Multiple Distribute Robotic System based on ROS

LIU YIFAN† TAKAHIRO KIKUCHI† MIDORI SUGAYA†

Currently, multiple robots are expecting to provide useful services corporately at the event of disaster and medical treatment, nursing care, home security etc.. Autonomic robots such as searching and following people should have the functionalities of collecting data from its physical environment and analyzing to share the information to each other. We build up these mechanisms with the ROS middleware that had been widely used as a software framework for robotic systems. ROS provides a structured communication layer above host operating system and can be used to build a distribute system. However, while we bring our ROS based distribute searching system from local network to the internet, it did not work as well as expected because of latency in this system which should be improved.

1. Introduction

Multiple robots' services are required for rescue, nursing, security and etc. These robots working together is more efficient than the robots that work separately. These cooperated works with information exchange could avoid repeated works while searching people. Moreover, it saves time to search survivals and is able to save more people. In other field such as nursing robot, security robot, there are also such demands existed for multiple robots to collect data so that we can know abnormal situation such as when nursing robots detect a falling down of person and invasion detected by security robot in time.

To achieve the goal to provide the service with multiple robots that can share information with each other to do their work efficiently, we consider that a software framework for building the distributed robotics service is required to satisfy the requirements. The one of the most widely used software framework for robotic systems is ROS (Robot Operating System) [1, 2], and it provides a structured communications layer above the host operating systems. It also provides libraries and tools to help software developers create robot applications. The ROS node makes no assumptions about where in the network it runs, allowing computation to be relocated at run-time to match the available resources, which makes it suitable for distribute robotic system.

†Shibaura Institute of Technology University

2. Issues

As we described in the section one, we had developed the distribute robot searching system with intelligent which used the advanced machine learning algorithms and it has the ability for searching efficiently for the target object based on target's features. The system is shown in the Figure 1, The first step, a cascade classifier [6] had been built up from a large amount of positive images and negative images. Then, while multiple robots are moving around, the images taken by the robot are sending to the server, and we use the cascade classifier to recognize whether there is a target thing in these images. At the same time, RGB images and depth information from RGBD camera was used to build up a map by SLAM algorithm on server. All of the calculation work was finished on server which leaded to the reduction of load on robots.

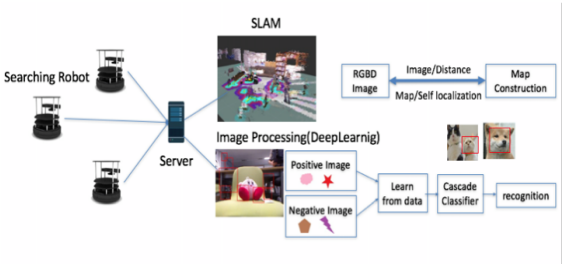

Figure 1 Distribute Robot System

It is realized with the ROS, RGBDSLAM, and deep learning libraries. However, there is not any approach about using multiple robots to improve the efficiency for searching.

ⓒ 2018 Information Processing Society of Japan

2.1. Calculation Load

As illustrated in Figure 2, the calculation load of building up a cascade classifier is high, it costs about 35 percent of the CPU and 20 percent of RAM resource in almost an hour for building up a cascade classifier which is not reasonable for robotic applications.

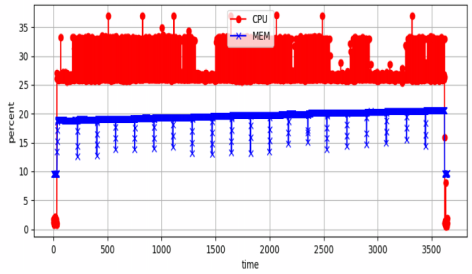

Figure 2 Calculation Load

- Using Raspberry pi to build up a cascade classifier
- CPU: Coretex-A53, 1.2Hz, 4 core, RAM: 1GB
- Time of building up cascade classifier: 1 hour

Therefore, in order to make this part of time short, it is necessary for the implementation to distribute this part of load to a powerful server.

2.2. Latency of network

The multiple service robotics system that we develop uses a small local network to communicate and share the information with each other. However, in the future, it is required to consider the wide area network distributed systems. Therefore, the network latency may become an issue. If the data collected by robots and the commands to control the robots may not arrive in time which can lead to a reduction in efficient and even some damage to the robots.

3. Proposed System

In this research, we propose a system that using multiple robots for searching efficiently with considering the performance problem of distributed robotics system with connection to a network.

In Fig 3, we show the whole system that is based on ROS middleware and cloud server. The multiple robot systems can share information through topics which means that when information exchange is required, the publish side (no matter server or robot) will publish a topic and the subscribe side will subscribe the topic that hold the same name as the publisher is publishing. For example, if robot(client) want the information that the server holds, the server will publish a topic and the robot can subscribe this topic and then information exchange can be finished. and all the robots are controlled by ROS master which is running on server. If robot wants to get data from others, the master node will tell the robot the IP address where the data is. While heavy load of calculation is needed, the data will be sent to server for calculation and server will send result back to the node that needs the result.

Currently, we consider to implement the cloud server by setting up a physical server consists of several virtual machines to make up a cluster for complex calculation. And in order to make the calculation quickly, we consider that building up a Hadoop cluster will be helpful. And because of the large amount of traffic which should be handled well, a Kafka cluster is also needed in this system. For image process, we plan to use Hadoop Image Process Interface(HIPI) for the procession of the large amount of image.

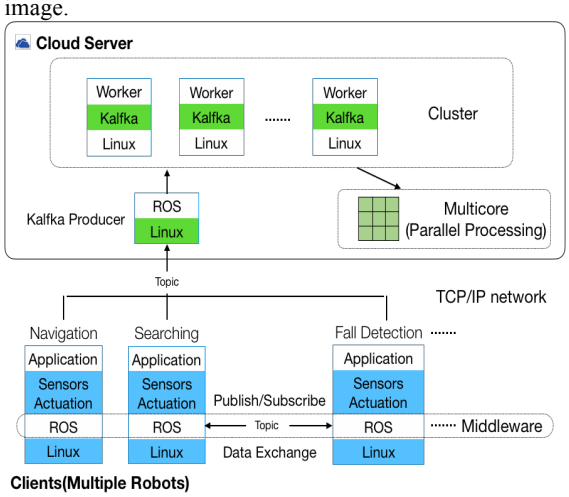

Figure 2 Proposed System

Reference

[1] “About ROS”. http://www.ros.org/about-ros/, (accessed 2017-11- 16).

[2] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng, “ROS: an open-source Robot Operating System”, ICRA Workshop on Open Source Software, 2009.

ⓒ 2018 Information Processing Society of Japan