Design and Implementation of Data Communication and Compression Methods in SIGMA Framework

Akira Matsuki
Graduate School of Information Science & Engineering Ritsumeikan University
Kusatsu, Japan
a-matuki@rm2c.ise.ritsumei.ac.jp

Tomonori Aritomi
Graduate School of Information Science & Engineering Ritsumeikan University
Kusatsu, Japan
aritomi@rm2c.ise.ritsumei.ac.jp

Kento Yamazaki
Graduate School of Information Science & Engineering Ritsumeikan University
Kusatsu, Japan
yamazaki@rm2c.ise.ritsumei.ac.jp

Asako Kimura
Graduate School of Information Science & Engineering Ritsumeikan University
Kusatsu, Japan
asa@rm2c.ise.ritsumei.ac.jp

Fumihisa Shibata
Graduate School of Information Science & Engineering Ritsumeikan University
Kusatsu, Japan
fshibata@is.ritsumei.ac.jp

***Abstract*—This paper introduces data communication and flexible data compression methods in SIGMA, the application framework we are developing. Recently, various kinds of wireless communication technologies and sensing devices have been developed. Hence, the opportunity of transmitting images and other large-volume data via the Internet is expected to increase. We consider specifying a region of interest (ROI) on spatiotemporal axes to reduce data size of unnecessary areas. This can be achieved by projecting a three-dimensional space which represents a ROI onto an image. To confirm the usefulness of this mechanism, we prototyped the ROI function in the WebP image format.**

Keywords—framework, image compression, data transmission, region of interest, WebP

I. Introduction

Recently, the price of devices which can be used for collecting information in public spaces, such as surveillance cameras installed at the side of a road and unmanned aerial vehicles (UAV) equipped with a camera, is decreasing. In the near future, the collected data from them will be widely used in Intelligent Transport Systems (ITS). For example, the data collected from sensors will be passed to a navigation system to estimate precise traffic conditions, or the data will be used to visualize blind areas around a vehicle based on diminished reality technology.

For the wireless communication studies recently, it is gathering attention that fifth-generation cellular wireless technology (5G) services have been launched in several countries, which is a new information communication infrastructure that enables non-congested data transferring of large-volume data. It is expected that 5G will provide a low-latency wireless data transmission channel [1]. Therefore, spreading the 5G network will make it possible to send data from sensing devices such as cameras and UAVs. However, mobile data traffic is increasing and the volume of mobile traffic in the 2020s would be at least 1000-fold larger than that of 2010 if the current rates of growth continue in the future [2]. In this point of view, data compression is required since transmitted data should be delivered with minimal delay for making it possible for application users to receive real-time fetched data. Moreover, it is supposed that transmission channel congestion may occur in some areas, which will cause high latency.

We are aiming at developing a system which can manage data acquired from sensing devices in practical use. Data in the framework is managed based on the time and space axes, hence we call it spatiotemporal data. In order to provide data delivery as a service, this system is proposed as an application framework, named SIGMA (Spatiotemporal Images with Generalized Management Architecture), in which developers can build various applications using spatiotemporal data. Additionally, the framework provides developers with various APIs to handle spatiotemporal data. These APIs will help developers with searching and processing spatiotemporal data. Spatiotemporal data stored and handled on this framework includes images and videos sent from camera devices on UAVs, roads, and vehicles and three-dimensional point cloud data measured by LiDAR sensors, which may be large in capacity.

For ITS-related studies recently, considering the vehicle communication system is the major interest, such as Vehicle-to-Vehicle (V2V) communication, Vehicle-to-Infrastructure (V2I) and Vehicle-to-Everything (V2X) communication. Conventional data storage servers supposed that they fetch data only from limited kinds of hardware that are configured when the server was newly established, hence it was hard to add an unsupported kind of hardware. In this situation, it is difficult to aggregate the servers which store different kinds of data. Along with that, if data from the sensing devices is not treated uniformly, it may not be able to distribute the data to each application in an optimal manner. We aim to solve problems caused by scattered data by fetching and delivering that kind of data. In addition, the framework server is supposed to have a high calculation capability. Since the various kinds of data gather in the framework server, the mechanism for processing a large amount of data will be required on it.

Our goal is to design data communication and flexible data compressing methods in SIGMA framework and enabling for framework users to develop various kinds of applications which employ spatiotemporal data. In this paper, we propose these methods and confirm their serviceability through experiments.

