Real-time Camera and Video Streaming Through Optimized Settings of Ethernet AVB in Vehicle Network System

Byoungman An[1], Youngseop Kim[1*]
[1] Electronic and Electrical Engineerin, Dankook University
152, Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, 16890, Korea
[e-mail: yesbman@naver.com, wangcho@dankook.ac.kr]
[*]Corresponding author: Youngseop Kim

Received March 16, 2021; revised June 27, 2021; accepted July 8, 2021;
published August 31, 2021

Abstract

This paper presents the latest Ethernet standardization of in-vehicle network and the future trends of automotive Ethernet technology. The proposed system provides design and optimization algorithms for automotive networking technology related to AVB (Audio Video Bridge) technology. We present a design of in-vehicle network system as well as the optimization of AVB for automotive. A proposal of Reduced Latency of Machine to Machine (RLMM) plays an outstanding role in reducing the latency among devices. RLMM's approach to real-world experimental cases indicates a reduction in latency of around 41.2%. The setup optimized for the automotive network environment is expected to significantly reduce the time in the development and design process. The results obtained in the study of image transmission latency are trustworthy because average values were collected over a long period of time. It is necessary to analyze a latency between multimedia devices within limited time which will be of considerable benefit to the industry. Furthermore, the proposed reliable camera and video streaming through optimized AVB device settings would provide a high level of support in the real-time comprehension and analysis of images with AI (Artificial Intelligence) algorithms in autonomous driving.

Keywords: Audio Video Bridge, AVB, Video Streaming, Autonomous, In-vehicle Network

A preliminary version of this paper was presented at ICONI 2020, and was selected as an outstanding paper. This version includes more technical details and supporting experimental results on AVB system in Vehicle Network Environment. This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2C2009717). We express our thanks to all members of ICONI 2020 Special Issue Editorial Board who checked our manuscript.

http://doi.org/10.3837/tiis.2021.08.018 **ISSN : 1976-7277**

1. Introduction

The application of automobiles differs significantly in the use cases and technical requirements of the vehicle assembler compared to the applications of other types of electronic products. Current technology specifications such as AVB (Audio Video Bridge) have arisen to address these varied needs. [1] This presentation starts with the background for the technological introduction into automotive network based on the features of Ethernet and in vehicle requirements. In automotive bridge LAN (Local Area Networks), the media is rapidly distributed in the automotive world which real-time multimedia streams necessitate the collection of specialized functions defined in different specifications. The stream of sound and video through the vehicle's interactive network demands the stable in-vehicle transmission technology as LVDS (Low Voltage Differential Signaling) system. Car assembly manufacturers and suppliers are keen to define standards for real-time audio and video processing in order to satisfy their customers' requirements. A simple multimedia transmission through various IEEE 802 network modes provides operational benefits. The AVB standard provides a transport mechanism for real-time camera, video, UI (User Interface) streams to all vendors developing vehicles, reducing the effort of creating units of work between different networks. [2] In this paper, we suggest an up-to-date overview of it as well as describes the basic of the key element of eAVB; in addition, the paper gives the reason that AVB is an excellent candidate for next generation automotive infrastructures.

Many technologies are being developed to study real-time network delay reduction for video transmission between equipment inside a vehicle. [3] Most of the studies describe variations of the architecture and test environment for various scenarios, from basic car networks with two end points, one controller to redundant in-vehicle networks with diverse end-nodes, and switches. To interconnect systems, there is a need for larger bandwidth with determinism and flexibility. Also, many subsystems are safety critical because the automotive industry has to deal with a lot of existing networking algorithms. Considering the vehicle's safety, it is influential to move from using numerous technologies to use the general standard. To increase safety as well as minimize the complexity, it is profitable to commit on a set of automotive networking technologies that can be adopted in most of the in-vehicle applications generally found in an automotive system. [4], [5] The present paper is focusing on investigations about the latency of AVB networking since it exerts a significant influence on communicating different automotive signals on time. Communication signals without timing boundaries can disable functions that cause serious situations for the automotive system. [6], [7], [8]

The suggestion of this paper is to provide a novel algorithm to use an optimized concept with the AVB networking. Optimizing and accelerating the GPU and effectively utilizing the configuration of the video encoder and AVB are expected to be used in many different applications in the vehicle. In addition, it can operate with low latency control and shape the video stream flow depending on the specific use case requirements.

This paper aims to offer a novel system design that uses an optimized framework for AVB in-vehicle network latency reduction. In Section 2, we describe the AVB standard with key concepts. In Section 3, we propose an algorithm for reducing the latency video stream of in-vehicle system and explain the detailed mathematical rationale. The experimental findings of the proposed design obtained by various studies are presented in Section 4. Finally, we present conclusions and future works with an outlook in Section 5.

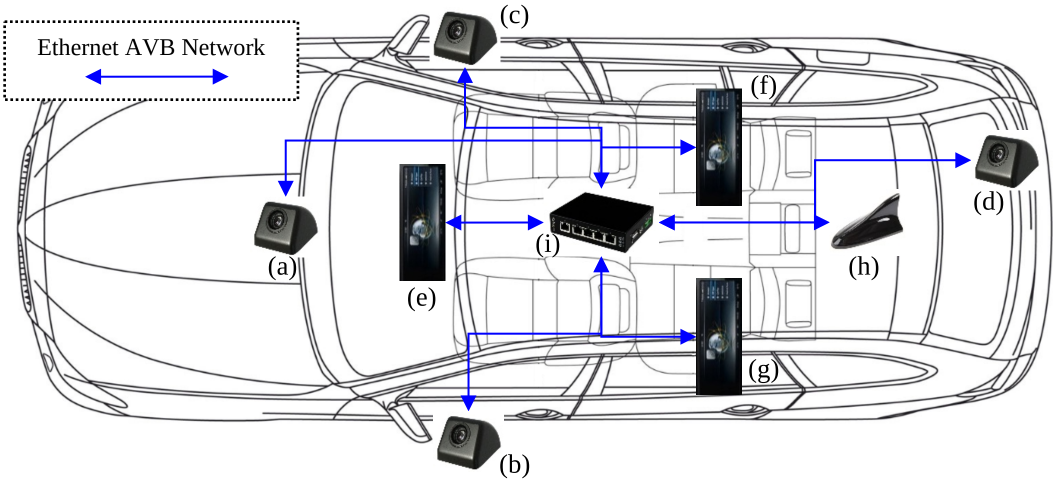


Fig. 1. The AVB system model design between in-vehicle network multimedia devices, (a) 1st device, (b) 2nd device, (c) 3rd device, (d) 4th device, (e) 5th device, (f) 6th device, (g) 7th device, and (i) 8th device (network hub).

2. Audio Video Bridge

2.1 A Standard of AVB

AVB is the most current latest mainstream technology for vehicle network connectivity, but there are few products, end-nodes device, network hub that support it. Especially, it has become a necessary technology as in-vehicle cameras for various processes increase to satisfy safety with the increasing level of autonomous driving. As an endpoint, various devices inside the vehicle are possible. Any digital equipment can be utilized as AVB such as speakers, cameras, digital signal processors, etc. For fully advanced digital communication systems in vehicles where additional data may also permit diverse control streams, progressive solutions such as AVB capable of satisfying traffic ratings and multiple streams are required. [9]

AVB supports accurate, low-latency networks in time to satisfy QoS (Quality of Service). **Fig. 1** shows the elements to visualize the AVB architecture design considering the hardware and software components involved in in-vehicle communication. Each packet's stream class packet guarantees a minimum latency for video data. [10] An example AVB system is shown in **Fig. 2**. The AVB network consists of three types of components:

1) AV (Audio and Video) Bridges

The AV Bridge is a relay device that conforms to IEEE 802.1BA. It acts as a switch that receives and forwards IEEE 802.1BA compliant data to the destination device.

2) End stations

An end station can be a talker, listener, or both. A talker is the end station that is the transmitter, the producer, or the source of the AVB system. A listener is an end station that is the receiver, the consumer, or the destination of the AVB system.

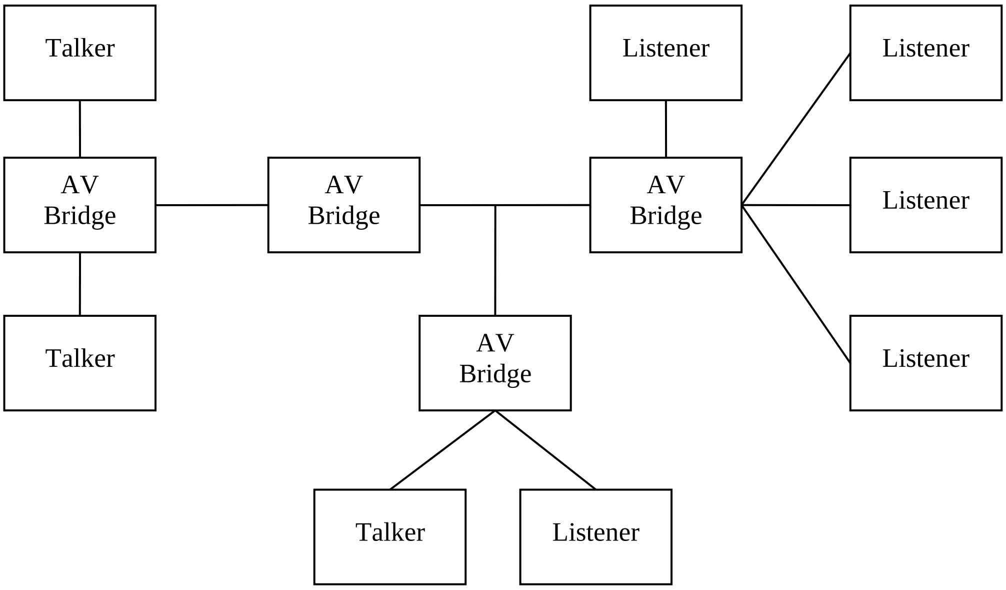


Fig. 2. Example high level AVB Network.

