A hardware intelligent processing accelerator for domestic service robots

doi: https://doi.org/10.1080/01691864.2020.1769726

To appear in *Advanced Robotics*
Vol. 00, No. 00, January 2013, 1–14

FULL PAPER

A Hardware Intelligent Processing Accelerator for Domestic Service Robots

Yutaro Ishida[a], Takashi Morie[a] and Hakaru Tamukoh[a]

[a] *Department of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, Japan*

(*v1.0 released April 2019*)

We present a method for implementing hardware intelligent processing accelerator on domestic service robots. These domestic service robots support human life; therefore, they are required to recognize environments using intelligent processing. Moreover, the intelligent processing requires large computational resources. Therefore, standard personal computers (PCs) with robot middleware on the robots do not have enough resources for this intelligent processing. We propose a "connective object for middleware to an accelerator (COMTA)," which is a system that integrates hardware intelligent processing accelerators and robot middleware. Herein, by constructing dedicated architecture digital circuits, field-programmable gate arrays (FPGAs) accelerate intelligent processing. In addition, the system can configure and access applications on hardware accelerators via a robot middleware space; consequently, robotic engineers do not require the knowledge of FPGAs. We conducted an experiment on the proposed system by utilizing a human-following application with image processing, which is commonly applied in the robots. Experimental results demonstrated that the proposed system can be automatically constructed from a single-configuration file on the robot middleware and can execute the application 5.2 times more efficiently than an ordinary PC.

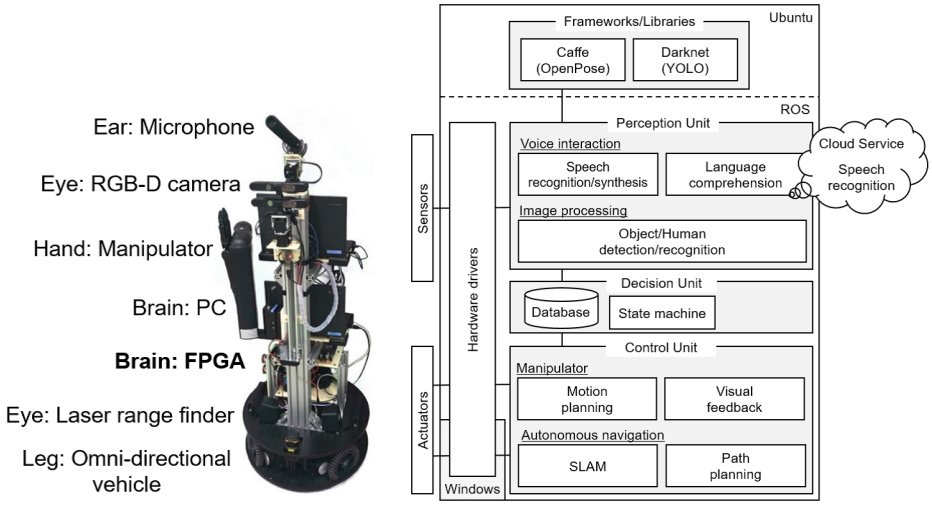
Keywords: Robot Operating System (ROS); field-programmable gate array (FPGA); domestic service robot; intelligent processing; RoboCup@Home;

1. Introduction

In various countries around the world, especially in Japan, the declining birth rates and aging populations have become a social problem. In such a social condition, the shortage of workers results in the collapse of social systems. Therefore, domestic service robots are expected to enrich the life by supporting the housework of the workers generation and take care of the elderly generation [1].

"Exi@" [2], which is a domestic service robot that we developed, is shown in Figure 1(a). Additionally, a block diagram of the software in the robot is shown in Figure 1(b). The robot was researched and developed through RoboCup@Home competitions [3, 4]. RoboCup@Home, which is an annual competition first held in 2006, is the largest worldwide competition for domestic service robots. In this competition, robots perform practical applications in residential and public environments, e.g., they can act as a housekeeper.

To obtain such applications, the robots are equipped with sensors and actuators which can accordingly approximate the human senses and motor functions (Figure 1(a)); the software application comprises perception, decision, and control units for processing sensory data and control actuators. In these units, a significant number of intelligent processes run at the same time. In addition, most processes have to be performed using high frequency based on the sensing/controlling frequency. Moreover, in ordinary robot systems, this software application can run on a personal computer (PC) connected to the robot. Thus, computing resources are always insufficient.



(a) (b)

Figure 1. Domestic service robot "Exi@": (a) hardware and (b) software block diagram

In addition, human/object detection/recognition technologies that utilize deep learning have been implemented on most robots [5, 6]. Graphics processing units (GPUs) are computing devices used to execute deep learning at high frequency. However, GPUs consume high power and exhaust heat. These cause problems by decreasing the operation time with the robots' limited battery and raising the temperature of the robots' body. Furthermore, cloud services are another way to implement deep learning and provide high-quality services [7, 8]. Even though most of the important functions of robots are implemented by cloud services, robots cannot operate when they are offline.

Since many different types of hardware and software must be integrated, robots' intelligent processing is becoming complex. For example, robotic engineers must consider relatively simple device drivers and much more complex perception-level software. This requires multiple technologies and is extremely time-consuming. Consequently, robot middleware which supports open sources, large-scale software integration, and prototyping is extensively applied in robots [9–14].

For these reasons, domestic service robots need a processing system that can perform intelligent processing at high speed with accelerators having dedicated architecture. Moreover, the system must satisfy robot hardware requirements, such as battery constraints and embedded implementation, and it should be easy to integrate with robot middleware. The expected applications of the system are robots' perception, such as image processing.

