High-resolution Multi-camera Analysis Infrastructure to Support Future Smart Cities

Keita Mikami, Ryosuke Kurebayashi, Xu Shi, Noriaki Inoue, Yoshinori Matsuo, and Ikuo Yamasaki

Abstract

In the IOWN (Innovative Optical and Wireless Network) era, smart cities will be built on a cyber-physical system (CPS) that makes all information in the city valuable and accessible. In this article, we introduce an artificial intelligence (AI) inference infrastructure that can efficiently process high-resolution, multi-camera images to support an urban-scale CPS. This infrastructure is based on the concept of event-driven inferencing to significantly reduce the processing loads and energy consumption of AI-inference processing executed with the infrastructure. The basic techniques of model cascading and inference resource sharing are discussed in the article.

Keywords: IOWN, smart city, AI

1. Smart city and cyber-physical system

When you hear the term smart city, what kind of city comes to mind? According to the definition by the Ministry of Land, Infrastructure, Transport and Tourism [1], a smart city is "a sustainable city or district that is managed to achieve total optimization while utilizing new technologies such as information and communication technology (ICT) to address the various issues facing the city." The Nomura Research Institute's definition [2] more specifically defines a smart city as "one that collects and integrates a variety of data, such as environmental data, facility operation data, consumer attributes, and behavioral data, through sensors, cameras, smartphones, and other devices installed throughout the city, and uses artificial intelligence (AI) to analyze the data, as well as remotely controlling facilities and equipment as necessary, with the aim of optimizing urban infrastructure, facilities, and operations, and improving convenience and comfort for businesses and consumers." Both definitions aim to achieve total optimization by using ICT technology to manage and operate a city. One concept, the goal with which is to achieve optimization by processing information from the real world, is the cyber-physical system (CPS). As shown in **Fig. 1**, a CPS takes information from the real world (physical), passes it to a virtual space (cyber), analyzes it through computing, then feeds back the results of the analysis to the real world to optimize real-world operations.

What will happen when reality becomes programmable, i.e., the *softwarization* of reality? In fact, the softwarization of reality is already happening everywhere. Take the telephone, for example. In the past, a telephone was a box with a handset and a dial, and cell phones, which have become smaller and more wireless, were initially just terminals to connect remote locations. However, the advent of the smartphone, a softwarized phone, changed the world. Physical buttons became icons on touchscreens, which eventually transformed freely into all sorts of palm-sized interfaces (phones, calculators, books, cameras, etc.). E-books and digital cameras are examples of books and cameras that have been softwarized. As a result, the static, one-way interface of

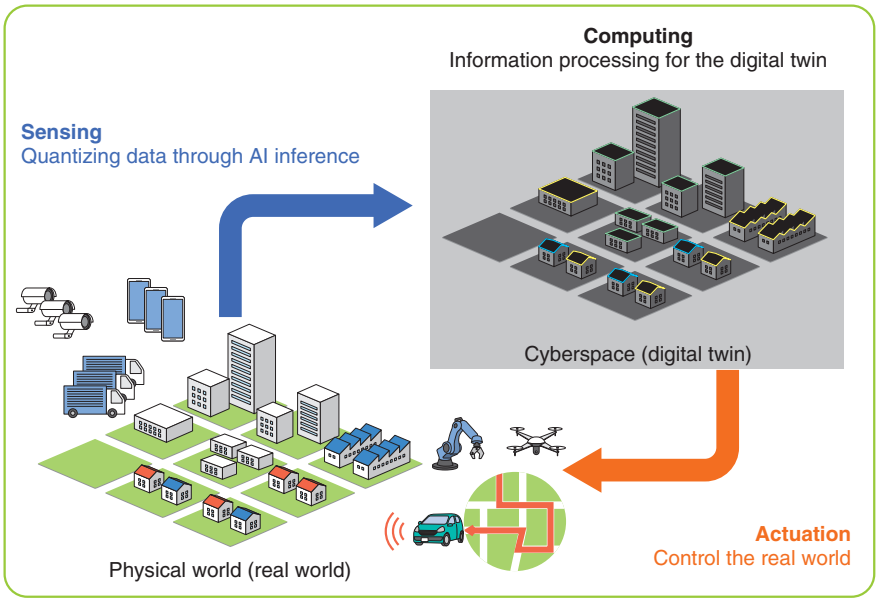


Fig. 1. A CPS.

the past has been transformed into a dynamic, interactive one, and the information presented can be personalized and recommendations can be made to suit the recipient. This yields dramatic improvements in convenience and comfort.