3) Ethernet LANs (Local Area Network)

Individual LANs interconnect the Bridges, Talkers and Listeners.

AVB has defined the standard with three additional sub-standards below to support time-sensitive in-vehicle applications in asynchronous in-vehicle networks, as shown in **Fig. 3**.

- IEEE 802.1 Qav (Queue and delivery rules) [11]
  IEEE 802.1Q defines a series of standards for separating time-sensitive and non-sensitive traffic into distinct traffic situations. The network egress port buffers are divided into several queues, each of which is allocated to a particular class. This design separates low and high-priority traffic. In addition, all network egress ports have a credit-based shaping function to avoid burst activity.

- IEEE 802.1 Qat (A protocol of bandwidth reservation) [12]
  It notifies reservation requests and defines a protocol for reserving resources for audio/video streams. It allocates buffers within the switch to provide the required QoS.

- IEEE 802.1 AS (A protocol of time synchronization) [13]
  This standard describes and defines precise time synchronization of network nodes to the global time.

- IEEE 1722 (A protocol of stream transmission)
  The recommended AVB stream transport protocol is IEEE 1722. From simple control data to large amounts of audio and video, it encapsulates them in Ethernet frames and adds presentation timestamps that correspond to the time that it takes to process the data.

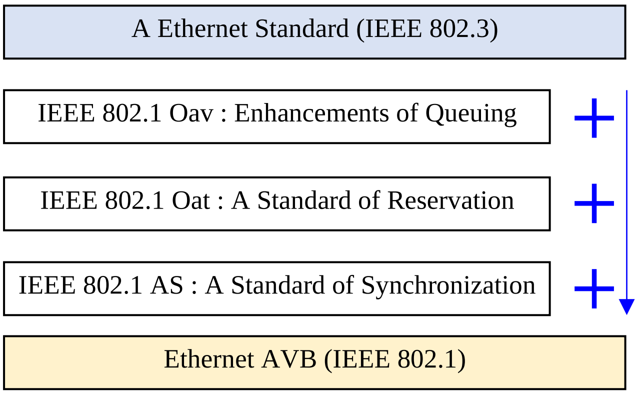

A Ethernet Standard (IEEE 802.3)
IEEE 802.1 Oav : Enhancements of Queuing
IEEE 802.1 Oat : A Standard of Reservation
IEEE 802.1 AS : A Standard of Synchronization
Ethernet AVB (IEEE 802.1)

Fig. 3. A standard of Ethernet AVB.

For multimedia data such as the audio and the video, a reference clock is sent from the transmitter of the stream to the receiver using the 1722 norm. The presentation time of audio and video is simultaneously provided by default for all listeners. This can lead to synchronous replay.

The AVB standard defines talkers, listeners, and bridges. AVB's talker and listener share gPTP (Generalized Precision Time Protocol) messages with each other. To compensate for this time error due to network communication, it is periodically transmitted. MMRP (Multiple MAC Registration Protocol) and MVRP (Multiple VLAN Registration Protocol) can also be used by the listener to register VLANs (Virtual Local Area Network) and its MAC (Media Access Control). The talker reserves the bandwidth required for communication by utilizing MSRP (Multiple Stream Reservation Protocol). The talker broadcasts a message to the network, and the bridge estimates the resources available on the outgoing ports to allocate the required resources. The results are sent to the listener. When resources are low, the listener sends an error message so that correction can be made. When the reservation is complete, the talker uses the transport protocol standard IEEE 1722 to transmit the streaming data and streams the required data in real-time from the bridge-based local network. When a communication with the IEEE 802.1 Qav protocol is sent over a high-speed network, per 7 hops, stream type A is transferred in 2 milliseconds. Class B flow is conveyed within the vehicle with forwarding and queuing laws within 50 meters every 7 hops. For QoS support, each node within AVB must support AVB standard, so AVB is not backwards compatible in terms of QoS, but it is compatible with standard Ethernet. The AVB transport protocol timestamps the transmitted video stream and activates it via global synchronization on the device; thus, the processing of data transfer runs concurrently. IEEE 802.1Qat blocks loss of parcels and IEEE 802.1Qav uses a system for traffic shaping to ensure timely traffic distribution. [14]

2.2 Related Works

For in-vehicle AVB communication, a variety of technologies have been developed and implemented. Since it ensures a significant amount of QoS for real-time multimedia transmission in a variety of circumstances, it is expected to become a promising technology in automotive communication networks. In-vehicle networks, on the other hand, typically use a single traffic class due to the complexity of configuring the stream reservation domain. Jitter restricting is often considered by AVB. To address these issues, an optimized Ethernet AVB system architecture with a single SR (Stream Reservation)-Class and time-triggered scheduling was proposed to satisfy QoS specifications while also providing many other advantages. [15]

Moreover, evaluating the availability of AVB is considered the next generation of in-vehicle network communications. [16] We can design a system that integrates the vehicle's antenna and tuner and check the potential performance of the system head unit. In [17], it describes how to enable fault tolerance in audio and video bridging Ethernet networks. The particular emphasis is placed on the specification of service interfaces that enable a layered approach using arbitrary redundant control protocols; also, The WCDs (Worst Case end-to-end Delays) of AVB flows in a TSN (Time-Sensitive Networking) are determined using a network calculus-based approach. [18] An analytical design of AVB is described in [19]. CBQ (Credit-Based Queuing) shaper algorithms (CBQ) and PQ (Priority Queuing) are evaluated as well as compared with legacy Ethernet technologies. It is practicable to figure upper bounds for all time sensitive traffic network streams with CBQ. Timing analysis techniques that reflect the capabilities of future AVB architectures such as static routing and stream reservation, fixed topologies and real-time applications are presented in [20]. It provides an evidence that the methodology provides much tighter boundaries than the calculations specified in AVB standards and previous work.

Recently, many studies on AVB's efficiency have been published. The increasing load on in-vehicle networks is due to the rapid development and adaption of ADAS (Advanced Driver Assistance Systems) and autonomous driving. Traditional protocols such as controller area networks, local interconnected networks, and FlexRay are beginning to reveal limitations as in-vehicle networks are now required to deliver a rapid data flow and constrained delay for real-time operation. The article introduces multiple Ethernet traffic scheduling standards based on TSN technology and proposes a heuristic-based Ethernet scheduling method. [21] The SRP (Stream Reservation Protocol) is a critical piece of work since it allows network resources to be reserved while maintaining the requisite QoS. The study investigates the termination and consistency of SRP using UPPAAL. It verifies that SRP lacks these qualities and suggests a new protocol called CSRP (Consistent Stream Reservation Protocol), which it officially verifies using UPPAAL. [22] The IEEE 802.1 standard expands the design area for automobile networks, resulting in a plethora of options. The study provides information on studies that were generated primarily for automobile networks, as well as open topics that require more investigation. [23] Furthermore, the research combines the sensor simulation software PreScan as the test platform with the vehicle's hardware-in-the-loop to link with reality sensors and virtual scenes (HIL). In a virtual road environment, simulation of vehicle auxiliary systems and sensors is useful and efficient. Industrial-grade Ethernet's high reliability satisfies automobile manufacturers' demands and Tier 1 safety criteria for original equipment function. The high-speed Ethernet AVB serves as a crucial data convergence route because the ADAS is equipped with many camera lenses. Advanced ADAS functionalities such as surround view, obstacle detection, and lane deviation warning are made possible by image sensor data. [24] The emergence of automated and autonomous driving necessitates the use of high-resolution sensor data to make decisions. Furthermore, these systems must respond safely and quickly to dynamically changing environmental conditions, such as traffic, weather, and so on. The paper discusses the many isolation mechanisms, standards, and concepts used to provide functional safety in automotive AVB systems. The research demonstrates that standardizations do not definitely mean safe "out of the box" nor performant as well as it is necessary to take the new design vectors into consideration. [25]

3. Machine to Machine Network Design and Optimization of Latency

3.1 Machine to Machine Network Design

The architectural approach between the components of the automotive environment is discussed in this section as illustrated in **Fig. 4**. The transmitted data is designed and distributed from network cameras, USB and antennas through AVB system. The network machines are transmitted directly to each machine or through a central switch to devices such as the head unit. The transfer of multimedia data directly to the head unit, such as audio and video could minimize the time lag between devices. However, there is a significant delay in case the video stream of camera and the UI are combined in order to share the monitor of head unit. In most of today's vehicles, it is difficult for all multimedia equipment to have its own display unless it is a top-of-the-line car. Therefore, the number of times that multiple devices use and share a limited display is increasing. As a result of combining images by exchanging the view as mentioned above, the display of the head unit displayed a delay of around 2 times when the UI was combined and transmitted in real time, compared to the image directly obtained from the camera.

