Ph.D. DISSERTATION

Operation of an Intelligent Space with Heterogeneous Sensors

이종 센서들을 이용한 지능형 공간의 운용

BY

DOOJIN KIM

AUGUST 2014

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstracts

A new approach of multi-sensor operation is presented in an intelligent space, which is based on heterogeneous multiple vision sensors and robots mounted with an infrared (IR) sensor. The intelligent space system is a system that exists in task space of robots, helps missions of the robots, and can self-control the robots in a particular situation. The conventional intelligent space consists of solely static cameras. However, the adoption of multiple heterogeneous sensors and an operation technique for the sensors are required in order to extend the ability of intelligent space.

First, this dissertation presents the sub-systems for each sensor group in the proposed intelligent space. The vision sensors consist of two groups: static (fixed) cameras and dynamic (pan-tilt) cameras. Each sub-system can detect and track the robots. The sub-system using static cameras localize the robot within a high degree of accuracy. In this system, a handoff method is proposed using the world-to-pixel transformation in order to interwork among the multiple static cameras. The sub-system using dynamic cameras is designed to have various views without losing the robot in view. In this system, a handoff method is proposed using the predictive positions of the robot, relationship among cameras, and relationship between the robot and

the camera in order to interwork among the multiple dynamic cameras. The robot's system localizes itself using an IR sensor and IR tags. The IR sensor can localize the robot even if illumination of the environment is low.

For robust tracking, a sensor selection method is proposed using the advantages of these sensors under environmental change of the task space. For the selection method, we define interface protocol among the sub-systems, sensor priority, and selection criteria. The proposed method is adequate for a real-time system, which has a low computational cost than sensor fusion methods.

Performance of each sensor group is verified through various experiments. In addition, multi-sensor operation using the proposed sensor selection method is experimentally verified in the environment with an occlusion and low-illumination setting.

Key words: intelligent space, sensor selection, heterogeneous sensors, camera handoff, multiple camera tracking

Student Number: 2009-30911

Contents

List of Figures

List of Tables

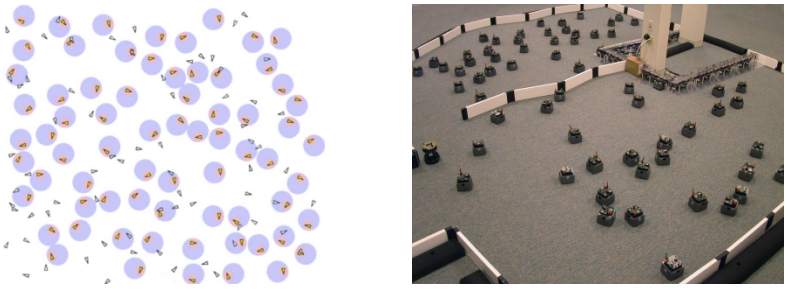
Chapter 1

Introduction

1.1 Background and Motivation

Existing mobile robot studies have focused on various technical issues such as control, obstacle avoidance, and localization of a single robot. In contrast, the interests of robotics researchers in recent years are moving towards the operation of multiple robots (agents) in complex environment. The change in interests has become possible as the robotics studies have been performed actively and the quantitative spread of robots has been enlarged. The development and dissemination of small robot such as the E-puck [1] and Elisa [2] also has helped researchers. These studies have considered the cost of the robots and the operation of multiple robots as a stumbling block. Many researchers have been attracted to the multiple robot studies in that the operation of multiple robots can divide a huge problem into several small

problems. In general, it takes too much times for a single robot to solve a huge problem. In the worst case, a single robot is failed to solve the huge problem. This division of the huge problem means that it takes a few times to solve the problem. The applicable area for robots has been enlarged from the factory to the military, home service, and hospital. Naturally, the task space of the robots have been changed from a simple to more complex, dense and cluttered environment such as a home, office, battle field, and space similar to nature. Rise in complexity of space, where robots perform their tasks or missions, have increased the necessity of research on multiple robots. There are many examples that are efficient and applicable from a solution in multiple robots: cleaning for huge space [3], scanning land mines [4], scanning minerals in oceans [5], securing buildings [6], managing warehouses [7], and encircling an intruder for security [8]. Figure 1.1 shows examples in the operation of multiple robots.



(a) (b)

Figure 1.1 (a) depiction of the collision avoidance problem [9] and (b) operation environment of multiple robots [10].

A high degree of intelligence is needed for an individual robot to perform various missions. Improving the intelligence of an individual robot requires the robot to be mounted with many sensors. In multiple agents operation, the robot sensors for intelligence may lead to increased costs. In addition, active sensors such as sonar or laser sensors have problems with the inter-sensor interference among multiple agents mounted with the sensors. Therefore, the possibility to encounter much more difficulties in a multiple agent mission would be higher if all relevant multiple robots are equipped with the necessary active sensors.

A sensor system mounted on the robots could be simple, if the environment itself has intelligence and is able to assist robots. Even if the robots have several sensors, usually an embedded and calibrated sensor environment is more efficient due to the immobility of the sensors. Such environment is called the smart environment or intelligent space (iSpace). The iSpace makes robot operation efficient by providing information for a task which requires high computing power to solve the complex problem associated with various robot sensors. The characteristics of the iSpace are the focus of this dissertation.

In the operation of the iSpace, there are basically two main frameworks: the centralized and distributed in a control aspect. The centralized framework could achieve optimality or sub-optimality of system performance because every part is controlled by only one main control supervisory center. However,

the centralized framework has limitations for scalability and robustness. On the other hand, the distributed framework has no main control center, and relies on the intelligence of each agent. Thus, a system using the distributed framework could be easily expanded. However, the efficiency of the system performance is hard to guarantee. In this dissertation, we focus on another framework for the iSpace operation: the hierarchical framework [11] which is combination of the centralized (top) and distributed (down) frameworks. The concept of the iSpace eases the organizing of the hierarchical framework. For example, robots have a certain intelligence (distributed) and the iSpace provides a mission or information obtained from an environment where robots conduct an allocated task (centralized).

Most iSpace studies have concentrated on control of devices or information fusion between devices. However, to my knowledge, there are no studies on operation of the intelligent space with heterogeneous sensors. Most studies have not utilized the characteristic of the iSpace because the studies used only static cameras, which are placed in a fixed position in an iSpace, as the main sensor. Thus, we compose an iSpace using two groups of static and dynamic camera and perform interworking among these sensor groups and robots mounted with an infrared (IR) sensor.

1.2 Related Work

Intelligent environments originated from ubiquitous computing. Studies

on intelligent environment began in the mid-1990s. Most ubiquitous computing studies on intelligent environment have been conducted on recognition and awareness of people in various environments [12-14].

Intelligent environments focus on human-centered systems. Furthermore, intelligent environments are appropriate to free people from artificial devices around their body. Small mobile devices are often used in ubiquitous computing for accessing computers. The EasyLiving project [15] at Microsoft is concerned with the development of architecture and technologies for intelligent environments. Visualization Space [16] at IBM is a visual computing system created as a testbed for a deviceless multimodal user interface. The Interactive Workspaces project [17] at Stanford focuses on augmenting a dedicated meeting space with large displays, wireless or multimodal devices, and seamless mobile appliance integration. The Classroom2000 at the Georgia Institute of Technology was designed and used for education environment [18]. The system is actually utilized in a real university classroom for three years. The KidsRoom [19] was proposed by the media laboratory at the Massachusetts Institute of Technology (MIT). The KidsRoom is a perceptually-based, interactive, and narrative play-space for children. The KidsRoom is implemented by various high-end technologies for children's fun. The Project Oxygen [20] was also proposed by the Artificial Intelligence laboratory at MIT. The proposed system is designed for computer-mediated presentations. The designed system understands

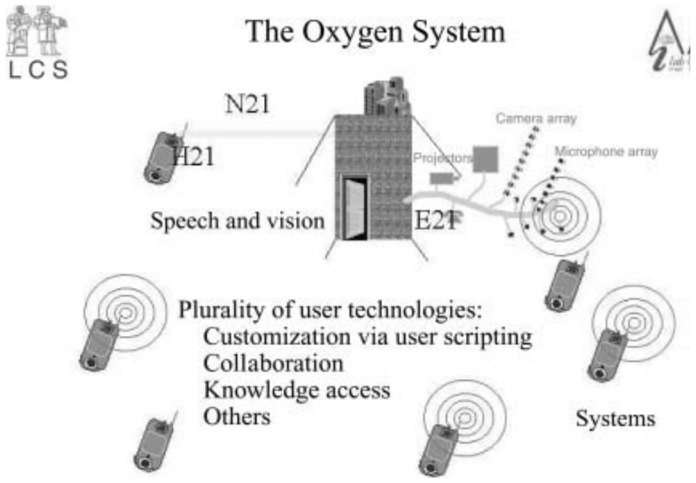


Figure 1.2 Conceptual figure of Project Oxygen [20].

naturally the presenter's speech and gesture. Sato [21] proposed the robotic room to offer indoor services for human living during human daily activities. The robotic room was composed of multiple robotic device and sensors surrounding the inhabitant. The ActiveCampus project [22] at the University of California, San Diego (UCSD) was developed to help classroom activities such as anonymous asking of questions, polling, and student feedback based on wireless location-aware computing.

Many studies on intelligent environment have been conducted. However, most studies are focused on the functional development of intelligent environment. There are only a few studies that proposed the architecture of intelligent environment with the consideration of adapting the architecture to a real environment. The objective of these particular studies is the interaction

between human and computer using face recognition [23], gesture recognition [24], speech synthesis [25], and speech recognition [26]. Several research groups are developing intelligent environments for specialized purposes.

However, there are other entities in the space of human activity. The entities could involve animals, robots, objects and other electronic devices. Thus applying computing resources to the other entities, except for humans, is not unreasonable. The application generates information about the entities, which could be helpful for humans. In this respect, expanding ubiquitous research from humans to other entities is a natural phenomenon. Thus, our research could be involved in the expansion to robots.

The iSpace is proposed by Lee [27] that is a supporting system for robots. The iSpace does not aim to rid sensors or autonomy from robots. Instead, the iSpace supports a robot by providing the resources it lacks to act as a normal robot, while helping a robot with good resources to act as an even better robot [28]. Each smart camera in this system localizes and tracks the robot independently. This distributed camera system can secure scalability of the system. However, there is a limitation to improve the efficiency of the system. And it is hard to extend ability of the system because the system only consists of the homogenous sensors.

1.3 Contributions

The original iSpace consists of homogenous sensors, which is a limitation.

Thus, this dissertation presents a new type of an iSpace system based on heterogeneous vision sensors and robots mounted with an IR sensor. We also present the design and implementation of an innovative iSpace system. The detail contributions of this dissertation are as follows.

First, this dissertation presents a low-cost localization method for robots. An existing localization system such as ViCON and Ubisense are very expensive. However, the proposed method is at least one-half times less expensive than ViCON and Ubisense, and offers performance similar to them.

Second, this dissertation presents a handoff method between static cameras. The handoff method of the iSpace in [27] utilizes a pre-calculated reliability map based on distance from a robot to the cameras. The handoff method is not desired because each camera must have a map, and the map is re-calculated and re-distributed when the height of the robot is changed. In order to overcome the limitation of the method, a handoff method using a world-to-pixel transformation is proposed in this dissertation. Using the transformation, the proposed handoff method generates and verifies a pixel coordinate of the robot. Therefore, our method is more efficient in memory usage.

Third, this dissertation presents localization and tracking method that uses a dynamic camera. Usually, localization using cameras requires more than two cameras. However, this dissertation proposes a localization method that uses a single dynamic camera. The localization is performed by using the

geometrical relationship between the camera and a robot. The tracking method is based on color information and tracks the target robot robustly despite a change of the background.

Fourth, this dissertation presents a handoff method between dynamic cameras. The handoff method considers prediction of the robot position as well as the distance from a sensor to a robot. The distance is the usual factor for handoff. This consideration makes the handoff process more reliable.

Fifth, this dissertation presents localization and tracking method using an IR localization sensor and IR tags. We developed a localization method using an IR sensor attached on the robot. A single IR tag cannot cover the entire area in the proposed iSpace. We decided on the adequate number of IR tags. The multiple IR tags are deployed to cover the entire area. Also, an adaptive extended Kalman filter is adopted to solve problems that are caused by using the multiple tags.

Sixth, this dissertation presents the design, implementation, and application of operation between heterogeneous vision sensors: static cameras and dynamic cameras. The complex configuration of the heterogeneous cameras also can overcome the disadvantages when a single type of camera is utilized. The static cameras have better performance in localization than dynamic cameras. However, dynamic cameras due to its motorized unit (pan-tilt unit) have larger coverage than static cameras. Therefore, we propose a complementary structure and interface protocol for the operation with static

and dynamic cameras.

Finally, this dissertation presents the design, implementation, and application of operation between the proposed iSpace and a robot. The proposed iSpace consists of multiple cameras. The iSpace could not perform its function according to the change of illumination in its environment. The IR sensor mounted on the robot can work in a low-illumination environment. Therefore, we propose a complementary structure and interface protocol for the operation using the iSpace and a robot for robust tracking.

1.4 Organization

This dissertation is organized as follows. Chapter 2 describes the original concept of intelligent space. Chapter 3 explains the intelligent system in the context of hardware and software. Chapter 4 explains localization and tracking methods when using each sensor. The handoff methods among homogeneous sensors are also proposed and explained. Chapter 5 provides an interface protocol among sensor systems and explains a proposed operation among the sensors and robots. Chapter 6 verifies the proposed intelligent space system for operating performance with discussion. Finally, Chapter 7 summarizes this dissertation and the implementation results with conclusions and future works.

Chapter 2

Overview of Intelligent Space

2.1 Original Concept of Intelligent Space

The Intelligent Space (iSpace) is a space that is equipped with sensors, which enable the space to understand a happened situation in the space. Originally, Lee [27] proposed the concept of iSpace which is basically a model of an ubiquitous computing system. While most previous ubiquitous computing studies focused only on humans, the iSpace is designed here for aiding robot navigation and especially is concentrated on interworking the sensory environment and a robot. Figure 2.1 shows the concept of the iSpace.

The main device of the original iSpace is the distributed intelligent network device (DIND). In [29], they developed the DIND, a kind of smart sensor, and constructed an iSpace based on the DINDs. The DINDs localize the position of human and robot using information from geometric camera

calibration information and the installed position of the DINDs. The robot in the iSpace has four color-code bars in order to be detected robustly by the iSpace. The DIND can also detect a human using hand and head shape detection after background subtraction. The DINDs are inter-connected through a network in order to cooperate between the DINDs. For this cooperation, a handoff method among DINDs is proposed, which is based on distance from DINDs to the robot. Using the handoff method, information sharing is possible between the DINDs. Figure 2.2 shows the picture of the DIND and the robot used in [29].

According to Lee [29], the iSpace was classified by information availability in three categories: potential information space, passive information space and active information space.

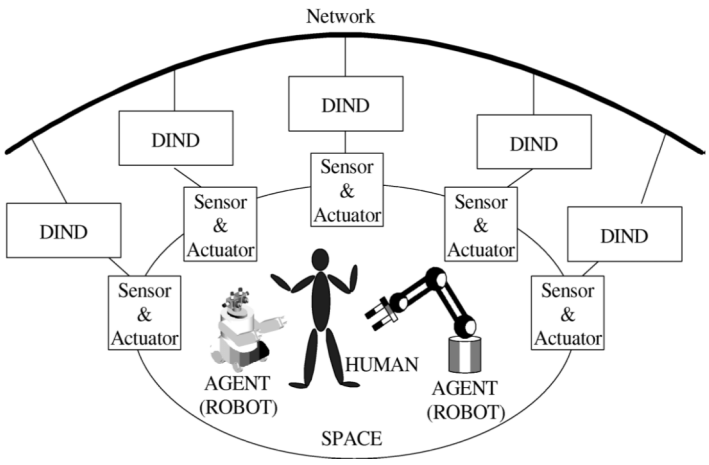
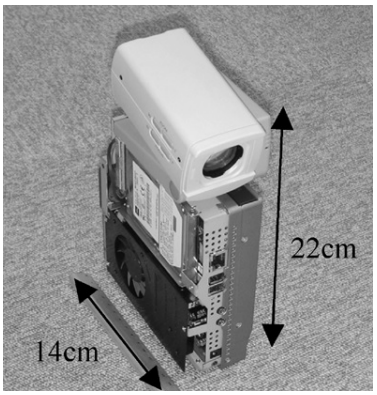
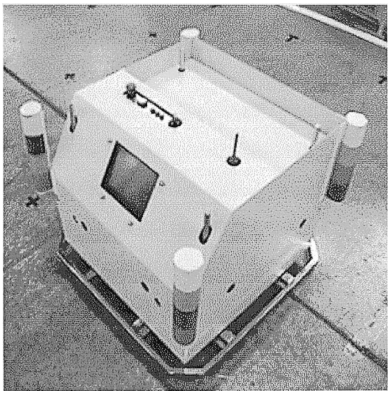


Figure 2.1 Conceptual figure of an Intelligent Space [27].



(a)



(b)

Figure 2.2 Picture of (a) DIND, and (b) robot with the color-coded bar [29].

The potential information space is an environment placed with natural objects. Robots could guess the category of an environment using little information from natural objects. The artificial information understood by agents is implanted in passive information space. Such examples include milestones, traffic signs, and signposts on the driveway. Robots must have devices or modules specially designed for understanding the implanted information because passive information space is usually centered on humans. The active information space provides custom-made information for each robot so that robots do not require particular efforts. This dissertation targets the active information space as in most iSpace studies.

2.2 Related Research

Most iSpace studies have utilized the installed positions and calibration information of sensors that are mainly cameras. Lee and Hashimoto [27] have proposed an early version of the iSpace concept and implemented the iSpace using a DIND-based sensor network. Lee et al. [29] also proposed a distance-based handoff method and suggested the reliability rank map, which is a tool to determine control authority. In [30], a human tracking technique using the mean-shift tracking method has been developed in a DIND-based iSpace. In [31], a localization and 3-D reconstruction method for a mobile robot has been proposed in the iSpace using four static cameras. The method uses motion segmentation and objective function minimization. The objective function consists of segmentation boundaries of the mobile robot, linear and angular velocity of the robot, and the depth. In [32], an auto-calibrated sensor network system has been proposed. This auto-calibration is performed by detecting a mobile robot in the overlap region between the field-of-views (FOVs) of the sensors.

There are some cooperative approaches to an iSpace and a mobile agent. In [33], an estimation method for a landmark position using information fusion has been proposed. The method fuses information from laser sensors mounted on the iSpace and a robot. The center of the sensor fusion was the extended information filter. In [34], research on localization and shape reconstruction of the mobile robot has been performed using an onboard sensor (wheel encoder) of a robot and external sensors (static camera). The

system uses the external sensors and tracks the robot using the SIFT (scale invariant feature transform)-supervised KLT (Kanade-Lucas-Tomasi) tracker. In [35], there was an attempt to fuse information from the iSpace to simultaneous localization and mapping (SLAM) using a laser range finder. Fernández et al. [36] conducted a study on the cooperation between the system with multiple static cameras and the mobile robot mounted with an active beacon and wheel encoder.

There are few studies on the iSpace systems using other sensors, rather than cameras. In [37], a triangulation-based positioning system has been proposed in a sonar-beacon-based iSpace. In [38], a sensor network using active radio-frequency identification (RFID) module has been used to construct an information system using heterogeneous sensors. The RFID module is added to temperature, humidity, and IR distance sensors. However, there is no structure to handle information from heterogeneous sensors. The discussion of the authors on the collected data only exists in [38]. In [28], a fuzzy algorithm and RF signals have been used for the algorithm estimating position of the target in a wireless sensor network. A reconfigurable iSpace has also been suggested using wall-climbing robots instead of static sensors [39]. A study on the interaction between robot and human has been conducted in an iSpace [40], where the projector-mounted robot was assisted by an iSpace for human gesture recognition using color-depth(RGB-D) camera and robot localization.

Researchers in the computer vision field have studied object tracking using a camera network [41-43]. However, these studies aimed at tracking the target object with no cooperation or interworking among the sensor network and the target object. This study aims at the development of an iSpace that targets the robot and interworks with the robot, and is different from the vision studies with a camera network.

2.3 Problem Statement and Objective

Most studies on iSpace have concentrated on the use of a static camera. A camera is cheap and provides much information, but has narrow FOV. Therefore, the iSpace must be equipped with other sensors in order to expand its ability.

The studies have mainly focused on localization and tracking of each sensor and sensor fusion. However, an additional study of a handoff method also is required to operate multiple homogenous sensors. The conventional handoff method of the iSpace is based on a pre-calculated reliability map using only a distance factor. In addition, it is not an efficient method in the context of memory usage because each camera must have different maps according to the height of the robots.

The original iSpace consists of the distributed sensors, DINDs. This is more scalable than a centralized system. However, systems based on DIND did not consider the efficiency of the entire system. Moreover, there are no

studies on interworking among sensors or the iSpace and a robot, and there is no consideration about the change of environmental conditions in the iSpace.

This dissertation aims at construction of the iSpace where operation with different sensors is possible. For this purpose, we first construct a multiple static camera system. The systems performs an accurate localization and tracking of robots. An improved handoff method is also developed for tracking that uses multiple static cameras. Second, we construct a multiple dynamic camera system. The systems performs the localization and tracking of a robot that uses changeable FOVs. Also, a handoff method is developed for the multiple dynamic camera system. Third, the self-localization system is constructed for a robot that uses an IR sensor. Finally, we define an interface protocol among sensor systems, design an operation that uses strengths of the each sensor system, and implement the proposed system.

Chapter 3

Architecture of a Proposed Intelligent Space

The iSpace is the space that is designed to communicate with robots using various sensors in space. We focus on the operation of multiple robots in terms of an iSpace. For this purpose, we need to know the position of the robots and then run the process to achieve a goal task by issuing commands to the robots, tracking robots using visual sensors, and controlling the robots if necessary. Thus, this dissertation addresses the achievement of robot localization, robot tracking, and interworking with multiple agents. Before the detail explanation of localization, tracking, and interworking techniques, this chapter is concerned with the structure and components in terms of hardware and software. In section 3.1, the hardware architecture of the proposed iSpace

is explained in terms of the physical components. In section 3.2 the software architecture is explained in terms of the functional components.