Thus, we propose an efficient (i.e., high-speed computation and low-power consumption) processing system that uses field-programmable gate arrays (FPGAs), which can be integrated with the robot middleware. To configure arbitrary architecture of FPGAs, the proposed system achieves a highly-efficient processing system by taking advantage of the high parallelism and the flexibility. In addition, the system demonstrates the effectiveness of easy integration by providing an interface between robot middleware and FPGAs. As far as we know, this has not been proposed before. Using the proposed system, our experimental results show the effectiveness of practical domestic service robots' application. The contributions of this paper are given by the following.

- The presentation of the proposed system implementations by utilizing a "connective object for middleware to an accelerator (COMTA)."
- The proposed system easily integration a hardware accelerator and robot middleware or sensors for intelligent image processing.
- The proposed system is evaluated in a practical human-tracking example of an FPGA accelerator applied in domestic service robots.

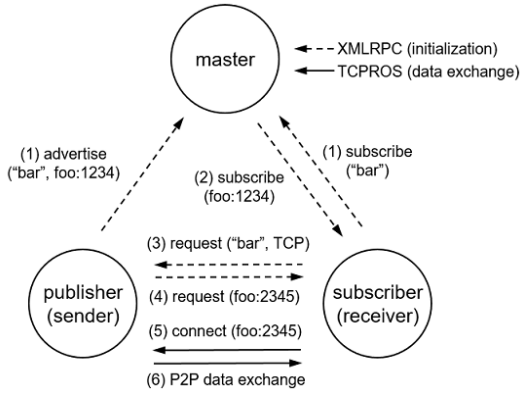


Figure 2. Typical Robot Operating System (ROS) Interface

2. Related Studies

2.1 *Robot Middleware*

A middleware mediates between an operating system and an application, and it provides data communication/management and debugging functions. Recently, various robots' middlewares have been developed, e.g., ROS [9, 10], OpenRTM-aist [11, 12], and V-Sido OS [13]. ROS has many users in the world [14]; it has been adopted by numerous research institutes and companies. Consequently, many robotic systems, which range from research to commercial robots, are implemented using ROS. ROS has attracted attention because it is easy to integrate using its own unified interfaces.

Figure 2 illustrates typical ROS interfaces relative to a publish/subscribe messaging model. It is noteworthy that ROS interfaces are based on a network connection. ROS has a master that manages the entire system. For this master, the publisher and subscriber register their request for data exchange by utilizing namespace (Figure 2(1)). If the name matches, then data exchange begins with a peer-to-peer (P2P) connection (Figure 2(2–6)). Therefore, users only need to manage the namespace for an automatic system integration.

For robotic engineers, it is important to automatically activate a processing system through the ROS interfaces to easily apply an accelerator. Meanwhile, the ROS interfaces use large computational resources for network communication in embedded systems, becoming a reason for intelligent processing to slow down [15]. Thus, an improved implementation for the embedded systems is required while leaving the goodness of the ROS interface.

2.2 *Domestic Service Robots*

An example of domestic service robots is Toyota human support robot (HSR) [1]. This robot approximates human senses and motor functions like Exi@, and it is expected to work on independent living, remote care, and housework support. Using sensors and controls actuators, a computer system on this robot executes various recognition tasks. This robot equips an embedded PC with Intel©Core i7-4700EQ 2.4 GHz. The domestic service robots can also be called mobile manipulators. One example of a mobile manipulator is the fetch mobile manipulator [16]. This robot has a similar hardware configuration as the Toyota HSR but is a bit large. The robot targets commercial applications, which are also available for research and development. The robot is equipped with Intel©i5 Haswell as a computer system. PAL Robotics Tiago has been mentioned as a different mobile manipulator [17]. This robot also has a similar hardware configuration as Toyota HSR, and the computer system is equipped with Intel©i5/i7 Haswell.

The computer system of domestic service robots or mobile manipulators is mainly a system with the computing ability of Intel Core i series scale. Toyota HSR and Tiago are considered to

develop applications using an external PC, and it shows that the computer resources which are implemented on the robot are insufficient. A seven degree of freedom arm of the fetch mobile manipulator consumes only 35 W even at maximum payload, showing that the computer system cannot consume over two digits of power. In addition, robots that are studied and developed at RoboCup@Home, similar to Exi@, are equipped with an Intel i series central processing units (CPUs) [18] and Nvidia GeForce series GPUs [19]. This demonstrates that the scale of computer resources required for domestic service robots is large. Consequently, an accelerator that can cooperate with such existing computer resources and offload large-processing loads into the external accelerator is required for the robots.

2.3 *Human Following*

Human following is an important application for domestic service robots. It can be utilized to carry an operator's luggage when the operator is following behind. Some recently conducted studies on human following through robots applied online Adaboost [20] and convolutional neural networks (CNNs) [21]. In addition, there is a study of the human following by applying Toyota HSR [22]. This study applies face recognition using the histogram of oriented gradients features and CNNs, and it also applies skeleton detection using OpenPose. Thus, for running the perception with high frequency, CPUs and GPUs that have large-computing resources such as Intel Core i series and Nvidia GeForce series should be used.

Moreover, as aforementioned, these large computer systems prevent long-term operation of robots regarding their battery and heat. Therefore, a hardware accelerator, which has higher speed and lower power consumption than that of ordinary systems, is needed while maintaining the same processing capacity.

2.4 *Accelerators Suitable for Robots*

GPUs, application-specific integrated circuits (ASICs), and FPGAs are all considered to be accelerator devices instead of CPUs. GPUs enable parallel computation by exploiting many computing cores, and an ASIC is an integrated circuit (IC) chip optimized for a particular application. In addition, FPGAs are reconfigurable digital circuits that can operate at a high speed and have low-power consumption. Each of the characteristics is presented in Table 1.

