Offloading Intellectual Processing from Home Service Robots to Edge Devices

Yuma Yoshimoto
NIT, Kitakyushu College, 5-20-1 Shii, Kokuraminamiku, Kitakyushu, Fukuoka, 802-0985 Japan

Mizuki Kawashima
NIT, Kitakyushu College, 5-20-1 Shii, Kokuraminamiku, Kitakyushu, Fukuoka, 802-0985 Japan

Shun Yonehara
NIT, Kitakyushu College, 5-20-1 Shii, Kokuraminamiku, Kitakyushu, Fukuoka, 802-0985 Japan
Email: yoshimoto@kct.ac.jp, k19053mk@apps.kct.ac.jp, k19206sy@apps.kct.ac.jp

Abstract

In this study, we focus on extending the operational time of home service robots by offloading intellectual processing to circuit devices such as Field Programmable Gate Arrays (FPGAs), which in turn reduces power consumption. The core of our approach involves developing a method for implementing intellectual processing on FPGAs, coupled with a dynamic circuit reconfiguration technique. This enables the FPGA to adaptively respond to frequent task changes. We present: (a) methods for transitioning circuits from a robot's control computer to FPGA in response to varying tasks, and (b) an evaluation of the effectiveness of using FPGA to extend operational time under these rapidly changing task conditions.

Keywords: Intelligent Processing, Edge Device, Service Robots, FPGA, Neural Networks

1. Introduction

In recent years, service robots have gained attention against the backdrop of a society experiencing declining birthrates and an aging population. Service robots are designed to perform a variety of tasks in homes and stores, such as tidying up and waitering, thereby reducing the workload on humans. These robots are equipped with various intelligent processing capabilities such as object recognition and voice recognition, allowing them to understand their surroundings and operate flexibly. Recently, Neural Networks (NNs) have been increasingly used for these intelligent processing tasks, and it is known that NNs require a substantial amount of computational power. The intelligent processing in service robots demands real-time operation, but achieving this through software alone is challenging. Therefore, hardware architecture acceleration is necessary.

Today, GPUs are primarily used as the hardware architecture for implementing NNs. However, running numerous NNs simultaneously leads to issues with insufficient hardware resources. Additionally, GPUs are generally known for their high-power consumption, which limits the operational time of robots.

This research aims to extend the operating time of robots by offloading the intelligent processing to circuit devices like Field Programmable Gate Arrays (FPGAs), thereby reducing power consumption. To achieve this, we propose the implementation of intelligent processing on FPGAs and a method to rewrite the circuits on the FPGA depending on the situation. This paper reports on the initial investigation, including (a) implementing state machines in robots and verifying the ability to construct appropriate state machines in response to user commands, (b) exploring methods to switch circuits from the robot control computer to the FPGA, and (c) examining the effectiveness of offloading the intelligent processing to the FPGA.

2. System Configuration of Service Robots

This section describes the configuration of the service robots targeted in this study.

2.1. *Behavior Planning with State Machines*

In service robots, processing is carried out through the construction and execution of state machines, as shown in Fig. 1. The process flow is as follows:

(1) Listen to the Commands

The robot listens to the user's voice and understands the commands as text.

(2) Construction of the State Machine

The robot interprets the heard commands and decides in what order and which intelligent processes to utilize.

(3) Execution of the State Machine

The robot executes the constructed state machine in sequence.

At this stage of executing the state machine (3), the state machine has already been constructed, and it is known in what order and which intelligent processes are needed.