II. Related Work

There has been a significant interest in applying 5G communications for ITS applications. Kombate *et al.*

 978-1-7281-9802-6/20/$31.00 ©2020 IEEE

proposed the advantages of 5G communication model for implementing the Internet of vehicles environment based on the specification of the 5G technology [3].

Creating vehicular social networks provides a wide variety of application fields including entertainment applications [4]. Sánchez *et al.* proposed a framework for deploying cooperative services for improving the safety of vulnerable road users [5].

Thanks to the development in sensing technology, 3D point cloud data is generated easily. However, this kind of data need to be compressed due to the bandwidth problem. Wiemann *et al.* presented an approach to compress 3D environment models using Draco, the open-source compression library developed by Google [6]. As an approach to compress 3D environment data using existing data compression technology, Tu *et al.* applied compression methods based on MPEG and JPEG to point cloud data [7].

Considering applying the region of interest (ROI) is often done in the field of image compression. By specifying ROI to an important area of the data, non-ROI area will be compressed strongly and efficient compression can be achieved. For ITS applications, Xia *et al.* presented an approach which extracts ROI based on vision image processing in an ITS system [8]. Basri *et al.* proposed ROI adjusting method under heavy traffic condition [9]. The compressing data with ROI is not limited to the image files. Sandri *et al.* introduced the point cloud data compression incorporating ROI coding [10]. Conventional methods of applying ROI are used for the general two-dimensional or three-dimensional spaces. Hence, we need to design the ROI applying method that matches the concept of the framework.

III. Data Communication Method

In this chapter, we consider the data communication method in SIGMA framework.

A. Overview of Data Communication

Our framework needs to be able to store and deliver spatiotemporal data comprehensively. Therefore, it is important for the framework to provide data delivery methods according to the data's usage and/or type so that the framework is applicable for the advanced use of the data from various sensors. Accordingly, we argue the data communication method on the framework with the following assumptions:

- Application developers implement features which they provide using this application framework.
- Implementations related to the network connection are done with APIs provided by the framework.
- The Internet connection is used to connect between the framework server and each client.

In this study, we aim to make a design of the connection flow on our framework by selecting existing mechanisms such as networking protocols. This will lead to ease of implementing applications with our framework and usefulness of data sensed on the environment where the application runs on.

B. Applying UDP-based Connections

For sensing devices connected to the framework server, it is required to upload sensor data with a minimal delay so that applications developed with the framework can get such data immediately. With reference to the study for the Internet connection faster, the QUIC transfer protocol is developed by Google [11]. This protocol runs on UDP, hence it will help its quickness by avoiding the initial TCP handshake mechanism, which causes high latency. In order to achieve faster data transferring, it is suggested to use UDP-based file transfer since TCP has its essential delay including its flow control and congestion control process for fulfilling connection-oriented communication. We consider supporting this protocol for faster connections when clients have the support for that.

C. Connection Flows

The connection cases on the framework are expected to be shown in *Fig. 1.*

First, clients (connection nodes with sensing devices) upload spatiotemporal data including image files, video files, and three-dimensional point data files to the framework server (hereinafter, this is called "case A"). Next, clients (applications developed with the framework) send a search request to the framework server (hereinafter called "case B"). After the existence of the requested file is confirmed by case B, connections for downloading data to clients (applications) are generated (hereinafter called "case C").

Among these three cases, the quick deliverability is especially required in case A. In case A, uploading data fetched by sensing devices should be sent to the framework server. This is because the framework should respond to the demand for using the newly sensed data by the framework users.

In case C, the connection type between the framework server and clients are selected from the real-time oriented connection or the conventional connection based on the specification of data and the environment of the client.

IV. Data Compression Method

In this chapter, we focus on the data compression method. For the first step of considering the data compression method, we chose image files for the subject to be compressed.

A. Overview of Data Compression

Since data treated on the framework is assumed to be transmitted over the Internet and the qualities of the connection channel vary from client to client, data compression will be needed to them. For lossy compression, there is a trade-off relation between the data quality and data size. The compression on the framework is suggested to be flexible depending on the characteristics of each data, as the framework is designed to provide plural data type suitable for the demand of the client. Accordingly, we adopted the ROI technique, which is used for wide applications, for instance, image or three-dimensional measuring point group data files.