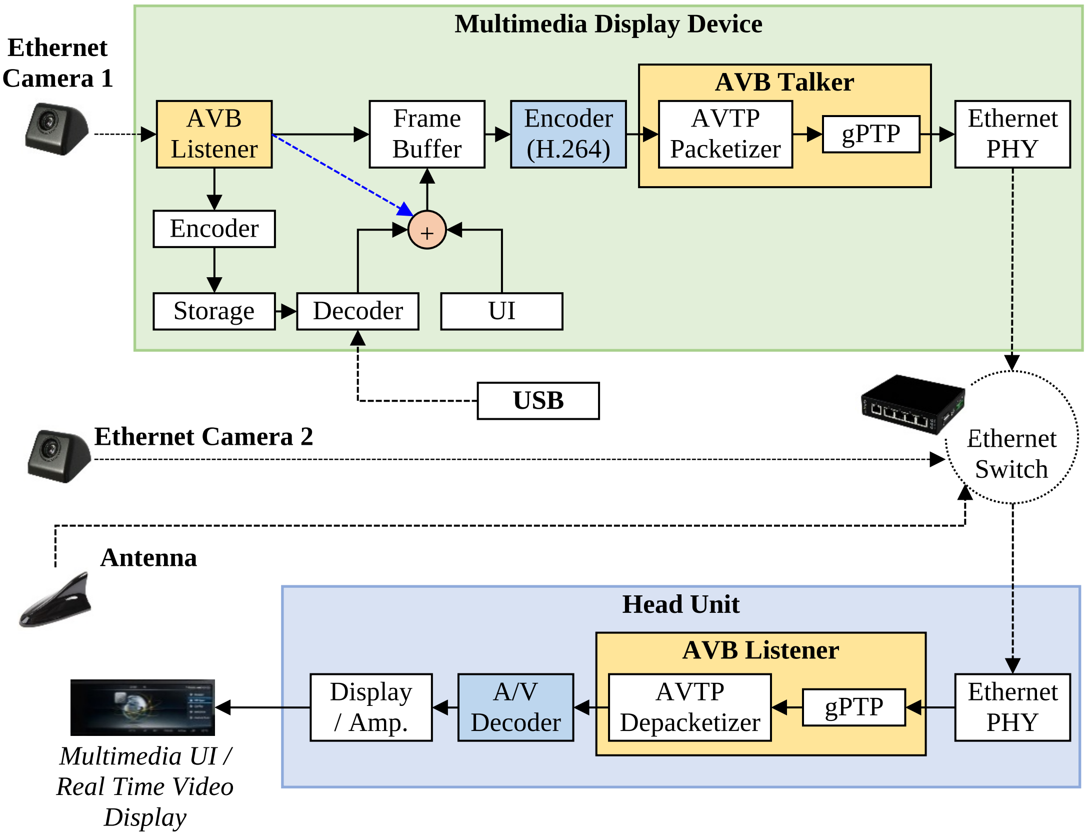


Fig. 4. In-vehicle AVB communication design example, blue dot is video transmission without UI transmission. (autonomous driving real-time video transmission situation)

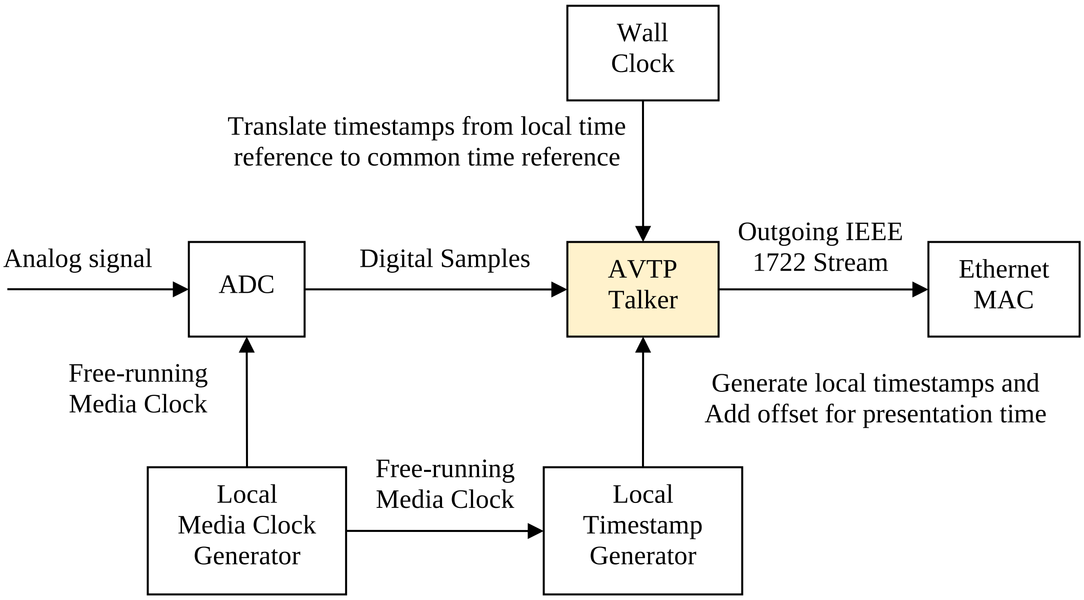

Wall Clock
Translate timestamps from local time reference to common time reference
Analog signal
ADC
Digital Samples
AVTP Talker
Outgoing IEEE 1722 Stream
Ethernet MAC
Free-running Media Clock
Generate local timestamps and Add offset for presentation time
Local Media Clock Generator
Free-running Media Clock
Local Timestamp Generator

Fig. 5. AVB talker simplified block diagram.

3.2 Reduced Latency of Machine to Machine (RLMM)

3.2.1 A Synchronization Between Talker and Listener

A Synchronization between talker and listener are crucial in the automotive system. In **Fig. 5**, it explains how the media clock source is embedded into AVTP (Audio Video Transport Protocol) packets. The media clock's source can be a simple free-running local oscillator because the media clock does not need to lock to any reference as long as the frequency of all media clocks in the same domain are synchronized. The media clock goes into a local timestamp generator that generates timestamps every certain number of media clock rising edges. These timestamps are generated based-off of a local time reference. The wall clock then translates the local time-based timestamps to gPTP timestamps. The AVTP talker adds a fixed offset (typically the maximum delay between the Talker and all Listeners) and generates presentation timestamps. AVTP Presentation Time represents the gPTP time at which a given media sample, or the event is sent to the time-sensitive application within each listener. This allows multiple listeners to display data at the same time, regardless of the listener's location on the network. The AVTP presentation time tells the listener when to start processing the stream's data. It is also used to recover the stream's media clock. [26]

In **Fig. 6**, it shows how a Listener extracts presentation timestamps and recovers the source media clock from incoming 1722 streams generated by the talker. The period of the source media clock in the gPTP time base can be estimated by the time difference between two presentation timestamps divided by the number of samples. Continuously performing this calculation and applying appropriate filtering techniques will allow the system to accurately measure the media clock period of the source. Similarly, the local media clock generator output can be timestamped with local time base which is then translated to gPTP

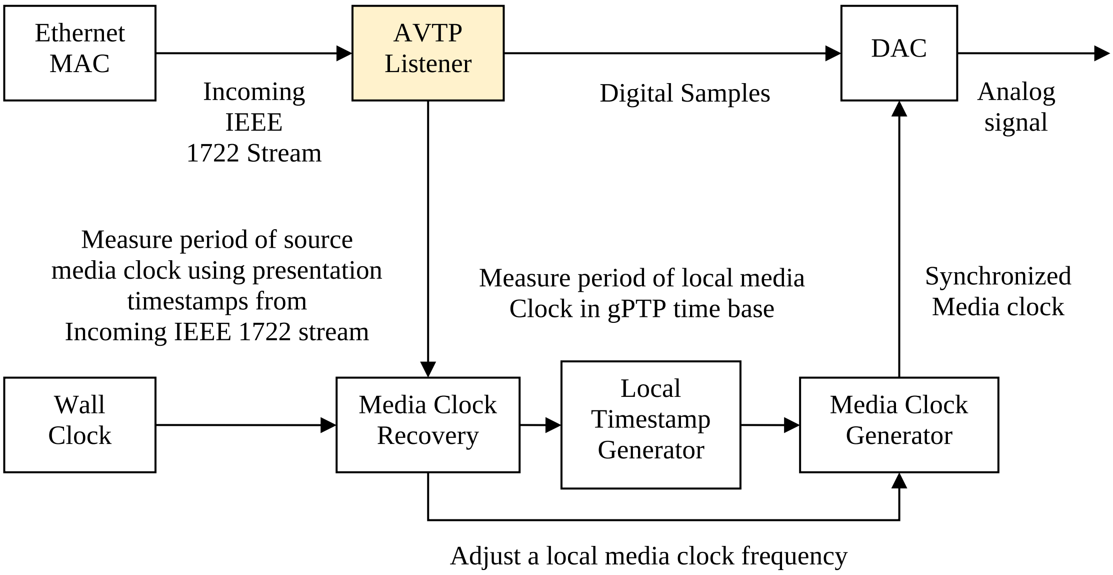

Ethernet MAC
Incoming IEEE 1722 Stream
AVTP Listener
Digital Samples
DAC
Analog signal
Measure period of source media clock using presentation timestamps from Incoming IEEE 1722 stream
Measure period of local media Clock in gPTP time base
Synchronized Media clock
Wall Clock
Media Clock Recovery
Local Timestamp Generator
Media Clock Generator
Adjust a local media clock frequency