Based on the benchmark results obtained using the CNN [23], FPGAs are more superior to embedded CPUs and mobile GPUs regarding both calculation speed and calculation performance per unit power. In addition, FPGAs generate less heat because of their low-power consumption; thus, they are reliable relative to thermal considerations in embedded environments. In this regard, the development efforts in man-hours required to implement them are considerably greater than those of the CPUs and GPUs [24]. A lot of time is required to prepare interfaces, check the operation of the FPGA, and implement an application on an FPGA takes time.

ASICs have an advantage over FPGAs with regards to computing performance and power consumption. In addition, for mass-produced applications, ASICs are also advantageous regarding cost compared to FPGAs. Since ASICs cannot be reconfigured, they cannot update an application after it has been implemented in a robot.

Herein, we consider an accelerator that is most suitable for robots. In general, robots utilize a limited power supply, such as a battery. Therefore, it is difficult to implement GPUs, which have high-power requirements, in robots. Since GPUs generate heat, it is very important to attach a cooling device to robots; otherwise, thermal throttling can reduce their processing performance. Furthermore, intelligent processing is progressing; thus, in a robot, functions which cannot be updated become obsolete after implementation. Robots' behavior is managed by the state machine, and the accelerator must be able to configure and deploy the process which is needed for current behavior.

FPGAs are considered to be suitable accelerators for robots in consideration of these factors.

Table 1. Comparison of Accelerators

Accelerators	High Processing Speed	Low-Power Consumption	Reconfigurable
CPUs/embedded CPUs			✓
GPUs	✓		✓
AISCs	✓	✓	
FPGAs	✓	✓	✓

However, the number of development efforts in man-hours is still a problem. Therefore, it is important to propose a high-affinity interface, which can connect FPGAs and robot middleware by reducing the workload burden for both circuit and robot engineers.

2.5 *Accelerator Applications*

A previously conducted study applied FPGAs in hardware/software (hw/sw) complex system [25], which combines hw (an FPGA) and sw (a CPU). High-speed parallel computation is performed for processes that FPGAs can efficiently execute, such as signal processing. Moreover, processes that FPGAs cannot efficiently execute are performed by CPUs, such as complex conditional branches. Consequently, this combines the advantages of FPGAs and CPUs by enabling the high-speed operation and reducing the power consumption of the entire system. Recently, system on chip (SoC), which is an IC that integrates an embedded CPU and an FPGA on a single chip, has been designed [26], and this has attracted a high level of attention among various researchers.

A previously conducted study integrated the SoC and ROS, i.e., ROS compliant FPGA components [15, 27, 28]. The study showed that ROS is implemented in the embedded CPU on the SoC, while intelligent processing is implemented in the FPGA on the SoC. However, from the perspective of computing resources, it is unrealistic to compute large-scale intelligent processing, which is installed in domestic service robots, using only the SoC. This is because these studies pointed out that even communication through the ROS interface consumes almost the same computing resources. Hence, it is important to propose a system which can integrate both the SoC and ROS installed on PCs.

A previously conducted study integrated the SoC and ROS installed on a laptop PC, i.e., FPGA ROS [29]. In the study, the SoC, laptop PC, and camera are connected to a robot, while the camera and robot are connected to the laptop PC. In the FPGA on the SoC, object detection/recognition is implemented, and the robot is controlled based on its results. In the system, the laptop PC is loaded to drive the camera, transfer an image to the SoC, and received the result from the SoC. To offload an additional processing load from the laptop PC to the SoC, the camera driver is removed from the laptop PC to the embedded CPU. Therefore, it is important to propose a system with a short-data path between sensors and the SoC.

3. Hardware-Accelerated ROS Package

As depicted in Figure 3, we propose a COMTA [30, 31], which is a hardware-accelerated intelligent processing system for domestic service robots. We now organize the problems to be solved in the proposed system. Using hardware description language (HDL) or high-level synthesis, circuit engineers design intelligent processing circuits inside FPGAs (Figure 3(1)). As shown in the right side of Figure 3 regarding the case of a hw/sw complex system, the engineers have to implement an interface that connects the circuit and an FPGA controller on an embedded CPU (Figure 3(2)). In this case, the system provide the interface to reduce the man-hour. In the same way, circuits engineers have to implement interfaces which connect the FPGA controller and ROS space/other processes (Figure 3(3)). The system should provide the interfaces which are familiar to the circuit engineers in this case. Furthermore, when robotic engineers utilize the hw/sw complex system, it is desirable that the system should be easy to maintain as a ROS package.

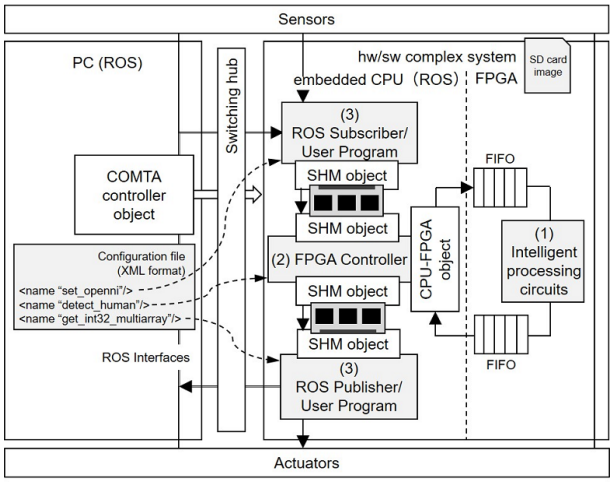


Figure 3. Block diagram of the proposed system (COMTA)