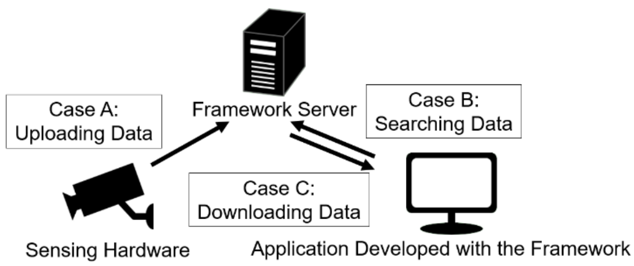

Fig. 1: Connection cases of the framework

978-1-7281-9802-6/20/$31.00 ©2020 IEEE

Since image files on the framework are subject to be used for various applications, these must be stored to the framework service in lossless or high-quality condition, which requires large file size. However, compressing the image files for the whole each image file will result in a bad image quality, which is not suitable for advanced use. On the other hand, delivering entire spatiotemporal files via the Internet will result in overspending the public communication channel, which causes the increasement of the cost of the server management.

Accordingly, we adopted the image compression method with the ROI function to the framework; thus, this will enable us to compress an image with plural compression parameters. For the first step of suggesting the application field of ROI, we decided that vehicles and road environments are subject to a supposed usage in this paper. With the ROI function, ITS-related applications can be widely developed.

B. Data Compression Provided on the Framework Server

The kinds of data provided on the framework server are suggested as below. The clients are allowed to use one of them reflecting their demands.

- *Original data:* The framework server sends the data stored on the server's database without any modification. This is used for applications which require high-quality data for their functions and don't have a limitation of processing time, or the data is not capable of compression.
- *Lossless compressed data:* This is applicable when the client needs high-quality data and the data type is able to be compressed with the existing compression methods. The connection time and bandwidth consumed will be reduced under the limited connection environment.
- *Lossy compressed data:* This type of data on the framework is broadly classified into compressing the whole data with one parameter and compressing the data with several parameters. The latter case includes ROI-enabled compression with axis and quality parameters.

For providing lossy compressed data, the ways to specify the ROI area are suggested as below.

- *Applying moving object detection:* This method is assumed to be used in the roadside camera module. Based on the image of the road with no vehicle or pedestrian, the region of vehicles and pedestrians can be detected.
- *Applying vehicle detection system:* With the analytics engine of vehicle detection [12], the specific region of vehicles is able to be decided.
- *Combining with detection of suspicious vehicles:* More than methods mentioned above, detection approaches of suspicious or dangerous vehicles are useful especially for violative vehicle photographing systems. Detecting over speeding vehicles is a typical example [13][14].
- *Applying a license plate recognition system:* Likely the suspicious vehicle detection, it is effective for the enforcement systems for traffic violations. In addition, the application of vehicle searching is suggested with image files fetched from camera hardware installed in city streets.
- *Applying object detection based on LiDAR on vehicles or distance sensors:* When this kind of hardware is equipped on vehicles for consumers, the ROI area can be selected for cameras installed on vehicles.
- *Selecting the road area of the camera's field of vision:* This method is effective for roadside cameras and other fixed camera hardware. The ROI in this way should be configurated and sent to the framework when the camera is newly installed or moved to the current place.
- *Request the application user to select the ROI area:* This approach is supposed to be used in the situation that the connection bandwidth is exceedingly limited. The workflow is as follows; First, the user gets the image which is sorely compressed. Then, the user selects a ROI area and sends it. Finally, the framework compresses the original image based on the ROI information sent.

Since these techniques enable the framework to reduce the bandwidth spent without extra processing such as setting compression parameters manually according to the weather or time of the day, the convenience of ROI feature on the framework is improved.

The ROI feature will be implemented on the framework server. This will be helpful for IoT hardware with minimal computing power by cloud computing. Additionally, developing applications with the framework will be easy since ROI-related implementation is not required on the clients.

C. Specifying ROI to Spatiotemporal Axes

Our framework has a function to extract data which is facing a specified area from the spatiotemporal database. Using this function, we implemented a mechanism to specify ROI on the spatiotemporal axes on our framework. Since the framework gathers a huge number of spatiotemporal data from many sensors, it is essential to specify ROI in the spatiotemporal axes. *Fig. 2* shows an example of specifying ROI in the spatiotemporal axes in which a cubic shape represents a three-dimensional ROI space and it is projected onto the camera image to specify the ROI area. This will make it possible for plural sensors to use the same ROI space. For instance, if the framework is used for developing a three-dimensional image reconstruction application for the accident black spots, only a tiny part of each image file is required for three-dimensional image reconstruction. The database connected to the framework server can extract sensor devices whose data include ROI areas. We implemented several APIs to filter data by specifying spatiotemporal axes in the real world.