Let us return to the topic of smart cities. Yes, a smart city is a *softwarized city*. The building blocks of a city had been physical such as concrete buildings, metal signs, and transportation systems. In a smart city, however, buildings will be smart buildings controlled by a building management system, signs will become digital signage, and transportation will have smart mobility such as connected cars, all of which are programmable. If cities become software-driven, always-crowded roads may be replaced by smart roads that optimize lanes and speed limits to optimize traffic conditions, and buses that seldom come may be replaced with self-driving cabs that stop in front of you the moment you want to go somewhere.

By constructing a city-scale CPS and using it to promote the softwarization of various urban services, we can expect dramatic changes and improvements in convenience and comfort on a city scale, similar to what happened when telephones became smartphones.

2. The key is to reduce the processing loads of and energy consumed by AI-inference processing

The question is, then, whether a city-scale CPS, which is the foundation supporting smart cities, can be readily implemented. A CPS consists of three major steps: sensing, computing, and actuation, and a variety of research is currently underway for each of these steps. The NTT Software Innovation Center (SIC) is developing an AI-inference infrastructure for sensing and computing. Sensing in a smart city is executed by analyzing a large amount of stream data continuously generated from cameras and various sensors placed throughout the city, as well as from connected cars and smartphones in the city, using inferencing based on deep learning (so-called AI inferencing) and converting the data into meaningful information. AI inferencing is also essential for reconstructing information in the form of a *digital twin* on a computer system and for computing the feedback that yields the desired results in the real world.

Traditionally, AI inferencing is a process that incurs excessive computation loads. Our solution is to develop a combination of stream merger and GPU (graphics processing unit) offloading to improve efficiency and capacity. However, to achieve the Innovative Optical and Wireless Network (IOWN) concept of implementing AI systems that can capture events

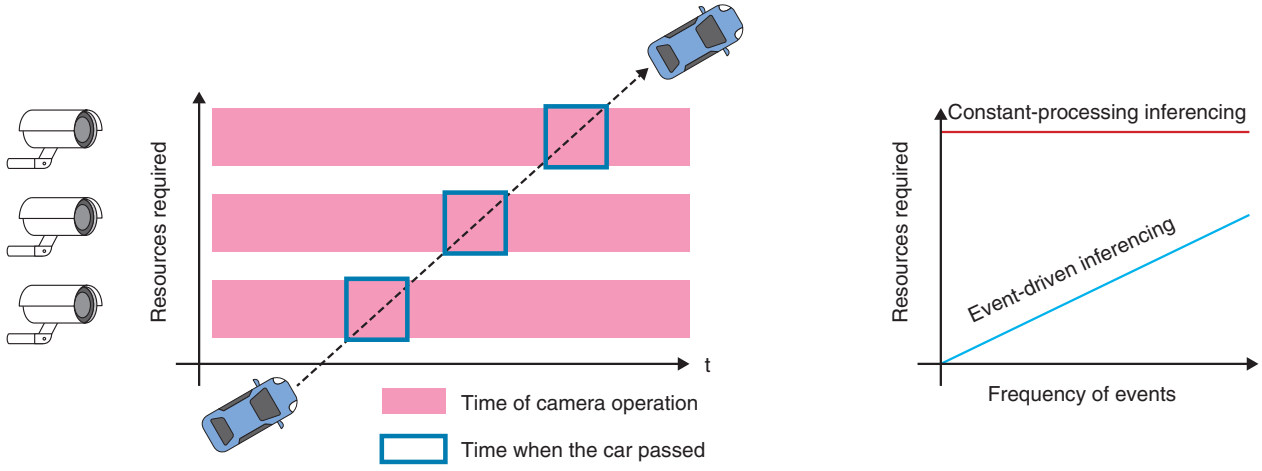


Fig. 2. Concept of event-driven inferencing.

on a scale that cannot be handled by humans and analyze and make decisions at a speed that exceeds that of humans, it is necessary to increase the resolution and frames per second (FPS) of input data and support the sheer number of cameras and sensors that the city scale demands [3]. In general, the processing loads of and energy consumed by AI inferencing are proportional to the amount of data to be analyzed, so an explosive increase in the amount of input data leads directly to an explosive increase in processing loads and energy consumption. To achieve a city-scale CPS that supports smart cities in the IOWN era, it is necessary to reduce the processing loads and energy consumed to a sustainable level.