Table 2. ROS space/other processes of the proposed system

Type	Name	Input	Output
ROS Subscriber	set_int32	std_msgs::Int32	SHM object
	set_int32_multiarray	std_msgs::Int32MultiArray	SHM object
	set_float32	std_msgs::Float32	SHM object
	set_float32_multiarray	std_msgs::Float32MultiArray	SHM object
ROS Publisher	get_int32	SHM object	std_msgs::Int32
	get_int32_multiarray	SHM object	std_msgs::Int32MultiArray
	get_float32	SHM object	std_msgs::Float32
	get_float32_multiarray	SHM object	std_msgs::Float32MultiArray
Preset User Program	set_openni	Connected to RGB-D camera	SHM object
	set_v4l2	Connected to Web camera	SHM object
User Program	Decided by user	Implemented by user	SHM object
		SHM object	Implemented by user

One of the use case of the COMTA is intelligent image processing on the robots. For this purpose, the system should acquire images from the ROS space or cameras and return the processing results to the ROS space. Additionally, the system should process images at the typical resolution (VGA, 640 * 320 pixels) and frame rate (30 fps) of the cameras installed on the robots.

3.1 *ROS Specialized Hardware Accelerator Constructor*

As demonstrated in Figure 3, we propose to automatically generate the hardware-accelerated intelligent processing system through ROS interfaces. This is achieved by utilizing various types of a program called "object."

A PC (Figure 3 left) and a hw/sw complex system (Figure 3 right) are applied in COMTA. An embedded CPU and an FPGA are integrated on the hw/sw complex system. Sensors and actuators basically employ the existing device drivers installed on the PC or the embedded CPU. Intelligent processing can be accelerated by utilizing a dedicated architecture implemented on the FPGA (Figure 3(1)).

A CPU-FPGA object is an interface which integrates circuits inside the FPGA and the FPGA controller on the embedded CPU. The intelligent processing circuits are connected to FIFO interfaces; therefore, circuits can be described using HDL, circuits can be generated by high-level synthesis, and any circuits can be connected.

A shared memory (SHM) object is an interface that integrates the FPGA controller and ROS space or other processes. Between the controller and ROS space/other processes, communication is performed by utilizing a shared memory communication model that will be described later.

The CPU-FPGA and the SHM objects can allow circuit engineers to implement intelligent

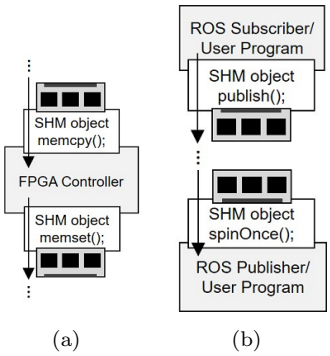


Figure 4. SHM object interface: (a) for circuit enginners and (b) for robotic engineers

processing on an FPGA without writing an interface from the FPGA to the ROS space. In addition, ROS engineers can write arbitrary programs and communication with the FPGA controller through the SHM object.

In the PC, a COMTA controller object is the core of the proposed system and activates the hw/sw complex system. The object is given a tiny configuration file defining the data flow through the hw/sw complex system. In this file, the names of the FPGA controller and the ROS space/other processes that input and output data to the FPGA are described in the same extensible markup language (XML) format as the ROS launch as shown in the left side of Figure 3 (Configuration file). Table 2 lists the processes that can be described in the file. Processes with "set" in their name input a ROS topic or sensor data and output it to the FPGA via the SHM object. Meanwhile, processes with "get" in their name input data via the SHM object and output it to a ROS topic. These processes are the basic data types for the use case of image processing. When the object is defined by the configuration file through the ROS interface, the hw/sw complex system is activated as follows: (1) ROS space/other processes and an FPGA controller are launched. (2) An SHM and a CPU-FPGA objects are activated for internal data exchange in the hw/sw complex system.

In summary, using the proposed system, ROS engineers can select the input and output data of inteligent processing on the FPGA only by describing the XML format and can automatically construct the hardware accelerator that is principally applied in robotic image processing. Consequently, the hardware accelerator can be simply handled just like an ordinary ROS system. For example, for a typical ROS navigation, users are mainly required to introduce hardware (Laser rangefinder) into their system and download the software [32–34] needed to run a nav-igation system. Similarly, users of the proposed hardware-accelerated ROS package only need to implement an FPGA board and download the hardware-accelerated ROS package from the Internet.

3.2 *Shared Memory Communication Model*

The computational resources of the embedded CPU are very limited; thus, on the embedded CPU, data exchange requires a communication model with a smaller computational load than that of the ROS interface, as illustrated in Figure 2. Moreover, the ROS interface is not well known among circuit engineers. Therefore, we propose a shared memory communication model instead of using the ROS interface. This model exchanges data by utilizing only internal memory; therefore, its computation load is less than that of the ROS interface, which exchanges data through a network. The model is implemented by an inter-process communication on UNIX, using "shmget", "shmat", "shmdt" and "shmctl" functions. As shown in Figure 4(a), an SHM object implemented in the model provides an interface to the FPGA controller. This interface is compatible with the "memcpy" and "memset" functions, in the C language, circuit engineers are familiar with. Additionally, as shown in Figure 4(b), the object provides an interface to the

ROS space/other processes that is compatible with "publisher" and "subscriber", which robotic engineers are familiar with, in ROS. Through the interface, the model can exchange the data of both circuit and robotic engineers through the shared memory model.