3.1 Hardware Architecture

A camera that is installed in a fixed position with the fixed FOV is called a "static camera." A camera installed in a fixed position and is capable of change with the FOV is called a "dynamic camera." Examples of a dynamic camera include pan-tilt cameras, zoom cameras, and pan-tilt-zoom cameras. The proposed iSpace consists of a control server computer and a metallic structure mounted with multiple sensors. Figure 3.1 shows the entire physical

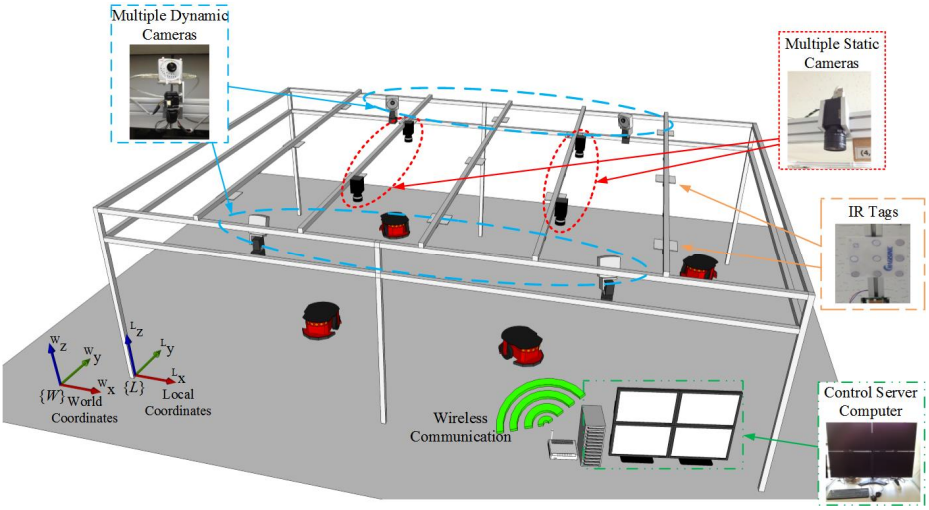


Figure 3.1 Depiction of the proposed iSpace.

structure of the proposed iSpace, and a situation where robots perform a mission in the iSpace. Many components such as a control server computer, multiple static cameras, multiple dynamic cameras, artificial infrared tags, and robots are shown in the figure. The control server computer communicates with multiple sensors and robots and processes information from multiple sensors. Thus, a multi-core processor is required because of the parallel processing ability. First, we explain a metallic structure required for attaching multiple sensors in the next section.

3.2.1 Metallic Structure

The proposed iSpace has a metallic structure as the frame for multiple sensors. The metallic structure is designed to attach various sensors at the desired positions in the environment. This metallic structure consists of iron beams that have a $6cm \times 6cm$ square-shaped horizontal cross-section. The height, width, and depth of the structure mounted multiple sensors are approximately $5.7m$, $8.2m$, and $2.4m$, respectively. The structure includes six columns, five horizontal direction supports, and twelve side supports. The length between the horizontal supports, where mounted with markers and static cameras, is approximately $1.28m$. Figures 3.2 and 3.3 show the side views of the metallic structure. Figure 3.4 shows the picture of the metallic structure.

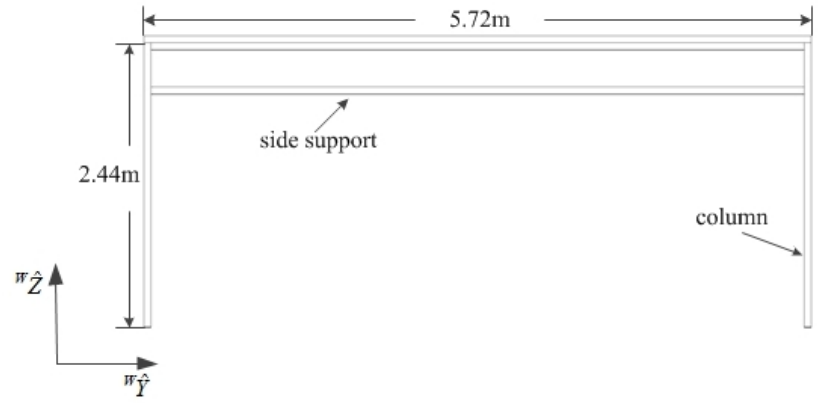


Figure 3.2 Side view of the metallic structure orthogonally projected on *y*-*z* plane of the world coordinates.

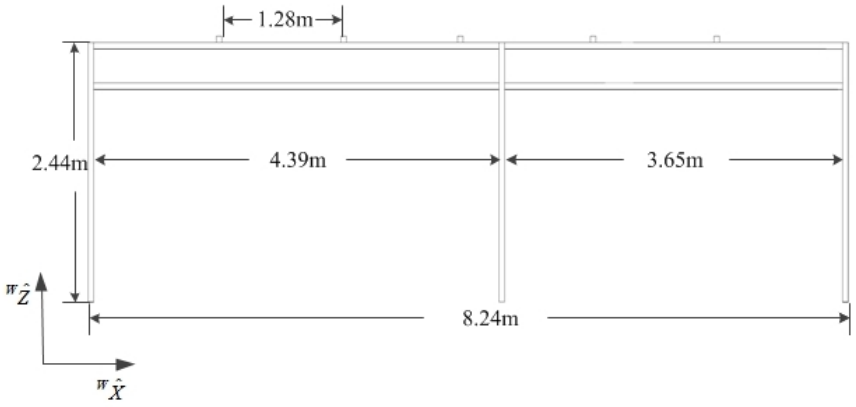


Figure 3.3 Side view of the metallic structure orthogonally projected on the *x*-*z* plane of the world coordinates.

Figure 3.4 Picture of the metallic structure.

3.2.2 Static Cameras

We adopt the GEViCAM GP-3780C camera using the Bayer color filter as the camera component of the static camera system for the proposed iSpace. This camera captures the color image with a one-third lower bit rate than the normal RGB camera. This advantage is useful to connect multiple cameras to a single computer. The camera has 30 frame per second (FPS) capturing speed and 1032×779 pixel resolution. Although the Bayer-color-filtered image requires more several conversion processes, the process time does not affect the total image acquisition processing time.

The camera's interface is the gigabit Ethernet that is designed to transmit the image data to approximately 100 meters. A multi-port Ethernet adapter is

needed to simultaneously transmit images to the main control server. The Intel pro 1000 series are usually chosen as the Ethernet adapter because of the high compatibility for many network cameras and the high stability for Ethernet communication. Figure 3.5 shows the static camera, the multi-port network adapter for multiple cameras, and the static camera attached to the environment. Figure 3.6 shows the system architecture of the static camera system.

The camera's lens has a 68° horizontal FOV and a 54° vertical FOV. We installed four static cameras on the horizontal-direction supports of the metallic structure. We set the height of the static cameras to the same value. This limits the scale space to a particular extent and the processing time of the robot detection is reduced.

(a) (b) (c)

Figure 3.5 Picture of (a) a static camera, (b) a quad port Ethernet adapter, and (c) the static camera installed on the metallic structure

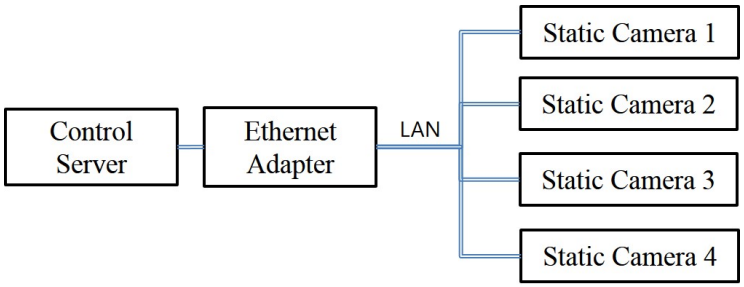


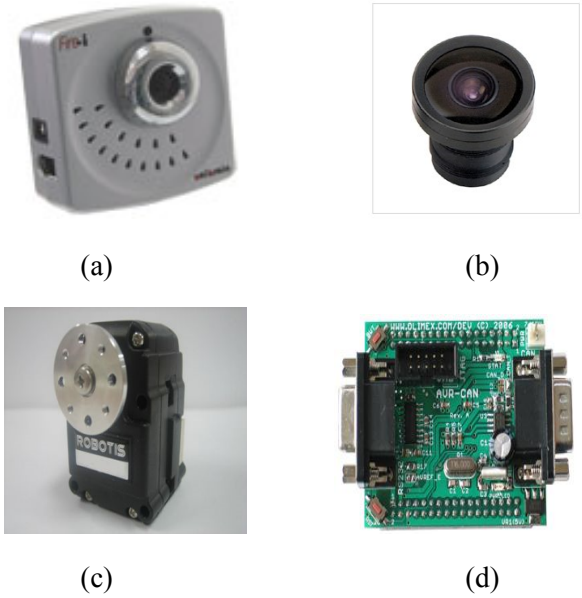
Figure 3.6 System architecture of four static cameras

3.2.3 Dynamic Cameras

The dynamic camera system consists of a camera and pan-tilt unit. We adopt the Unibrain Fire-i webcam as the camera part of the dynamic camera. Figure 3.7(a) shows a picture of the camera part of the dynamic camera system. The camera has 30 FPS capturing speed and 320×240 pixel resolution. The camera has an 82° horizontal FOV and a 66° vertical FOV. In general, webcams have narrow horizontal FOV, approximately 40°~45°. In order to cover a larger area and maximize advantage of the tracking ability of the dynamic camera system, the adopted camera is mounted with the lens of a large FOV. We installed four dynamic cameras on the side supports of the metallic structure to cover a large area. Figure 3.7(b) shows the picture of the camera lens.

The camera's interface is the b-type of IEEE 1394. The four cameras are serially connected by the daisy chain scheme, which is one of the multiple device connection type of the IEEE 1394.

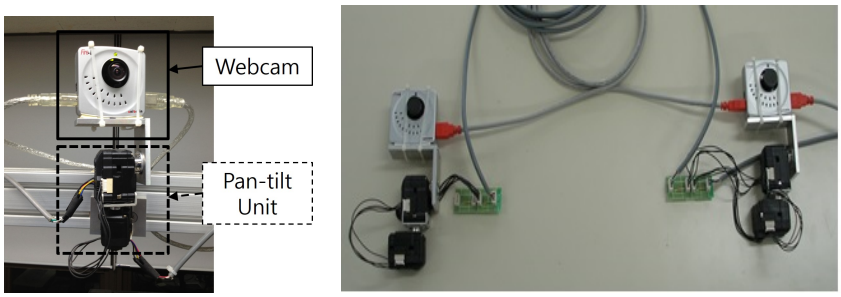
We built a pan-tilt unit that uses two Robotis dynamixel CX-28 servomotors. Figure 3.7(c) shows a picture of the servomotor. The motor has



(a) (b)

(c) (d)

Figure 3.7 Parts of dynamic cameras: (a) Fire-i camera, (b) servo motor, (c) 2.1 mm wide lens, and (d) AVR board for control area network communication.



(a) (b)

Figure 3.8 Picture of (a) dynamic camera and (b) two dynamic cameras wired by the daisy chain scheme

an encoder that provides the pan or tilt angle information to the control server computer. The motor is controlled through the control area network (CAN) communication. For CAN communication, the motors are given an identification (ID) number. The communication can reduce the cost and complexity of installation because of the serial connection scheme. In addition, the communication is advantageous for real-time control. If the multiple commands for multiple motors are given, each of the motors only responds to the command, which has the motor's ID number. We adopted two AVR boards to control the eight servomotors via the CAN communication. The AVR board is shown in Figure 3.7(d).

Figure 3.8 shows a picture of the dynamic camera and the daisy-chained dynamic cameras. The panning range of the unit is -170° ~ 170°, and the tilting

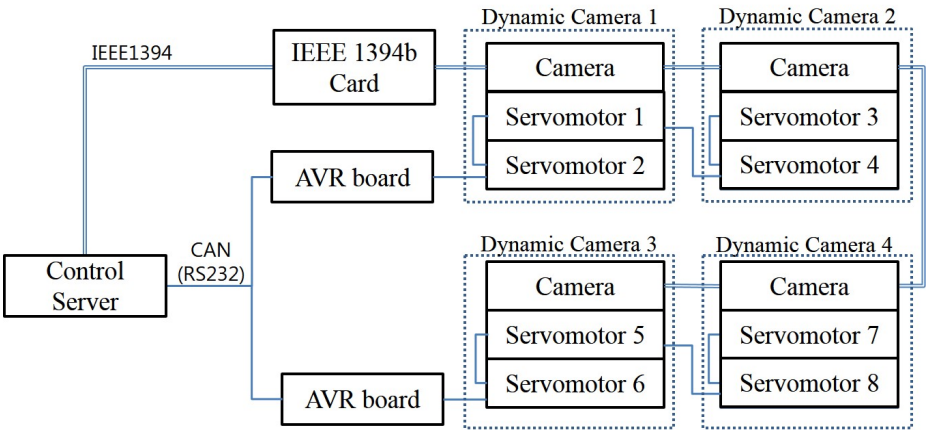


Figure 3.9 System architecture of dynamic cameras

range of the unit is -90˚ ~ 30˚. Figure 3.9 shows the system architecture of the dynamic camera system.

3.2.4 Infrared (IR) Camera and Passive IR Tags

The proposed iSpace has passive IR tags that reflect IR light. We attach thirteen IR tags on the bottom of the metallic structure's horizontal supports. The IR tags have a pattern consisting of eight IR dots. Figure 3.10(c) shows

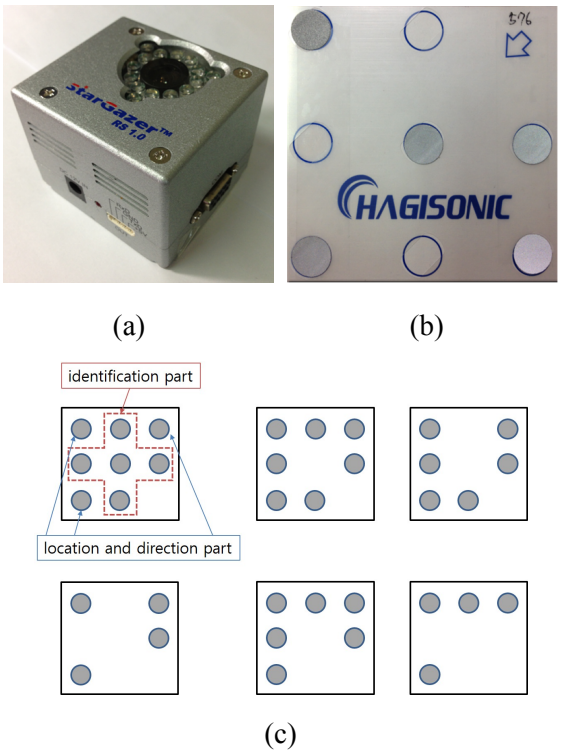


Figure 3.10 Pictures of (a) IR sensor, (b) IR Tag, and (c) example of a IR tag's pattern

that the IR tag consists of the identification part and the location and direction part. Dots at the center cross area are the identification of the tag. Dots at the four corners of the tag represents the location and direction of the tag.

The robots are mounted with the Hagisonic IR sensor that emits IR light and captures an IR image reflected from the tags. The robots can calculate the position of themselves based on an IR reflection tag captured by the IR sensor. The update rate of the IR sensor is 10 FPS. Figure 3.10 shows pictures of the IR sensor and IR tag.

3.2.5 Mobile Robots

In general, detection methods of a robot include two methods that use color information and geometric information. The method using color information has fast processing time, but is weak for illumination change and is difficult to intuitively determine the accuracy of detection results. The method using geometric information is robust to illumination change and easy to intuitively determine the accuracy of detection results, but requires more processing time than the method using color information. Thus, robots have a geometric marker to help detection by the static cameras.

We attached the marker on the rotation axis of the robots. Figure 3.11(a) represents examples of the marker for robot detection. The size of the marker has 158*mm* width $\times$ 158*mm* height. The marker consists of an identification part and a detection part. The marker has the identification part because of the

pattern of squares in the marker. When the mobile robot moves, the marker may be blurred. Thus, a long and narrow square is placed at the right side of the marker as the detection part. The angle of the marker is determined by the combination of the identification part and the detection part of the marker. The center of the marker is the origin of the coordinates for the marker. The center of the marker is the same as the origin of the robot coordinates. The

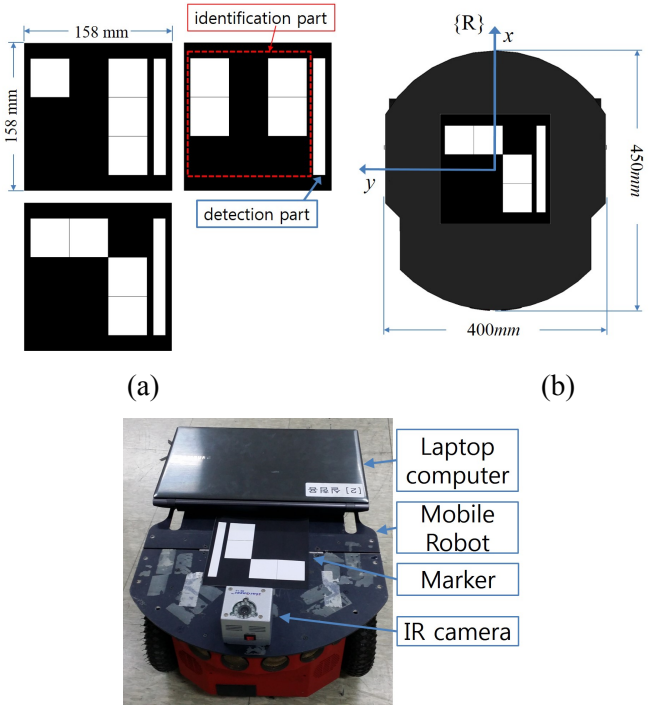


Figure 3.11 (a) examples of the marker, (b) axis of the mobile robot, and (c) the robot attached the marker on the middle of upper plate of the robot and mounted with the IR sensor

angle of the marker increases counter-clockwise from the x-axis of the marker. Figure 3.11(b) represents the axis of the mobile robot. Figure 3.11(c) shows a picture of the mobile robot attached the marker and mounted with the IR sensor and a notebook computer for mobile robot control.

The Adept MobileRobots Pioneer 3-DX is employed as the mobile robot in the iSpace. The robot is controlled by the notebook computer on the upper plate of the robot via serial communication such as RS232. The notebook computer communicates with the control server computer via IEEE 802.11n wireless local area network (WLAN). The detail specifications of the robot are described in Table 3.1.

Table 3.1 Specifications of the mobile robot

Feature	Specification	Feature	Specification
Length	450 *mm*	Runtime	30 hours
Width	400 *mm*	Drive wheel diameter	200 *mm*
Weight	9 *kg*	Payload	17 *kg*
Max. translation velocity	1500 *mm/s*	Max. rotation velocity	360 deg/s
Max. translation acceleration	2000 mm/s^2	Max. rotation acceleration	2000 mm/s^2
Drive	Two-wheel differential drive		

3.2 Software Architecture

The proposed software system consists of three subsystems according to the device type. Each system focused on not only cooperation of its devices, but also interworking of the other devices. The device of the first system is static cameras. The device of the second system is dynamic cameras. The device of the third system is a mobile robot. Each system has individual localization and tracking modules. Figure 3.12 shows the software architecture of the propose iSpace.

Interworking Static Camera System: The Interworking Static Camera System (ISCS) performs cooperation among multiple static cameras for

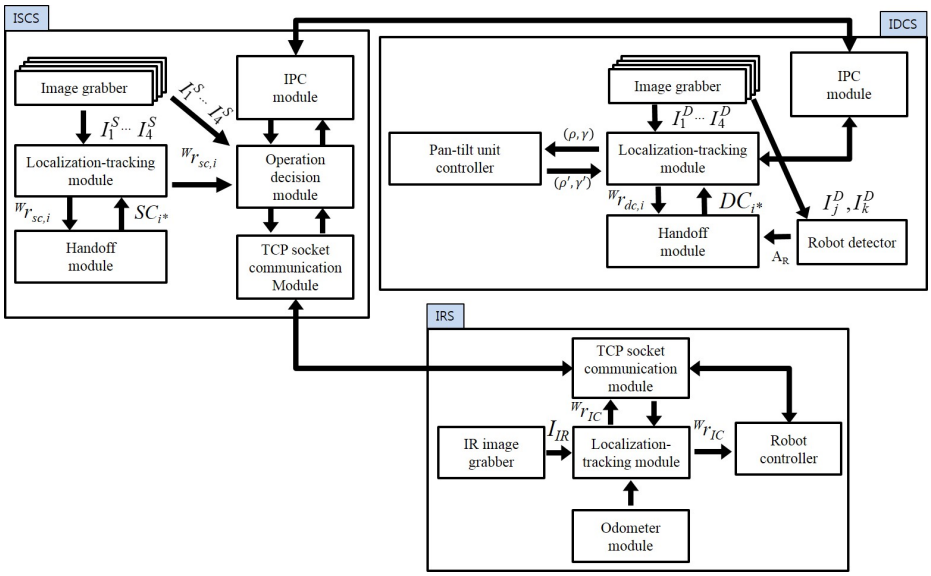


Figure 3.12 Software architecture of the proposed iSpace

localization and tracking of target robots. This subsystem consists of three software modules for localization and tracking: an image grabber, localization-tracking module, and handoff module. The image grabber per camera captures images I_i^S from the camera. The localization-tracking module calculates the robot position ${}^W r_{sc,i}$ using pixel-world mapping information, camera positions, and the height of the target. In addition, this module detects the marker that represents the center of the robot. The handoff module determines the tracking camera SC_{i*} among the static cameras. The handoff module assigns a cooperative attribute to the localization-tracking module in the ISCS.

This subsystem also includes three modules for interworking with the other subsystems: Inter-Process Communication (IPC) module, operation decision module, and Transmission Control Protocol (TCP) socket communication module. The IPC module transmits commands or data between the ISCS and the proposed Interworking Dynamic Camera System (IDCS). In addition, this module interprets the commands by a command parser.

The operation decision module determines the interworking among the ISCS, IDCS, and proposed Interworking Robot System (IRS) by collecting and analyzing data from each subsystem. After this determination, the module generates an adequate command for the current situation in the iSpace.

The TCP socket communication module is designed to transmit

commands to the robots and receive data from the robots. TCP is an adequate communication protocol for a wireless communication environment because of its robustness of communication error and bidirectional attribute. Also, this module interprets commands between the ISCS and IRS.

Interworking Dynamic Camera System: The Interworking Dynamic Camera System (IDCS) performs cooperation among multiple dynamic cameras for localization and tracking of a target robot. This subsystem consists of five software modules for localization and tracking: an image grabber, localization-tracking module, handoff module, pan-tilt unit controller and robot detector.