D. Implementation Details of ROI Feature

By compressing data on the framework with the ROI feature, data communication can be made efficient. In this study, we chose to compress image files with ROI feature based on WebP lossy image compression [15]. This is because WebP image format is released as open-source and made for compressing image files better than conventional image formats such as JPEG or GIF using some techniques including adopting the prediction coding. The original WebP image compression does not implement ROI feature, therefore we implemented that to WebP lossy compression by reusing the existing segments system on the WebP. The segment is originally used for adaptive block quantization, which yields efficient compression by redistributing data to the complex area of the image. WebP lossy compression is based on VP8 video compression format, whose bitstream is designed to have a maximum of four segments [16].

We implemented the ROI feature on the WebP lossy image format by allocating the specific area of the image data to the first segment which is used for the ROI area. For this segment, compression parameters for higher quality are given. On the other hand, compression parameters for lower quality are given for the second segment which is used for the non-ROI area for data reducing with the unimportant area of the image. This method was implemented by modifying the quantization part of the libwebp-1.0.3 WebP library. Macroblocks which have ROI area are set to the first segment, while the other macroblocks are set to the second segment.

V. Experiments

For the first step of considering the data compression method with ROI on the spatiotemporal axes, we chose image files as the subject to be compressed. In this experiment, we specified a three-dimensional ROI by a cubic shape as a simple model. By projecting the cubic shape onto the image, we got a rectangle ROI area. We conducted an image compression experiment with the implemented ROI-supporting WebP image encoder. We compressed an original image with the ROI space which includes a license plate of a vehicle. Then, we also generated another compressed image without the ROI space for comparison which has almost the same file size of the ROI-specified image by "-size" option implemented on an official libwebp library.

The cropped images of the license plate are shown in *Fig. 3*. For validation of the image quality, we calculated the structural similarity (SSIM) index between the original image and compressed images, with or without ROI.

Based on the results of compressed images, we have concluded that image compressing with the ROI feature can compress image files lossy while keeping the image quality of ROI. Under the almost same file size, the cropped field of the ROI-enabled result was extremely close to 1.0 in the SSIM index, which means that the quality of the original image is represented on the compressed image. On the other hand, the result of non-ROI compressed indicated the SSIM of 0.8365 which shows the loss of the image quality. Therefore, this ROI feature is effective as a method for communicating the stored data sensed from the real world without losing the important part of it.

For the validation of the effectiveness of the ROI-enabled image compression for plural image files, we applied the ROI-enabled compression for 78 PNG image files (60,207,153 bytes in total) of a clock tower *(Fig. 4)*. We manually set the ROI area for each image file. As a result, the set of the compressed image file was 6,191,306 bytes. We confirmed that each image file of the result has a ROI, where the image quality of the clock tower is kept as the original image file.

VI. Conclusion

This paper describes data communication and compression methods for our application framework which provides developers with various APIs for advanced use of the spatiotemporal data. For the data communication method, we argued that data transmission with the UDP-based connection is effective for real-time data uploading to the framework server. For the data compression method, we verified the useableness of image encoding with the ROI feature on spatiotemporal axes. In the future, we will design and implement ROI feature for three-dimensional point cloud data to achieve efficient communication on the framework.

Acknowledgement

This work was supported in part by a Grant-in-Aid for Scientific Research (B), Grant Number 17H01747.

References

[1] J. M. Khurpade, D. Rao, and P. D. Sanghavi, "A survey on IOT and 5G network," in *Proceedings of 2018 International Conference on Smart City and Emerging Technology (ICSCET)*, 2018, pp. 1–3.

[2] NTT DOCOMO, INC.: DOCOMO 5G White Paper 5G Radio Access: Requirements, Concept and Technologies, https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/whitepaper_5g/DOCOMO_5G_White_Paper.pdf, 2014. (Accessed: Aug. 24, 2020.)

[3] D. Kombate and Wanglina, "The Internet of Vehicles based on 5G communications," in *Proceedings of 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016, pp. 445–448.