3. Concepts and elemental techniques of event-driven inferencing

To reduce the processing loads and energy consumption described above, we are developing the implementation concept of AI inferencing of stream data called *event-driven inferencing*. Event-driven inferencing makes the processing loads incurred by AI inferencing proportional to the amount of valuable information rather than the amount of input data (**Fig. 2**).

Conventional constant-processing AI inferencing treats all frames as equally important, so the processing loads and energy consumption depend on frame characteristics (resolution, FPS, number of streams, etc.). Therefore, increasing the resolution, FPS, and number of cameras and sensors deployed will directly increase the processing loads and energy consumption. At first glance, this may seem unavoidable, but consider the mechanism of human cognition. Humans use event-driven cognition, which lightly monitors the whole field of perception, and when a significant event (such as a sudden movement or sound) is noticed, closer attention is paid to that event. In this case, the processing loads and energy consumed for cognition depend on the number of objects in the field of perception important to the individual, i.e., the amount of valuable information. Event-driven inferencing involves the same principle to make AI-inferencing implementation practical.

There are several possible approaches to achieving event-driven inferencing. A typical approach is model cascading. Model cascading analyzes the same frame in multiple steps, which is detailed in the next section. Other approaches include temporal control and spatial control. In temporal control, the analysis parameters of subsequent frames are controlled on the basis of the analysis results of the previous frame. For example, in temporal control, the analysis is executed at 5 FPS in normal operation, and only when a person is detected, the frequency of analysis is increased to 15 FPS. Spatial control involves the topology of cameras and sensors and uses the analysis results of one input source to control the analysis parameters of other input sources. For example, "Analyze the camera images in the area only where a person is detected by the camera at the entrance of the monitored area."

To actualize this concept, SIC is investigating model cascading and inference-resource sharing.

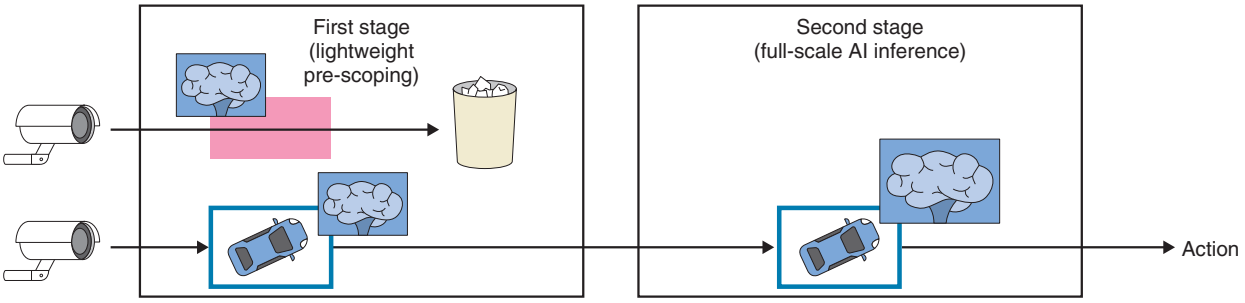


Fig. 3. An example of model cascading using a pre-scoping model.

4. Model cascading

To achieve event-driven inferencing, events must be detected in some way. Model cascading combines a lightweight pre-scoping model for event detection in the first stage with an AI-inferencing model for full-scale analysis in the second stage; the AI inferencing model is activated in the second stage only when an event occurs, thereby reducing overall processing loads (**Fig. 3**). There are several variations of pre-scoping [3], including one that uses event detection to winnow the frames sent to the second stage, one that uses a lighter, lower-resolution model to infer the same task as the second stage and sends it to the second stage only when the confidence in the processing result is low, and one that divides AI inferencing into separate parts and uses the first part for pre-scoping and sends the intermediate output to the second stage. The optimal configuration will vary depending on the use case and hardware configuration.

5. Inference-resource sharing

Inference-resource-sharing technology is needed when the valuable information-dependent approach is used for processing in event-driven inferencing. Conventional constant-processing AI inferencing requires relatively constant amounts of hardware (resources), so it is easy to accommodate the demands. In event-driven inferencing, on the other hand, the required hardware resources change depending on the event, and it is necessary to prepare resources that can cope with an increase in load due to a concentration of events, but it is undesirable to leave resources reserved for peak times idle during normal times.