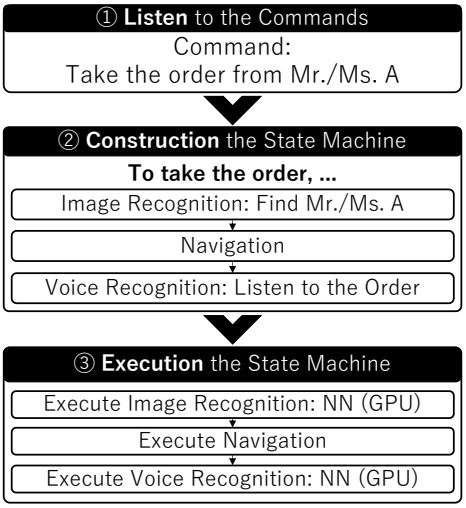


Fig.1 Constructing and Executing the State Machine

2.2. *Required Intelligent Processing*

Various intelligent processes are necessary for service robots. For example, Tsuji et al. have implemented the Neural Networks (NN) shown in Table 1 for the realization of versatile robots. In conventional methods, responding to human commands requires the simultaneous activation of numerous NNs.

Table 1. AI Implemented in the Robot [1]

AI	Model
Object Detection	Mask R-CNN, UOLS
Object Classification	CLIP (ViT-B/32)
Speech Recognition	Whisper
NLP and NLU	GPT-3

2.3. *Pathways Language Model (PaLM [2]), SayCan [3]*

There are developed by Google, this robot-specific AI allows robots to autonomously plan actions when given ambiguous verbal commands. For instance, if input like 'I spilled my drink. Help me,' is given, PaLM-SayCan can make decisions such as bringing a sponge or disposing of an empty can. The number of parameters required for PaLM is said to be 540B [2]. If these parameters are managed as float64, the memory size required for PaLM amounts to 216 GB. However, the memory in GPUs used for AI processing in service robots ranges from 12 GB to 24 GB. Therefore, it is not feasible to implement an AI of size 216 GB in the GPU of a robot control computer. Thus, integrating this method into the edge side is challenging. Additionally, incorporating this method into robots would rely on external computing devices like cloud services, which introduces instability due to network connections, making it difficult to achieve a stable system.

3. Field Programmable Gate Array

An FPGA is a semi-custom LSI composed of a grid of reconfigurable logic gates. FPGAs can achieve high computational performance by reconfiguring their internal logic circuits. Unlike GPUs, which are used for AI processing and consist of block-based arithmetic architectures like multiplication and division, FPGAs construct their architecture with logic operation units, enabling lower power consumption and faster processing. Therefore, it is conceivable that using FPGAs allows for real-time processing.

3.1. *Comparison of FPGA with GPU and CPU*

Nakahara et al. implemented CNNs on FPGA and compared it with CPU and GPU [4]. The comparison table is shown in Table 2. They used the NVIDIA Jetson TX1 board for comparison, which includes an embedded CPU (ARM Cortex-A57) and an embedded GPU (Maxwell GPU). The CPU and GPU used Caffe (version 0.14) and were compared when running VGG-11 with a batch size of 1 for latency measurement. As shown in Table 2, FPGA is faster and consumes less power than the embedded CPU and GPU. Thus, a binary CNN on FPGA is more suitable for embedded systems than CPU and GPU.

Table 2. Comparison with Embedded Platforms [5]

Platform	**Embedded CPU**	**Embedded GPU**	**FPGA**
Device	ARM Cortex-A57	Maxwell GPU	Zynq 7020
FPS [s^{-1}]	0.23	36.7	421.9
Power Consumption [W]	7	17	2.3

3.2. *FPGA Hardware Resource Utilization*

Masatomo et al. implemented a model based on YOLOv3-tiny with Depth wise Separable Convolution on FPGA (Zynq-7020). They reported a BRAM utilization rate of 85.71%, indicating that implementing just one CNN on FPGA already exceeds 50% of the necessary resources. Therefore, it is difficult to implement other NNs simultaneously.

