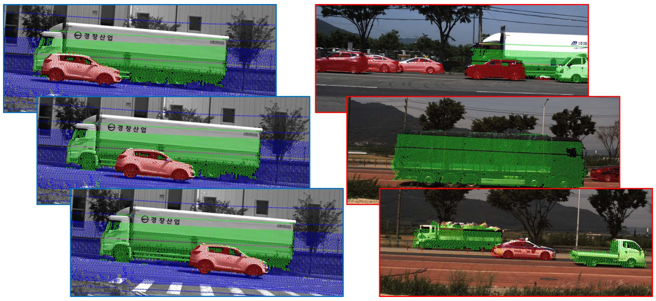

Fig. 11. Fusion of LiDAR and camera frames captured in **proposed** synchronization mode.

3) Hardware Trigger-Based Synchronization: As discussed in the previous, the hardware-based triggering approach emerges as the most promising alternative for addressing the issues highlighted in software and network-based synchronization methods. In this approach, the LiDAR and camera are integrated into a hardware synchronization system, where the LiDAR sends an external trigger signal to the camera to initiate the capture. While hardware triggering approach offers several advantages, it also comes with certain limitations that directly affect on synchronization accuracy. Achieving high precision in synchronization time is challenging due to the latency between the trigger signal and the actual capturing time of the camera, which impacts the alignment accuracy. Although it is possible to mitigate latency issues by configuring constant compensation parameters, these parameters may fluctuate over time due to factors such as processing overhead and sensor temperature. Therefore, a hardware triggering-based synchronization approach with dynamic delay compensation would be ideal for achieving precise LiDAR and camera synchronization in various applications.

The statistical analysis presented in Table III illustrates the projection error and time gap error estimated from the frames captured over a duration of 30 mins. Fig. 10 shows projection of LiDAR points on camera image caputued in hardware triggering mode. The estimated mean and variance of the projection error consistently increase over time, particularly in relation to the speed of the moving object.

4) Hardware Triggering With Delay Compensation: Addressing the limitations of the synchronization approach based on hardware triggering, the proposed method introduces automatic delay compensation to mitigate dynamic latency issues arising from various factors. In the proposed method, the time gap between the LiDAR and camera frames is observed at

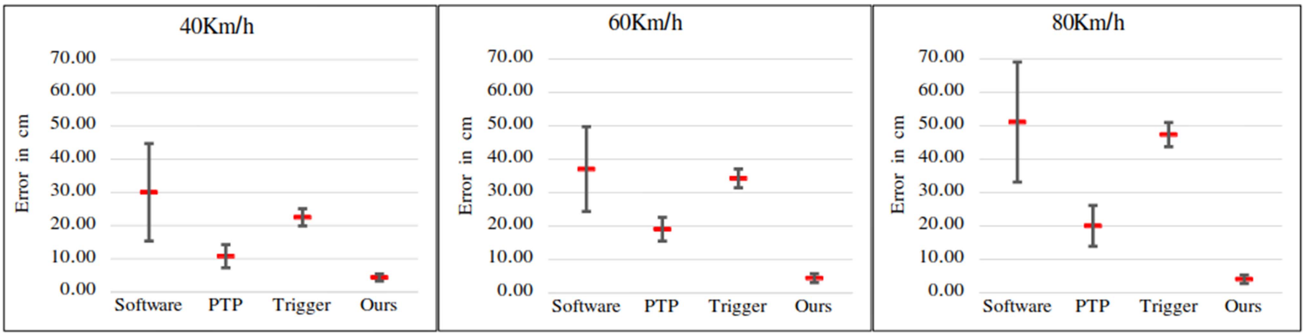

Fig. 12. Comparison of the mean and variance of projection error when surrounding objects, captured at different speeds using four synchronization approaches.

TABLE II
EXPERIMENTAL EVALUATION OF A **NETWORK** SYNCHRONIZATION APPROACH

No. of Trials	Projection Error	40 kmh	60 kmh	80 kmh	Time Gap	Frame Loss(%)
Trial 1	p_{d3D}	11.12	18.93	21.10	10.72	24.6
	p_{d2D}	10.78	19.08	20.02		
Trial 2	p_{d3D}	15.08	20.35	**22.33**	11.08	23.8
	p_{d2D}	12.07	**21.02**	**22.89**		
Trial 3	p_{d3D}	16.08	**21.66**	20.58	12.74	24.5
	p_{d2D}	12.95	**22.33**	**23.05**		
Trail 4	p_{d3D}	13.58	17.98	19.57	11.57	24.6
	p_{d2D}	11.25	19.24	20.14		