The image grabber per camera captures image I_i^D from the camera. The localization-tracking module calculates the robot position ${}^W r_{dc,i}$ using pan-tilt information, camera geometric information, and camera positions. This module pursues the robot using the mean-shift tracking algorithm based on a color histogram. While the module tracks the robot, the module also generates the pan-tilt control input ρ, γ and transfers the control input to the pan-tilt controller.

The handoff module determines the tracking camera DC_{i*} among the dynamic cameras using robot dynamics and the geometrical relationship among the robot and cameras. This handoff module assigns a cooperative attribute to the localization-tracking module in the IDCS.

The pan-tilt unit controller conducts panning and tilting of the motor units

based on the input and transfers the current panning and tilting values ρ', γ' to the localization-tracking module.

The robot detector searches the robot using the handoff-candidate camera when the handoff process is triggered. The adjacent-frame difference method is used to detect the target robot.

This subsystem also includes the IPC module for interworking with the ISCS. The IPC module in the IDCS is similar to the IPC module in the ISCS. The IPC module transmits commands or data between the IDCS and the ISCS. In addition, this module interprets the commands by a command parser.

Interworking Robot System: The subsystem Interworking Robot System (IRS) consists of five modules: IR image grabber, localization-tracking module, odometer module, robot controller, and TCP socket communication module.

The IR image grabber captures the reflected image of the IR tags and transfers the image to the localization-tracking module in this system. The localization-tracking module calculates the robot position ${}^{W}r_{IC}$ using the wheel encoder values and the geometrical information and pattern of the IR tags. The odometer module transfers the wheel encoder values to the localization-tracking module. The robot controller moves the robot based on the command from the ISCS. The TCP socket communication module is similar to the TCP socket communication module in the ISCS. The TCP socket communication module is designed to transmit data to the ISCS and receive commands from

the ISCS. Also, this module interprets the received commands from the ISCS.

The commands among ISCS, IDCS, and IRS follows the interface protocol explained in section 5.1.

Chapter 4

Localization and Tracking of Mobile Robots in a Proposed Intelligent Space

4.1 Localization and Tracking with an IR Sensor Mounted on Robots

4.1.1 Deployment of IR Tags

The sensing range of each IR tag is about 1.2~1.5*m* radius when the height of the IR tag is 2.5*m* [44]. The IR tags have to be arranged in order to find the global coordinates of the sensor because the proposed iSpace is larger than this coverage.

The uncertainty of the values calculated by using the IR sensor increases as the distance between IR tags and the sensor increases [45]. In addition, we figured out that the error is increased as the distance between the IR sensor

and the IR tag is increased. Therefore, the error can be reduce by densely placing IR tags. However, this can cause frequent change of the detected IR tag. Thus, the discontinuity of the tracking occurs more than before.

Thus, the deployment of the IR tags is the crucial problem to reduce errors. We found the appropriate interval is about 80% of the tag's height by experiments. When the height of the IR tag is 2.5*m*, the radius of sensing coverage is 1~1.2*m*. Thirteen IR tags is the solution to cover the space of 7 × 5*m* as depicted in Figures 4.1 and 4.2.

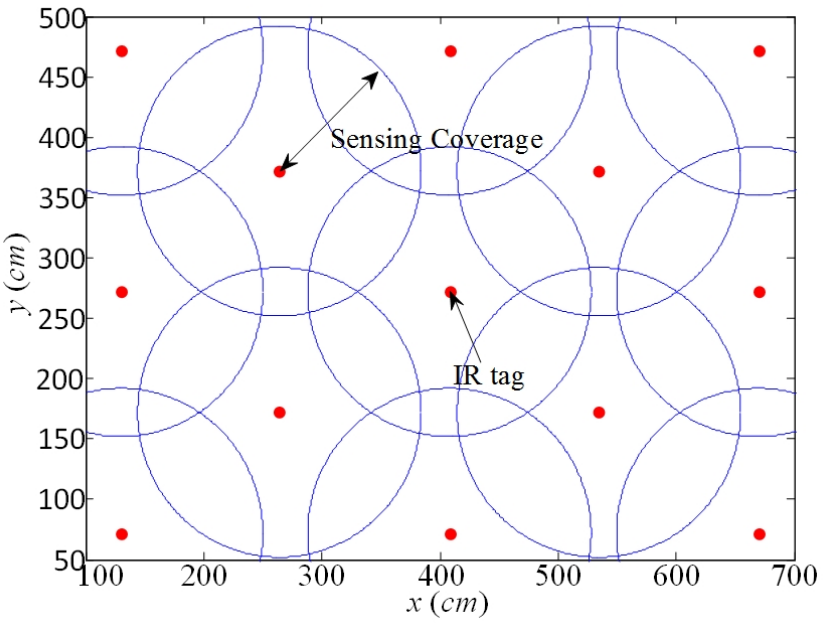


Figure 4.1 Deployment of IR tags.

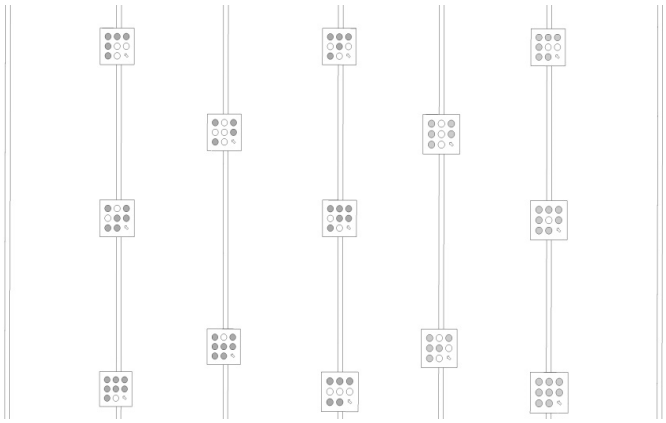


Figure 4.2 IR tags attached on the metallic structure (bottom view)

4.1.2 Localization and Tracking Using an IR Sensor

The IR sensor emits IR light that is reflected by passive IR tags with an independent ID pattern. Then, the IR sensor observes the reflected light and analyzes the IR images to determine the relative position of the sensor from the IR tag. Figure 4.3 explains how the sensor finds the location of itself.

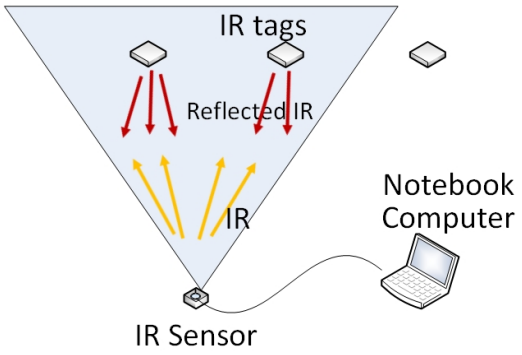


Figure 4.3 Localization process of the IR sensor

The IR sensor can obtain the relative position and bearing from each IR tag. If we know the position value of each IR tag in global coordinates, the global location of the sensor can be calculated by adding these values. Therefore, the sensor calculates the global location from each tag. If the sensor receives the data from several tags, the location is estimated by averaging these data.

This localization using IR tags has three shortcomings: uncertainty problem as distance increases, discontinuity problem, and noise problem. The uncertainty of the sensor value increases as the distance between the IR tag and the sensor increases as explained in section 4.1.1. This is the general problem of distance sensors.

As the coverage of the IR tag is limited, the detected tag will keep changing when the sensor moves from one place to another place. The data from each detected tag should be interpreted as the same location in ideal case. However, the result from newly detected tag is different from the existing data. The discontinuity in a tracking result is occurred when the reference IR tag is changed.

The specification of the IR sensor implements that the repetitive precision is approximately 2*cm*. However, in an actual test, there were some noises caused by the misidentification of the tag or the failure of detecting the IR tag. The sensor determines the relative position from the IR tag. Therefore, if the IR tag identification is misunderstood, the results would become inordinate

values. The failure of detecting the IR tag also gives unreliable data and makes the data difficult to understand.

In this study, we apply an adaptive extended Kalman filter (AEKF) [46] to estimate the position of robots equipped with an IR sensor in order to overcome the aforementioned shortcomings of the sensor. The data provided by the wheel encoder of the robot and the IR sensor are fused together by means of the EKF. To be specific, denote with the robot state $\mathbf{x}_k = [x(k), y(k), \theta(k)]^T$ at time k that is obtained from the IR sensor and the robot control input $\mathbf{u}_k = [\Delta s_r(k), \Delta s_l(k)]^T$ at time k, where Δs_r and Δs_r are displacement of the left and right wheel encoders respectively. The system is described by

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k\text{-}1}, \mathbf{u}_{k\text{-}1}, \mathbf{w}_{k\text{-}1}), \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k), \end{aligned} \tag{4.1}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are the process and observation noises that are assumed to be zero mean multivariate Gaussian noises with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively, and $\mathbf{z}_k$ is the odometer measures. Then, the performance of the filter can be degraded according to the noise statistics. We have to adjust these statistics according to the environment and data we received from the sensor. Therefore, the AEKF can adaptively estimate the correct locations even if we received poor sensor values. Measures are discharged if the difference exceeds a threshold. The structure of the proposed localization algorithm is

reported in Figure 4.4.

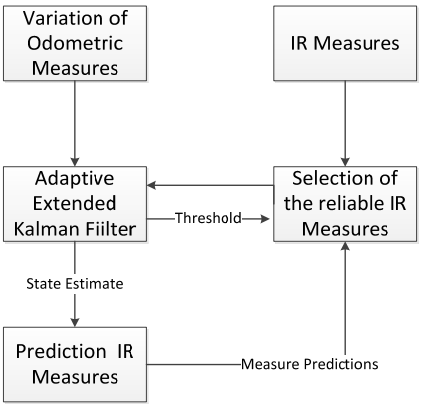


Figure 4.4 Estimation process using AEKF

4.2 Localization and Tracking with Multiple Dynamic Cameras

4.2.1 Localization and Tracking based on the Geometry between a Robot and a Single Dynamic Camera

Generally, the robot localization using a single camera is not possible because the data from a normal camera with a single lens does not contain depth information. However, a robot's position can be calculated using two assumptions with a single dynamic camera. First, we assumed that the robot exists on the optical axis of the camera. In our dynamic camera system, we have an interest only about an image in the center of the image because it is the place where the lowest distortion image of the robot is obtained. Second,

we assumed that the robot's height, position, pan angle, and tilt angle of the dynamic cameras are known to the proposed iSpace. These assumptions turn the localization problem using the dynamic camera to a simple geometric problem.

The calculation of the robot position is related to four coordinates: world coordinates, camera coordinates, pan coordinates, and tilt coordinates. The transformation ${}^{P}_{T}\boldsymbol{M}_{dc,i}$ from the tilt coordinates to the pan coordinates of the i^{th} dynamic camera is defined as

$$ {}^{P}_{T}\boldsymbol{M}_{dc,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_{T,i} & \sin\theta_{T,i} & 0 \\ 0 & -\sin\theta_{T,i} & \cos\theta_{T,i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2} $$

where $\theta_{T,i}$ is the tilt angle of the i^{th} dynamic camera. The transformation ${}^{C}_{P}\boldsymbol{M}_{dc,i}$ from the pan coordinates to the camera coordinates of the i^{th} dynamic camera is defined as

$$ {}^{C}_{P}\boldsymbol{M}_{dc,i} = \begin{bmatrix} \cos\theta_{P,i} & -\sin\theta_{P,i} & 0 & 0 \\ \sin\theta_{P,i} & \cos\theta_{P,i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.3} $$

where $\theta_{P,i}$ is the pan angle of the i^{th} dynamic camera. The transformation ${}^{W}_{C}\boldsymbol{M}_{dc,i}$ from the pan coordinates to the camera coordinates of the i^{th} dynamic

camera is defined as

$$ {}^{W}_{C}\boldsymbol{M}_{dc,i} = \begin{bmatrix} \cos\theta_{C,i} & -\sin\theta_{C,i} & 0 & x_{dc,i} \\ \sin\theta_{C,i} & \cos\theta_{C,i} & 0 & y_{dc,i} \\ 0 & 0 & 1 & H_{dc,i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.4} $$

where $\theta_{C,i}$ is the reference bearing of the dynamic camera relative to the world coordinates. $[x_{dc,i},\ y_{dc,i},\ H_{dc,i},\ 1]^{\mathrm{T}}$ is the installed position of the i^{th} dynamic camera.

Therefore, the calculated robot position from the i^{th} dynamic camera ${}^{W}r_{dc,i}$ is

$$ {}^{W}r_{dc,i} = {}^{W}_{C}\boldsymbol{M}_{dc,i}\,{}^{C}_{P}\boldsymbol{M}_{dc,i}\,{}^{P}_{T}\boldsymbol{M}_{dc,i}\,{}^{T}r_{dc,i}\,, \tag{4.5} $$

where ${}^{T}r_{dc,i}$ is the robot position calculated in the tilt coordinates. Figure 4.5 shows the relationship between the i^{th} dynamic camera and the robot. The distance $d_{rdc,i}$ between the robot and the dynamic camera is calculated by

$$ d_{rdc,i} = \frac{H_{dc,i} - h}{\sin\theta_{T,i}}\,. \tag{4.6} $$

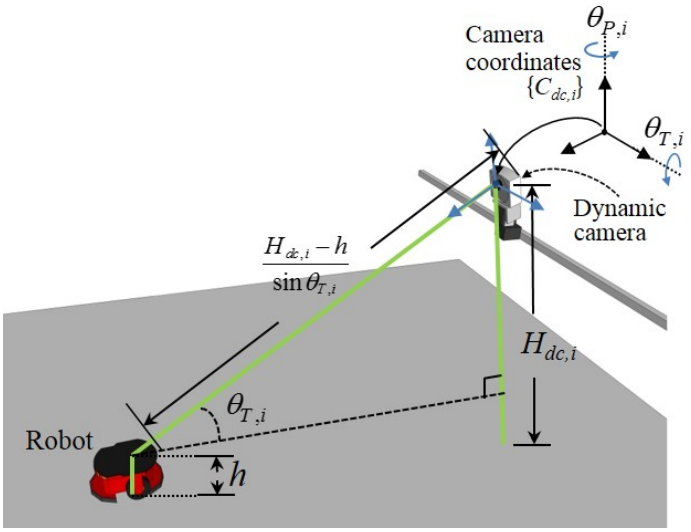


Figure 4.5 Relationship between the i^{th} dynamic camera and the robot

Thus,

$$ {}^{T}r_{dc,i} = \left[0, \frac{H_{dc,i} - h}{\sin\theta_{T,i}}, 0, 1 \right]^{T}. \tag{4.7} $$

Finally, ${}^{W}r_{dc,i}$ is calculated by the substitution of Equation 4.7 into Equation 4.5.

The robot tracking using a dynamic camera is performed in the image space of the dynamic camera. We employed the mean-shift tracker [47] based on the color histogram of the robot's image. The mean-shift tracker generates a vector based on the gradient ascent method in the image space. The mean-

shift tracker uses the local window and search only in the image space. Therefore, the mean-shift tracker has low computational cost relative to the sliding window approach for detecting the robot in the static camera system. We weighted the center of the target model at the initialization of the mean-shift tracker. The weighted target model makes the tracker more robust. In addition, the wide FOV of the lens adopted for the dynamic cameras prevents losing the target robot. When the tracker decides the current robot position in the image space, our system controls the motor units to place the target on the center of the image.

4.2.2 Proposed Predictive Handoff among Dynamic Cameras

The handoff method for dynamic cameras consists of four steps. First, the handoff process is triggered when the robot is moving out of the coverage of the tracking camera. Second, the robot's reachable points are predicted. Third, the next adequate camera to track is selected by evaluating adequacy of the cameras at the predicted points. Finally, if the robot is in the coverage of the selected camera, the handoff is executed.

For continuous robot tracking, the handoff process has to be triggered before the robot entirely escapes from the trackable coverage of the tracking camera. Figure 4.6 shows the geometrical relationship between the robot and the tracking camera DC_i. Table 4.1 summarizes the notation in Figure 4.5.

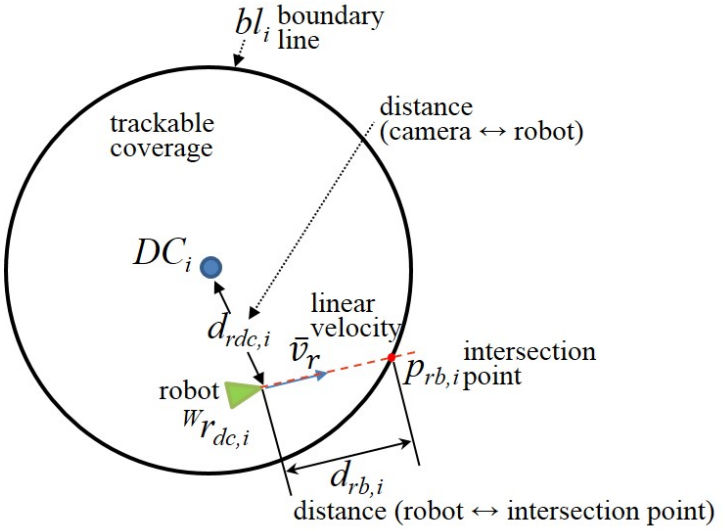


Figure 4.6 Geometry of the robot and tracking camera.

Table 4.1 Notations shown in Figure 4.6

Notation	Description
DC_i	the i^{th} dynamic camera
${}^W r_{dc,i}$	the robot position tracked by DC_i
$\bar{v}_r$	the robot's linear velocity vector
bl_i	the boundary line of the trackable coverage of DC_i
$p_{rb,i}$	the intersection point between the extension of $\bar{v}_r$ and bl_i
$d_{rb,i}$	the distance between ${}^W r_{dc,i}$ and $p_{rb,i}$
$d_{rdc,i}$	the position between ${}^W r_{dc,i}$ and DC_i

The handoff trigger criterion TC_i is defined as

$$TC_i = \left[\left(d_{rb,i} \le T_r \right) \wedge \left(\frac{dS_{dst,i}}{dt} < 0 \right) \right], \quad T_r = c_h t_h \left| \overline{v}_r \right|, \tag{4.8}$$

where $\wedge$ is the logical AND operator, and $S_{dst,i}$ is the evaluation function based on $d_{rdc,i}$. In the trigger threshold T_r, t_h is the average value of the required time to perform the handoff process, and c_h is a scaling factor. If TC_i is true, the handoff process is triggered.

When the handoff process is triggered, robot positions are predicted up to n-steps on the basis of a robot motion model using the curvature velocity [48]. The predicted position $r_{i,j}$ means that the predicted position generated in the j^{th}. $r_{i,j}$ are calculated by

$$r_{i,j} = {}^{W}r_{dc,i} + \begin{bmatrix} -\sqrt{\frac{2v_{r,j}}{w_{r,j}}} \sin\left(\frac{\pi - w_{r,j}\Delta t}{2} \right) \\ \sqrt{\frac{2v_{r,j}}{w_{r,j}}} \cos\left(\frac{\pi - w_{r,j}\Delta t}{2} \right) \end{bmatrix}, \begin{cases} \left| \overline{v}_r \right| - a_m \Delta t < v_{r,j} < \left| \overline{v}_r \right| + a_m \Delta t \\ w_r - b_m \Delta t < w_{r,j} < w_r + b_m \Delta t \end{cases}, \tag{4.9}$$

where a_m and b_m are the maximum linear and angular velocities of the robot, respectively, and $r_{i,j}$ is generated by using various inputs: the linear velocity $v_{r,j}$ and angular velocity $w_{r,j}$. Figure 4.7 shows an example of $r_{i,j}$ with $v_{r,j} \in [0,$

1.5] *m/s* and $w_{r,j} \in$[-0.35, 0.35] *rad/s*.

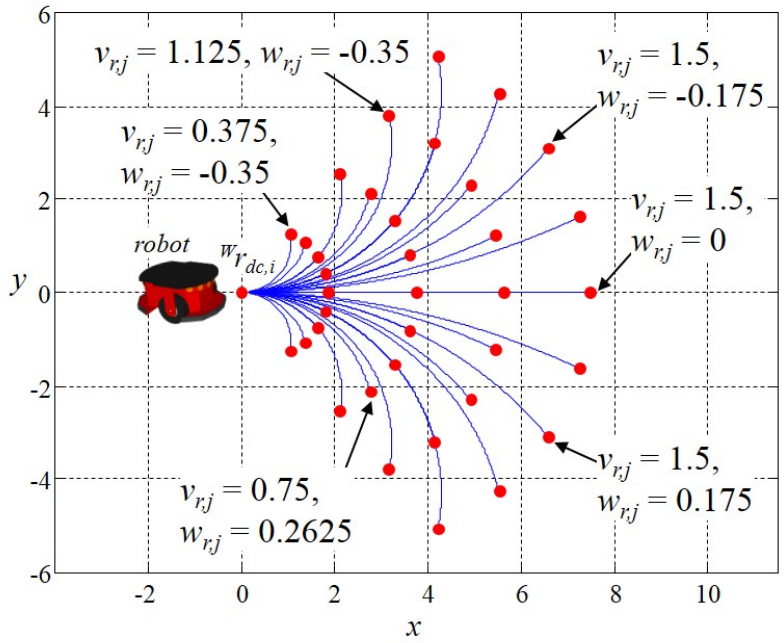


Figure 4.7 Examples of the predicted positions (red dots).