4. Proposal

We propose a method to control FPGA with a state machine. Fig. 2 shows the diagram of the proposed system. The operating procedure of the system is as follows:

① A control computer's state machine sends an instruction for the AI mode to be executed to the System on a Chip Manager (SoC Manager), which then instructs the FPGA Manager to rewrite.
② The FPGA Manager calls pre-defined circuit modules and rewrites the intelligent processing circuits on the FPGA.
③ The FPGA Manager sends input data for inference to the intelligent processing circuit on the FPGA.
④ The intelligent processing circuit on the FPGA completes the inference and sends the results back to the FPGA Manager.
⑤ The FPGA Manager sends the inference results to the SoC Manager, which then sends them to the state machine.

By repeating these steps, intelligent processing is executed on the FPGA while time-sharing its resources. Executing intelligent processing on the FPGA resolves the issue of power consumption, and time-sharing the FPGA resolves the resource issue.

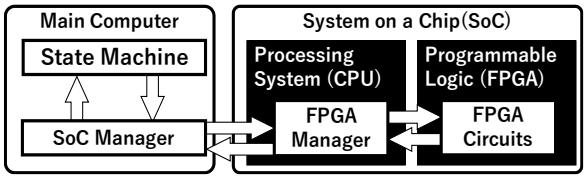


Fig.2 Overview of the Proposed System

5. Experimental

5.1. *Construction of the State Machine*

A system as shown in Fig. 1 was constructed, and it was verified that it could (1) listen to commands, (2) construct a state machine, and (3) execute the state machine. In this verification, the following technologies were implemented as elemental technologies:

- Recognition of opening and closing doors through Depth image processing
- SLAM for creating environmental maps and estimating self-position
- Navigation allowing the robot to autonomously move to a target location
- Person and posture detection using skeleton estimation AI
- Object detection with Faster-RCNN
- Depth image processing to estimate the three-dimensional position of objects
- Speaking function using gTTS and HSR-API
- Voice recognition to understand human speech using Whisper
- Context understanding that interprets the meaning from recognized text strings

The system's operation was also confirmed. The operational verification was conducted at GPSR task of RoboCup JapanOpen 2022 @Home League, a benchmark competition for service robots. This task competes on how well robots can respond to input from people, evaluating the constructed state machine. The robot was given instructions in English, and its ability to go to a designated location and perform specific actions was tested.

As a result of the experiment, the constructed state machine was given the input 'Go to the Living room, find Jennifer and answer her question.' First, the command was correctly processed using the voice recognition function, and then the robot moved to the living room using the autonomous movement algorithm. Next, it used person recognition to find Jennifer and moved to her location. After moving, it answered questions using the voice recognition AI. It was confirmed that the state machine was appropriately constructed for these states up to the movement, and the states changed according to the situational changes.

5.2. *Consideration of FPGA Circuit Switching Method*

As a preliminary step to constructing the system shown in Fig. 3, a method to switch the FPGA circuit from the control computer was realized. The specific configuration is shown in Fig. 3. This was implemented on the KV260 evaluation board equipped with FPGA-SoC, and its operation was verified. The operation flow is the same as described in Chapter 4. In the experiment, the inference process was confirmed to be possible by switching between two models of YOLO v3-Tiny [6], each trained on different datasets.

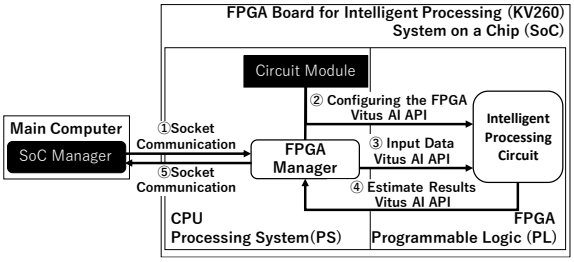


Fig.3 System Configuration

5.3. *Consideration of the Effectiveness When Implemented on FPGA Circuit*

The object recognition AI was implemented on AMD's KV260 FPGA evaluation board and an MSI-manufactured notebook PC equipped with RTX3060. The power consumption of each was measured both in a steady state and during the operation of the object recognition AI. The results are shown in Table 3. The difference between the power consumption during the operation of the object recognition AI and in the steady state was 2 W for the FPGA and 96 W for the GPU, which is a reduction to 1/48th. From this, it can be understood that the implementation of intelligent processing on FPGA is advantageous in terms of power consumption compared to GPU implementation.