TABLE III
EXPERIMENTAL EVALUATION OF A **HARDWARE-TRIGGERING** SYNCHRONIZATION APPROACH

No. of Trials	Projection Error	40 kmh	60 kmh	80 kmh	Time Gap	Frame Loss(%)
Trial 1	p_{d3D}	22.55	35.17	**47.06**	25.35	13.7
	p_{d2D}	22.50	34.28	**47.33**		
Trial 2	p_{d3D}	21.98	34.83	45.66	25.35	14.8
	p_{d2D}	22.01	33.29	**46.74**		
Trial 3	p_{d3D}	20.74	35.24	45.89	25.33	12.9
	p_{d2D}	21.58	34.16	**47.21**		
Trail 4	p_{d3D}	21.35	33.61	44.67	25.37	12.4
	p_{d2D}	22.04	34.55	**46.62**		

TABLE IV
EXPERIMENTAL EVALUATION OF A **PROPOSED** SYNCHRONIZATION APPROACH

No. of Trials	Projection Error	40 kmh	60 kmh	80 kmh	Time Gap	Frame Loss(%)
Trial 1	p_{d3D}	5.24	4.92	**5.96**	0.51	10.8
	p_{d2D}	4.37	4.44	4.06		
Trial 2	p_{d3D}	5.39	4.59	5.29	0.54	11.3
	p_{d2D}	4.98	4.52	4.96		
Trial 3	p_{d3D}	4.32	4.84	5.13	0.50	9.7
	p_{d2D}	4.65	4.81	5.02		
Trail 4	p_{d3D}	4.88	4.67	**5.59**	0.57	11.8
	p_{d2D}	4.29	4.35	**5.86**		

regular intervals, and precise compensation parameters are computed to adjust the synchronization accordingly.

Table IV presents the results of the experimental analysis conducted with the proposed hardware triggering with delay compensation approach, aimed at improving synchronization accuracy. Delay compensation is computed whenever there is a time gap between LiDAR and camera frames exceeding a predefined threshold. The camera trigger signal is then generated based on the estimated compensation parameter to initiate the capturing. Fig. 11 shows the projection output of LiDAR points on camera image captured in proposed synchronization mode.

V. RESULTS & DISCUSSION

Derived from the experimental analyses conducted under critical scenarios that significantly impact the fusion of LiDAR and camera data for surrounding environment perception in autonomous driving, the proposed approach proves promising in addressing various challenges. The ideal fusion LiDAR and camera data was achieved with the proposed approach, irrespective of the surrounding object speeds and distances, without requiring any additional hardware interfaces.

Furthermore, the synchronization module can identify and rectify the time gap between the LiDAR and multiple cameras in real-time. The quantitative comparisons of the mean and variance of the projection error plot in Figs. 12 and 13 shows a very small errors in the proposed hardware-based triggering method compared to the other three synchronization approaches. While the software synchronization-based method proves to be more prone to projection error due to the random selection of LiDAR and camera frames based on the ROS software callback function. In the case of network synchronization, though theoretically timestamp-based frame matching appears promising, its performance is affected in real-time applications, especially when the system involves multiple sensors, due to network overload and transmission delays. In the case of hardware triggering-based synchronization approach, it is found to be a potential solution to address synchronization issues. However, variations in the trigger signal transmission and small delays in data transmission from the sensor contribute to error accumulation in the projection. Finally, in the proposed synchronization approach, real-time monitoring and delay correction significantly improved synchronization accuracy by reducing projection errors to less than 10 cm.

VI. CONCLUSION

To address the challenge of LiDAR and multiple camera synchronization for the benefit of sensor fusion-based algorithms in the autonomous driving domain, a precise time

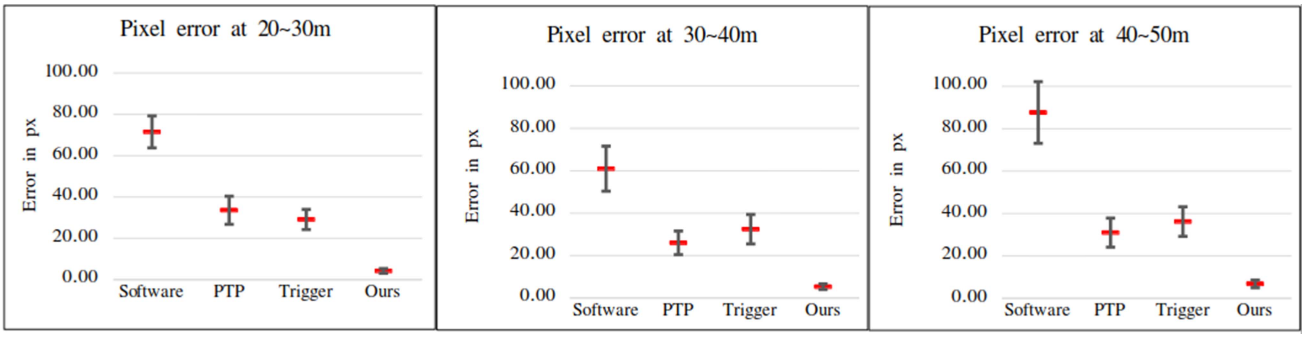

Fig. 13. Comparison of the mean and variance of projection error when surrounding objects, captured at different distances using four synchronization approaches.