The handoff method between dynamic cameras has two factors to evaluate the handoff adequacy of the cameras. The first factor is the distance from a camera to the robot, which is a basic evaluation factor in most handoff methods. If the candidate camera for handoff is the k^{th} camera DC_k, the evaluation function $S_{dst,k}$ of DC_k is defined as

$$S_{dst,k} = \frac{1}{N}\sum_{j=1}^{N} S_{dst,k,j},\ S_{dst,k,j} = \frac{1}{1+\left\| {}^{W}p_{dc,k} - r_{i,j} \right\|}, \tag{4.10}$$

where $S_{dst,k,j}$ represents the evaluation function for DC_k at $r_{i,j}$, and N is the number of the predicted positions. The second factor is the robot's movement direction. In [49], the direction factor considered the relationship between the robot direction vector and the vector from ${}^{W}r_{dc,i}$ to the position ${}^{W}p_{dc,k}$ of DC_k. However, this method does not properly work in a hall-way situation where the cameras are deployed side by side. Thus, the relationship between the robot's direction vector and the vectors from DC_i to the other cameras is chosen as the direction factor in this dissertation. The evaluation function based on the proposed direction factor is defined as

$$S_{vel,i,k} = \sum_{j=1}^{N} S_{vel,i,j,k},\ S_{vel,i,j,k} = \frac{1}{2}\left(1 + \frac{\bar{v}_{r,j} \cdot \bar{d}_{i,k}}{\left\|\bar{v}_{r,j}\right\|\left\|\bar{d}_{i,k}\right\|}\right), \tag{4.11}$$

where $\bar{d}_{i,k}$ is the vector from ${}^{W}p_{dc,i}$ to ${}^{W}p_{dc,k}$, and $\bar{v}_{r,j}$ is the linear velocity vector of $r_{i,j}$. The total evaluation function for handoff is defined as Equation 4.12. The adequate camera for handoff has the maximum evaluation value of all the cameras, expressed in Equation 4.13.

$$S_{total,k} = w_{dst}S_{dst,k} + w_{vel}S_{vel,i,k}, \tag{4.12}$$

$$k^* = \arg\max(S_{total,k}), \tag{4.13}$$

where w_{dst} and w_{vel} are weighting values for each evaluation factor, and k^* is the index of the selected camera. Figure 4.8 shows the direction factor for handoff evaluation when the index of the tracking camera is 1.

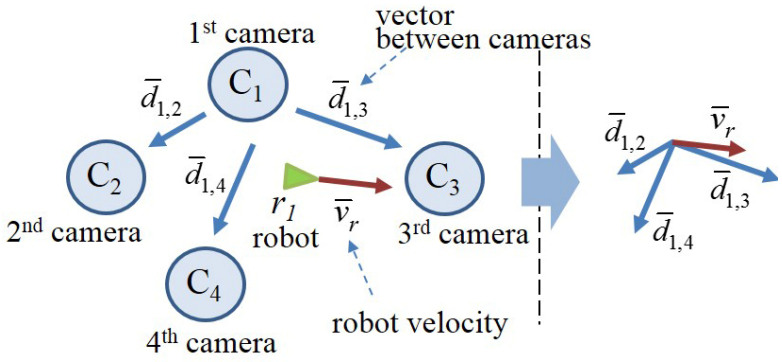


Figure 4.8 Relationship between the robot's movement direction and the vectors from the 1st dynamic camera to the other cameras.

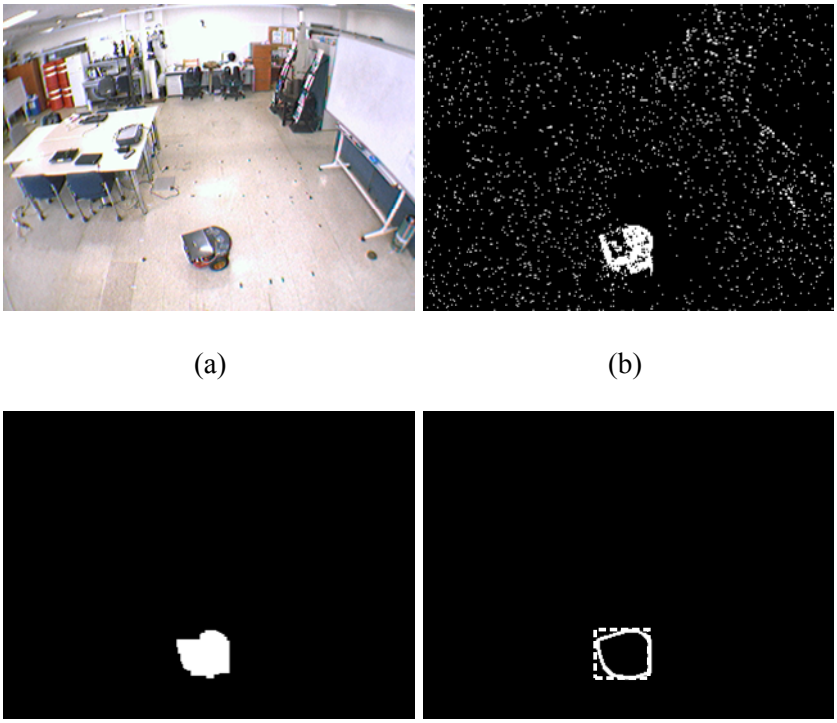
Although the next camera for handoff is selected, the handoff should be executed when the robot is inside the trackable coverage of the next camera DC_{k^*}. The robot should also be approaching and facing the selected camera. Therefore, the handoff execution criterion EC_{k^*} is defined as

$$EC_{k^*} = \left[SC(r_i, C_{k^*}) \wedge \left(\frac{dS_{\text{total},k^*}}{dt} > 0 \right) \right], \tag{4.14}$$

where $SC(a, b)$ is the function that determines whether a is in the trackable coverage of b. If EC_{k^*} is true, the handoff process is executed. If the handoff is executed, the selected camera gazes at the position, ${}^{W}r_{dc,i} + \eta \bar{v}_r$ ($\eta \geq 0$), ahead of the moving robot. At this position, the input for the robot tracking using the selected camera is generated by using the robot detection method. When the tracking using the selected camera is successfully started, the entire handoff process is finished.

Robot detection is required in order to initialize the target model for robot tracking. The robot's scale in the image from the dynamic camera will continuously change. This scale change requires much computational cost. Therefore, we do not use the marker to detect the robot in dynamic camera system. In general, robot detection for dynamic camera is achieved by using the feature (e.g., color, points, and edges) matching method and the frame difference method. The frame difference method has a simple process and requires low computational cost relative to other methods. These advantages are adequate in operating multiple cameras. In general, the frame difference method uses a fixed camera in a fixed background. In order to apply for our dynamic camera system, we used frame differencing of adjacent image frames. When our system is performing the detection of robots, the pan-tilt unit of the dynamic camera is fixed. We used the morphological processes, such as opening and closing, to reduce the noise of the result image from the frame difference. The result of the robot detection is then an edge contour. We

calculate the center point as the robot position in image space and the bounding box for the edge contour as the initial region for the target tracking. We also set a region of interest (ROI) in the image space. This ROI means the priority region in the image space. A candidate inside the ROI obviously has higher priority than a candidate outsize the ROI. Figure 4.8 shows results from the robot detection process.



(a) (b)

(c) (d)

Figure 4.9 Robot detection using morphology operation and contour extraction. (a) input image, (b) frame difference image, (c) result after applied morphology operation, and (d) object contour and bounding box

We explained the proposed predictive handoff method. Finally, Table 4.2 summarizes the differences between the proposed method and the conventional method [29].

Table 4.2 Comparison between the proposed handoff method and the conventional method

	Applicable to the dynamic cameras	Use distance factor	Use velocity factor
Proposed Method	Yes	Yes	Yes
Conventional Method	No	Yes	No

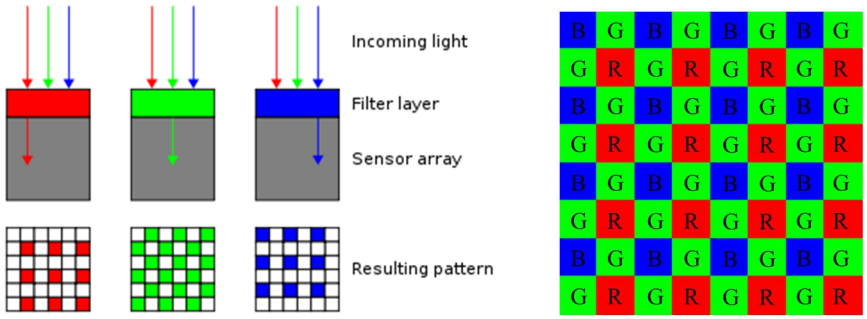
4.3 Localization and Tracking with Multiple Static Cameras

4.3.1 Preprocess for Static Cameras

Several preprocesses for static cameras are needed in order to localize and track the robot using the cameras. The exposure time is generally adjusted because the exposure time is related to the success rate of detecting robots. However, when the exposure time is adjusted, the flicker effect should be avoided. The flicker effect refers to a phenomenon that occurs when the camera's image acquisition cycle and the fluorescent lamp emitting cycle do not match each other. If the flicker effect occurrs, the captured images by the camera have clear difference in brightness. Therefore, the exposure time

should be adjusted in order to match a divisor of 120 because the normal fluorescent lamp emitting cycle in our environment is 120Hz.

The adopted camera as the static camera has the Bayer pattern filter. The Bayer-pattern color filter array is designed to generate color images from a camera that has a monochrome image sensor. Thus, the generated color images require one-third lower bandwidth than the color image from the RGB image sensor. This lower bandwidth is proper for the system simultaneously using multiple cameras. Figure 4.10(a) shows the structure between the Bayer pattern color filter array and image sensor.

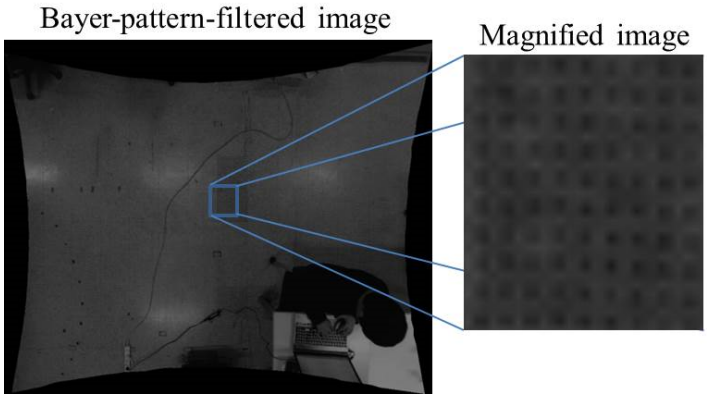


(a) (b)

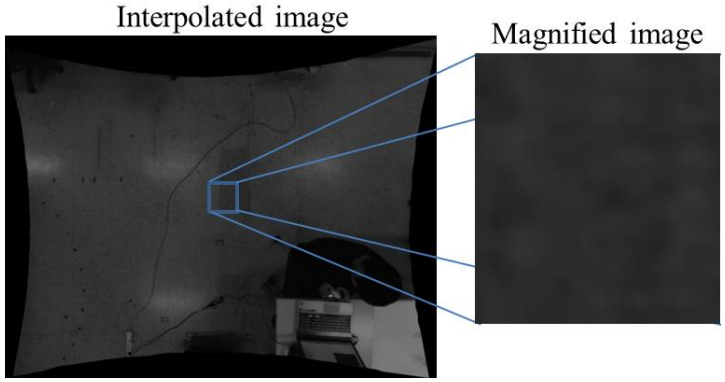
Figure 4.10 (a) Structure between the Bayer pattern and the image sensor [50], and (b) example of the BGGR pattern

There are several patterns of Bayer pattern. Among those patterns, our static camera uses the Blue-Green-Green-Red (BGGR) pattern. Figure 4.10(b) represents the example of the BGGR pattern.

However, the raw image from the camera requires an interpolation to generate a normal image. Because the raw image has dots in the image caused by the filter array, the filtered image is difficult to use for detection or tracking. In order to generate a color image from a Bayer-pattern-filtered



(a)



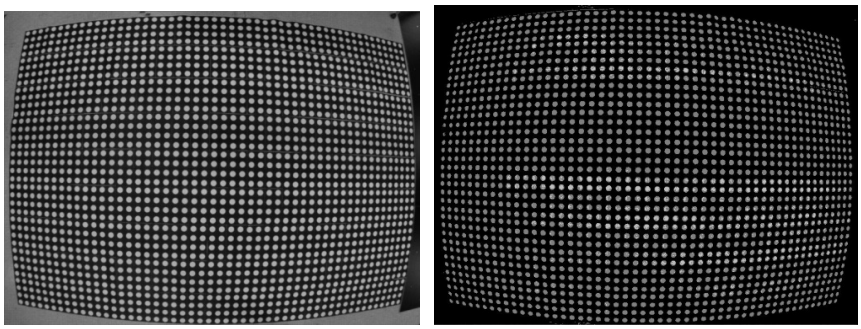
(b)

Figure 4.11 (a) Bayer-pattern-filtered image, and (b) Interpolated Image

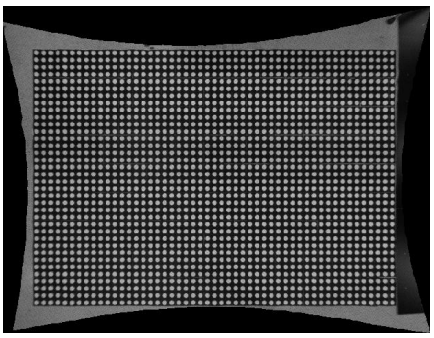
image, we used the bilinear interpolation method [51] that generates a color image. After interpolation, we convert the color image to the gray image because the proposed static camera system uses only geometrical features of markers. Figure 4.11 shows the Bayer-pattern-filtered image and the interpolated image. The interpolated image has no dots caused by the color filter array.

The static camera should be calibrated to undistort the image from the camera and obtain mapping information between world coordinates and pixel coordinates of the camera. Camera calibration is categorized in two main: color calibration and geometric calibration. In this dissertation, we only use the geometric feature of the camera. Thus, the geometric camera calibration is required. The geometrical parameters of the camera are divided into intrinsic and extrinsic parameters. The intrinsic parameters are inherent information of the camera's body and lens. The extrinsic parameters refer to a position and a viewing direction (orientation) of the camera. The geometric intrinsic parameters can be calculated by using multiple images containing an object with a pattern called the calibration pattern. In general, a geometrical camera calibration includes three steps. First, a few images including the calibration object is acquired. Second, a calibration pattern in the calibration object is detected. Finally, the transformation between the pixel and world coordinates (or the pixel and camera coordinates) is estimated by using the relationship between the detected patterns.

In the case of the static camera, it is easy to figure out the mapping relationship between the world coordinates and the pixel coordinates, because the position and orientation of the camera is fixed. Figure 4.12 shows the rectangular calibration object for the static camera calibration. For calibration of the static camera, one image is only used. The pattern on the calibration object consists of white circles and a black background. The white circles are



(a) (b)



(c)

Figure 4.12 (a) Calibration pattern for a static camera, (b) Binary image of (a), and (c) Undistorted image of (a) after the calibration process

placed 48 per column and 32 per row. The space between the circles is 7.5cm per column and row. The binarization of the image with the calibration object is performed to detect the pattern. Then, the mapping matrix can be calculated by using the image from the binarization and calibration method in [52]. Figure 4.12(b) shows the image that resulted from the binarization, and Figure 4.12(c) shows the image undistorted using the parameter obtained by the calibration.

4.3.2 Marker-based Localization and Tracking of Multiple Robots

As mentioned in Chapter 3, the geometrical marker is attached to the robot. Detection using the shape or feature extracted from the robot image [34] is more desirable. However, the detection has a performance limitation by environmental changes and the robot's appearance change. The strategy using the marker provided a more robust detection method against these changes.

We build the model for robot detection using the extracted edges from the marker. Figure 4.13 shows an example of a marker and an extracted edge from the marker. The edge of the model is extracted by the canny edge detector [53]. The score function evaluates the candidates to match using the edge-based model. If the score of the matched edge is bigger than threshold α, we consider that the marker is detected. The score function S [54] is thus

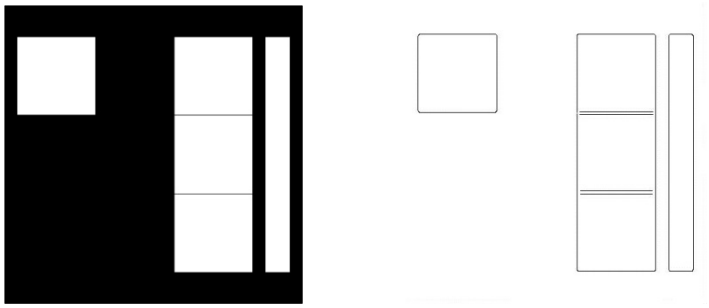
$$S = C(1 - w_{fe} \cdot Err_{norm}), \tag{4.15}$$

where C is the model coverage of the marker, Err_{norm} is the normalized fitting error of the marker, and w_{fe} is the weight of the fitting error. The model coverage explains the occlusion area of the detected marker using edge contour lengths. The model coverage function is defined as

$$C = \frac{LC_{found}}{LC_{total}}, \tag{4.16}$$

where LC_{found} and LC_{total} indicate the edge contour lengths from the matched edge and the marker model respectively. The larger occlusion induces the smaller model coverage.

The fitting error function is defined as



(a) (b)

Figure 4.13 (a) the marker for detecting the mobile robot, and (b) the edge

$$Err = \frac{\sum_{k}^{N_{edl}} \left({e_{k,x}}^2 + {e_{k,y}}^2 \right)}{N_{edl}}, \tag{4.17}$$

where N_{edl} is the total number of edgels (edge pixels) in the marker model, and $e_{k,x}$ and $e_{k,y}$ are the orthogonal distance between the k^{th} edgel and the matched edge in x-direction and y-direction respectively. Using the score function, we find the matched edge that has the lowest occlusion and fitting error.

Using the geometric camera calibration method [52], we generate the matrix ${}^{W}_{P}M_{sc,i}$ which explains the transformation between the world coordinates $\{W\}$ and pixel coordinates $\{P_{sc,i}\}$ of i^{th} static camera. However, the transformation matrix, ${}^{W}_{P}M_{sc,i}$ does not consider the height of the robot. The inconsideration of the robot's height causes an error of the robot's position in the world coordinates. Thus, we design an additional correction method using a geometrical relationship between a static camera and the robot. Figure 4.14 shows the relationship between the robot's height and the position error.

Let $d_{e,i}$ be the distance error from the i^{th} dynamic camera caused by the robot's height h. We can express the relationship between $d_{e,i}$ and h as the below proportional expression:

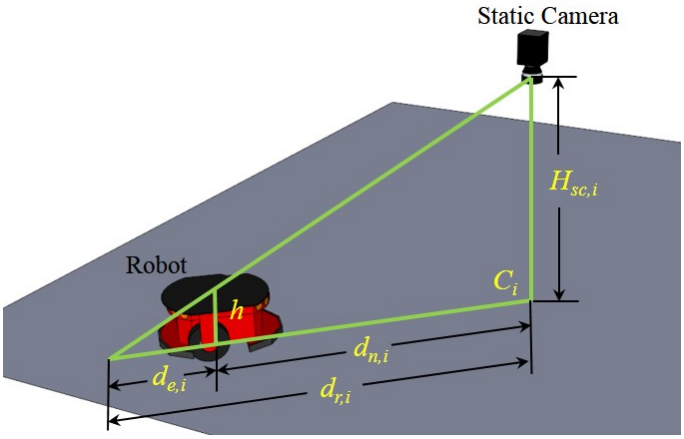


Figure 4.14 Relationship between the robot's height and calculated position

$$d_{e,i} + d_{n,i} : H_{sc,i} = d_{e,i} : h, \tag{4.18}$$

where $d_{n,i}$ is the distance from the point C_i projected on the x-y plane of the static camera position ${}^{W}p_{sc,i}$ to the actual position of the robot. Equation 4.18 is reformulated to

$$d_{e,i} = \frac{d_{n,i}h}{H_{sc,i} - h} = kd_{n,i}, \tag{4.19}$$

where $H_{sc,i}$ and h are not zero. As shown in Equation 4.19, the error is proportional to the distance $d_{n,i}$.

The robot's position ${}^{P}r_{sc,i}$ in pixel coordinates of i^{th} static camera is mapped to ${}^{W}r_{sc,i}$ in the world coordinates using the following equation:

$$ {}^{W}r_{sc,i} = {}^{W}_{P}M_{sc,i}\ {}^{P}r_{sc,i}\ . \tag{4.20} $$

Let ${}^{W}r_{sc,i}=[{}^{W}x_{r,sc,i,}\ {}^{W}y_{r,sc,i,}\ h,\ 1]^{\mathrm{T}}$, then we transform $[{}^{W}x_{r,sc,i,}\ {}^{W}y_{r,sc,i}]$ in Cartesian coordinates to the polar coordinates centered on the projected static camera center as

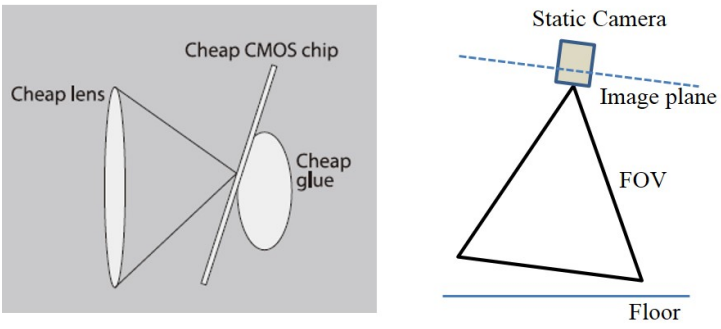
$$ \begin{cases} d_{r,i} = \sqrt{({}^{W}x_{r,sc,i} - c_{x,sc,i})^2 + ({}^{W}y_{r,sc,i} - c_{y,sc,i})^2} \\ \phi_{r,i} = \tan^{-1}\left(\dfrac{{}^{W}y_{r,sc,i} - c_{y,sc,i}}{{}^{W}x_{r,sc,i} - c_{x,sc,i}}\right), \end{cases} \tag{4.21} $$

where $d_{r,i}$ and $\phi_{r,i}$ is the distance and angle of the robot's position in the polar coordinates. Using Equation 4.21, we calculate the corrected distance $d_{n,i}$. Then, we transform the corrected robot position $r_{pol,i}=[d_{n,i},\ \phi_{r,i}]$ in the polar coordinates to Cartesian coordinates as the following in Equation 4.22. Finally, we obtain the corrected robot's position ${}^{W}r_{rn,i} = [{}^{W}x_{rn,sc,i},\ {}^{W}y_{rn,sc,i},\ h,\ 1]^{\mathrm{T}}$ in Cartesian coordinates using

$$ d_{n,i} = d_{r,i} - d_{e,i}\ , \tag{4.22} $$

$$ \begin{cases} {}^{W}x_{rn,sc,i} = d_{n,i}\cos\phi_{r,i} + c_{x,sc,i} \\ {}^{W}y_{rn,sc,i} = d_{n,i}\sin\phi_{r,i} + c_{y,sc,i} \end{cases}. \tag{4.23} $$

After the aforementioned correction method is performed, the error is still



(a) (b)

Figure 4.15 Comparison between (a) the tangential distortion [55] and (b) the non-parallel between the camera image plane and the floor of the iSpace.

remained. The error can be reduced by utilizing the geometrical camera error model. According to Equation 4.19, the error increases as the robot moves away from the center of the camera. Even the error is corrected through Equations 4.18~4.23, the remaining error also tends to increase as the distance between the projected center of the camera and the robot is extended. This is very similar to the radial distortion error of the camera geometric error. The radial distortion error occurs by the spherical aberration of the camera lens. In addition, the aforementioned correction method is based on the assumption that the image plane of the camera is parallel to the floor of the iSpace. However, since it is difficult in practice to make them completely parallel, a localization error still occurs. The trend of this error is similar to the tangential distortion error of the camera geometric error. The tangential distortion error occurs because the image chip and the lens of the camera are not parallel.