3.3 *Discussion*

When compared with the other previously conducted studies discussed above, the proposed system has the following novelty, efficacy, and contribution [15], [27–29]. By adopting the shared memory communication model, the proposed system provides high-speed communication with ROS space on the embedded CPU. The system enables the configuration of the hw/sw complex system from ROS space with the single-configuration file, and robotic engineers can easily utilize the FPGA. In addition, circuit engineers are provided with interfaces that can connect the FPGA, the embedded CPU, and ROS space, and they can also focus on implementing the application circuit only. As depicted in Figure 3, the proposed system has a short data path between sensors and the FPGA in the proposed architecture that can implement arbitrary software on the embedded CPU. The path achieves efficient communication among the various sensors and memories, and it can offload the processing load of sensors from the PC.

4. Experiments

We implement a human-tracking application to validate the effectiveness of the proposed system. This application is commonly applied at domestic service robots to carry an operator's luggage. The flow process of the application is described in Figure 5. The numbers in the parentheses correspond to those in Figure 5.

(1) A depth image is obtained from an RGB-D camera using OpenNI [35].
(2) We resize the depth image by clipping based on the distance.
(3) A binary image is generated from the depth image with two threshold distances.
(4) The region of interest images is extracted from the binary image by utilizing sliding windows.
(5) We extract multiresolution cooccurrence histograms of oriented gradients (MRCoHOG) features [36] from the region of interest images.
(6) We determine whether the features are human by applying a real AdaBoost [37].
(7) The wheel base of the robot is controlled to approach the target human.
(8) We set the distances of the human region as the threshold for generating the next binary image.