Fig. 6. AVB listener simplified block diagram.

time base, and its period can also be accurately measured. After comparing the two clock periods, the Media Clock Recovery module continually generates to increment or decrement output frequency of the device so that the local media clock is synchronized to the source media clock. [27]

3.2.2 Latency Computation

The delay in AVB *lQueue(m, q)* is calculated using the traffic class's position, queue *q*'s reserved bandwidth, and the message *m*. Consider a series of AVB queues that will pass *Q* through *m* (see, for example, Eq. (1), (2), (3), and (4)).

$$lQueue(m, q) = d_s(m, q),\ if\ q = q_s \tag{1}$$
$$lQueue(m, q) = d_0(m, q),\ if\ q = Q_0 \tag{2}$$
$$lQueue(m, q) = d_n(m, q),\ if\ q = Q_n \tag{3}$$

The queue q_s with the smallest bandwidth requirement for the traffic class must be considered. If the traffic bandwidth allotted to the bottom of the queue is the same, so q_s is the highest. At q_s, the cumulative shaping delay is applied. Where Q_0 is a list of AVB queues that routes *m* (the message) ahead of q_s. m (the message) and Q_n, which is routed over after q_s, are the AVB queues. Equation (4) explains it as following,

$$\begin{gathered} Q = Q_0 \cup \{q_s\} \cup Q_n\ with \\ Q_0 \cap \{q_s\} = Q_0 \cap Q_n = \{q_s\} \cap Q_n = \varnothing \end{gathered} \tag{4}$$

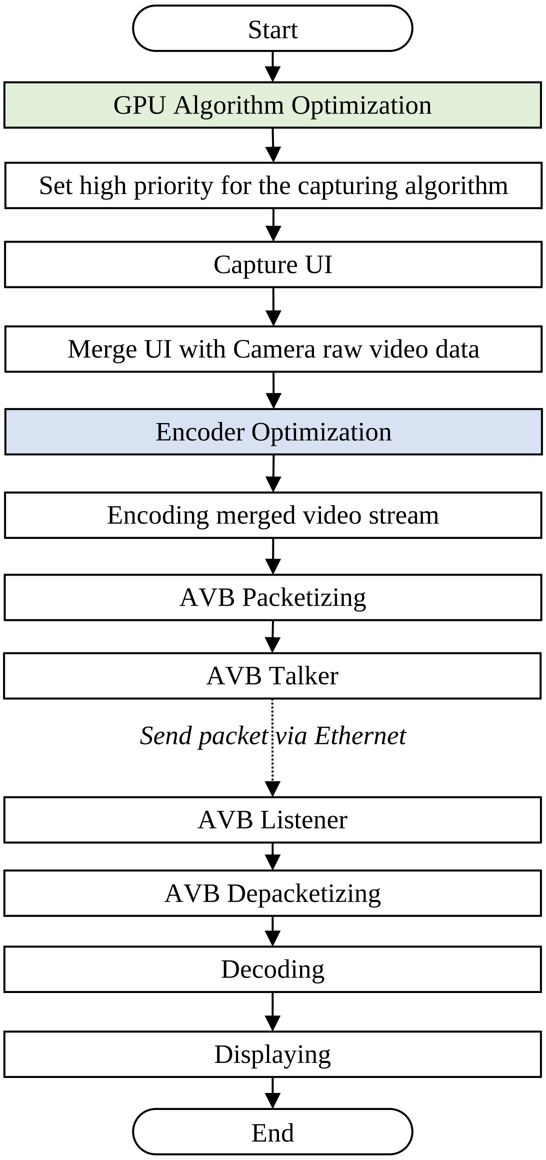

Start
GPU Algorithm Optimization
Set high priority for the capturing algorithm
Capture UI
Merge UI with Camera raw video data
Encoder Optimization
Encoding merged video stream
AVB Packetizing
AVB Talker
Send packet via Ethernet
AVB Listener
AVB Depacketizing
Decoding
Displaying
End

Fig. 7. The proposed RLMM flowchart.

3.2.3 RLMM

We suggest an integrated approach based on in-vehicle interface experiments. The suggested scheme is represented in condensed form in **Fig. 7**. The usage of CPU rises by more than 65 percent while the AVB talker is running, according to test results in an embedded multi-core ARM environment. In this case, the CPU consumption was deemed undesirable, and tests have shown that optimizing the GPU will minimize the latency. By accelerating the GPU, it is possible to reduce the resource consumption of UI capture and audio/video decoding while also satisfying the device specifications of multimedia equipment. It uses a GPU optimization algorithm to minimize CPU consumption. [28] The UI capture time is greatly decreased by speeding and optimizing the GPU, as seen in **Fig. 8**.

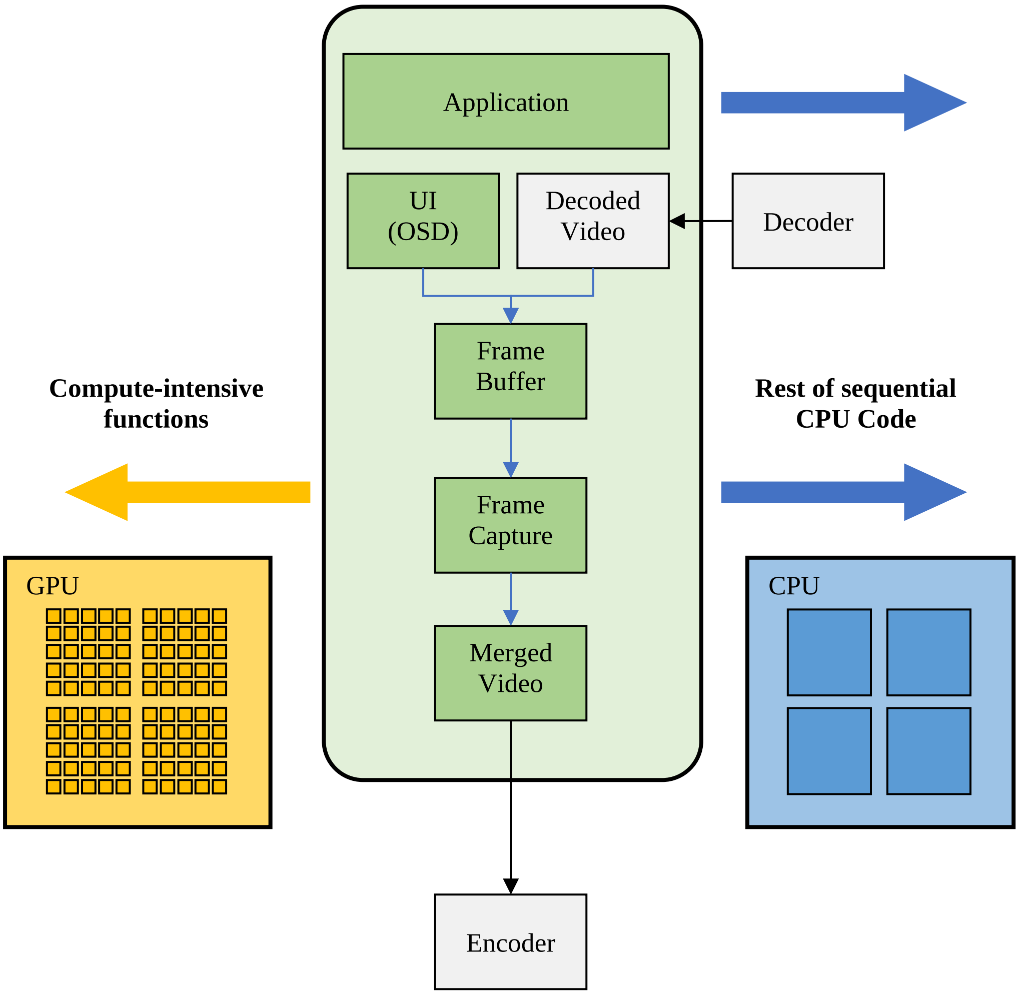
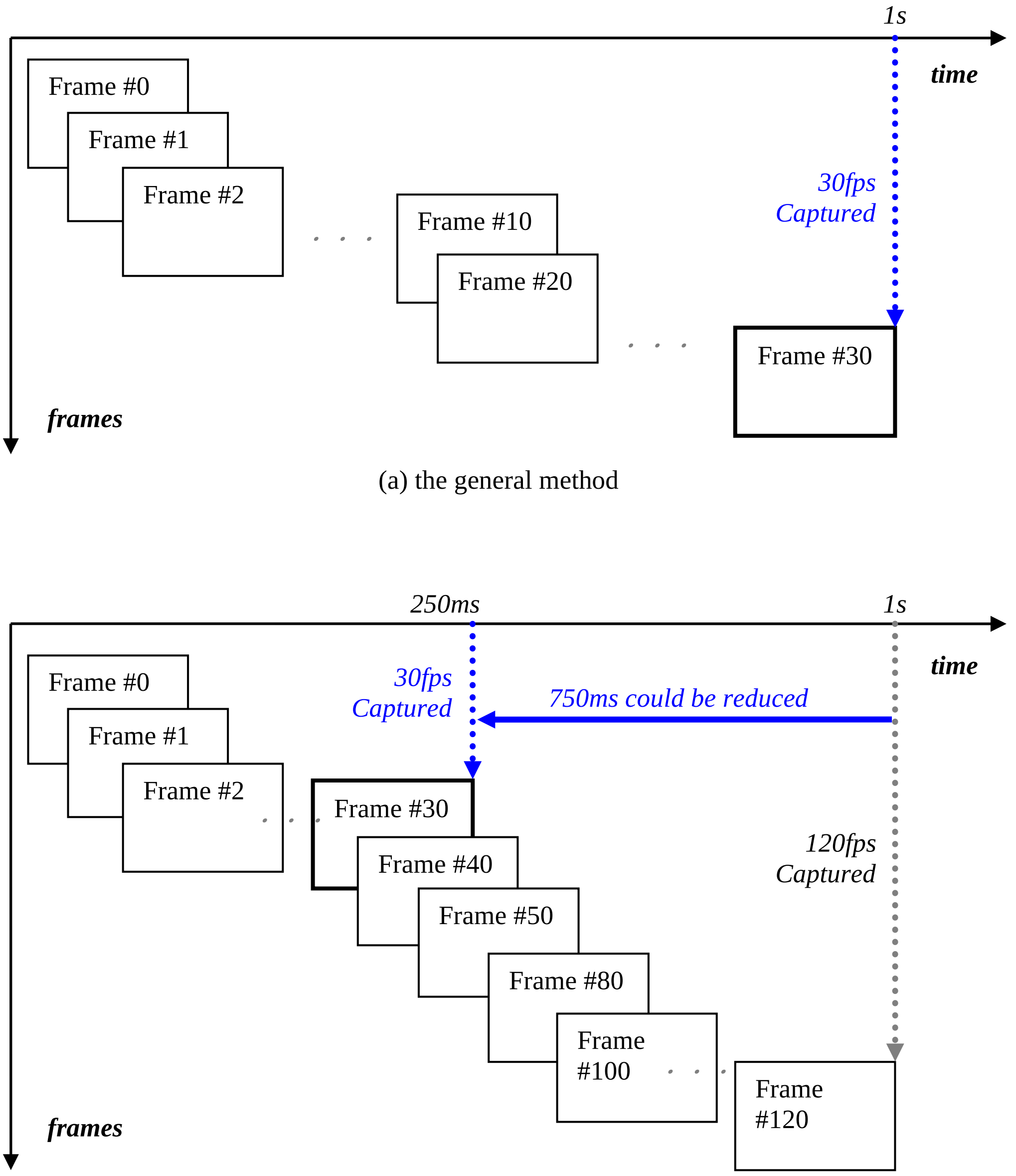