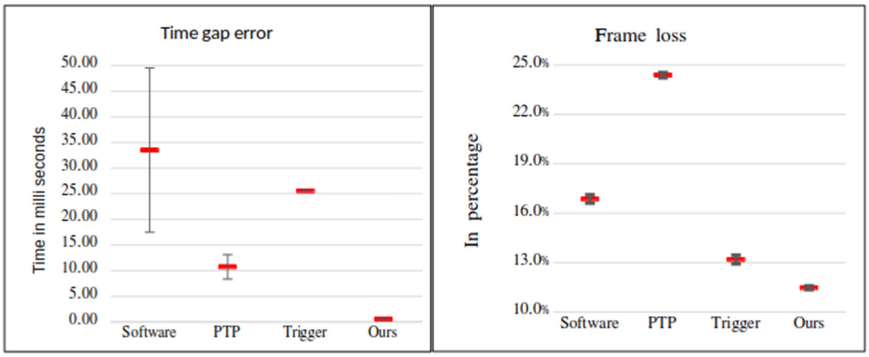

Fig. 14. Comparison of the estimated frames time gap and frame loss among four synchronization approaches.

synchronization approach is introduced. The proposed approach offers detailed insight into the practical challenges involved in establishing precise synchronization among multiple sensors used for environment perception in autonomous driving. Building upon existing synchronization approaches, our method aims to eliminate key limitations that directly impact the overall accuracy of data fusion. Based on a hardware synchronization architecture, the proposed real-time delay estimation and correction algorithm significantly improve fusion accuracy by reducing the projection error to less than 10 cm. Extensive experimental analysis performed to evaluate the significance of our proposed synchronization approach along with existing synchronization approaches. Comparing with the existing synchronization approach, the significance of the proposed method is demonstrated through key challenging scenarios that would significantly impact synchronization accuracy. Additionally, we provided an analysis of frame loss rates among the other four synchronization approaches in comparison with our proposed approach as shown in Fig. 14. Overall, the practical applicability of our proposed system is demonstrated in an urban scenario with a real vehicle equipped with Level 4 autonomous driving technology.

References

[1] H. Hu, J. Wu, and Z. Xiong, "A soft time synchronization framework for multi-sensors in autonomous localization and navigation," in *Proc. 2018 IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2018, pp. 694–699.

[2] N. Kaempchen and K. C. J. Dietmayer, "Data synchronization strategies for multi-sensor fusion," 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:15483874

[3] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[4] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *Proc. 2014 IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1836–1843.

[5] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7337–7345.

[6] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, "Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles," *Sensors*, vol. 19, no. 20, 2019, Art. no. 4357. [Online]. Available: https://www.mdpi.com/1424-8220/19/20/4357

[7] H. Yoon, M. Jang, J. Huh, J. Kang, and S. Lee, "Multiple sensor synchronization with therealsense RGB-D camera," *Sensors*, vol. 21, no. 18, 2021, Art. no. 6276. [Online]. Available: https://www.mdpi.com/1424-8220/21/18/6276

[8] Y. Li et al., "DeepFusion: LiDAR-camera deep fusion for multi-modal 3D object detection," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17182–17191.

[9] S. Schneider, M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Fusing vision and LiDAR - synchronization, correction and occlusion reasoning," in *Proc. 2010 IEEE Intell. Veh. Symp.*, 2010, pp. 388–393.

[10] H. G. Norbye, "Camera-LiDAR sensor fusion in real time for autonomous surface vehicles," M.S. thesis, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2019. [Online]. Available: https://folk.ntnu.no/edmundfo/msc2019-2020/norbye-lidar-camera-reduced.pdf

[11] P. An et al., "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D–3D point correspondences," *Opt. Exp.*, vol. 28, no. 2, pp. 2122–2141, Jan. 2020. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-28-2-2122

[12] L. Yang, Q. Jia, R. Wang, J. Cao, and W. Shi, "HydraView: A synchronized 360°-view of multiple sensors for autonomous vehicles," in *Proc. IEEE 2020 Int. Conf. Connected Auton. Driving*, 2020, pp. 53–61.

