

# A Role Based Framework for Robot Sensor Management

Ran WANG<sup>1</sup>, Xiao-Dong LV<sup>2</sup>, Dian-Xi SHI<sup>3</sup>, Zhe-Hang TONG<sup>4</sup>

<sup>1</sup>National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China

<sup>2</sup>National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China

<sup>3</sup>National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China

<sup>4</sup>National Lab for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China  
<sup>1</sup>532576041@qq.com, <sup>2</sup>lxdnogfly@sina.com, <sup>3</sup>dxshi@nudt.edu.cn, <sup>4</sup>zhehangt@163.com

**Abstract:** Sensors play an important role in the execution of robot tasks as an important component of understanding the world. In this paper, we propose a role-based sensor management framework on ROS (Robot Operating System), in which the role is a set of information needed for robot sensor development. Firstly, we develop a sensor management interface to facilitate the development of sensor applications on robot. And a subscriber module is proposed to shield the details of the sensor driver and the communication between processes, the sensor drivers on the robot are effectively managed. Robot sensors have different executive ability in different situations when robot performing tasks. For example, we know that RGB camera can collect a clear image when the light is strong, while generate a blurred image in weak light environment. Secondly, we propose an environment based sensor role dynamic evaluation mechanism to get the confidence of sensor in a certain environment. The confidence represents the executive ability of sensor when performing tasks. The confidence provides the basis for switching the task schemes. In the mechanism we can dynamically configure the environment information and confidence calculation method. In the experiment phase, we verify the effectiveness of the framework through a robot patrol task. Experiment results show that our framework can effectively manage the robot sensors with a certain environmental adaptability.

## 1. Introduction

With the rapid development of robot technology, different types of robots in different platforms emerge in an endless stream. Robots are used in different fields, such as industry, education, entertainment, service and so on [1]. Sensors play an important role in the execution of robot tasks as an important component of understanding the world. The development of sensor technology have greatly improved the perception ability of robot. Intelligent robots are often equipped with a large number of different types of sensors to detect the environments and collect the data. It is of great significance for development of robot sensor application to manage sensors on the robot effectively. Robot needs an intelligent sensor interface for gathering the data [2]. As the most popular robot operating system, ROS (robot operating system) [3] [4] provides a unified communication

architecture for heterogeneous sensors data acquisition. However, as far as the author knows, ROS does not provide a unified management method or framework for different robot sensors. We do not know a lot of information for the development of sensor applications, such as sensors the robot has, related information of sensors, sensors that are performing tasks and so on. In ROS, the process of inter-process communication is carried out through the publish/subscribe mode to get the sensor data. The developer needs to know the related topics and messages of the hardware driver. Different sensors have different hardware drivers and publish different topics and messages. Sensors of the same type may also publish different topics due to different manufacturers and driver developers. We need to understand information of underlying drivers to obtain the sensor information and data. It is advisable for developers to get sensor information directly without knowing the details of the inter-process communication. Shielding details of

hardware driver and communication between processes are critical to a sensor management framework. We present the concept of sensor roles to represent the set of information needed in the development of robot sensors applications and a subscriber module to shield the sensor hardware driver details.

The environments in which the robot performs the task are complex. In the real world, it is necessary to take into account the robot's attributes and environmental uncertainties [5]. Task execution ability of sensors varies in different environments [6] [7]. For example, when robot performs patrol and reconnaissance tasks, in the case of adequate light the RGB camera has a strong task execution ability and can collect better data; while in the case of insufficient light, it is better to use infrared camera to collect data. In order to express the task execution ability of sensor in a certain environment, sensor confidence is added to the sensor role. The confidence information can provides basis for switching task schemes and multi-sensor data fusion.