Figure 4.15 shows the similarity between the cause of the tangential distortion and the non-parallel between the camera image plane and the floor of the iSpace.

Let $X_{rc,\mathrm{i}}^{rd}$ be the *x-y* position of the robot corrected by using the radial distortion model. If the result of Equation 4.22 and 4.23 is substituted into the radial distortion model, the correction model is as follows:

$$X_{rc,\mathrm{i}}^{rd} = \left[x_{rc,i}^{rd}, y_{rc,i}^{rd} \right],$$

$$\begin{cases} x_{rc,i}^{rd} = {}^{W}x_{rn,sc,i}(1 + k_1 {d_{n,i}}^2 + k_2 {d_{n,i}}^4 + k_3 {d_{n,i}}^6) \\ y_{rc,i}^{rd} = {}^{W}y_{rn,sc,i}(1 + k_1 {d_{n,i}}^2 + k_2 {d_{n,i}}^4 + k_3 {d_{n,i}}^6) \end{cases}, \tag{4.24}$$

where k_1, k_2, and k_3 are the error coefficients for this model. Let $X_{rc,\mathrm{i}}^{td}$ be the *x-y* position of the robot corrected by using the tangential distortion model. If the results of Equations 4.22 and 4.23 are substituted into the tangential distortion model, the correction model based on the tangential distortion model is as follows:

$$X_{rc,i}^{td} = \left[x_{rc,i}^{td}, y_{rc,i}^{td} \right],$$

$$\begin{aligned} x_{rc,i}^{td} &= [2p_1 {}^{W}y_{rn,sc,i} + p_2({d_{n,i}}^2 + 2 {}^{W}x_{rn,sc,i}{}^2)] \\ y_{rc,i}^{td} &= [p_1({d_{n,i}}^2 + 2 {}^{W}y_{rn,sc,i}{}^2) + 2p_2 {}^{W}x_{rn,sc,i}] \end{aligned}, \tag{4.25}$$

where p_1 and p_2 are the error coefficients for this model. If positions measured at multiple points in the iSpace are substituted into Equation 4.26 that were combined with Equation 4.24 and 4.25, Equation 4.26 becomes the linear equation such as Equation 4.27. Let $X_{rc,i}$ be the *x*-*y* position of the robot corrected by using the tangential distortion model.

$$X_{rc,i} = [x_{rc,i}, y_{rc,i}],$$
$$X_{rc,i} = X_{rc,\mathrm{i}}^{rd} + X_{rc,\mathrm{i}}^{td}. \tag{4.26}$$

$$b = Ak, \tag{4.27}$$

where

$$A = \begin{bmatrix} {}^{W}x^{1}_{rm,sc,i}(d^{1}_{n,i})^2 & {}^{W}x^{1}_{rm,sc,i}(d^{1}_{n,i})^4 & {}^{W}x^{1}_{rm,sc,i}(d^{1}_{n,i})^6 & 2\,{}^{W}y^{1}_{rn,sc,i} & (d^{1}_{n,i})^2 + 2({}^{W}x^{1}_{rn,sc,i})^2 \\ {}^{W}y^{1}_{rm,sc,i}(d^{1}_{n,i})^2 & {}^{W}y^{1}_{rm,sc,i}(d^{1}_{n,i})^4 & {}^{W}y^{1}_{rm,sc,i}(d^{1}_{n,i})^6 & (d^{1}_{n,i})^2 + 2({}^{W}y^{1}_{rn,sc,i})^2 & 2\,{}^{W}x^{1}_{rm,sc,i} \\ {}^{W}x^{2}_{rm,sc,i}(d^{2}_{n,i})^2 & {}^{W}x^{2}_{rm,sc,i}(d^{2}_{n,i})^4 & {}^{W}x^{2}_{rm,sc,i}(d^{2}_{n,i})^6 & 2\,{}^{W}y^{2}_{rn,sc,i} & d^{2\,2}_{n,i} + 2({}^{W}x^{2}_{rn,sc,i})^2 \\ {}^{W}y^{2}_{rm,sc,i}(d^{2}_{n,i})^2 & {}^{W}y^{2}_{rm,sc,i}(d^{2}_{n,i})^4 & {}^{W}y^{2}_{rm,sc,i}(d^{2}_{n,i})^6 & d^{2\,2}_{n,i} + 2({}^{W}y^{2}_{rn,sc,i})^2 & 2\,{}^{W}x^{2}_{rm,sc,i} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^{W}x^{M}_{rm,sc,i}(d^{M}_{n,i})^2 & {}^{W}x^{M}_{rm,sc,i}(d^{M}_{n,i})^4 & {}^{W}x^{M}_{rm,sc,i}(d^{M}_{n,i})^6 & 2\,{}^{W}y^{M}_{rn,sc,i} & (d^{M}_{n,i})^2 + 2({}^{W}x^{M}_{rn,sc,i})^2 \\ {}^{W}y^{M}_{rm,sc,i}(d^{M}_{n,i})^2 & {}^{W}y^{M}_{rm,sc,i}(d^{M}_{n,i})^4 & {}^{W}y^{M}_{rm,sc,i}(d^{M}_{n,i})^6 & d^{M\,2}_{n,i} + 2({}^{W}y^{M}_{rn,sc,i})^2 & 2({}^{W}x^{M}_{rm,sc,i}) \end{bmatrix}$$

$$b = \begin{bmatrix} {}^{W}x^{1}_{rn,sc,i}\, x^{1}_{rc,i} \\ {}^{W}y^{1}_{rn,sc,i}\, y^{1}_{rc,i} \\ \vdots \\ {}^{W}x^{M}_{rn,sc,i}\, x^{M}_{rc,i} \\ {}^{W}y^{M}_{rn,sc,i}\, x^{M}_{rc,i} \end{bmatrix}, \qquad \boldsymbol{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ p_1 \\ p_2 \end{bmatrix}$$

Here, the superscription M means the index number of a position input in ${}^{W}x^{M}_{rm,sc,i}$, ${}^{W}y^{M}_{rn,sc,i}$, and $d^{M}_{n,i}$. If the least square method is applied to Equation 4.12, as follows:

$$\boldsymbol{k} = (A^T A)^{-1} A^T b \tag{4.28}$$

The value of $X_{rc,i}$ can be calculated by using the obtained $\boldsymbol{k}$ and Equations 4.24 ~ 4.26.

Our strategy to track the robot using static cameras is the tracking-by-detection using the sliding window search method [56]. In other words, the tracking method is continuously searching and detecting the robot in the image stream captured by the static cameras. The global detection for the robot tracking in image and scale spaces requires high computational time. The computational time increases in proportion to the extent of the scale space and the image space. However, the proposed iSpace has very narrow scale space to search the robot because of the assumptions that the terrain of the iSpace is planar, and the image plane of the static camera is parallel to the planar terrain. Therefore, the iSpace avoids the computational load caused by search in the scale space.

Our iSpace has four static cameras, which means that we have four times larger area to search in image space than the case of a single static camera. In order to decrease the computational time for tracking, we set the local window

that consists of the robot position ${}^{P}r_{sc,i}$ and the width wnd_{wd} and the height wnd_{ht} of the window.

4.3.3 Proposed Reprojection-based Handoff among Static Cameras

The handoff for the multiple static camera tracking is divided into three steps. First, we need to know whether the robot is in the overlap region between static cameras. When the robot is in the overlap region, the handoff is triggered. The handoff trigger and execution is usually based on the distance from the robot to the cameras [29]. In this step, we use the transformation between the world coordinates and pixel coordinates of a static camera. Second, we need to decide which camera is adequate. The adequate camera selection is based on the score of a detected marker. Finally, the handoff is executed when the robot is only detected by the selected camera.

[29] uses the distance from cameras to the robot for the handoff trigger and execution. However, the distance measure is hard to apply when robots have various heights because the overlap region between cameras usually varies according to the height of the robot. [57] uses the frame difference method for handoff trigger. In [57], all of the cameras perform the frame difference continuously. However, our handoff method is only triggered when the robot is in any of the overlap regions.

When the robot is inside the overlap region between the i^{th} and j^{th} static

cameras, the location of the robot would be appeared in both images from the cameras. Based on this concurrent appearance, the handoff is triggered. The robot position ${}^{W}r_{sc,i}$ needs to be remapped on the image space of each camera as follows:

$$ {}^{P}r_{sc,i} = {}^{P}_{W}\boldsymbol{M}_{sc,i}\,{}^{W}r_{sc,i} \tag{4.29} $$

$$ {}^{P}r_{sc,i,j} = {}^{P}_{W}\boldsymbol{M}_{sc,j}\,{}^{W}r_{sc,i}, \quad i \neq j, \tag{4.30} $$

where ${}^{P}_{W}\boldsymbol{M}_{sc,i}$ is the transformation matrix from the world coordinates to the pixel coordinates of the i^{th} static camera SC_i. ${}^{P}r_{sc,i,j}$ is the transformed pixel position of the robot in the pixel coordinates of the SC_j. We define the valid region VR_i of SC_i as

$$ VR_i = \{(u_i, v_i) \in P_i \mid 0 \le u_i < wd_i, 0 \le v_i < ht_i\}, \tag{4.31} $$

where wd_i and ht_i is the width and height of an image captured by the SC_i. The trigger criterion for handoff between static cameras is defined as

$$ TC_{sc,i} = \left[\left({}^{P}r_{sc,i} \in VR_i\right) \wedge \left({}^{P}r_{sc,i,j} \in VR_j\right)\right]. \tag{4.32} $$

If one of the remapped positions is valid for the i^{th} camera, the camera is in the image of the i^{th} camera. Therefore, if two or more of the remapped

positions are valid, we know that the robot is under the handoff situation. The cameras with the valid positions are chosen as the candidate cameras for handoff. Figure 4.16 illustrates the position of the robot in the pixel coordinates when the robot is observed in the overlapped FOV between the static cameras. The robot exists on the valid regions of $\{P_1\}$ and $\{P_2\}$.

The detection score of the robot is a useful criterion of selection for the adequate camera to track because the score is decreased as the robot is close to the border of the image. The distortion of the camera lens is getting worse from the center of the image to the border of the image. Although we undistorted the image using the camera calibration, there still remains a little distortion. Thus, when the robot's image is close to the border of image, the detection score decreases. In addition, when the robot moves from the i^{th} camera to the j^{th} camera, a part of the robot's image could only be captured through the i^{th} camera. This case causes the score to decrease. On the contrary, the robot detection score increases in the image captured by the j^{th} camera. Let $S_{sc,j}$ be the detected score by the candidate camera SC_j. then the adequate camera is chosen by

$$j^* = \arg\max(S_{sc,j}), \quad S_{sc,j} \geq kS_{sc,i} \tag{4.33}$$

where k is a constant.

When the system detects the robot in the overlap region, the system

generates the local tracking windows centered on the projected position of the robot in the images. Then, when the robot is out of the overlap region, the valid region is the only one left. Then the handoff is executed. The execution criterion EC_{sc,j^*} for the handoff is define as

$$EC_{sc,j^*} = \left[\left({}^{P}r_{sc,i} \notin VR_i \right) \wedge \left({}^{P}r_{sc,i,j^*} \in VR_{j^*} \right) \right]. \tag{4.34}$$

We explained the proposed re-projection based handoff method. Finally,

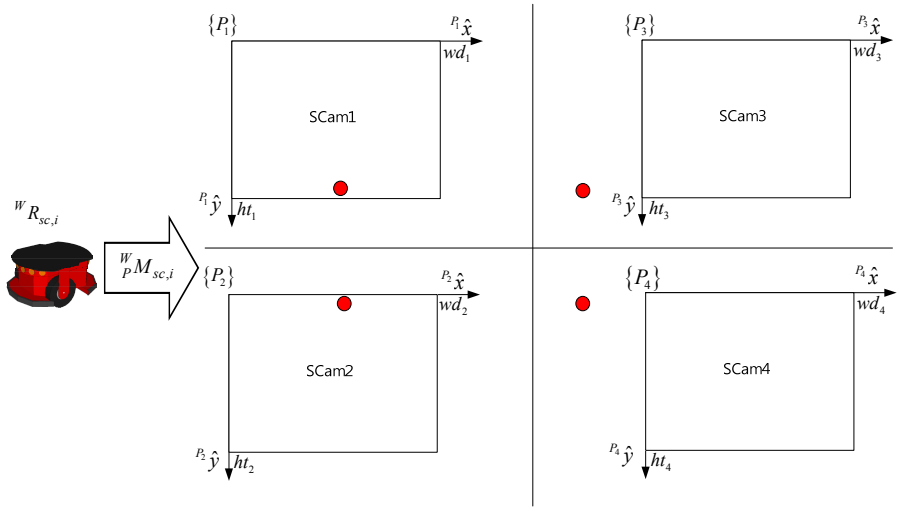


Figure 4.16 An example of a handoff situation between the static cameras. The robot exists between SC_1 and SC_2. $\{P_i\}$ is the pixel coordinates system of the i^{th} static camera. The red circles represent the robot position in each pixel coordinates.

Table 4.3 summarizes the differences between the proposed method and the conventional method [29].

Table 4.3 Comparison between the proposed handoff method and the conventional method for the multiple static cameras

	Applicable to the static cameras	Required memory	Dependency of the robot's height
Proposed Method	Yes	Fixed	No
Conventional Method	Yes	Depends on the type of robots	Yes

Chapter 5

Operation with Heterogeneous Sensor Groups

5.1 Interface Protocol among Sensor Groups

In computing, a process is an instance of a computer program that is being executed. The program contains the program code and its current activity. The proposed interworking static camera system (ISCS) and interworking dynamic camera system (IDCS) are executed as independent processes. Because a process is not able to access memory of other processes, a particular method is needed for the exchange of data between processes, which is called inter-process communication (IPC). Thus, an IPC method is required in order to communicate between the ISCS and IDCS.

IPC methods include file mapping, mail slot, anonymous pipe, and named

pipe. File mapping enables a process to treat the contents of a file as if they were a memory allocated in the process. However, file mapping is very slow and unstable to concurrent access by multiple processes. The mail slot is a type of shared memory among processes. However, this method only provides one-way communication. In contrast, the anonymous pipe and named pipe provide two-way communication. The proposed iSpace employs the named pipe as the IPC because the anonymous pipe provides data exchange between a child process and parent process. Figure 5.1 shows the communication between ISCS and IDCS.

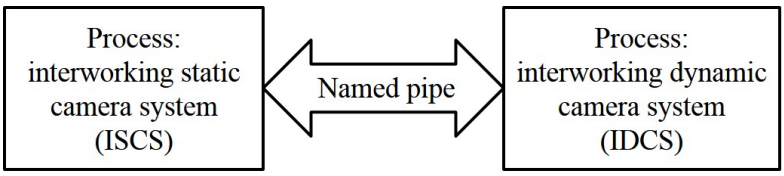


Figure 5.1 Diagram of communication between ISCS and IDCS.

Cooperation between the proposed iSpace and a robot can be classified by determining whether communication is possible between them. Generally, if communication is possible between a robot and an iSpace, then it is more efficient or beneficial than the case where communication between them is not possible. The methods that can be used for communications between robots and the iSpace include characters, sounds (voice), and data transmissions. Characters and voices are natural communication methods that

are used by humans. However, robots require additional abilities in order to process voices or characters. Therefore, data communications over computer networks are the most efficient method for communications between the iSpace and robots. As a result, this type of communication was employed in the present study. Communications between the iSpace and robots are bidirectional. The iSpace can send information to the robot and the robot can send information to the iSpace.

As shown in Figure 5.2, a network topology can be classified as a ring, mesh, star, fully connected, line, tree, or bus topology. In the present study, the star and fully connected network topologies are appropriate in terms of connectivity, because it is assumed that a single iSpace is present in a given space. Mesh networks have an advantage in that they enable direct communications with all entities. However, it is difficult to configure a mesh

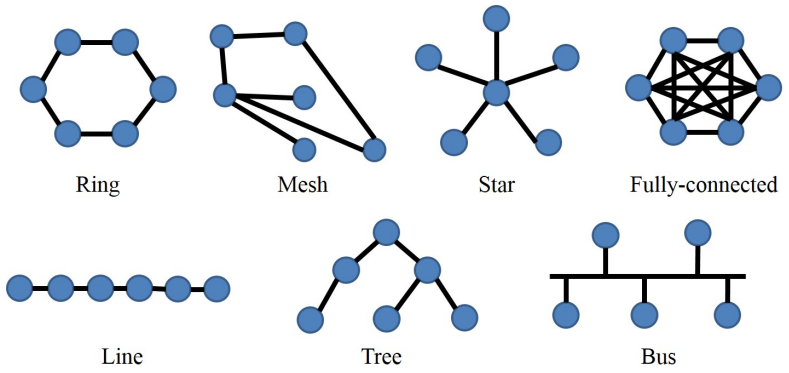


Figure 5.2 Diagrams of network topologies

network because it is necessary to maintain connectivity with all of the entities in the network. A star network was selected as the network topology for the proposed iSpace, because communications do not occur frequently between robots. Figure 5.3 shows an example of the configured topology.

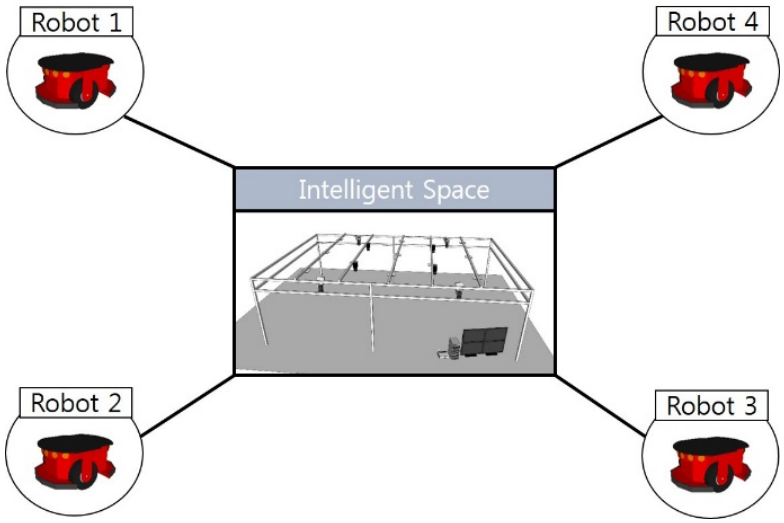


Figure 5.3 The configured topology between the iSpace and robots

The general models for network architectures include a server-client model, host-terminal model, and peer-to-peer model. The relationship between an iSpace and a robot follows the hierarchical control method [11] that allows the iSpace to send information and instructions to the robot and also allows the robot to process its own data. Therefore, the server-client model is appropriate for the network connection structure in the iSpace. In this dissertation, it is assumed that the iSpace acts as a server and the robot

acts as a client.

A communication protocol is a digital protocol system for mutual exchanges of data between entities that are connected to the same network. In particular, when data is exchanged through a computer network, the aforementioned protocol can also be called a network protocol. Today, most communication between computers is based on the internet protocols. In general, the Transmission Control Protocol (TCP or TCP/IP) and User Datagram Protocol (UDP) can be used as the internet protocol. The TCP/IP provides bidirectional communications and error correction. As a result, TCP/IP was adopted as the network protocol for the proposed iSpace.

The internet protocol (IP) address of an iSpace and a robot can be determined via automatic settings or via random allocation. However, if the complexity of a system is high, it becomes difficult to identify the addresses. Therefore, the network IPs for an iSpace and a robot are configured as shown in Table 5.1. The IP addresses consist of four numbers separated by a dot. Because the IP address for the iSpace is the internal address and is not the actual internet address, the iSpace and the robots maintain the first two numbers of the IP address (e.g., 192 and 168). The third number of the IP address varies based on the type of device that is connected to the network. The iSpace always has a value of 0 for the third number and robots of the same type have the same value for the third number. The fourth number of the IP address is also used for identifying robots of the same type.

Table 5.1 Example of the allocated IP addresses for the iSpace and robots

Type of device		IP	Port	Description
iSpace		192.168.0.3	8080	Third number of IP must be 0
A-type robot	#1	192.168.1.1		Third number of IP: Type of robot, Fourth number of IP: Index number of robot
	#2	192.168.1.2		
	#3	192.168.1.3		
	#4	192.168.1.4		
B-type robot	#1	192.168.2.1		
	…			

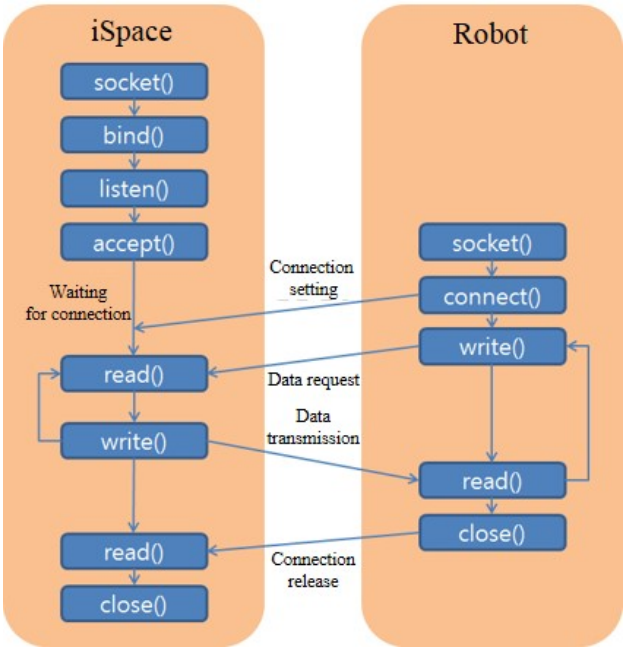


Figure 5.4 Basic flow for socket communication between the iSpace and the robot.

Figure 5.4 shows the basic flow for socket communications over the TCP. To begin with, if the iSpace opens a socket and begins to wait for a connection request, the robot also opens a socket and determines the connection settings. Once the connection settings have been transferred to the iSpace, the connection is established. Once the connection is established, data is transmitted by the request of the robot. If further operations are not required, the connection between the iSpace and the robot is released.