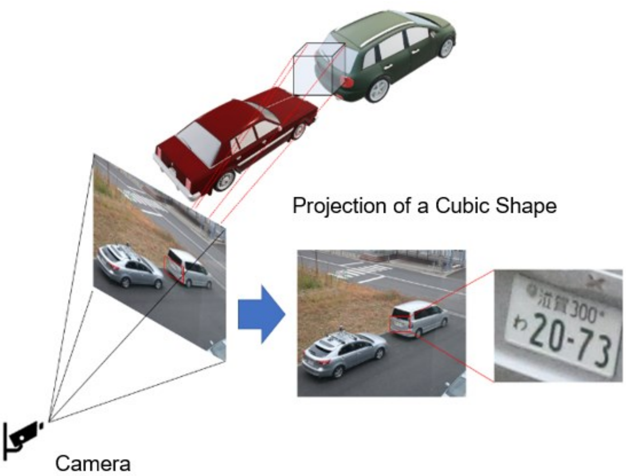

Fig. 2: Specifying ROI in spatiotemporal axes

SSIM: 1.0 (a) Cropped image | SSIM: 0.8365 (b) ROI OFF | SSIM: 0.9888 (c) ROI ON

Fig. 3: Experimental results: the license plate

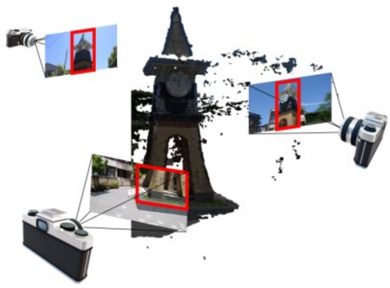

Fig. 4: A set of image files for the experiment

978-1-7281-9802-6/20/$31.00 ©2020 IEEE

[4] A. M. Vegni and V. Loscrí, “A survey on vehicular social networks,” *IEEE Communication Surveys & Tutorials*, vol. 17, no. 4, pp. 2397–2419, 2015.

[5] N. Sánchez, J. Alfonso, J. Torres, and J. M. Menéndez, “ITS-based cooperative services development framework for improving safety of vulnerable road users,” *IET Intelligent Transport Systems*, vol. 7, no. 2, pp. 236–243, 2013.

[6] T. Wiemann, F. Igelbrink, S. Pütz, M. K. Piening, S. Schupp, S. Hinderink, J. Vana, and J. hertzberg, “Compressing ROS sensor and geometry messages with Draco,” in *Proceedings of 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019, pp. 243–248.

[7] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, “Compressing continuous point cloud data using image compression methods,” in *Proceedings of 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1712–1719.

[8] Y. Xia, M. Huang, X. Chen, and W. Li, “A region of interest extraction approach of on-Road object,” in *Proceedings of 2009 WRI World Congress on Computer Science and Information Engineering*, vol. 6, 2009, pp. 519–522.

[9] Basri, Indrabayu, and A. Achmad, “Gaussian mixture models optimization for counting the numbers of vehicle by adjusting the region of interest under heavy traffic condition,” in *Proceedings of 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 245–250.

[10] G. Sandri, V. F. Figueiredo, P. A. Chou, and R. de Queiroz, “Point cloud compression incorporating region of interest coding,” in *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 4370–4374.

[11] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, “The QUIC transport protocol: Design and internet-scale deployment,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM ’17)*, 2017, pp. 183–196.

[12] B. F. Momin and T. M. Mujawar, “Vehicle detection and attribute based search of vehicles in video surveillance system,” in *Proceedings of 2015 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2015)*, 2015, pp. 1–4.

[13] A. Tungkastan, N. Jongsawat, and W. Premchaiswadi, “A proposed framework for automated real-time detection of suspicious vehicles,” in *Proceedings of 2014 Twelfth International Conference on ICT and Knowledge Engineering*, 2014, pp. 90–93.

[14] T. Umedu, K. Isu, T. Higashino, and C. K. Toh, “An intervehicular-communication protocol for distributed detection of dangerous vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 627–637, 2010.

[15] Google Inc., “A new image format for the Web,” https://developers.google.com/speed/webp/ (Accessed: Aug. 24, 2020.)

[16] J. Bankoski, P. Wilkins, and Y. Xu, “Technical overview of VP8, an open source video codec for the web,” in *Proceedings of 2011 IEEE International Conference on Multimedia and Expo*, 2011, pp. 1–6.

 978-1-7281-9802-6/20/$31.00 ©2020 IEEE