In this paper, we propose a role-based sensor management framework on ROS, which shields the details of sensor hardware drivers and the inter-process communication. In order to facilitate the development of sensor applications we present the concept of sensor role which is a set of information for sensor development. In order to express the capability of task execution, confidence information is added to the sensor role. This paper designs a sensor dynamic evaluation mechanism, which can obtain the real time role confidence by configuring the related environment information and the confidence calculation function in the confidence rule file. The structure of this paper is as follows: Section 1 introduces the research background and organizational structure; Section 2 analyzes the related work; Section 3 outlines the overall design of this framework; Section 4 details the relevant modules of this framework; Section 5 verifies the effectiveness of the framework by a robot patrol task; Section 6 summarizes the thesis.

## 2. Related Work

### 2.1 ROS

ROS is the most popular open source robot operating system, the emergence of ROS has greatly promoted the research of robot and code reuse in robot application development. Process exists as a node in ROS and the communications infrastructure of ROS provides the ability to communicate across processes for nodes. A robot system consists of multiple nodes, and the most common communication pattern between nodes is publish/subscribe mode, as shown in Figure 1. For sensor data acquisition, the sensor drivers are equivalent to the Pub nodes and the applications are equivalent to the Sub nodes. In addition, ROS also provides a service communication mechanism which is similar to RPC (remote procedure call).

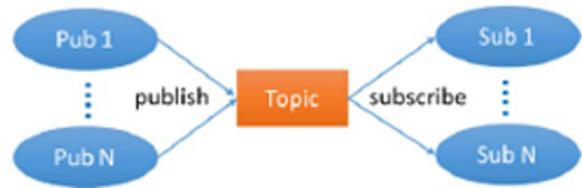


Figure 1. Publish subscribe communication

### 2.2 Android Sensor Management

Android as the most popular portable device operating system provides a good manner to get the sensor information for different hardware devices. And android provides a unified user access interface [8]. Figure 2 shows a very simplified sensor management architecture in android. Android shields the hardware heterogeneity through HAL layer (hardware abstraction layer). The well-established sensor architecture makes Android a robust platform for perceptual applications.

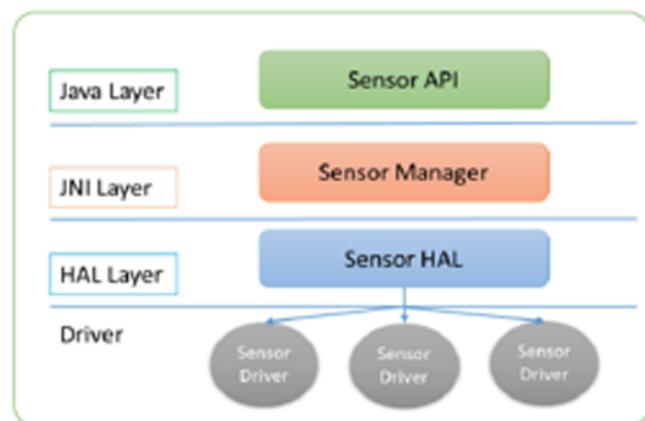


Figure 2. Publish subscribe communication

### 2.3 Classic Sensor Management

A good sensor management system can dynamically manage the sensor resources and change sensor configuration through the results of sensing [9]. There are many methods for sensor management. In [10] and [11], the temporal characteristics of sensor management are presented and the sequence decision process is described as a general Markov process. In [12], a sensor management framework for mobile robot is established using Bayesian Confidence Network and decision theory. In [13], the Boolean matrix is defined based on the sensor capability and effectiveness, and predict the task performance using the sensor performance model and then select the sensor model beneficial to the goal. In [14], knowledge reasoning and fuzzy theory are applied to sensor management which achieve some effects. There are many classic sensor management methods, and the goal of sensor

management system is to complete the task effectively and improve the accuracy of the information fusion system [15]. We have done the following work with reference to the idea of Android sensor management architecture and the classic sensor management methods: (1) we provide a subscriber module to shield sensor hardware driver details on ROS; (2) the sensor confidence is introduced to represent the task execution ability of the sensor; (3) an environment based sensor dynamic evaluation mechanism is designed to obtain the confidence.