Table 3. Difference in Power Consumption between FPGA and GPU

	Power Consumption [W]		
	Steady State	During Object Recognition AI Operation	Difference
KV260	10	12	2
MSI Stealth-15M-B12UE-012JP	44	140	96

5.4. *Verification Using Images Captured by the Robot*

Images were captured using an RGB-D camera (Xtion Pro) connected to the robot, and these images were input into the system. As a result, an output was obtained that shows what appears where in the images.

6. Conclusion

In this paper, we proposed a method for offloading power-intensive intelligent processing to FPGA circuit devices, aiming for extended operational times of service robots. In our verification, as a first step towards realizing the system, we (a) implemented a state machine in the robot and confirmed the ability to construct an appropriate state machine in response to user commands, (b) explored a method for switching circuits implemented on FPGA from the robot control computer, and (c) investigated the effectiveness of offloading intelligent processing to FPGA. The results confirmed that (a) the construction and execution of a state machine are possible, as demonstrated in the RoboCup@Home DSPL's GPSR, (b) it is feasible to switch actual circuits from the robot control computer to FPGA, (c) significant power reduction can be achieved when running the object recognition AI on FPGA compared to GPU. In addition (d) the edge device (FPGA) estimated the images from robot camera (Xtion Pro). Future work will focus on integrating these developments with the state machine and extending support to other intelligent processes.

References

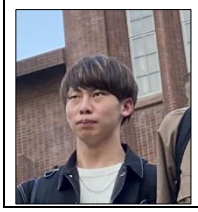
1. C. Tsuji, D. Komukai, et al, "TRAIL 2023 Team Description Paper," https://trail.t.u-tokyo.ac.jp/project/robocup2023/.
2. A. Chowdhery, S. Narang, et al., "PaLM: Scaling Language Modeling with Pathways," arXiv:2204.02311.
3. M. Ahn, A. Brohan, et al., "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," arXiv:2204.01691.
4. H. Nakahara, H. Yonekawa, T. Fujii, M. Shimoda, S. Sato, "GUINNESS: A GUI Based Binarized Deep Neural Network Framework for Software Programmers," IEICE Transactions on Information and Systems," E102.D, 5, pp.1003-1011, 2019.
5. M. Matsuda, Y. Araki, et al., "An FPGA Implementation of Object Recognition System with Low Power Consumption using YOLOv3-tiny-based CNN," 2022 "Hinokuni - Land of Fire" Information Processing Symposium (HINOKUNI2022), 2022.
6. J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," IEEE Computer Vision and Pattern Recognition, pp. 1-8, 2018.

Authors Introduction

Dr. Yuma Yoshimoto

He received his B.Eng. degree from National Institute of Technology (KOSEN), Maizuru College, Japan, in 2016. He received his M.Eng. and D. Eng. degrees from Kyushu Institute of Technology, Japan, in 2018 and 2021, respectively. And he was JSPS researcher, in 2019 - 2021. He was a post-doctoral researcher at the Kyushu Institute of Technology, Japan in 2021-2022. Currently, he is assistant professor at the National Institute of Technology (KOSEN), Kitakyushu College, Japan. His research interests include deep learning, robot vision and digital hardware design. He is a member of IEICE, IEEE.

Mr. Mizuki Kawashima



He enrolled at National Institute of Technology (KOSEN), Kitakyushu College, Japan, in 2019. In 2021, he pursued the Information Systems course, focusing on algorithms and control. He commenced his research in robotics in 2022.

Mr. Shun Yonehara

He enrolled at National Institute of Technology (KOSEN), Kitakyushu College, Japan, in 2019. In 2021, he pursued the Information Systems course, focusing on algorithms and control. He commenced his research in robotics in 2022.