Fig. 8. GPU Acceleration.

In this paper, we propose a frame capture method as shown in **Fig. 9**. In general, the frames are captured at the frame buffer in proportion to the input image. If the input is 30fps, the captured frames could be 30fps. We found that the usage CPU is increased during the 30fps capture procedure while the usage of GPU is used than we are expected. We have investigated the optimization of system if there is any way to reduce the latency by using the GPU. [29] The proposed capturing of frame method shows that it captures four times more frames at the same time (1s) than before. After capturing 120fps, we dropped it into 30fps. The result shows that the proposed algorithm is possible to reduce the capture time by 4 times, Eq. (5).

$$\begin{aligned}&\textit{Original Capture Time : } FC(t)\\&\textit{Proposed Capture Time : } FC(t)\,/\,\textit{GPU acceleration time}\end{aligned} \tag{5}$$

Where, FC(t) : Frame Capture Time



(a) the general method

(b) the proposed method

Fig. 9. GPU acceleration with the proposed capturing of frame method.

In addition, we used a system with optimized AVB configuration to implement the design in the encoder. The proposed algorithm allows the delay reduction of all system aspects. There are many options for the encoding method, in consideration of the compression rate and resolution of the image according to the specifications requested by the automotive OEM (Original Equipment Manufacturer), it was confirmed that setting the specific coefficient in agreement with the vendor can reduce the delay. The encoder setting's chosen SDME (Speed Dependent Motion Estimation) methodology could reduce computational processing time. [30]

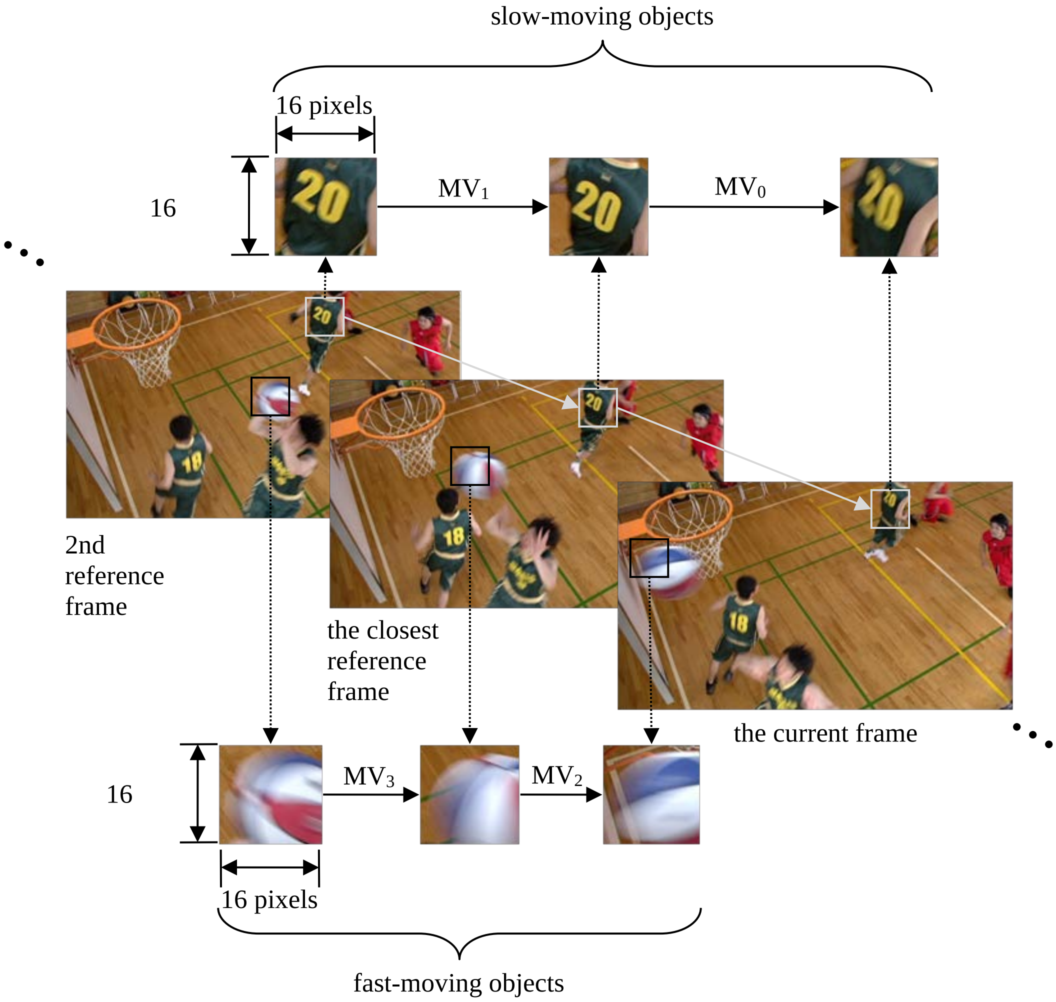


Fig. 10. Video coding sequences provide a perceptual graphical element.