Server-oriented techniques (e.g., Triton Inference Server [4], KServe [5]) can be used to dynamically allocate hardware resources to suit changing inference requirements, but such techniques are generally designed for specific on-demand use cases. At SIC, we are researching and developing inference-resource sharing, which is an extension of server-oriented techniques for real-time stream processing. By using inference-resource sharing, it becomes possible to share inferencing resources among streams and obtain statistical multiplexing benefits by bundling multiple streams with different peaks in real-time stream-processing use cases.

6. Future directions

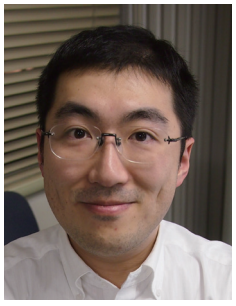
In this article, we introduced the world view of a software-driven city (i.e., a smart city) achieved with a city-scale CPS, AI-inferencing infrastructure that can efficiently process high-resolution and multi-camera images, concept of event-driven inferencing, and its basic techniques of model cascading and inference-resource sharing.

By using model cascading and inference-resource sharing, and by introducing the concept of event-driven inferencing, we can expect to significantly reduce processing loads and energy consumption to practical levels, enabling the implementation of AI inferencing required by smart cities in the IOWN era. By accumulating the basic techniques for IOWN, we can also implement AI systems that can analyze and make decisions faster than humans. We will then create services that are safer, more accessible, more sustainable, and more comfortable for everyone and solve a variety of social issues.

References

[1] The website of Cabinet Office, Government of Japan, “Smart City” (in Japanese), https://www8.cao.go.jp/cstp/society5_0/smartcity/index.html

[2] Nomura Research Institute, “Smart City” (in Japanese), https://www.nri.com/jp/knowledge/glossary/lst/sa/smart_city
[3] T. Eda, R. Kurebayashi, X. Shi, S. Enomoto, K. Iida, and D. Hamuro, “An Efficient Event-driven Inference Approach to Support AI Applications in IOWN Era,” NTT Technical Review, Vol. 19, No. 2, pp. 41–46, Feb. 2021. https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr202102fa4.html
[4] Triton Inference Server, https://developer.nvidia.com/nvidia-triton-inference-server
[5] KServe, https://kserve.github.io/website/



Keita Mikami
Senior Research Engineer, AI Application Platform Project, NTT Software Innovation Center.
He received a B.S. in information science from Waseda University, Tokyo, in 2004 and M.S. in information engineering from Faculty of Science and Engineering, Waseda University in 2007. He joined NTT in 2007, and his current research interests include deep learning and computer vision. He is a member of the Information Processing Society of Japan (IPSJ).

Ryosuke Kurebayashi
Senior Research Engineer, Supervisor, AI Application Platform Project, NTT Software Innovation Center.
He received a B.S., M.S., and Ph.D. from University of Tsukuba, Ibaraki, in 1998, 2000, and 2003. He joined NTT in 2003, and his research interests include computing and networking technologies for Internet of Things and AI. He is a member of IPSJ and the Institute of Electronics, Information and Communication Engineers (IEICE).

Xu Shi
Engineer, AI Application Platform Project, NTT Software Innovation Center.
She joined NTT in 2014. Her current research interests include deep learning and computer vision.

Noriaki Inoue
Research Engineer, AI Application Platform Project, NTT Software Innovation Center.
He received a B.S. and M.S. from Osaka University in 1995 and 1997 and joined NTT in 1997. His current research interests include computing platform, deep learning, and computer vision.

Yoshinori Matsuo
Senior Research Engineer, Supervisor, AI Application Platform Project, NTT Software Innovation Center.
He received a B.S. and M.S. in physics from Tokyo Institute of Technology in 1999 and 2001. He joined NTT Cyber Space Laboratories in 2001, and his research interests include a wide range of topics in network security and system engineering.

Ikuo Yamasaki
Senior Research Engineer, Supervisor, AI Application Platform Project, NTT Software Innovation Center.
He received a B.S. and M.S. in electronic engineering from the University of Tokyo in 1996 and 1998. He joined NTT in 1998 and was involved in research and development (R&D) activities regarding OSGi. In 2006, he was a visiting researcher at IBM Ottawa Software Laboratories, IBM Canada. Currently, he is leading and managing R&D activities related to the AI application platform at NTT laboratories.