The sub-systems of the iSpace exchange a message over the IPC or TCP in order to operate interactively. The message is basically composed of three terms: message index (MIDX), device index (DIDX), and command (CMD). Table 5.2 summarizes the components of the message.

Table 5.2 Basic components of a message that is exchanged between sub-systems

Name	Description	Value
MIDX	Cumulative count of communication	001, 002, ...
DIDX	Index number of reception target system or robot	100: static camera system, 200: dynamic camera system, 300: robot, 301, 302, ... : Robot index
CMD	Content requested to the reception target device	

When an iSpace sends a request to a robot for a data transmission based

on the message above, the robot responds as illustrated in Table 5.3. The robot's response includes two steps. The first step is to acknowledge that the command has been received and the second step is to determine whether the commands have been read and executed. This can be determined based on the value of an RTN_CODE. Based on the results for these two responses, the server can determine whether the client is receiving and performing the commands. When the commands are performed correctly, the iSpace can obtain the desired value via parameters that are included in the second response.

Table 5.3 Example of communications between the iSpace and a robot: (a) MOVE_VEL command and (b) GET_STARGAZER command

(a)

iSpace> MIDX=001,DIDX=301,CMD=MOVE_VEL,VEL=100,ACC=0.2
robot1> MIDX =001,DIDX=301,STEP=1,RTN_CODE=0001
robot1> MIDX =001,DIDX=301,STEP=2,RTN_CODE=0001

(b)

iSpace> MIDX =001,DIDX=301,CMD=GET_STARGAZER
robot1> MIDX =001,DIDX=301,STEP=1,RTN_CODE=0001
robot1> MIDX =001,DIDX=301,STEP=2,RTN_CODE=0001,X=532.0,Y=-313.1,THETA=59.3

Tables 5.4 – 5.11 describe the commands or responses that are transmitted between the ISCS and IDCS or between the iSpace and the robot.

Table 5.4 Commands transmitted from the ISCS to IDCS

Name	Value	Description
CMD	GET_PAN_ANG	Obtaining a pan angle of a dynamic camera
	GET_TILT_ANG	Obtaining a tilt angle of a dynamic camera
	SET_PAN_ANG	Specifying a pan angle of a dynamic camera
	SET_TILT_ANG	Specifying a tilt angle of a dynamic camera
	GET_PAN_VEL	Obtaining a panning speed of a dynamic camera
	GET_TILT_VEL	Obtaining a tilting speed of a dynamic camera
	SET_PAN_VEL	Specifying a panning speed of a dynamic camera
	SET_TILT_VEL	Specifying a tilting speed of a dynamic camera
	GET_ROBOT_POS	Obtaining the position of the robot tracked by the IDCS
	GET_TRACK_CAM	Obtaining the index number of the tracking camera
	SET_GAZE	Getting a dynamic camera gaze at a position in iSpace
	STOP	Stopping a dynamic camera
	SET_AUTO	Commanding the IDCS to track the robot automatically

Table 5.5 Detailed parameters for commands transmitted from the ISCS to IDCS

CMD	Parameters	Description	Return value
GET_PAN_ANG	CIDX	Target camera	THETA: -170° ~ 170°
GET_TILT_ANG	CIDX	Target camera	THETA: -90° ~ 90°
SET_PAN_ANG	CIDX	Target camera	
	THETA	Pan angle	
SET_TILT_ANG	CIDX	Target camera	
	THETA	Tilt angle	
GET_PAN_VEL	CIDX	Target camera	VEL:0~1.0
GET_TILT_VEL	CIDX	Target camera	VEL:0 ~ 1.0
SET_PAN_VEL	CIDX	Target camera	
	VEL	Panning speed	
SET_TILT_VEL	CIDX	Target camera	
	VEL	Tilting speed	
GET_ROBOT_POS	-		X, Y
GET_TRACK_CAM	-		CIDX
SET_GAZE	X	*x*-position in iSpace	
	Y	*y*-position in iSpace	
STOP	-		
SET_AUTO	STATE	True / false	

Table 5.6 Commands transmitted from the IDCS to ISCS

Name	Value	Description
CMD	GET_TRACK_CAM	Obtaining the index number of the tracking camera
	GET_ROBOT_POS	Obtaining the position of the robot monitored by the ISCS
	GET_ILLUM_LEVEL	Obtaining the degree of illumination in iSpace

Table 5.7 Detailed parameters for commands transmitted from the IDCS to ISCS

CMD	Parameters	Description	Return value
GET_TRACK_CAM	-	-	X, Y
GET_ROBOT_POS	-	-	CIDX
GET_ILLUM_LEVEL	-	-	LEVEL:0~1.0

Table 5.8 Commands transmitted from the iSpace to robots

Name	Value	Description
CMD	MODE_AUTO	Setting to automatic mode
	MODE_MANUAL	Setting to manual mode
	MOVE_POS	Moving to a specific location
	SET_THETA	Specifying theta value of the robot
	MOVE_VEL	Moving at the specified linear velocity
	MOVE_VEL_ANGULAR	Moving at the specified angular velocity
	STOP	Stopping the robot
	GET_SONAR	Obtaining ultrasonic sensor values
	GET_LASER	Obtaining laser scanning values
	GET_STARGAZER	Obtaining Stargazer value
	GET_ENCODER	Obtaining encoder values
	GET_EKF	Obtaining values estimated by EKF
	SET_WANDERING	Random movement while avoiding collisions
	SET_IDLE	Setting to an idle state

Table 5.9 Detailed parameters for commands transmitted from the iSpace to robots

CMD	Parameters	Description	Return value
MODE_AUTO	-		
MODE_MANUAL	-		
MOVE_POS	X	*x*-position in iSpace	
	Y	*y*-position in iSpace	
	THETA	Robot angle	
SET_THETA	THETA	Robot angle	
MOVE_VEL	VEL	Linear velocity	
	ACC	Acceleration	
MOVE_VEL_ANG	RAD	Radius	
	W	Angular velocity	
	ACC	Acceleration	
STOP	-		
GET_SONAR	SID	Sensor index	VALUE:0~1.0
GET_LASER	-		VALUE:0~4.0
GET_STARGAZER	-		X, Y, THETA
GET_ENCODER	-		R_W, L_W
GET_EKF	-		X, Y, THETA
SET_WANDERING	SPEED	Wandering speed	
SET_IDLE	-		

Table 5.10 Commands transmitted from a robot to the iSpace

Name	Value	Description
CMD	GET_POS	Obtaining the observed position to the robot
	GET_POS_ROBOT	Obtaining the observed positions of other robot to the robot

Table 5.11 Detailed parameters for commands
transmitted from a robot to the iSpace

CMD	Parameters	Description	Return value
GET_POS	-		X, Y, THETA
GET_POS_ROBOT	ROBOT	Index number of other robot	X, Y, THETA

5.2 Sensor Selection for an Operation Using Heterogeneous Sensors

Sensors in the proposed iSpace have different characteristics and accuracy. Thus, if each sensor is used according to its characteristic, performance will improve in the iSpace. For this purpose, we set a priority for each sensor as an accuracy of the sensors. Table 5.12 summarizes the sensor priorities.

Table 5.12 Priorities of sensors

Priority	Device
1	static camera system
2	robot system (IR sensor)
3	dynamic camera system
4	robot system (wheel encoder)

As summarized in Table 5.12, the static camera system has the first priority because of its high degree of accuracy. The IR sensor mounted on the mobile robot has the second priority, the dynamic camera system has the third priority, and the onboard wheel encoder of the mobile robot has the fourth

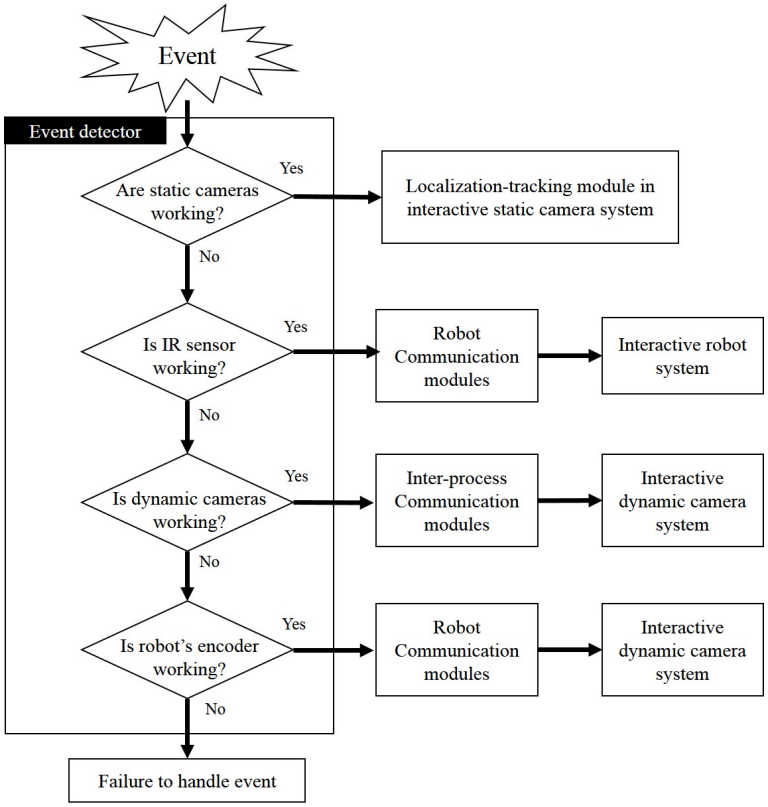


Figure 5.5 Sensor selection process in the event detector

priority. When a sensor with a higher priority temporarily malfunctions, the iSpace tries to utilize the next higher priority sensor. This sensor selection process is shown in Figure 5.5. The sensor selection is decided in the module, event detector. The event detector confirms the function of sensors according to the sensor priorities. Then, if the occurred situation (event) satisfies one of the sensor selection criteria, the event detector selects the sensor related with the satisfied criterion and activates the modules related with the selected sensor. The IR sensor's selection criterion is defined as

$$AC_{IR} = \left[\left(S_{sc} < T_{sc} \right) \wedge \left(L_{avg} < T_L \right) \wedge IR\_Checker(^W r_{sc}) \right], \qquad (5.1)$$

where S_{sc} is the detection score attached on the robot body, L_{avg} is the average of pixel values in the image from the tracking static camera, *IR_Checker* checks the function of IR sensor at $^W r_{sc}$, T_{sc} is a threshold value of detection score, and T_L is a threshold value to check the degree of illumination.

The selection criterion of the dynamic camera system is defined as

$$AC_{dc} = \left[\left(S_{sc} < T_{sc} \right) \wedge \left(S_{dc} \geq T_{dc} \right) \wedge \neg IR\_Checker(^W r_{sc}) \right], \qquad (5.2)$$

where T_{dc} is a threshold value of robot detection in the dynamic camera system, and S_{dc} is defined as

$$S_{dc} = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_{n=1}^{N} \sqrt{H_1(n) \cdot H_2(\mathrm{n})}}\ , \qquad (5.3)$$

where H_1 and H_2 are color histograms of the model and input, and N is the length of the color histogram.

The selection criterion of the robot's onboard wheel encoder is defined as

$$AC_{odo} = \left[\left(S_{sc} < T_{sc} \right) \wedge \left(S_{dc} < T_{dc} \right) \wedge \neg IR\_Checker(^W r_{sc}) \right]. \qquad (5.3)$$

5.3 Proposed Operation with Static Cameras and Dynamic cameras

For operation between the interworking static camera system (ISCS) and the interworking dynamic camera system (IDCS), the ISCS and IDCS should be aware of a situation in the iSpace and be able to communicate with each other. In order to conduct this operation, we propose a software structure that consists of the operation decision module and inter-processing communication (IPC) module for the ISCS. The operation decision module detects an event in the iSpace and generates an appropriate message for the interworking operation. The IPC module is responsible for communication with the IDCS. Figure 5.6 shows the modules related to operation with the IDCS in the ISCS.

For the operation with the IDCS, the operation decision module internally consists of an event detector, event buffer, and IPC message generator. The event detector collects information acquired from the image grabbers and the IDCS and finds out the event in the iSpace. The detected event is accumulated in the event buffer and the IPC message generator produces an IPC message for the IDCS according to the detected event.

The IPC module in the ISCS consists of an IPC sending message buffer, IPC message sender, IPC message receiver, IPC received message buffer, and IPC message interpreter. The generated message in the operation decision module is added in the IPC sending message buffer, and the IPC message sender transmits the message to the IDCS. A message responded from the

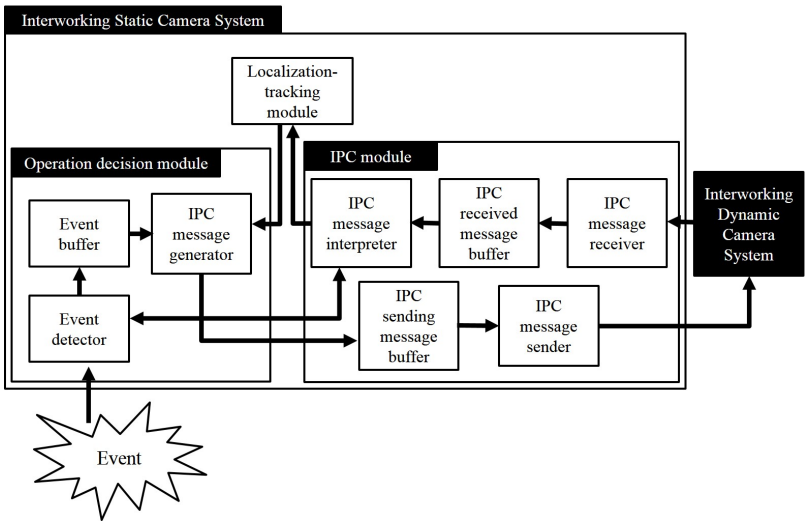


Figure 5.6 Software modules in an interworking static camera system for the operation with the interworking dynamic camera system.

IDCS is transmitted to the IPC message receiver. The received message is added in the IPC received message buffer. The IPC message interpreter fetches a message from the buffer, analyses the message, extracts data from the message, and sends the data to the localization-tracking module or the event detector in the ISCS.

The IDCS also has an IPC module for interworking with the ISCS. The IPC module in the IDCS consists of an IPC message generator, IPC sending message buffer, IPC message sender, IPC message receiver, IPC received message buffer, and IPC message interpreter, which is analogous to the module in the ISCS. A message from the ISCS is transmitted to the IPC message receiver, and the receiver adds the message to the IPC received

message buffer. The IPC message interpreter fetches the message from the buffer, analyses the message, and sends the data included in the message to the localization-tracking module or pan-tilt unit controller in the IDCS. After the reception, the IDCS produces an appropriate message through the IPC message generator using information from the pan-tilt unit controller and the localization-tracking module. The produced message is added in the IPC sending message buffer, which is transmitted to the ISCS by the IPC message sender. Figure 5.7 shows the modules related to operation with the ISCS in the IDCS.

The ISCS has a high degree of accuracy in localization of a robot. However the viewing direction of static cameras is parallel, and each static

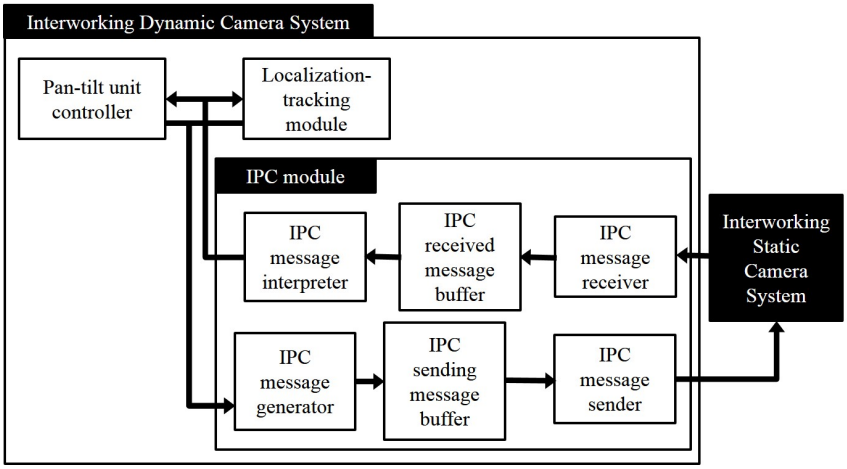


Figure 5.7 Software modules in an interworking dynamic camera system for the operation with the interworking static camera system.

camera cannot cover a large environment because the viewing direction is fixed. Therefore, when the robot is occluded by an obstacle in the view of the ISCS, the ISCS may fail to track the robot. In contrast, the dynamic cameras in the IDCS have different viewing direction because the installed position of the dynamic cameras is the sidewall of our iSpace. In addition, the viewing direction of the cameras is changeable. Thus, it is possible for the IDCS to track the robot when the ISCS fails. Therefore, an application of the operation between the ISCS and the IDCS in this dissertation is the continuous tracking of the robot when the robot is occluded. The detection of occlusion in view of the ISCS is decided by the detection score of the marker attached on the center of the robot's upper plate. When the robot is occluded, the ISCS using the aforementioned modules for operation is provided with the robot position by the IDCS, and the ISCS considers the provided position as the current robot position.

5.4 Proposed Operation with the iSpace and Robots

For operation between the iSpace and the interworking robot system (IRS), the iSpace should be aware of a situation in the iSpace and be able to communicate with the IRS. The ISCS in the iSpace is responsible for the operation between the iSpace and the IRS. The TCP module is activated instead of the IPC module when the iSpace interacts with the IRS. In addition, a TCP message generator instead of the IPC message generator is activated in

the operation decision module. The event detector collects information acquired from the image grabbers and the IRS finds out the event in the iSpace. The detected event is appended to the event buffer, and the TCP message generator produces a TCP message for the IRS according to the detected event.

The TCP module in the iSpace consists of a TCP sending message buffer, TCP message sender, TCP message receiver, TCP received message buffer, and TCP message interpreter. The generated message by the message generator is added in the TCP sending message buffer, and the TCP message sender transmits the message to the IRS. A message responded from the IRS is transmitted to the TCP message receiver. The received message is added in the TCP received message buffer. The TCP message interpreter fetches a message from the buffer, analyses the message, and sends the data extracted from the message to the localization-tracking module or the event detector in the iSpace. Figure 5.8 shows the modules related to the operation with the IRS in the iSpace.

The IRS also has a TCP module for interworking with the iSpace. Analogous to the iSpace, the TCP module in the IRS consists of a TCP message generator, TCP sending message buffer, TCP message sender, TCP message receiver, TCP received message buffer, and TCP message interpreter. A message from the iSpace is transmitted to the TCP message receiver, and the receiver adds the message to the TCP received message buffer. After the

message is analyzed in the TCP message interpreter, the data included in the message is sent to the localization-tracking module or robot controller in the IRS. After the reception, the IRS produces an appropriate message through the TCP message generator using information from the odometer module and the localization-tracking module. The generated message is added in the TCP sending message buffer, and it is transmitted to the iSpace by the TCP message sender. Figure 5.9 shows the modules related to the interworking operation with the iSpace in the IRS. .

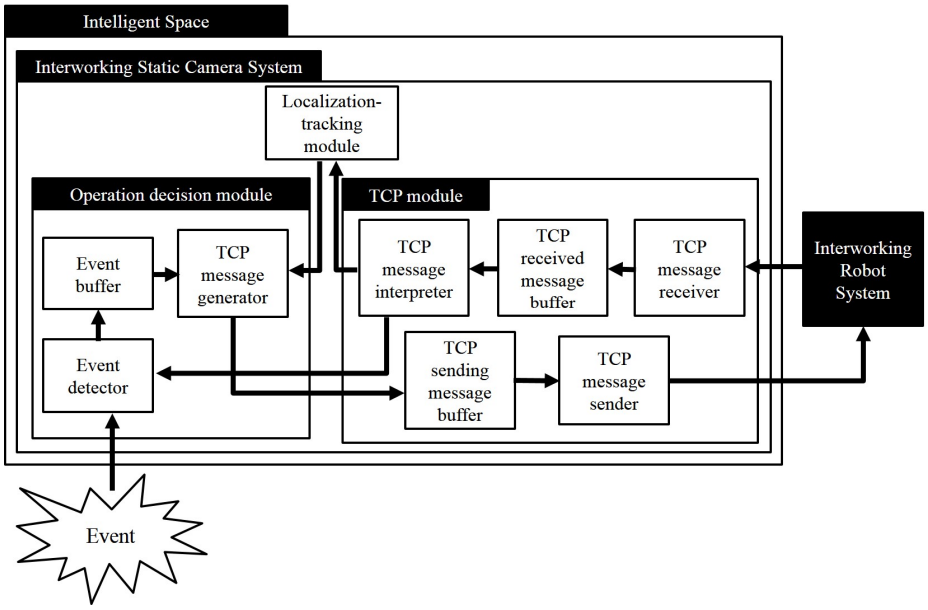


Figure 5.8 Software modules in the iSpace for the operation with the interworking robot system.

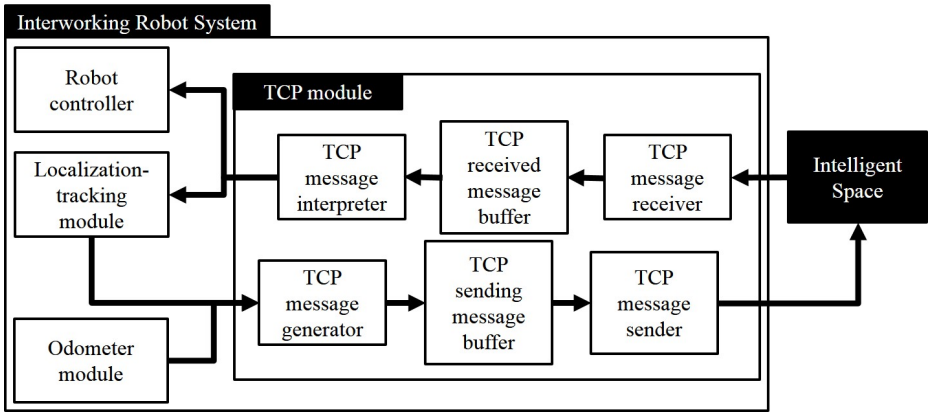


Figure 5.9 Software modules in the IRS for the operation with the iSpace.