For a long time, the perceptual video coding depending on ROI (Region of Interest) technology has been created. In the past, ROI-based video coding methods had limitations when defining ROI. The foreground/background video coding algorithm adopts this method. [31] Visual focus approaches have been created in recent years to recognize more versatile ROIs or ROA (Region of Areas) of concern than previously. Smaller quantization parameters are used in the ROI video coding process to reflect a ROI with less distortion, which can vastly increase the overall video's subjective consistency. On the other hand, using different quantization parameters for within and outside the ROI will result in objects at the ROI boundary. The emphasis and characteristics of HVS (Human Vision System) are not taken into account by some current ROI video coding algorithms. The quantization parameters for the ROI and context regions are coded using fixed weights. As a result, the ROI video coding system is inadequate for eliminating perceptual redundancy. ROI-based video coding can improve coding quality by using the perceptual features of HVS by integrating with the foveation model. The visual attention system has recently allowed the identification of multiple ROIs. The perceptual and visual features of a video coding sequence are shown in **Fig. 10**. The key point here is that the human perception is susceptible to well-structured entity motion (MV_0 which MV_1) (Motion Vector) and makes significant illusions in areas that shift with random structures (MV_2 and MV_3). Fast objects have greater motion vectors than slow objects since the size of the motion vector is proportional to the distance between the actual video frame and the closest reference video frame. Through identifying MBs (Motion Block) ahead of time, the proposed algorithm reduces numerical processing. The proposed SDME aims to minimize computation time by missing some MBs while maintaining the best possible RD (Rate Distortion) efficiency. The properties of HVS are used to design SDME. To start, the motion of the motion vector can be calculated as shown in Equation (6) below to see if the motion of MB is wide enough to induce motion blur.

$$\mathrm{MVDistV} = |\,\overrightarrow{MV}\,| \tag{6}$$

Where MV denotes the motion vector used to encode the MB difference between the present and reference video frames. This covers the volume and speed of movement. Following that, the maximal scale value is represented by the DV (Distance Value) as follows:

$$\mathrm{DV} = \sqrt{(width)^2 + (height)^2} \tag{7}$$

The horizontal and vertical measurements of the input image are height and width, respectively. These parameters are calculated using an OV (Offset Value) and can be obtained using Eq. (8) to quantify MVDistV as a percentage of DV.

$$\mathrm{OV} = \frac{\mathrm{MVDistV}}{\mathrm{DV}} = \begin{cases} 1, & 0 \leq \mathrm{OV} \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where OV is restricted to a scale of 0 to 1.

Then we set the LT (Lowest Threshold) and HT (Highest Threshold) to see if SDME was turned on. Since the SDME algorithm is only enabled when the OV is between LT and HT, Equation (8) is crucial in implementing SDME. The thresholds of LT and HT are the set range of the input sequence before coding, the minimum LT is 0 (0%), the maximum HT is 1 (100%), and is determined taking into account the HVS and various experimental variables, which are the overall speed. Frames and search range of test sequences:

$$\text{Lowest Threshold (LT)} \;\leq\; \mathrm{OV} \;\leq\; \text{Highest Threshold (HT)} \tag{9}$$

When the suggested SDME method is turned on, it collects MVDistV, DV, and OV and compares them to LT and HT. If the OV is between LT and HT, the largest block mode is selected as the best mode for calculating MB with the least amount of computation. The SDME algorithm is critical in determining the difference between the suggested and normal methods. The best findings are explained in Section 4.

4. Experimental Result and Analysis

4.1 Test Environment

In this paper, we designed to be able to test the various AVB systems introduced in the previous section. As explained in Fig. 11, the test environment for the external camera, the non-display device, and the display device (head unit) among several systems are described. The linux-based benchmark devices were used for HW, open-sources, Gstreamer were adopted for SW to create pipelines for each structure. In the experiment, OpenAvnu was used as the software for AVB environment. The Avnu Alliance is the project sponsor for OpenAvnu. The goal is to offer components for the development of AVB/TSN systems. The repository primarily provides network building block components - drivers, libraries, sample applications, and daemon source code - involved in building an AVB/TSN system. It is intended to ultimately contain different packet encapsulation types, protocol discovering daemons, and algorithms to convert media clocks to AVB clocks and vice versa. Visit the Avnu Alliance webpage at www.avnu.org for further information about AVB/TSN. The open-source website of GitHub for OpenAvnu could be found here. [32] Table 1 describes the experimental environment and specifications in detail. We have established some typical test environment for conducting tests in a well-defined manner so that the results can be compared fairly.

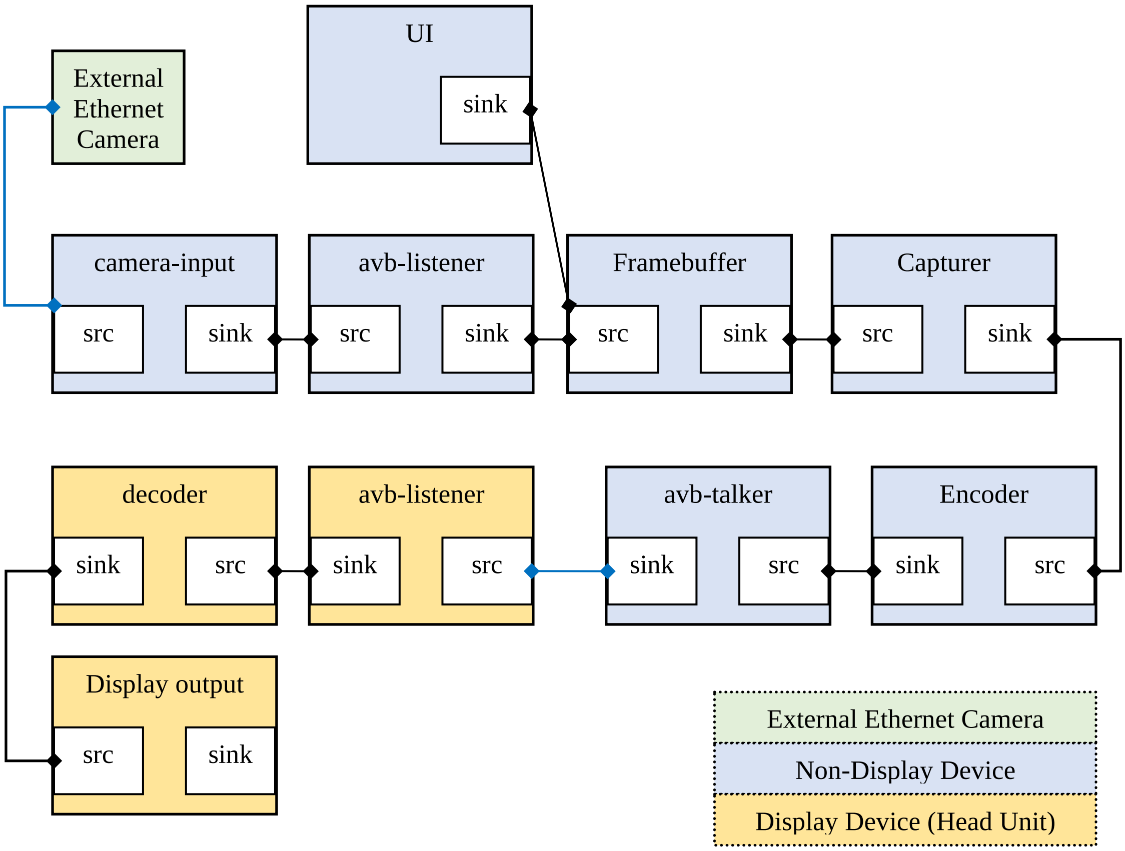


Fig. 11. A system pipeline structure of Gstreamer among AVB devices.

Table 1. The test environment and specifications of devices

Devices	**Components**	Specifications
Ethernet Camera	Resolution	H.264 (1920x1080p@30fps)
LVDS Camera	Resolution	H.264 (1920x1080p@30fps)
LVDS to HDMI Converter	Resolution	H.264 (1920x1080p@30fps)
Ethernet Switch	Speed	10/100/1000Mbps
Multimedia Device (Qualcomm Evaluation Kit)	CPU	QCS605, Kryo^{TM} 300, Octa-core, 64bit, 2.5GHz
	GPU	Adreno 615, 64bit, 780MHz
	DDR	4GB (LPDDR4)
	Video Encoder	H.264 (1920x1080p@30fps)
	Video Decoder	H.264 (1920x1080p@30fps)
	OS	Kernel 4.9 (Android Oreo 8.1)
	Gstreamer	v1.14.5
	Computer Language	C / C++ / Shell Script
	SDK	Provided by EVK
	AVB Software	OpenAvnu
Head Unit (PC)	CPU	Intel(R) Core (TM) i7-8550U, 1.80GHz
	DDR	16GB (LPDDR4)
	GPU	NVIDIA QUADRO P2000 (Memory 5GB GDDR5, i/f 160-bit, B/W 140GB/s, CUDA Cores 1024)
	OS	Ubuntu v18.04 LTS
	NVIDIA SMI Driver	v440.000
	NVIDIA CUDA	V10.2.89
	NVIDIA Video Codec SDK	V9.0.20
	Gstreamer	v1.14.5
	Computer Language	C / C++ / Shell Script
	AVB Software	OpenAvnu

4.2 SR class latency target

One of the goals of the AVB specification is to make it possible to configure an AVB network that meets standard end-to-end latency metrics in the worst-case scenario of a stream being transmitted between the talker and the listener. For SR classes A and B, the target value for the maximum delay time [802.1 BA] from talker to listener is defined as shown in **Table 2**. The SR Class A and B latency targets embody the specifications of many common implementation situations and must not be treated as a rigid cap on the video network's end-to-end latency. However, it provides a firm base for pursuing "plug & play" interoperability. [1] The 802.1Q specification defines variables that lead to a bridge's latency contributions. SRP needs talkers including bridges to record the network's worst latency for each hop. The worst-case latency for a class A stream can be calculated using Equation (10) as following:

Table 2. MTT (Maximum Transit Time)

A Standard of MTT	Time *(ms)*
SR Class A	2
SR Class B	10

$$\begin{aligned} Latency\ maximum &= t_{Device} + t_{MaxPacketSize} + IPG + (t_{AllStreams} - t_{StreamPacket} + IPG) \\ &\times Rate/MaxAllocBand + t_{StreamPacket} \end{aligned} \tag{10}$$

where,

- t_{Device}

  : The device's internal latency
- $t_{MaxPacketSize}+IPG$

  : The transmission time for maximum size interfering frame and preamble and SFD (Start of Frame Delimiter) and subsequent IPG (Inter-packet Gap)
- $t_{StreamPacket}$

  : The transmission time, preamble and SFD for the maximum frame size of the reserved stream
- $t_{StreamPacket}+IPG$

  : The transmission time, preamble and SFD, and next IPG for the maximum frame size of the reserved stream
- ***Rate***

  : The port's transmission rate
- ***MaxAllocBand***

  : The maximum bandwidth that can be allocated, the maximum bandwidth that the AVB system can allocate to the port's stream.
- $t_{Interval}$

  : The observation interval, or 125 μs
- $t_{AllStreams}$

  : $(MaxAllocBand \times t_{Interval}) / t_{Rate}$ = The sum of the transmission times of all stream frames that the AVB system can allocate in the observation interval (125 μs) on the port. [1]

Table 3. Comparison of proposed RLMM and existing AVB system

Devices	**Components**	**Without RLMM *(ms)***	**Proposed RLMM *(ms)***	**Decreased Latency *(ms)***	***Δ (%)***
Network Camera		61.3	61.3	-	-
Network Switch		1.52	1.52	-	-
Multimedia Device (Share the display of H/U)	AVB Listener	10.1	10.1	-	-
	UI + Camera video	22.2	7.1	-15.1	*68.0%*
	UI Capture	115.4	15.2	-100.2	*86.8%*
	Video Encoder	62.1	27.2	-34.9	*56.2%*
	AVB Talker	52.7	52.7	-	-
Network Switch		1.52	1.52	-	-
Head Unit	AVB Listener	10.3	10.3	-	-
	Video Decoder	22.2	22.2	-	-
Monitor		5.1	5.1	-	-
Latency Total		364.4	214.2	-150.2	*41.2%*

4.3 Analysis of Experimental Results

A system model was created to test, verify, and analyze AVB's experimental results against existing Ethernet requirements in order to explore the viability of using AVB standards. [11], [12], [13], [26], [27] The component and the test system model configuration are visualized in **Fig. 4**, which finds the hardware and software modules appropriate for in-vehicle network connectivity. A single central network switch, diverse cameras, and digital devices make up the in-vehicle network. A full-duplex connection connects the switches and equipment.

Table 3 displays the experimental results of the proposed system. In our exploration, we experiment to get the same results at network cameras, switches, head units, and monitors (displays) that have not been improved in our experiments. Major improvements have been achieved in multimedia devices. In particular, it showed great improvement in the process of merging UI and camera video, the UI capture process of capturing the merged image in the frame buffer, and the video encoder optimization process. First, when RLMM was not utilized, the speed of the real-time merging of UI and the camera video was 22.2 milliseconds. When RLMM was used in the same scenario, the result was 7.1 milliseconds and a speed improvement of -15.1 milliseconds (68.0 percent). Second, when RLMM was not utilized, a delay of 115.4 milliseconds was observed in the process of capturing the merged images on the frame buffer in real-time which had the largest latency of the AVB system. The suggested technique yielded an excellent result of 15.2 milliseconds. When it comes to the improvement of latency, the RLMM achieves a reduction of -100.2 milliseconds (86.8 percent).

Table 4. The PSNR result of decoded video frames

Decoded Frames	**Average PSNR (dB)**			
	Original Method	**Proposed RLMM**	**△ (dB)**	**△ (%)**
1 – 30	35.26	35.22	*-0.04*	*-0.11*
31 – 60	35.25	35.20	*-0.05*	*-0.14*
61 – 90	35.39	35.23	*-0.16*	*-0.45*
91 – 120	35.24	35.18	*-0.06*	*-0.17*
121 – 150	35.20	35.10	*-0.1*	*-0.28*
151 – 180	35.11	35.05	*-0.06*	*-0.17*
181 – 210	35.10	35.02	*-0.08*	*-0.23*
211 – 240	35.25	35.20	*-0.05*	*-0.14*
241 – 270	35.31	35.23	*-0.08*	*-0.23*
271 – 300	35.15	35.04	*-0.11*	*-0.31*
Total Average	35.226	35.147	*-0.079*	*-0.22*

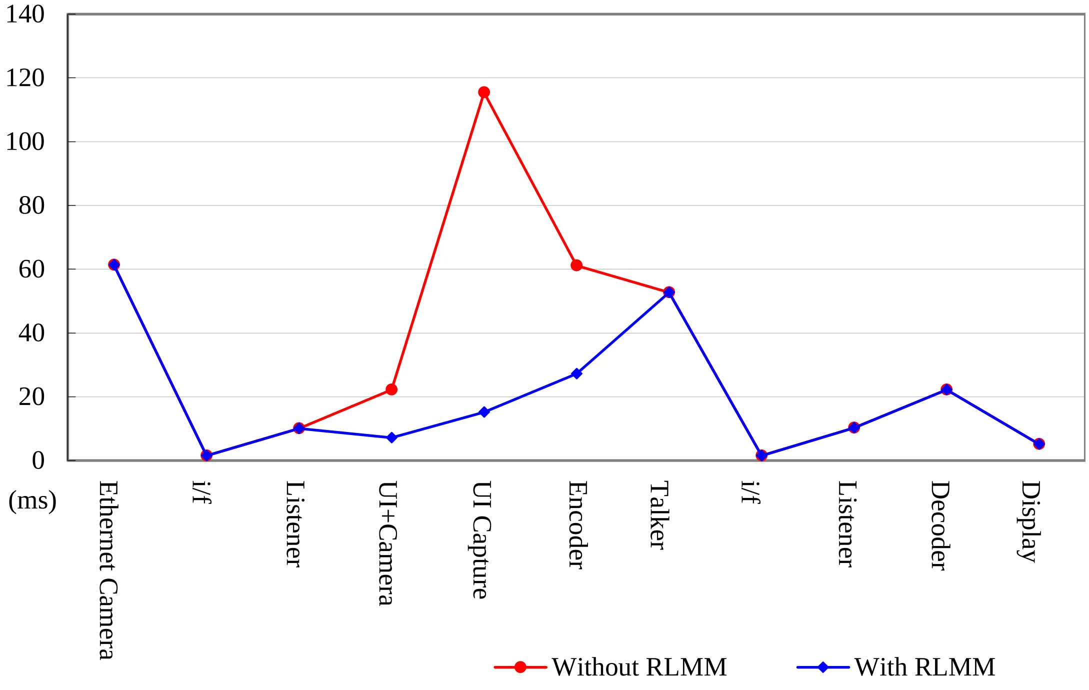


Fig. 12. Comparison between with RLMM and without RLMM.

A significant reduction was obtained through optimization of the GPU algorithm, but the concern about frame drop had to be taken into account. **Table 4** compares the results of the objective image quality evaluation in the proposed RLMM and the existing method. As a result of the experiment, we compared the PSNR (Peak Signal-to-Noise Ratio) of the decoded frames. There was an average of -0.079dB (-0.22%) degradation in image quality, but it was not at the level we were concerned about. The proposed method had a slight deterioration in image quality through image quality comparison, but only a slight difference was seen; therefore, it was confirmed that the proposed low-delay algorithm has an influence on reducing the latency. Third, we compared the optimization results of video encoders. The transmission rate was 62.1 milliseconds when RLMM was not used, but it was 27.2 milliseconds when RLMM was used. A delay of -34.9 milliseconds (56.2 percent) was decreased according to video encoder improvement. Overall, End-to-end processing rates are substantially faster than in previous models for the same payload as shown in **Fig. 12**. As a result, the suggested RLMM method significantly reduces the latency of AVB system from 364.4 milliseconds to 214.2 milliseconds. The overall system achieved an average improvement of about 42.1 percent.

5. Conclusion

We evaluated the AVB network latency in this article, developed a new approach based on the study result and suggested the optimization algorithm of AVB system on Linux embedded devices. The focus of this paper was to analyze the effect of delay reduction between components of in-vehicle network equipment in detail by function. For the level of practical use available for commercial products, the proposed RLMM has shown that there is a delay reduction of about 41% across the entire AVB system. The suggested method of adapting to the automotive network world is supposed to minimize the time in the design and production phase considerably. In addition to a latency review evidence for AVB, we have proposed alternatives for actual trials. In this project, a vehicle multimedia camera was generally tested using 2M (1920x1080). The inclusion of images over 8M (3840x2160) in vehicles using the proposed algorithm is supposed to make a larger impact. More study will lead to further advancements in AVB system, although it may be difficult to considerably exceed the suggested optimized GPU method design. The proposed approach has shown tremendous promise in the image processing for automotive artificial intelligence technology, although this study is just beginning to develop. Consequently, it is anticipated that AI algorithms in autonomous driving are a great aid in the real-time interpretation and analysis of videos through the optimized camera and video streaming using the enhanced AVB system. In the future research, we plan to investigate the high-speed and low cost for wide-ranging deployment in various video applications.