### 3. Overall Framework Design

#### 3.1 Role Definition

In order to provide a general sensor management framework for robots, we present the concept of sensor role. Sensor role is a set of information needed for robot sensor application development and we introduce confidence to sensor role to express the task execution capability of sensor. Sensor role is dynamic and environment related. Sensor role contains five parts, namely, attributes, capabilities, platform, configuration information and confidence, as shown in Figure 3. The role attributes are inherent information of sensor, including sensor name, identifier, device manufacturer, version, and encapsulated sensor topic with corresponding message. The role ability represent the function and strength of sensor, including sensor type, detection range, maximum sampling frequency, maximum resolution and power. The role platform represents the platform where the sensor performs the task, including land, sea and air. The role configuration information represents the configuration parameter for data acquisition, including sampling frequency and resolution. The role confidence indicates the task execution capability of the sensor in the current environment which includes the confidence value and the name of confidence rule file. The confidence rule file includes the name of environment information and confidence calculation method. The sensor role has two states, the development state and running state. The sensor role exists in the form of XML file during developing state and in the form of object during running state. The sensor role file corresponds to the sensor on the robot.



Figure 3. Sensor role definition

#### 3.2 Overall Architecture Design

The overall architecture of the role-based robot sensor management framework is shown in Figure 4. The framework includes three parts, role catalog module, sensor manager module and dynamic evaluation mechanism module. Role catalog module is responsible for three parts, conversion between role file and role object, management of the sensor role object and roles in performing the task. All the role objects are added to the role list. And sensors that need to acquire the data are added to the task list. We manage the sensor role object and roles in performing the task through processing these lists.

Sensor manager module directly interacts with the application layer, and with this module developers can easily get the role-related information. Sensor manager provide four kinds of operations, including get the information about the sensor, register and unregister the sensor, select the sensor to be evaluated, and set the environment monitoring frequency. If we select the role to be evaluated, the dynamic role evaluation mechanism will be activated, and then we can get the real time confidence of the sensor. We can set the environment monitoring frequency, this frequency indicates how long it will take to reevaluate the sensor.

The dynamic evaluation mechanism includes two parts, role evaluation module and environment monitor. Role evaluation module is used to calculate the confidence of the sensor. And environment monitor is a thread, which is used to inform the role evaluation module to reevaluate the sensor at a fix frequency. Role evaluation involves two aspects, environment abstract and confidence rules. Environment abstract module is used to get the value of the concrete environment information. For example how much Lux is the light intensity, how high is the barometric altitude. The confidence rules include what environment information and confidence calculation method we need to calculate the confidence of the sensor. The developer can customize the environment information and confidence calculation method. In Section 4 we will detail the implementation of related modules.

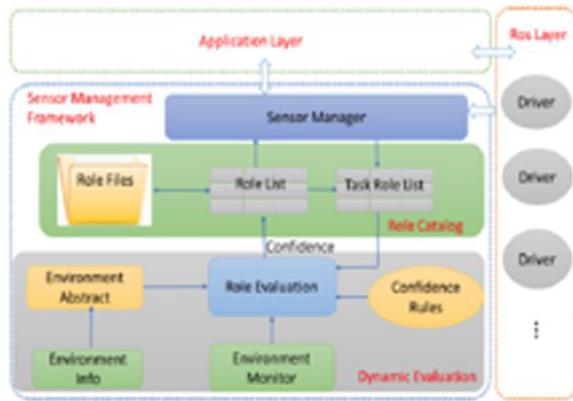


Figure 4. Overall architecture of the framework

## 4. Implementation Details

In this section we will detail the implementation of the framework. The whole framework is written in Python.