The proposed iSpace consists of multiple cameras. The cameras are inexpensive and acquires much information. However, the cameras are sensitive to change of illumination in its surrounding environment. If illumination of the environment is low when the iSpace is tracking the robot, the iSpace would fail to track. In contrast to the iSpace, the IR sensor mounted on the robot can track the robot by itself in low-illumination environment. Therefore, an application of the operation between the iSpace and the IRS is the continuous tracking of the robot in a low-illumination environment. The measure of illumination is performed by the average of the pixel values in the image captured by the static cameras. When the iSpace decides that illumination of the environment is low, the iSpace requests the robot position from the IRS, and the iSpace considers the provided position as the current robot position.

Chapter 6

Experimental Results

6.1 Experimental Setup

Static cameras in the proposed iSpace are mounted on the horizontal supports (ceiling part of the iSpace) of the metallic structure mentioned in Chapter 3. Table 6.1 summarizes the mounted position of the static cameras. The static cameras have the identical z-coordinate.

Table 6.1 Position of Static Cameras (unit: *cm*)

Camera index	*x*	*y*	*z*
1	273	160	245
2	269	384	245
3	536	162	245
4	532	385	245

Dynamic cameras in the proposed iSpace are mounted on the side supports (sidewall part of the iSpace) of the metallic structure. Table 6.2 summarizes the mounted position of the dynamic cameras. The dynamic cameras have the identical z-coordinate.

Table 6.2 Position of dynamic cameras (unit: *cm*)

Camera index	*x*	*y*	*z*
1	190	540	210
2	189	17	210
3	600	540	210
4	599	17	210

The EFM Networks A2004NS dual-band wireless router is employed as the access point for the robots. The control server computer has the Intel quad core processor i7-3770 3.4GHz and 8GB memory, and the laptop computer for robot control has the Intel quad core mobile processor i7-3610QM 2.3GHz and 8GB memory. The IR sensor mounted on the robot is placed at 15*cm* forward from the center of the robot's upper plate.

6.2 Experimental Results of Localization

6.2.1 Results using Static Cameras and Dynamic Cameras

We measured positions at 88 points between 100 – 450*cm* in the *x* direction and 250 – 500*cm* in the *y* direction in order to verify localization

performance of the interworking static camera system (ISCS). We obtained 100 measurements at each position. The observed target's height is 24*cm*. Figure 6.1 shows the measurement spots (blue dots) and the average values (red crosses) of the measurements at each point. A red circle in the center of the figure is the position of static camera 2. Table 6.3 represents the error statistics for the position measurements before errors are corrected. The average error of x and y are 0.77*cm* and 0.95*cm*, respectively. The standard deviation of the errors of x and y is 0.68*cm* and 0.94*cm*, respectively. The maximum errors in x and y direction are 3.91*cm* and 3.34*cm*, respectively.

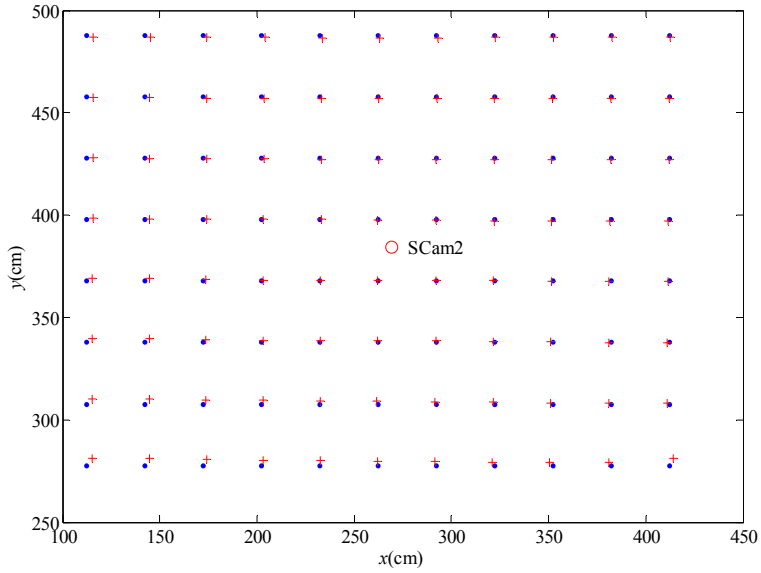


Figure 6.1 True positions (blue dots) and average points of measurements (red crosses) using static camera 2.

Table 6.3 Errors in the static camera system before correction (unit: *cm*)

	Avg. error	Std. error	Max. error
x	0.77	0.68	3.91
y	0.95	0.94	3.34

We analyzed the errors on the distance from the position of static camera 2. Figure 6.2 shows the distance errors sorted in the order of the distance from the projected center of static camera 2. The red line in this figure is the trend line of the distance errors. As shown in the trend line, the errors increase as the distance increases from static camera 2.

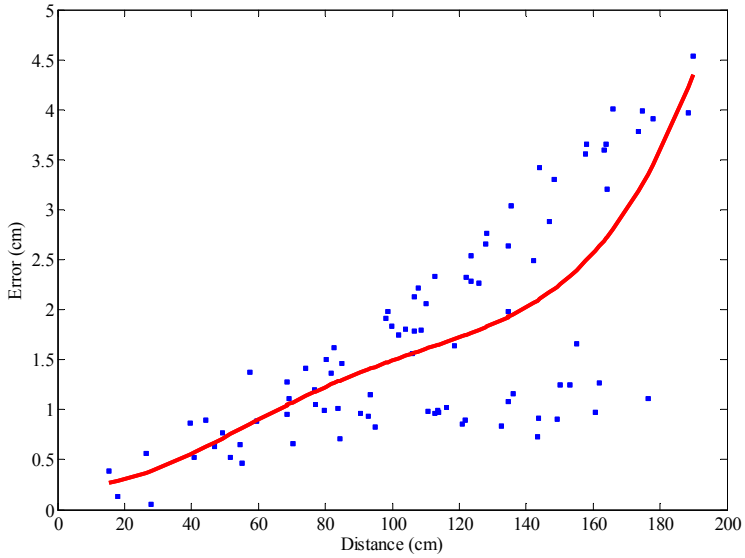


Figure 6.2 Distance errors sorted in order of the distance from static camera 2 to the measured points.

Figures 6.3 and 6.4 show the results of applying the correction method to the measured positions mentioned in Chapter 4. Figure 6.3 confirms that the corrected positions are closer to the true positions than before the correction method is applied. As shown in Tables 6.3 and 6.4, the average errors in the *x* and *y* directions are reduced by approximately 51% after the correction. The standard deviations of the error are reduced by approximately 53% and 64% in the *x* and *y* directions respectively. The maximum errors in the *x* and *y* directions are also reduced by approximately 43% and 56% respectively. In addition, the trend line in Figure 6.4 has a lower slope than the trend line in Figure 6.2.

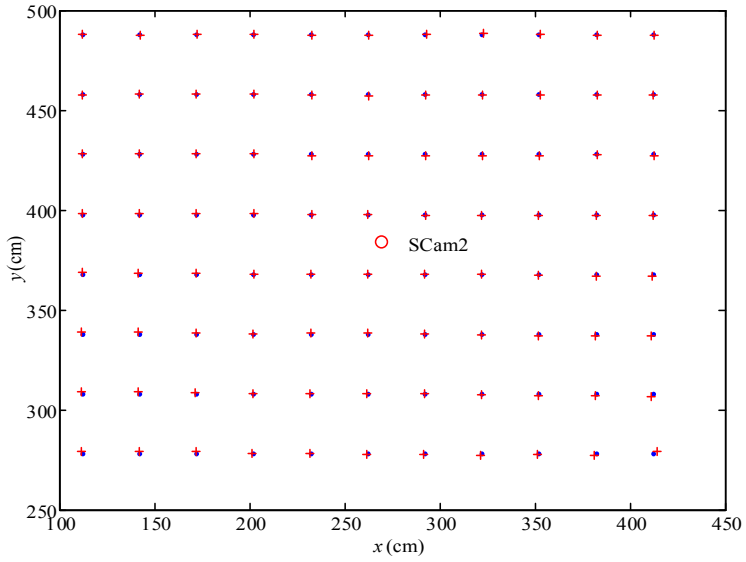


Figure 6.3 True positions (blue dots) and the average points of corrected measurements (red crosses) using static camera 2.

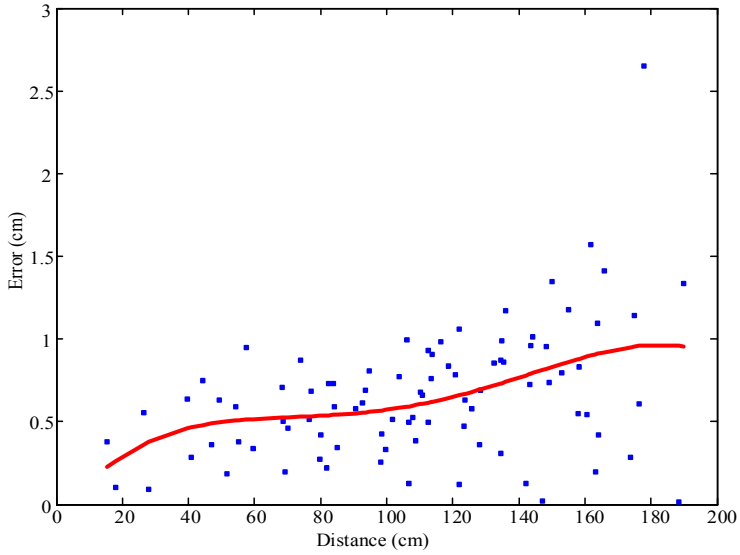


Figure 6.4 Distance errors sorted in order of the distance from static camera 2 to the corrected measured points.

Table 6.4 Errors in the static camera system after correction (unit: *cm*).

	Avg. error	Std. error	Max. error
x	0.38	0.32	2.22
y	0.47	0.34	1.46

Table 6.5 Parameters for localization correction in the static camera system (unit: *cm*)

Parameter	*k1*	*k2*	*k3*	*p1*	*p2*
value	2.517e-06	-1.337e-10	2.093e-15	-1.091e-05	-2.248e-05

Table 6.5 shows the calculated parameter for the correction method. The errors are relatively low, therefore the static camera system is adequate for the high accuracy tracking or localization.

We measured positions at 64 points between $100 - 350cm$ in the x direction and $250 - 500cm$ in the y direction in order to verify localization performance of the interworking dynamic camera system (IDCS). We obtained 100 measurements at each position. The measured target's height is $24cm$. Each measurement is obtained after the movement of the dynamic camera became stable. Figure 6.5 shows the measurement spots (blue dots) and the average values (red crosses) of the measurements at each of the spots.

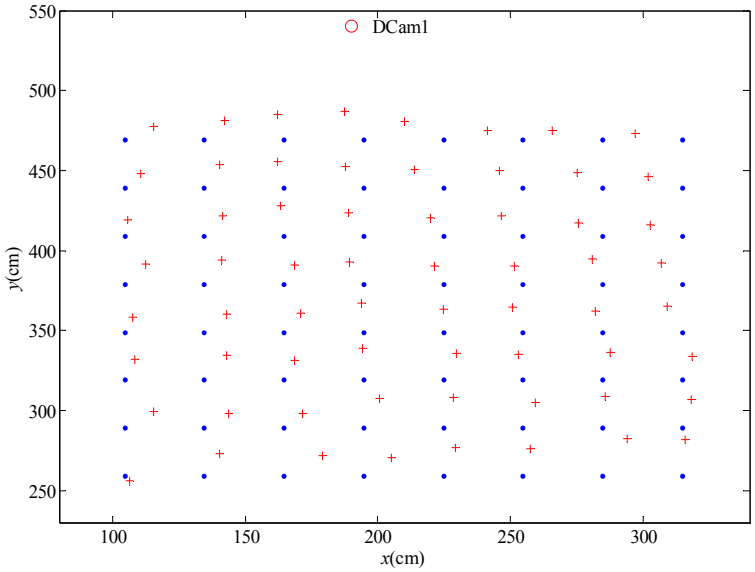


Figure 6.5 True positions (blue dots) and the average points of measurements (red crosses) using dynamic camera 1.

A red circle is the position of dynamic camera 1. Table 6.4 represents the error statistics for the position measurements. The average error of x and y are 8.03cm and 12.78cm respectively. These are approximately 21.13 and 27.19 times higher than the results of the static camera. The standard deviation of the errors of x and y is 5.35cm and 6.04cm respectively. These are approximately 16.71 and 17.76 times higher than the results of the static camera. The maximum errors in x and y directions are 20.03cm and 28.53cm, respectively. These are approximately 9.02 and 19.54 times higher than the results of the static camera.

We analyzed the errors for distance from the position of dynamic camera

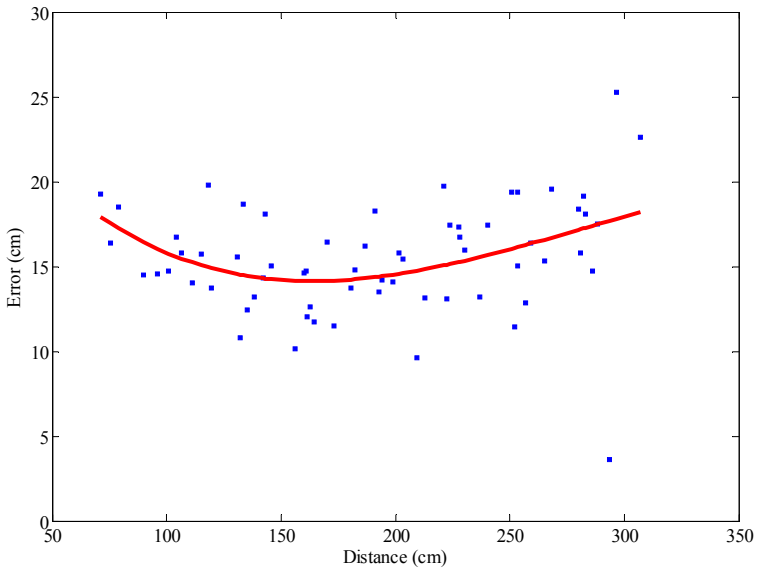


Figure 6.6 Distance errors sorted in order of the distance from the dynamic camera 1 to the measured points.

1. Figure 6.6 shows the distance errors sorted in the order of the distance from the center of dynamic camera 1. The red line in Figure 6.6 is the trend line of the distance errors. The errors decrease until approximately 160 and increase from approximately 160 along the increase of the distance from the camera position.

Table 6.6 Error in the dynamic camera system (unit: *cm*)

	Avg. error	Std. error	Max. error
x	8.03	5.35	20.03
y	12.78	6.04	28.53

6.2.2 Results using the IR Sensor

We measured positions at 54 points between 300 − 450*cm* in the *x* direction and 150 – 390*cm* in the *y* direction in order to verify localization performance of the IR sensor in the interworking robot system (IRS). We obtained 100 measurements at each position. Figure 6.7 shows the measurement spots (blue dots) and the average values (red crosses) of the measurements at each spots. A red circle is the position of IR tag 8. Table 6.7 represents the error statistics for the position measurements. The average error of *x* and *y* are 3.46*cm* and 2.71*cm*, respectively. These are approximately 9.11 and 5.76 times higher than the results of the static camera. The standard deviation of the errors of *x* and *y* is 2.22*cm* and 1.81*cm*, respectively. These are approximately 6.94 and 5.32 times higher than the results of static camera.

The maximum errors in the x and y directions are 9.93cm and 6.65cm, respectively. These are approximately 4.47 and 4.55 times higher than the results of the static camera.

We analyzed the errors for distance from the position of the IR tag. Figure 6.8 shows the distance errors sorted in the order of the distance from the center of the IR tag. The red line in Figure 6.8 is the trend line of the distance errors. The errors increase as the distance from the IR tag increases.

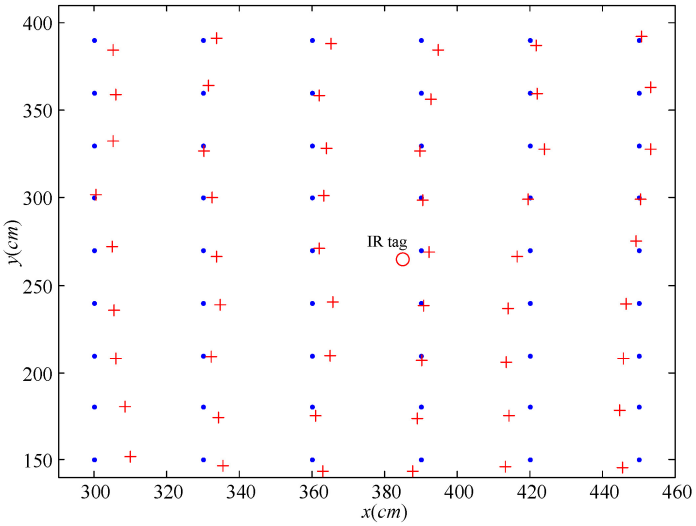


Figure 6.7 True positions (blue dots) and the average points of measurements (red crosses) using the IR sensor.

Table 6.7 Error of the IR Sensor mounted on the robot. (unit: *cm*)

	Avg. error	Std. error	Max. error
x	3.46	2.22	9.93
y	2.71	1.81	6.65

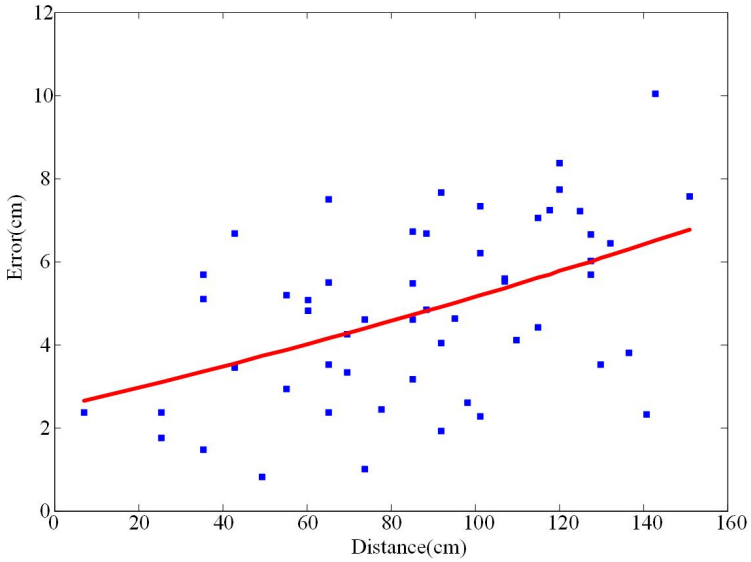


Figure 6.8 Distance errors sorted in order of the distance from the dynamic camera 1 to the measured points.

6.3 Experimental Results of Tracking

In order to verify the tracking performance of our system, we move a mobile robot in a rectangular path. The given start position (495, 105) of the robot is almost the same as the goal position (495, 90). Each of the sub-systems tracks the robot simultaneously. The following subsections explain the results of the tracking using each sensor system.

6.3.1 Results using Static and Dynamic Cameras

The tracking method using static cameras is the tracking-by-detection through the sliding window search method. The tracking method continuously

searches and detects the robot in images captured by static cameras. Figure 6.9 shows the tracking result for the mobile robot using the static camera system. The black dot is the robot's trajectory from the robot's wheel encoder. The blue × means the robot's trajectory from the static cameras. As might be expected, the results of the encoder shows the incremental errors. The interval in the trajectory using the static camera is caused by the handoff between adjacent static cameras.

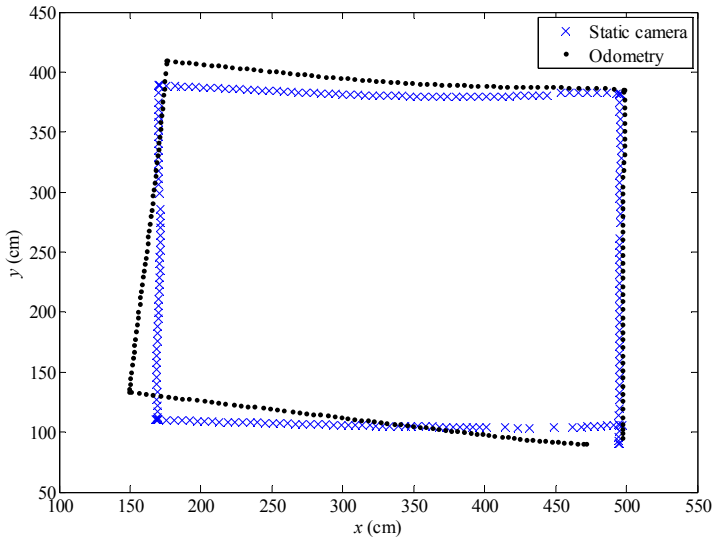


Figure 6.9 Tracking results using static cameras. Blue ×'s are the results from static cameras. Black dots are wheel-encoder data (odometry) of the mobile robot.

The robot tracking using a dynamic camera is actually performed in image space of the dynamic camera. The mean-shift tracker tracks the robot's image

using the color histogram. Therefore, the *x*-*y* tracking results are generated by the localization method using the geometric relationship between the robot and the dynamic camera and the predictive handoff method. Figure 6.10 shows the tracking results of the robot using multiple dynamic cameras. The tracked robot moved the same rectangular path shown in Figure 6.9. The red square is the trajectory generated by the dynamic cameras. The result from dynamic cameras has fluctuations. This fluctuation may be caused by encoder's resolution limitation of pan-tilt unit, trembling of the mean-shift tracking, and discontinuity in the handoff.

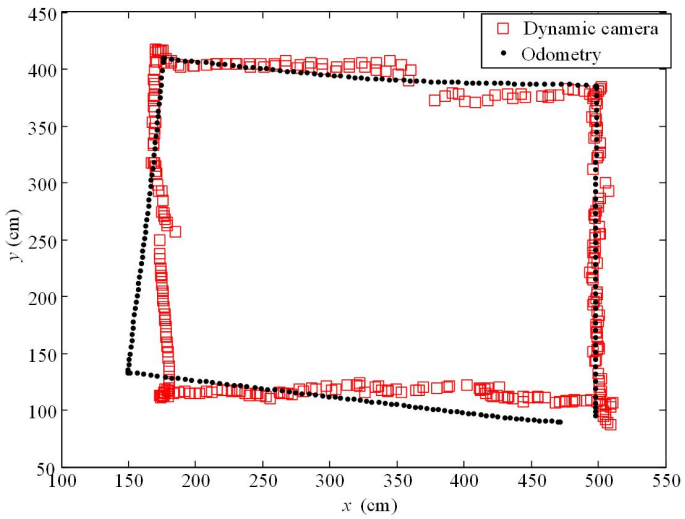


Figure 6.10 Tracking results using dynamic cameras. Red squares are the result from the dynamic cameras. Black dots are the wheel-encoder data (odometry) of the mobile robot.