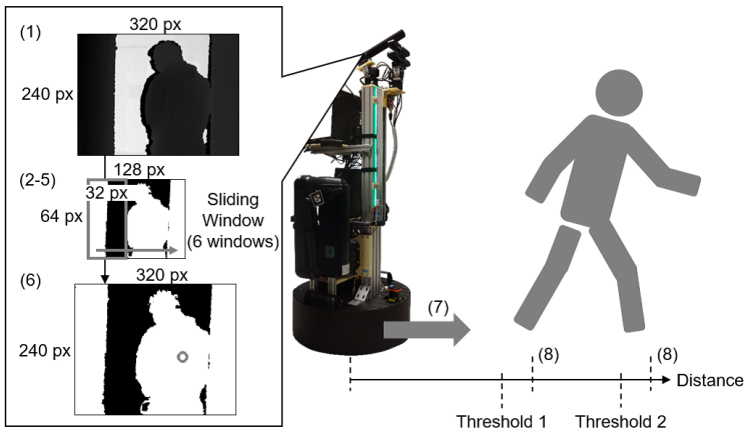


Figure 5. The flow of the human-tracking application

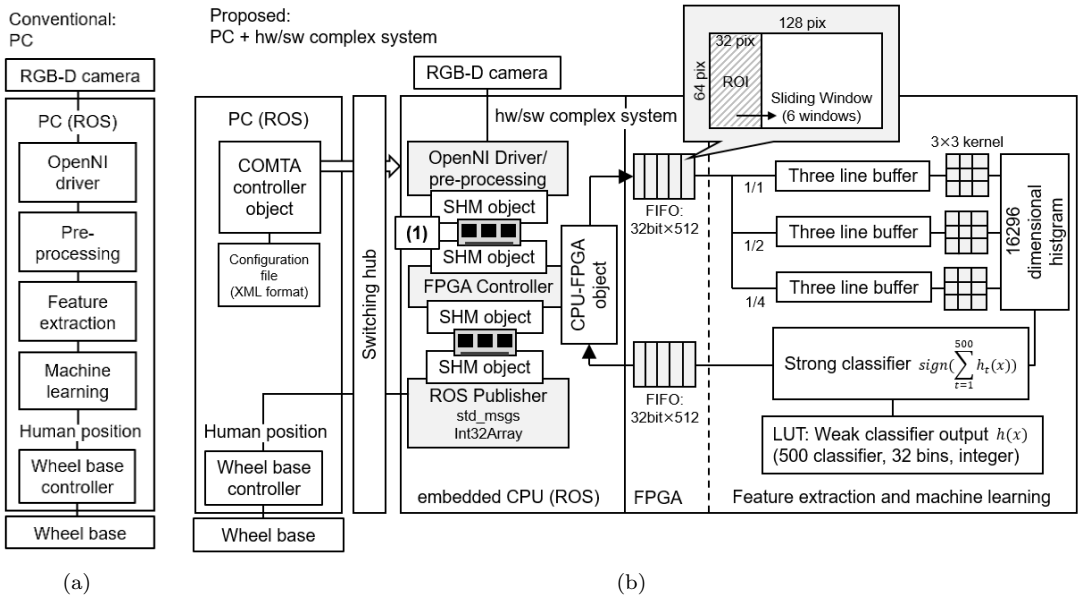


Figure 6. Block diagram: (a) conventional systems, (b) the proposed systems

Figure 6 depicts a block diagram of the system constructed for the experiment. Note that Figure 6(a) illustrates a conventional system that is implemented by a laptop PC only. In this system, all operations (i.e., sensing to control operations) were performed on a PC. Figure 6(b) demonstrates the proposed system. In this proposed system, operations were performed by utilizing a PC and the hw/sw complex system. We explain how the above application is executed in the proposed system. First, based on the behavior of the hw/sw complex system defined in an XML format configuration file, the COMTA controller object activates the embedded CPU's software. In this experiment, the file is defined to utilize/activate OpenNI driver/pre-processing, ROS publisher, and FPGA controller programs in the embedded CPU. Then, the operation flows 1, 2, and 3 are executed using an OpenNI driver/pre-processing program. At this time, the region of interest for all images are written to a FIFO through an SHM object, an FPGA controller, and a CPU-FPGA object. It is noteworthy that we can speed up flows 2 and 3 using pointers when accessing the images data. In the FPGA, the region of interest for all the images are sent to three-line buffers at an original, a half, and quarter resolutions. The buffered images are calculated by luminance gradients using the upper, lower, left and right of a 3×3 kernel, and vote on a histogram block. The histogram is evaluated by utilizing the output of the weak classifiers implemented in a LUT. Finally, the results, whether human or not, are sent to the embedded CPU through the FIFO and sent to the PC through the shared memory. To turn and get closer to human, the robot is controlled using wheel base controller depending on the results. Thus, the PC is only employed to activate the hw/sw complex system and control the wheel base of the robot, and the computational load of the intelligent processing is offloaded to the hw/sw complex system.

In this experiment, we used a Xilinx XC7Z020 SoC and its evaluation board [26, 38]. To synthesize the internal circuit of the FPGA, we utilized a Xilinx Vivado HLS 2016.2, which is a high-level synthesis tool [39]. The synthesis results of feature extraction and machine learning are presented in Table 3. Moreover, the minimum and maximum latency of the internal circuit for one region of interest are 0.257 and 0.565 ms, respectively.

4.1 *Evaluation of Human Detection Accuracy*

In this experiment conducted, we implemented human detection using MRCoHOG and real Adaboost, which require large computational resources among the human-tracking algorithms, in

Table 3. Synthesis Results

Resource	Available	Used	Utilization
BRAM_18K	280	0	0%
DSP_48E	220	0	0%
FF	106400	8970	8%
LUT	53200	16942	31%

(a) (b)

Figure 7. INRIA person dataset: (a) positive (human), (b) negative (not human)

Table 4. Parameters of MRCoHOG, real Adaboost, and dataset

MRCoHOG	Size of Original Image	64(H) * 32(W) pix
	Size of Resized Image 1	32(H) * 16(W) pix
	Size of Resized Image 2	16(H) * 8(W) pix
	Size of Block	8(H) * 8(W) pix
	Offset Distance	1
	Number of Gradient Direction	8
	Threshold of Gradient Magnitude	15.0
Real AdaBoost	Number of Histogram Bin	32
	Trials of Train	250
Dataset	Train Positive (Human)	2416 Images
	Train Negative (Not Human)	12288 Images
	Test Positive (Human)	1126 Images
	Test Negative (Not Human)	4840 Images

Table 5. The results of human detection of a PC and an FPGA

		Input			
		PC		FPGA	
		Positive	Negative	Positive	Negative
Predict	Positive	1014	69	1021	90
	Negative	112	4771	105	4750
Accuracy		96.97%		96.73%	

the FPGA on the proposed system. To implement the human-tracking algorithm on the FPGA, a hardware-oriented algorithm is required to maximize the performance of the FPGA. It is different from the original software algorithm for optimizing the number of quantization bits and avoid multiplications and divisions. Therefore, we compared the accuracy of the human detection implemented on the PC (original software) and the FPGA (hardware-oriented algorithm).

In this experiment, we evaluate INRIA person dataset [40] (Figure 7) and compare with detection accuracy by determining whether the input image is a human or not. The parameters of MRCoHOG and real Adaboost, and the number of data sets used for training and testing are demonstrated in Table 4.

Table 5 illustrates the accuracy of the human detection implemented in the PC and the FPGA. From these results, the accuracies of the PC and FPGA are 96.97% and 96.73%, respectively. The proposed MRCoHOG and real Adaboost are designed for the FPGA implementation, and it has a high accuracy which is equivalent to that of the PC.

4.2 *Evaluation of the Embedded CPU Internal Communication*

Data exchange in the embedded CPU is shown in Figure 6(b)(1). The performance of the conventional ROS interfaces is compared with that of the proposed shared memory model. In the former,

Table 6. Comparison of the embedded CPU internal communication methods

Method	Send/ Receive	Average frame rate [fps]	
		QVGA	VGA
ROS interfaces	Send	309	65
	Receive	244	60
Shared memory	Send	1261703	822011
	Receive	70550	77393

Table 7. The execution results of the human-tracking application

	Conventional Core i5 5200U 2.2GHz	Proposed ARM Dual Core 667MHz/FPGA
Frame rate [fps]	29.06	27.55
Power consumption [W]	26	4.7
Efficiency [fps/W]	1.12	5.86

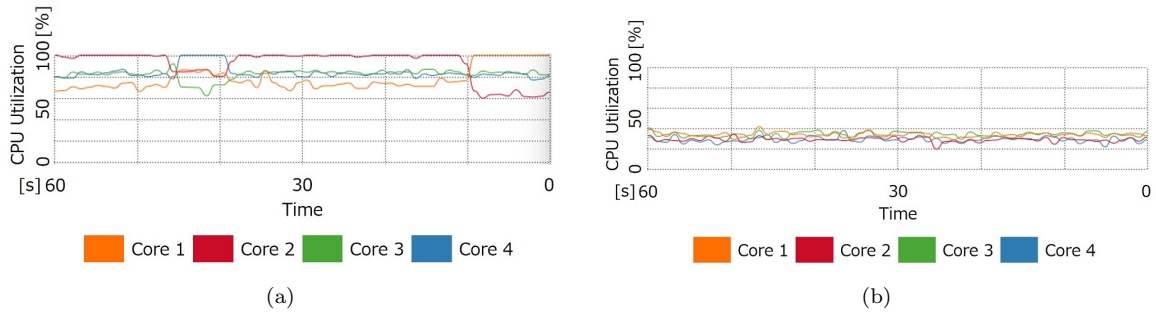


Figure 8. CPU utilization on a PC running (a) the conventional system and (b) the proposed system. The lines represent the utilizations of four CPU cores.

data exchange is performed via a local network inside the embedded CPU, while the latter is often used for inter-process communication with the data exchange performed using memory. In this experiment, QVGA/VGA, RGB 8 bit, and three-channel images were exchanged. Moreover, we measured the average frame rate. The results are presented in Table 6. The experimental results demonstrated that the shared memory can exchange data at a high speed with a smaller calculation load.