[13] H. Usami, H. Saito, J. A. Kawai, and N. Itani, "Synchronizing 3D point cloud from 3D scene flow estimation with 3D LiDAR and RGB camera," *Electron. Imag.*, vol. 2018, 2018, Art. no.426. [Online]. Available: https://api.semanticscholar.org/CorpusID:69443235

[14] A. Noda, Y. Yamakawa, and M. Ishikawa, "Frame synchronization for networked high-speed vision systems," in *Proc. IEEE SENSORS*, 2014, pp. 269–272.

[15] E. Olson, "A passive solution to the sensor synchronization problem," in *Proc. 2010 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1059–1064.

[16] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul./Aug. 2004.

[17] M. Faizullin, A. Kornilova, and G. Ferrer, "Open-source LiDAR time synchronization system by mimicking GNSS-clock," in *Proc. 2022 IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun.*, 2022, pp. 1–5.

[18] K. Yuan, L. Ding, M. Abdelfattah, and Z. J. Wang, "LiCaS3: A simple LiDAR–camera self-supervised synchronization method," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3203–3218, Oct. 2022.

[19] L. Fridman, D. E. Brown, W. Angell, I. Abdić, B. Reimer, and H. Y. Noh, "Automated synchronization of driving data using vibration and steering events," *Pattern Recognit. Lett.*, vol. 75, pp. 9–15, May 2016, doi: 10.1016/j.patrec.2016.02.011.

[20] A. Rangesh, K. Yuen, R. K. Satzoda, R. N. Rajaram, P. Gunaratne, and M. M. Trivedi, "A multimodal, full-surround vehicular testbed for naturalistic studies and benchmarking: Design, calibration and deployment," 2019, *arXiv:1709.07502*.

[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.

[22] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. IEEE INFOCOM*, 2004, vol. 1, Art. no. 574.

[23] D. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[24] P. Hyla, "Multi camera triggering and synchronization issue : Case study," *J. KONES*, vol. 23, no. 3, pp. 193–200, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:189955402

[25] S. Liu et al., "The matter of time–A general and efficient system for precise sensor synchronization in robotic computing," 2021, *arXiv:2103.16045*.

[26] Hesai Pandar, "Pandar 64 lidar sensor, Hesai technology (NASDAQ: HSAI)," Shanghai, Palo Alto and Stuttgart, 2021. [Online]. Available: https://www.hesaitech.com/downloads/#pandar64/UserManual.pdf

[27] FLIR Balck Fly S, "Teledyne technologies," Wilsonville, OR, USA, 1978. [Online]. Available: https://www.flir.eu/products/blackfly-s-gige/?vertical=machinevision&segment=iis

[28] G. A. Kumar, J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon, "LiDAR and camera fusion approach for object distance estimation in self-driving vehicles," *Symmetry*, vol. 12, no. 2, 2020, Art. no. 324. [Online]. Available: https://www.mdpi.com/2073-8994/12/2/324

Ajay Kumar Gurumadaiah received the bachelor's and master's degrees from the Department of Computer Science, University of Mysore, Mysore, India, in 2012, and the Ph.D. degree from Chung-Ang University, Seoul, South Korea, in 2018. From 2018 to 2023, he was with the Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea, as a Postdoctoral Researcher in the Future Automotive Division. His research interests include autonomous vehicles (including UAVs and UGVs), SLAM, map matching, IMU odometry, sensor fusion, and deep learning.

Jaehyeong Park received the B.S. and M.S. degrees in electronic control engineering from the Daegu University of Korea, Gyeongsan, South Korea. He is currently a Researcher with the Division of Automotive Technology, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, South Korea. His research interests include deep learning-based scene perception, autonomous driving, and sensor fusion.

Jin-Hee Lee received the M.S. and Ph.D. degrees in computer and information engineering from the Inha University of Korea, Incheon, South Korea. She is currently a Senior Researcher with the Division of Automotive Technology, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea. Her research interests include deep learning-based scene perception, autonomous driving, and sensor fusion.

JeSeok Kim received the Ph.D. degree from the Department of Automotive Engineering, Hanyang University, Seoul, South Korea, in 2015. He is currently a Senior Researcher with the Division of Automotive Technology, and Director of AI.Drive Research Lab, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea. He is also a Board Member of the Institute of Embedded Engineering of Korea. His research interests include path planning, vehicle control, and autonomous vehicle.

Soon Kwon (Member, IEEE) received the B.S. degree in electric & electronic engineering from Korea University, Seoul, South Korea, in 2003, and the M.S. degree in electronic engineering from Seoul National University, Seoul, in 2006. Since 2006, he has been with the Daegu Gyeongbuk Institute of Science and Technology, where he is currently a Principal Researcher with the Division of Automotive Technology. His research interests include deep learning-based scene perception and autonomous driving, real-time stereo vision, parallel processing, and visual sensing system.