### 4.1 Role Catalog Module

Role catalog is responsible for three parts, conversion between role file and role object, management of the role list and management about the task role list. In the framework initialization phrase, role catalog module will resolve the role files and convert the files to objects. Developers can easy get the information of sensors with the object. The role list contains the objects of all the sensors on the robot. Once the developer set the sensor role needed to be evaluated, the object of the sensor role will be added to the task role list. The role evaluation related module will get the list, and makes a corresponding evaluation.

### 4.2 Sensor MANAGER

Sensor manager module is the unified interface interact with the application layer. The architecture of sensor manager is shown in figure 5. Sensor manager include three parts: unified info manager, subscribe module and the evaluation switch. The unified info manager can get the sensor related information from the role list, such as attributes, ability, confidence, and so on. Subscriber module is responsible for generate and start the subscribe thread and get the sensor data. Developer can use evaluation switch to get the sensor role evaluated. If we open the switch of the role sensor, the role will be added to the task role list. And roles in the task list will be evaluated. When we need to get data, we just need to register sensor with the sensor name and the callback function pointer without caring about the topic name and message of the sensor driver. We don't need to change the code of data acquisition when the hardware driver changes. The timing diagram of data acquisition is shown in Figure 6. When we initialize the framework, at first the unified info manager will

inform the catalog module to create the role object collection. Then the unified info manager will get the topic and message from the sensor object, subscriber module will generate the subscriber thread object with the topic and other parameters configured. At last of the initialization phrase the unified info will get the role subscriber thread objects of all sensor roles on the robot. When we register the sensor, the unified info manager will set the listener which is the pointer of callback function, and then start the subscriber thread. The subscriber module will get the sensor data from the topic and pass it to the listener.

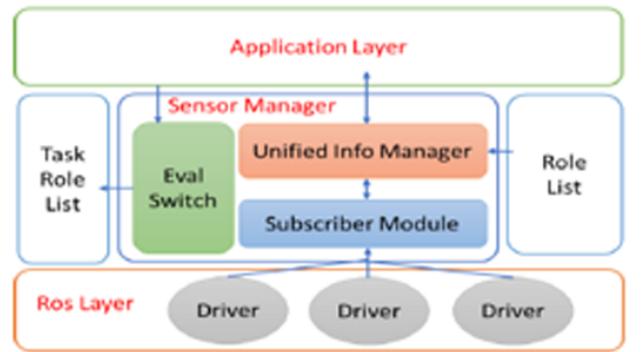


Figure 5. Sensor manager architecture

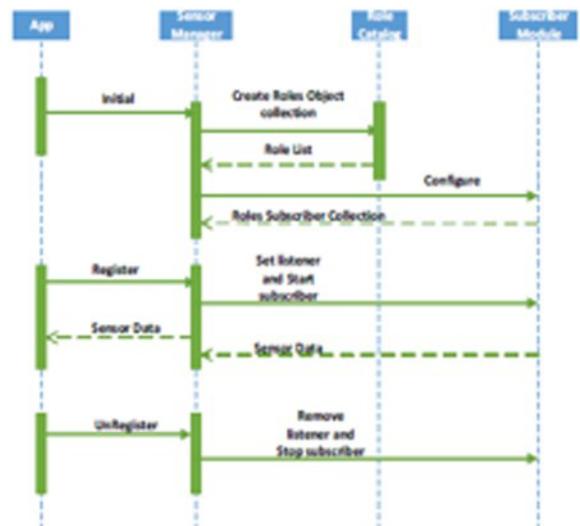


Figure 6. Timing diagram of data acquisition

### 4.3 Dynamic Evaluation Mechanism

The evaluation mechanism is a real time mechanism because we are monitoring the environment and constantly reevaluate the sensor. This mechanism includes two parts: sensor role evaluation and environment monitor. The process of role evaluation is as follows, at first the role evaluation module