4.3 *Evaluation of the Proposed System Using Real Robot Application*

As aforementioned, the performances of the conventional system (Figure 6(a)) are compared with those of the proposed systems (Figure 6(b)) using a human-tracking application with Exi@ robot. In this experiment, frame rate, power consumption, power efficiency, and CPU utilization on a PC were measured.

The experimental results are illustrated in Table 7. In the conventional system, the frame rate was the same as that of the RGB-D camera. Moreover, the conventional system consumed 26 W; its power efficiency decreased. In the proposed system, the frame rate is almost equivalent to that of the conventional system. The proposed system consumed only 5 W; its power efficiency was 5.2 times better than that of the conventional system.

The Figure 8 shows the experimental results of CPU utilization on the PC. In Figure 8, the vertical axis represents CPU utilization, the horizontal axis represents time, and lines represent the utilization of four CPU cores. As shown in Figure 8(a), tracking a person by the conventional system utilized approximately 80% of the CPU. The same task in the proposed system, utilized approximately 30% of the CPU (Figure 8(b)). In these experiments, the CPU utilization was 50 points lower in the PC of the proposed system than in the PC of the conventional system, demonstrating that the heavy load incurred by the human-tracking application was off-loaded from the PC to the proposed hw/sw complex system.

5. Conclusion

In this study, we proposed a COMTA hardware-accelerated robot middleware package. The proposed system can automatically generate an intelligent processing system by utilizing a tiny configuration file in ROS space. Moreover, we focused on communication and proposed a shared memory communication model. The experimental results verified that the communication of the proposed system has a better performance than that of the conventional ROS interfaces. In a human-tracking application which is commonly used as domestic service robots, the human detection accuracy of the proposed system is equivalent to that of a general PC system. The power efficiency of the proposed system was found to be 5.2 times better than that of the general PC system. In addition, the CPU utilization on the PC of the proposed system was 50 points less than that of the conventional system. By utilizing deep neural networks in the future, we hope to improve the efficiency of intelligent processing and implement it further.

Acknowledgment

This research is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and partially supported by JSPS KAKENHI grant number 17H01798.

References

[1] Yamamoto T, Terada K, Ochiai A, Saito F, Asahara Y, Murase K. Development of Human Support Robot as the research platform of a domestic mobile manipulator. ROBOMECH Journal. 2019.
[2] Hori S, Yutaro I, Kiyama Y, Tanaka Y, Kuroda Y, Hisano M, Imamura Y, Himaki T, Yoshimoto Y, Aratani Y, Hashimoto K, Iwamoto G, Fujita H, Morie T, Tamukoh H. Hibikino-Musashi@Home 2017 Team Description Paper. arXiv:1711.05457. 2017.
[3] Wisspeintner T, Van der Zant T, Iocchi L, Schiffer S. RoboCup Home: Scientific Competition and Benchmarking for Domestic Service Robots. Interaction Studies. 2009;10;3;392-426.
[4] Iocchi L, Holz D, Ruiz-del-Solar J, Sugiura K, Zant T. Analysis and results of evolving competitions for domestic and service robots. Artificial Intelligence. 2015;229;258-281.
[5] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. p. 779-788.
[6] Cao Z, Simon T, Wei S, Sheikh Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. p. 7291-7299.
[7] Yamauchi Y, Kato Y, Yamashita T, Fujiyoshi H. Cloud Robotics Based on Facial Attribute Analysis for Human-Robot Interaction. In: Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication. 2016.
[8] Sugiura K, Zettsu K. Rospeex: A Cloud Robotics Platform for Human-Robot Spoken Dialogues. In: Proceedings the IEEE/RSJ International Conference on Intelligent Robots and Systems. 2015. p. 6155-6160.
[9] Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A. ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software. 2009.
[10] ROS Wiki. Available from: `http://wiki.ros.org/`
[11] Ando N, Suehiro T, Kitagaki K, Kotoku T, Yoon W. RT-middleware: distributed component middleware for RT (robot technology). In: Proceedings the IEEE/RSJ International Conference on Intelligent Robots and Systems. 2015. p. 3555-3560.
[12] OpenRTM-aist official website. Available from: `http://www.openrtm.org/openrtm/ja/content/openrtm-aist-official-website`
[13] V-Sido OS. Available from: `https://www.asratec.co.jp/v-sido-os/`