Acknowledgement

This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2C2009717).

References

[1] IEEE Standard for Local and metropolitan area networks - Audio Video Bridging (AVB) Systems, in IEEE Std 802.1BA-2011, pp.1-45, 30 Sept. 2011. Article (CrossRef Link).

[2] M. Booysen, J. Gilmore, S. Zeadally and G. Rooyen, "Machine-to-Machine (M2M) Communications in Vehicular Networks," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 2, pp. 529-546, 2012. Article (CrossRef Link).

[3] J. Eveleens, "Ethernet AVB Overview and Status," in *Proc. of SMPTE 2014 Annual Technical Conference & Exhibition*, Hollywood, CA, USA, pp. 1-11, 2014. Article (CrossRef Link).

[4] C. Herber, A. Saeed and A. Herkersdorf, "Design and Evaluation of a Low-Latency AVB Ethernet Endpoint Based on ARM SoC," in *Proc. of 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, New York, NY, pp. 1128-1134, 2015. Article (CrossRef Link).

[5] T. Nolte, H. Hansson and L. L. Bello, "Automotive communications-past, current and future," in *Proc. of 2005 IEEE Conference on Emerging Technologies and Factory Automation*, Catania, pp. 8 pp.-992, 2005. Article (CrossRef Link).

[6] R. Queck, "Analysis of Ethernet AVB for automotive networks using Network Calculus," in *Proc. of 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, Istanbul, pp. 61-67, 2012. Article (CrossRef Link).

[7] J. Park, B. Cheoun and J. Jeon, "Worst-case analysis of ethernet AVB in automotive system," in *Proc. of 2015 IEEE International Conference on Information and Automation*, Lijiang, pp. 1696-1699, 2015. Article (CrossRef Link).

[8] X. Liu, Z. Nie, D. Li and H. Yu, "Design of An Improved Ethernet AVB Model for Real-time Communication in In-Vehicle Network," in *Proc. of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, pp. 6-10, 2019. Article (CrossRef Link).

[9] Diarra, A., Hogenmueller, T., Zimmermann, A., Grzemba, A., Khan, U.A., “Improved Clock Synchronization Start-up Time for Ethernet AVB-based In-vehicle Networks,” in *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation*, ETFA, October, 2015, art. no. 7301412. Article (CrossRef Link).

[10] S. Jeon, J. Lee and S. Park, "Dual-path method for enhancing the performance of IEEE 802.1 avb with time-triggered scheme," in *Proc. of 2015 21st Asia-Pacific Conference on Communications (APCC)*, Kyoto, pp. 519-523, 2015. Article (CrossRef Link).

[11] IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks, in IEEE Std 802.1AS-2011, pp.1-292, 30 March 2011. Article (CrossRef Link).

[12] Virtual Bridged Local Area Networks – Amendment 9: Stream Reservation Protocol (SRP), in IEEE P802.1Qat/D6.1 edition, June 2010. Article (CrossRef Link).

[13] IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams, in IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005), pp.C1-72, 5 Jan. 2010. Article (CrossRef Link).

[14] A. Kern, H. Zinner, T. Streichert, J. Nöbauer and J. Teich, "Accuracy of Ethernet AVB time synchronization under varying temperature conditions for automotive networks," in *Proc. of 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, New York, NY, pp. 597-602, 2011. Article (CrossRef Link).

[15] Park, C., Lee, J., Tan, T., Park, S., “Simulation of Scheduled Traffic for the IEEE 802.1 Time Sensitive Networking,” in *Proc. of International Conference on Information Science and Applications*, ICISA, Vol. 376, pp. 75-83, 2016. Article (CrossRef Link).

[16] D. An, "Design and implementation of the car multimedia data transmission system using Ethernet AVB," in *Proc. of 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, pp. 481-483, 2016. Article (CrossRef Link).

[17] O. Kleineberg, P. Fröhlich and D. Heffernan, "Fault-tolerant Audio and Video Bridging (AVB) Ethernet: A novel method for redundant stream registration configuration," in *Proc. of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, Krakow, pp. 1-8, 2012. Article (CrossRef Link).

[18] L. Zhao, P. Pop, Z. Zheng and Q. Li, "Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus," in *Proc. of 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Porto, pp. 25-36, 2018. Article (CrossRef Link).

[19] J. Imtiaz, J. Jasperneite and L. Han, "A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication," in *Proc. of 2009 IEEE Conference on Emerging Technologies & Factory Automation*, Mallorca, pp. 1-8, 2009. Article (CrossRef Link).

[20] F. Reimann, S. Graf, F. Streit, M. Glaß and J. Teich, "Timing analysis of Ethernet AVB-based automotive E/E architectures," in *Proc. of 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Cagliari, pp. 1-8, 2013. Article (CrossRef Link).

[21] Kim, H.-J., Choi, M.-H., Kim, M.-H., Lee, S., "Development of an Ethernet-based Heuristic Time-sensitive Networking Scheduling Algorithm for Real-time In-vehicle Data Transmission," *Electronics (Switzerland)*, 10 (2), pp. 1-11, 2021, Art. no. 157. Article (CrossRef Link).

[22] D. Bujosa, I. Álvarez and J. Proenza, "CSRP: An Enhanced Protocol for Consistent Reservation of Resources in AVB/TSN," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3640-3650, May 2021. Article (CrossRef Link).

[23] A. Sabry, A. Omar, M. Hammad and N. Abdelbaki, "AVB/TSN Protocols in Automotive Networking," in *Proc. of 2020 15th International Conference on Computer Engineering and Systems (ICCES)*, pp. 1-7, 2020. Article (CrossRef Link).

[24] Z. -Y. Xu, J. -F. Jiang, H. -Y. Wei and K. -S. Hsu, "Driving Simulation of Autonomous Vehicle with ADS Control," in *Proc. of 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 374-377, 2020. Article (CrossRef Link).

[25] A. Kostrzewa, D. Stöhrmann and R. Ernst, "Towards Safety in Automotive Ethernet-based Networks with Dynamic Workloads," in *Proc. of 2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pp. 1-6, 2020. Article (CrossRef Link).

[26] IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks, in IEEE Std 1722-2016 (Revision of IEEE Std 1722-2011), pp.1-233, 16 Dec. 2016. Article (CrossRef Link).

[27] IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722(TM) Based Devices - Corrigendum 1:Technical and Editorial Corrections, in IEEE Std 1722.1-2013/Cor 1-2018 (Corrigendum to IEEE Std 1722.1-2013), pp.1-69, 15 March 2019. Article (CrossRef Link).

[28] B. An and Y. Kim, "Optimization of Vehicle Communication Networking Parameters in eAVB," in *Proc. of 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, Seoul, Korea (South), pp. 1-4, 2020. Article (CrossRef Link).

[29] J. M. Ready and C. N. Taylor, "GPU Acceleration of Real-time Feature Based Algorithms," in *Proc. of 2007 IEEE Workshop on Motion and Video Computing (WMVC'07)*, Austin, TX, USA, pp. 8-8, 2007. Article (CrossRef Link).

[30] B. An, Y. Kim and O. Kwon, "Low-Complexity Motion Estimation for H.264/AVC Through Perceptual Video Coding," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 8, pp. 1444-1456, 2011. Article (CrossRef Link).

[31] D. Chai and K. N. Ngan, "Foreground/background video coding scheme," in *Proc. of 1997 IEEE International Symposium on*, vol. 2, pp. 1448-1451, 1997. Article (CrossRef Link).

[32] OpenAvnu, "A github based website for OpenAvnu,". Article (CrossRef Link).

Byoungman An received the B.S. and M.S. in Electronics & Electrical Engineering from Dankook University, Korea in 2010 and 2012, respectively. He is also currently pursuing an Ph.D. degree in the school of Electrical & Electronics Engineering. His research interests include image/video compression, automotive network, in-vehicle network and image processing.

Youngseop Kim received the M.S in Computer Engineering from the University of Southern California in 1991, and the Ph.D. in E.S from Rensselaer Polytechnic Institute in 2001. He was a manager at Samsung SDI until 2003. He developed the image-processing algorithm for PDP TV while at Samsung. Currently he is a Professor at Dankook University in Korea. He was the resolution member and the Editor of JPsearch part 2 in JPEG, the co-Chair of JPXML in JPEG, and is still Head of Director (HOD) of Korea. He is also Editor-in-chief of the Korea Semiconductor and Technology Society. His research interests are in the areas of image/video compression, pattern recognition, communications, stereoscopic codecs, augment reality, and multimedia with deep learning. They include topics such as object-oriented methods for image/video coding, fake media, pattern recognition, image processing with deep learning, and feature extraction.