will get the sensor to be evaluated from the task role list and then get the name of confidence rule file. In order to dynamically change the environment information and confidence methods, we use XML file to define the confidence rules. The role evaluation module will analyze the file and get the environment name and the confidence method name. And then the role evaluation module will find the related functions with these names. Functions about environment information acquisition are in the environment abstract module and confidence calculation methods are in the role evaluation module. After get the related function, the role evaluation module can get the environment value from the environment abstract module, and then input the environment value to the confidence calculation method to get the confidence. If the framework have integrated enough environment abstract and confidence calculation methods, the developer could get the evaluation results just through configuring the confidence rule file. By now we have developed some related method for some tasks. For example, related methods about collecting image data under different illumination conditions and high accuracy height measurement of low level UAV (unmanned aerial vehicle).

## 5. Experiments and Results

In the experiment phrase, we verify the effectiveness of the framework through a robot patrol and image collection task.

### 5.1 Experiment Setup

There are three equipment in the experiment, a Turtlebot robot equipped with a Kinect, a laptop with ROS installed, and an android phone. The model of the android is Samsung galaxy S5. The experimental scenario is shown in Figure 7. In the room, the light intensity is strong in one side and weak in the other side. Smartphone has equipped with a lot of sensors, in this experiment we will use some sensor of the smartphone to sensing the environment.

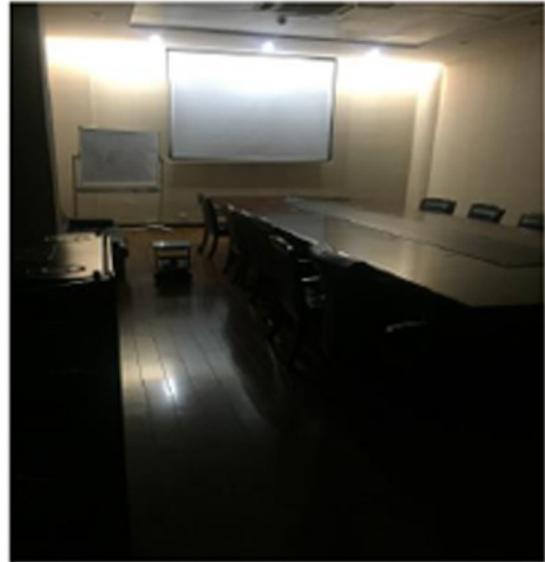


Figure 7. Experimental scenario

### 5.2 Experiment Details

The robot will patrol to collect image data in the room. The robot will collect RGB image with RGB camera in the side where the light intensity is strong and disparity map with depth camera in the other side.

In the development phrase, at first we need to configure the environment information and confidence calculation method in the confidence rule file. The confidence rule file of RGB camera is shown in Figure 8. By now, we just use light information to evaluate the camera. And the light data is sent by the phone with standard ROS message format. Our core work is not to study specific evaluation algorithms, therefore our evaluation method is relatively simple. In confidence calculation method, we take 24 Lux as the threshold of light intensity. When the light intensity is higher than the threshold, we will set the confidence of RGB camera to 1.0, and otherwise the confidence will be set to 0.0. The depth camera is just opposite. We will get the data and other information with the unified sensor manager interface. The collected image data is shown in Figure 9.

Results of the experiment show that our sensor manager framework could effectively facilitate the development of sensor application.

```

rgb_cam_confidence.xml x
1  <?xml version="1.0" encoding="UTF-8"?>
2  <confidence>
3    <name>rgb_sensor</name>
4    <environment>
5      <name>light</name>
6    </environment>
7    <model>rgb_sensor_confidence</model>
8  </confidence>