[14] ROS Community Metrics Report 2015. Available from: http://download.ros.org/downloads/metrics/metrics-report-2015-07.pdf
[15] Ohkawa T, Yamashina K, Kimura H, Ootsu K, Yokota T. FPGA Components for Integrating FPGAs into Robot Systems. IEICE Transactions on Information and Systems. 2018;E101-D;2;363-375.
[16] Wise M, Ferguson M, King D, Diehr E, Dymesich D. Fetch and freight: Standard platforms for service robot applications. In: Proceedings Workshop on Autonomous Mobile Service Robots. 2016.
[17] Jordi P, Luca M, Francesco F. TIAGo: the modular robot that adapts to different research needs. In: Proceedings International Workshop on Robot Modularity. 2016.
[18] Seib V, Manthe S. Memmesheimer R, Polster F, Paulus D. Team Homer@UniKoblenz – Approaches and Contributions to the RoboCup@Home Competition. RoboCup 2015: Robot World Cup XIX. 2015;83-94;
[19] RoboCup@Home 2018 Rules. Available from: http://www.robocupathome.org/rules
[20] Bao C, Raghavender S, John T. Person Following Robot Using Selected Online Ada-Boosting with Stereo Camera. In: Proceedings Conference on Computer and Robot Vision. 2017.
[21] Bao C, Raghavender S, John T. Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot. Computer Vision Systems. 2017;300-313;
[22] Minkyu K, Miguel A, Nick W, Yuqian J, Justin H, Peter S, Luis S, An Architecture for Person-Following using Active Target Search. arXiv:1809.08793. 2018.
[23] Yonekawa H, Nakahara H. On-Chip Memory Based Binarized Convolutional Deep Neural Network Applying Batch Normalization Free Technique on an FPGA. In: Proceedings the IEEE International Parallel and Distributed Processing Symposium Workshops. 2017. p. 98-105.
[24] Benkrid K, Akoglu A, Ling C, Song Y, Liu Y, Tian X. High performance biological pairwise sequence alignment: FPGA versus GPU versus cell BE versus GPP. International Journal of Reconfigurable Computing. 2012;1-15.
[25] Tamukoh H, Sekine M. Design of Networked hw/sw Complex System using Hardware Object Model and Its Application. In: Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society. 2013. p. 2250-2255.
[26] Zynq-7000 All Programmable SoC. Available from: https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html
[27] Yamashina K, Ohkawa T, Ootsu K, Yokota T. Proposal of ROS-compliant FPGA Component for Low-Power Robotic Systems -case study on image processing application-. arXiv:1508.06320. 2015.
[28] Ohkawa T, Yamashina K, Matsumoto T, Ootsu K, Yokota T. Automatic Generation Tool of FPGA Components for Robots. IEICE Transactions on Information and Systems. 2019;E102-D;5;1012-1019.
[29] Cheng H, Sato S, Nakahara H. A Performance Per Power Efficient Object Detector on an FPGA for Robot Operating System. In: Proceedings of the 9th International Workshop on Highly-Efficient Accelerators and Reconfigurable Technologies. 2018. p. 1-4.
[30] Ishida Y, Morie T, Tamukoh H. A Hardware Accelerated Robot Middleware Package for Intelligent Processing on Robots. IEEE International Symposium on Circuits and Systems. 2018.
[31] Ishida Y, Morie T, Tamukoh H. Live Demonstration: A Hardware Accelerated Robot Middleware Package for Intelligent Processing on Robots. IEEE International Symposium on Circuits and Systems. 2018.
[32] hokuyo_node. Available from: http://wiki.ros.org/hokuyo_node
[33] amcl. Available from: http://wiki.ros.org/amcl
[34] mode_base. Available from: http://wiki.ros.org/move_base
[35] OpenNI 2 Downloads and Documentation. Available from: https://structure.io/openni
[36] Iwata S, and Enokida S. Object Detection Based on Multiresolution CoHOG. In: Proceedings of International Symposium on Visual Computing. 2014. p. 427-437.
[37] Shapire R, Singer Y, Improved Boosting Algorithms Using Confidencerated Predictions. Machine Learning. 1999;37;297-336.
[38] Xilinx ZedBoard. Available from: http://zedboard.org/product/zedboard.
[39] Xilinx. Available from: https://www.xilinx.com/
[40] INRIA Person Dataset. Available from: http://pascal.inrialpes.fr/data/human/

Authors

Yutaro Ishida received his BE and ME degree from Kyushu Institute of Technology, in 2015 and 2017. He is currently in a doctoral program at the graduate school of Life Science and Systems Engineering, Kyushu Institute of Technology. His research interest includes hardware/software complex system and service robots. He is a member of the RSJ, ISCIE, IEICE and IEEE.

Takashi Morie received the B.S. and M.S. degrees in Physics from Osaka University, Osaka, Japan, and the Dr. Eng. degree from Hokkaido University, Sapporo, Japan, in 1979, 1981 and 1996, respectively. From 1981 to 1997, he was a member of the Research Staff at Nippon Telegraph and Telephone Corporation (NTT). From 1997 to 2002, he was an associate professor of the department of electrical engineering, Hiroshima University, Higashi-Hiroshima, Japan. Since 2002 he has been a professor of Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, Kitakyushu, Japan. His main interest is in the area of VLSI implementation of neural networks, mixed/merged analog-digital circuits, and new functional devices. He is a member of IEEE, IEICE, IEEJ, the Japan Society of Applied Physics and the Japanese Neural Network Society.

Hakaru Tamukoh received the B.Eng. degree from Miyazaki University, Japan, in 2001. He received the M.Eng and the Ph.D. degree from Kyushu Institute of Technology, Japan, in 2003 and 2006, respectively. He was a postdoctoral research fellow of 21st century center of excellent program at Kyushu Institute of Technology, from April 2006 to September 2007. He was an assistant professor of Tokyo University of Agriculture and Technology, from October 2007 to January 2013. He is currently an associate professor in the graduate school of Life Science and System Engineering, Kyushu Institute of Technology, Japan. His research interest includes hardware/software complex system, digital hardware design, neural networks, soft-computing and home service robots. He is a member of IEICE, SOFT, JNNS, IEEE, JSAI and RSJ.