```

Figure 8. Confidence rule file of RGB camera

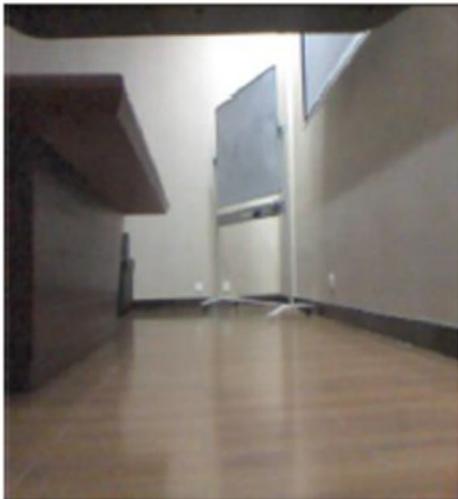


Figure 9. (a) RGB image



Figure 9. (b) Disparity map

## 6. Conclusions and Future Work

This paper proposed a role-based unified sensor management framework on ROS to facilitate the development of sensor applications. We propose the concept of sensor role to represent the set of information needed for the development of the robot sensor. Developers can get the related information through the unified interface named sensor manager. We proposed a subscriber module to shield the details of topic and message. Developers can use the same data acquisition code without caring the type of the driver and the topics. We introduce confidence to represent the task execution ability of the sensor and propose an environment-related dynamic evaluation mechanism to get the real time confidence of the sensor. We development some functions of evaluation abstraction and confidence calculation, developers can use different confidence calculation function just through configuring the confidence rule file. We have verified the effectiveness of the framework through a robot patrol task. In the future, we will develop more experiment abstraction and confidence calculation functions to support more tasks.

## References

- [1] Al-Razgan M, Alfallaj L F, Alsarhani N S, et al. Systematic Review of Robotics Use Since 2005[J]. International Journal of Mechanical Engineering and Robotics Research, 2016, 5(2): 129.
- [2] Kumar Y B, Ilaiyaraja S. Smart interface for sensors in robot[C]// Sixth International Conference on Advanced Computing. IEEE, 2014:145-149.
- [3] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System[C]//ICRA workshop on open source software. 2009, 3(3.2): 5
- [4] "ROS" [Online]. Available: Available: <http://www.ros.org>
- [5] Suzuki S, Habib M K, Iijima J, et al. How to describe the mobile robot's sensor-based behavior?[J]. Robotics & Autonomous Systems, 2015, 7(2-3):227-237.
- [6] Murphy T B, Morison A. Affordances as a Means to Assess Human-Sensor-Robot Performance[C]// Human Factors and Ergonomics Society Meeting. 2016:1610-1611.
- [7] Tsung T T, Hoai N. Dynamic Performance Measurement of Proximity Sensors for a Mobile Robot[J]. Key Engineering Materials, 2014, 625(625):683-688.
- [8] Yan Y, Cosgrove S, Blantont E, et al. Real-Time Sensing on Android[C]// International Workshop on Java Technologies for Real-Time and Embedded Systems. ACM, 2014:67.
- [9] Hero A O, Cochran D. Sensor Management: Past, Present, and Future[J]. IEEE Sensors Journal, 2011, 11(12):3064-3075.

- [10] Malhotra R. Temporal considerations in sensor management[J]. 1995, 1:86-93 vol.1.
- [11] Meier L, Peschon J, Dressler R. Optimal control of measurement subsystems[J]. IEEE Transactions on Automatic Control, 1967, 12(5):528-536.
- [12] Schenker P S. Sensor models and a framework for sensor management[J]. Proceedings of SPIE - The International Society for Optical Engineering, 1993.
- [13] Rothman P L, Bier S G. Evaluation of sensor management systems[C]// Aerospace and Electronics Conference, 1989. NAECON 1989. Proceedings of the IEEE 1989 National. IEEE, 1989:1747-1752 vol.4.
- [14] Molina Lopez J M, Jimenez Rodriguez S J, Corredera J R C. Fuzzy reasoning for multisensor management[C]// IEEE International Conference on Systems, Man and Cybernetics, 1995. Intelligent Systems for the Century. IEEE Xplore, 1995:1398-140
- [15] Xiong N, Svensson P. Multi-sensor management for information fusion: issues and approaches[J]. Inf
- [16] ormation Fusion, 2002, 3(2):163-186.