We also compared the results from the static and dynamic cameras. Figure

6.11 shows the trajectories from both the static and dynamic cameras. As shown in this figure, the trajectory from dynamic cameras is inaccurate relative to the static cameras.

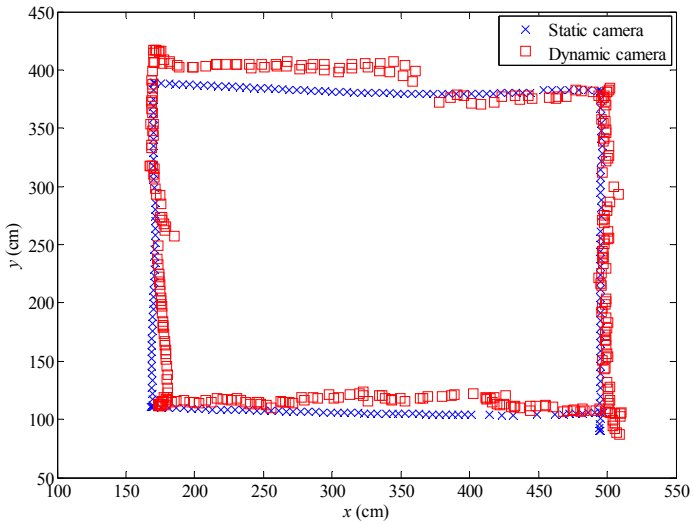


Figure 6.11 Comparison of tracking results. Blue ×s are the tracking results using static cameras. Red squares are the tracking results using dynamic cameras.

The ISCS can track multiple robots. In order to verify this, we provide the start positions and the goal positions for three robots. The start positions (s) and goal positions (g) are s1 (210, 144), g1 (552, 342), s2 (239, 302), g2 (474, 105) and s3 (496, 396) g3 (157, 150) for Robot 1, Robot 2, and Robot 3, respectively. Figure 6.12 shows the trajectories of the robots that move successfully to the goals. However, each robot has some broken intervals. These are caused by the handoff between the static cameras.

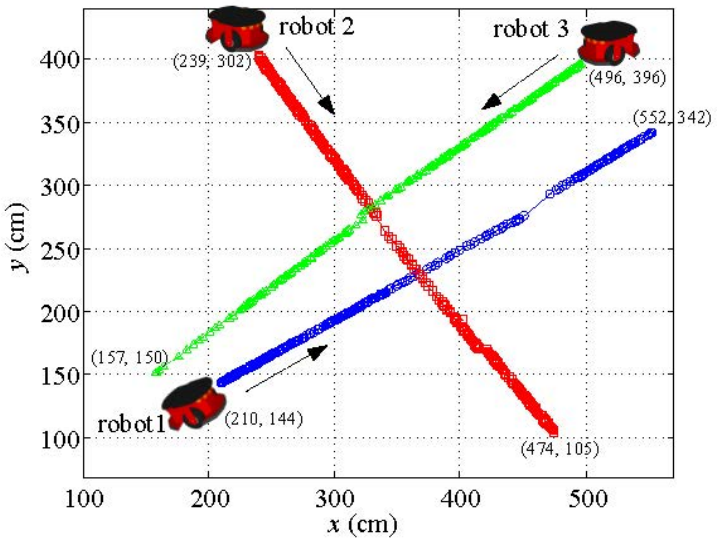


Figure 6.12 Tracking results for multiple robots using the static camera system.

6.3.2 Results using the IR Sensor

We used thirteen IR tags for the Stargazer to overcome the coverage limitation of a tag. We also employed the adaptive extended Kalman filter (AEKF) to estimate the position of a robot equipped with the IR sensor. Figure 6.13 shows the tracking result of the robot using the IR sensor, and the tracked robot moved the same rectangular path as shown in Figures 6.9 and 6.10. The green triangle is the trajectory tracked from the IR sensor. The trajectory from the IR sensor is very smooth.

We tried to compare the trajectories from the IR sensor and the dynamic

cameras. Figure 6.14 shows the trajectories from the IR sensor and the dynamic cameras. The resulted trajectory from the IR sensor is more smooth and continuous than the result from the dynamic cameras. We also tried to compare the resulted trajectories from the IR sensor and the static cameras. Figure 6.15 shows the trajectories tracked by the IR sensor and the static cameras. The trajectory result from the static cameras is very similar to the results from the IR sensor. However, as shown in the figure, the trajectory of the IR sensor shows a diminutive curved path when the robot moved straight.

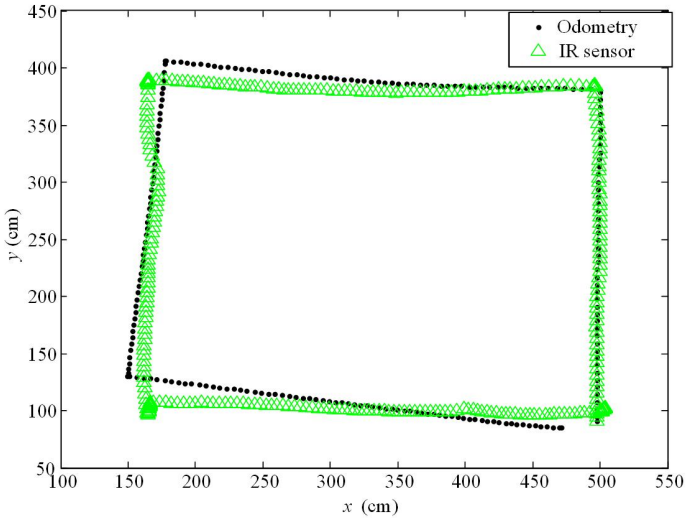


Figure 6.13 Tracking result using the IR sensor. Green triangles are the result from the IR sensor. Black dots are wheel-encoder data (odometry) of the mobile robot.

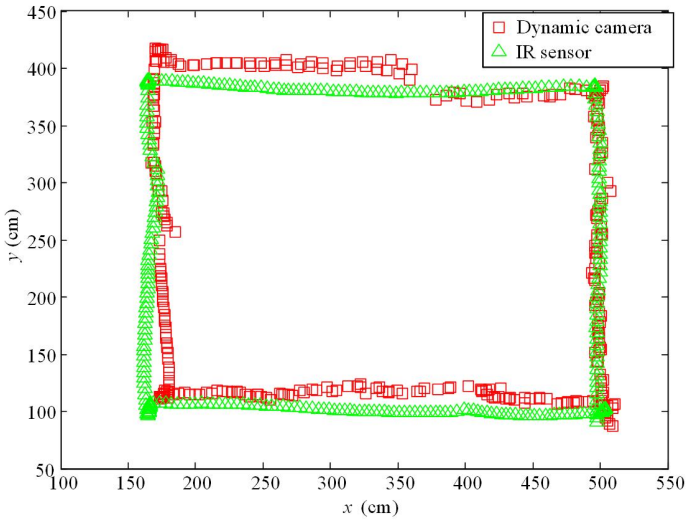


Figure 6.14 Comparison of the tracking results from the dynamic camera and IR sensor. Green squares are the tracking result using the IR sensor. Red squares are the tracking result using dynamic cameras.

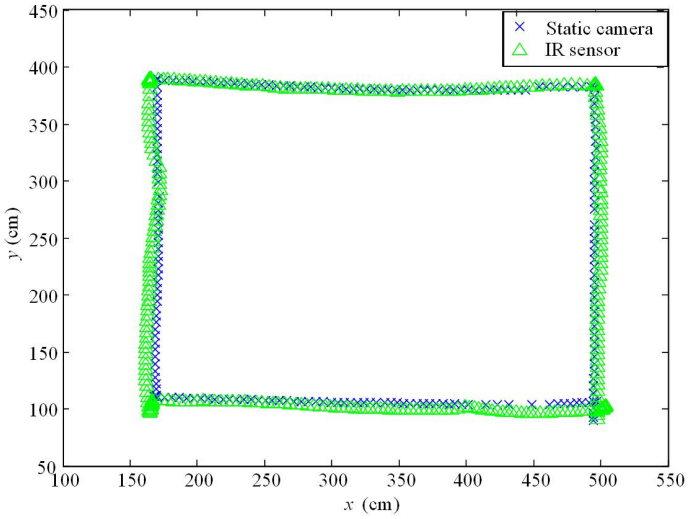


Figure 6.15 Comparison of tracking results from the static camera and IR sensor. Blue ×'s are the tracking result using static cameras. Green squares are tracking result using the IR sensor.

We compare the final positions of the results using other sensors with the final position of the result using the static cameras because the static camera is considered the most accurate in the iSpace. Table 6.6 summarizes the final position of the tracking results.

Table 6.8 Errors in final position (the reference: final position in the results of the static camera)

	wheel encoder	dynamic camera	IR sensor
x direction	-25.62	13.55	5.66
y direction	-16.10	0.01	-4.24
distance	30.26	13.55	7.08

6.4 Experimental Results using Heterogeneous Sensors

This dissertation designed the structure of operation between heterogeneous sensors and proposed applications for operation with heterogeneous sensors: tracking a robot in environment with occlusion and tracking a robot in low-illumination environment. Subsequent experiments is designed for verification of the proposed operation structure.

6.4.1 Results in Environment with Occlusion

It is possible for the dynamic camera system to observe the robot when the robot is occluded by an obstacle in the view of the static camera system. Thus, the dynamic camera system can track the robot when the static camera

systems fails because of the different viewing direction of the dynamic camera system. This dissertation proposed operation between the ISCS and IDCS when the robot is occluded. In order to verify the operation, we set up an experiment where four occluded regions are created. Figure 6.16 shows a depiction of the experiment of tracking in an occlusion environment. The robot moves through the occluded regions, and return to a point near the start position. Figure 6.17 shows the robot detected by the static camera system at the start point in this experiment. The white square means the detected robot. Figure 6.18 and 6.19 are the images from the static camera system and the dynamic camera system when the robot is occluded in the view of the static cameras. Figure 6.19 shows that when the robot is occluded, the robot is continuously tracked by the dynamic camera system.

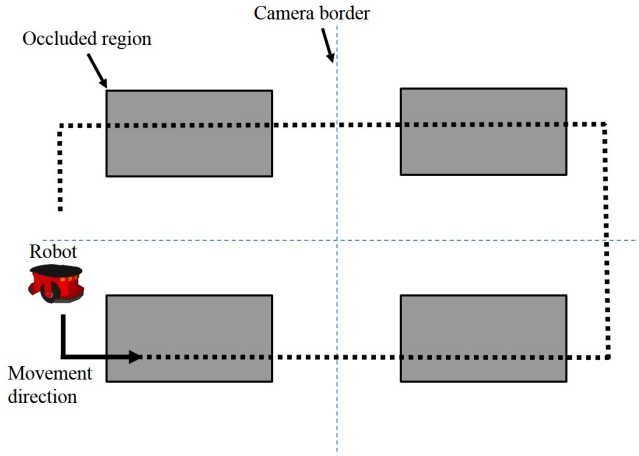


Figure 6.16 Depiction of the experiment of tracking in environment with occlusion

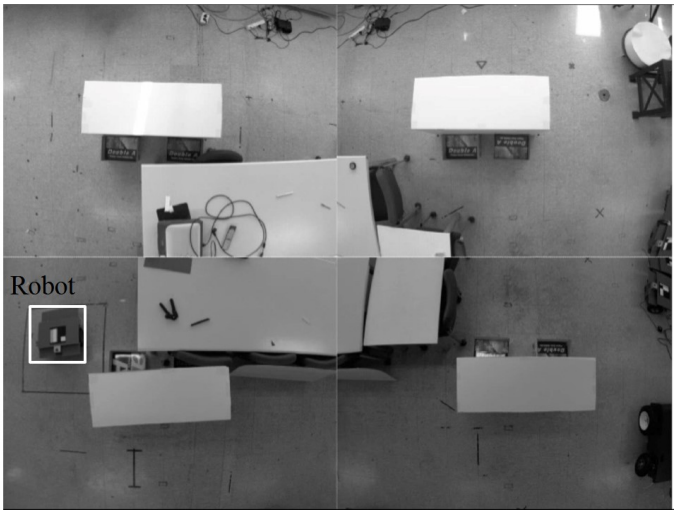


Figure 6.17 Robot detected by the ISCS at the start point

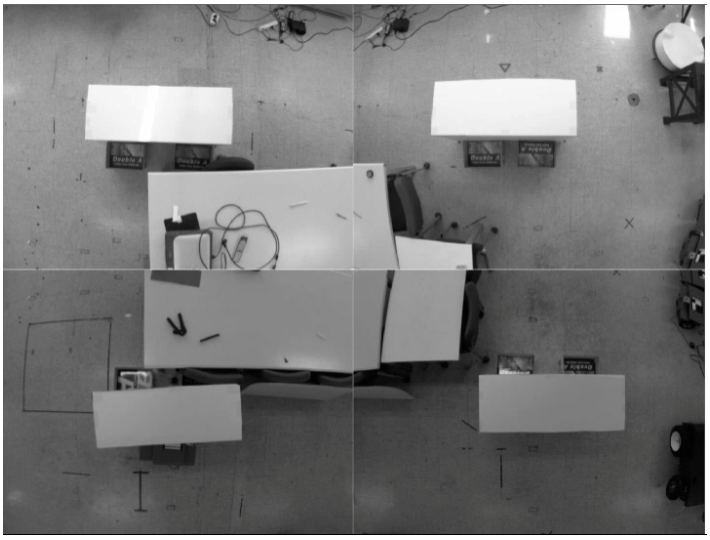


Figure 6.18 Situation in which the robot is occluded

Figure 6.19 Robot detected by the IDCS in the occluded situation.

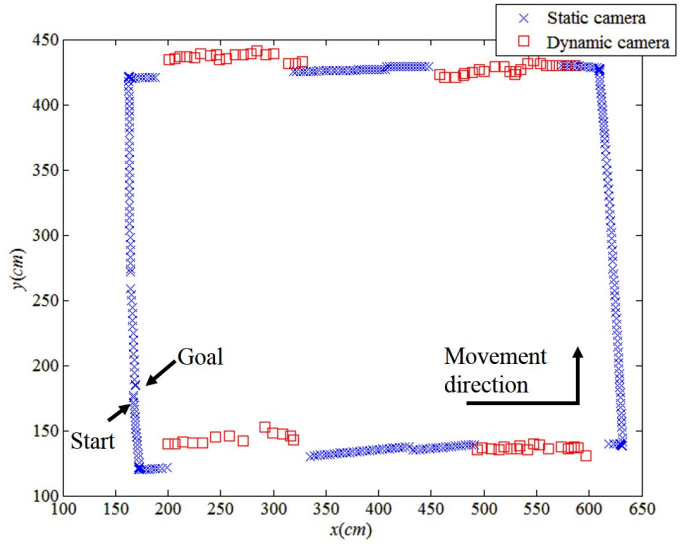


Figure 6.20 Results of tracking through the operation between the ISCS (static camera) and the IDCS (dynamic camera).

Figure 6.20 shows the result of tracking through the operation between the ISCS and the IDCS. The red squares represent the trajectory obtained from the IDCS, and the blue ×'s represent the trajectory obtained from the ISCS.

6.4.2 Results in Low-illumination Environment

The IR sensor in the interworking robot system (IRS) can track the robot even if illumination in the robot's surrounding environment is low. However, the ISCS and the IDCS may fail in low-illumination environment. This dissertation proposed the operation between the IRS and the iSpace when the robot moves into a low-illumination region. In order to verify the operation,

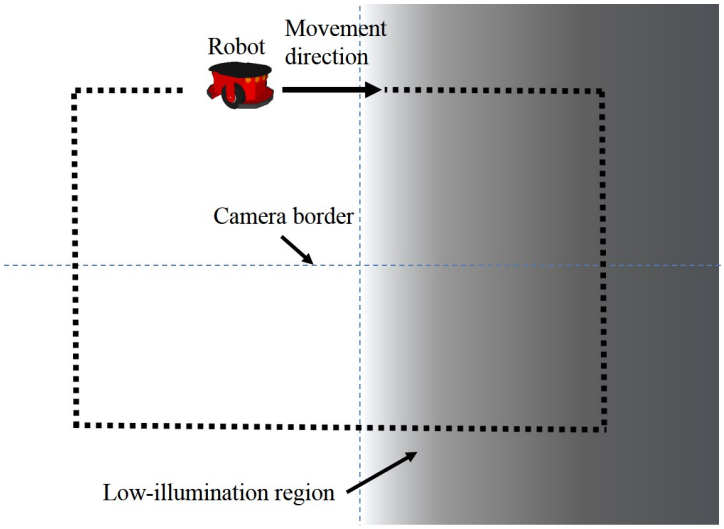


Figure 6.21 Depiction of the experiment of tracking in low-illumination environment.

we set up an experiment where almost half region of the environment is dark. Figure 6.21 shows depiction of the experiment. The robot moves through the low-illumination region, and return to a point near the start position.

Figure 6.22 shows the robot detected by the static camera system at the start point in this experiment. The white square is the detected robot. Figure 6.23 shows the image from the ISCS when the robot moves through the low-illumination region. Tracking the robot by the ISCS failed.

Figure 6.22 Robot detected by the ISCS at the start point

Figure 6.23 Situation in which the robot moves through the low-illumination region.

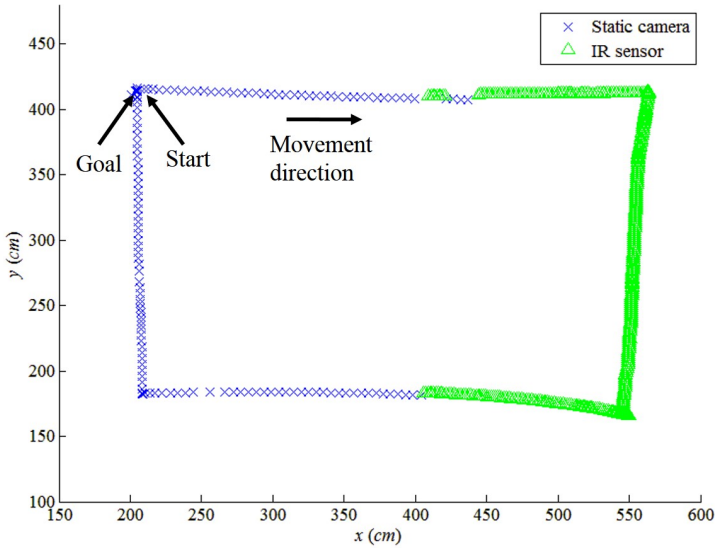


Figure 6.24 Results of tracking through the operation between the IRS (IR sensor) and the ISCS (static camera) of the iSpace.

Figure 6.24 shows the results of tracking through the operation between the IRS and the iSpace. The green triangles represent the trajectory obtained from the IRS, and the blue ×'s represent the trajectory obtained from the ISCS of the iSpace.

6.5 Discussion

The proposed iSpace in this dissertation requires the sub-systems to localize the robot in the iSpace. The ISCS showed a high degree of accuracy in localization. The ISCS also showed comparable performance with the representative commercial sensor systems ViCON and Ubisense, which have 1mm and 1.5cm in accuracy. Unfortunately, the ISCS tends to increase the average errors as the distance from the center of the static camera to the robot increases. Possibly, the localization trend of the ISCS is caused by the remaining error of the lens distortion. On the other hands, the dynamic camera of the IDCS shows inaccurate results in localization. The average errors of localization using the IDCS are more than twenty times greater than the ISCS. This may be caused by the inaccuracy of the mean-shift tracker and limited resolution of encoders in the pan-tilt unit. Meanwhile, localization using the IR sensor in the IRS is more accurate than the IDCS but less accurate than the ISCS. The average errors of localization using the ISCS are more than five times greater than the ISCS. Analogous to the ISCS, localization using the IR sensor tends to increase errors as the distance from the IR tag's position to the

robot increases. Supposedly, this also is caused by the lens distortion's effect in the IR sensor.

The tracking performance is verified by the above experiments using each sub-system in the iSpace. All sub-systems track the robot well and the tracking performance of the static camera system is the highest among the sub-systems because tracking is related with localization performance. In the result of tracking using the IR sensor, we can confirm the reduction of the inherent errors, shown in the result of localization using IR sensor, by AEKF.

The iSpace must be equipped with other sensors in order to expand the iSpace ability. Even if the iSpace is equipped with multiple sensors, the iSpace cannot perform efficiently except with interworking among the sensors. Thus, this dissertation designed and implemented the operation between heterogeneous sensors, which is verified by experiments. Although the result of tracking in an environment with occlusions is not smooth, the operation is successful between the ISCS and the IDCS. In the experiment under in a low-illumination environment, the resulted trajectory from the operation between the IRS and the iSpace is continuous compared to the result in environment with occlusions because of the smoothing by the AEKF.

Chapter 7

Conclusions

Intelligent space (iSpace) is a system that is embedded in environment and designed to track and control a robot. The conventional iSpace consists of multiple static cameras. The iSpace has focused on localization and tracking a single robot. However, this dissertation presents an intelligent space (iSpace) using robots, multiple static cameras, and multiple dynamic cameras. In addition, this dissertation presents operations with heterogeneous sensor groups. Operation of the conventional iSpace would fail in the low-illumination environment and environment with occlusion. However, the proposed iSpace is designed and developed to operate in the low-illumination environment and environment with occlusion.

This dissertation presents localization and tracking method using each sensor. First, the proposed static camera system localizes robots using edge

model of a marker attached on the robots and tracks robots using tracking-by-detection strategy with local tracking windows. The proposed static camera system showed similar performance to the expensive commercial localization system such as Ubisense and ViCON. Tracking using multiple static cameras requires a handoff method. This dissertation divides a handoff process using static cameras into three sub-processes and formulates each sub-process mathematically. The sub-processes consist of the handoff trigger, adequate camera selection, and handoff execution. The handoff method in the conventional iSpace requires multiple handoff map as change of the overlap region according to height of the robot. This is not desired because the method needs large memory space. This dissertation proposes a handoff method based on the transformation between the world coordinates and pixel coordinates to overcome the disadvantage.

Second, we develop a dynamic camera using a cheap pan-tilt unit and webcam. The proposed dynamic camera system localizes the robot using a single dynamic camera and tracks the robot with the mean-shift tracker using a color histogram of the robot. As in the case of the static cameras, this dissertation also divides a handoff process using dynamic cameras into three sub-processes and mathematically formulates each sub-process. The sub-processes consists of the handoff trigger, adequate camera selection, and handoff execution. This proposed method considers not only distance from the camera to the robot, but also the velocity of the robot. Then, we proposed

evaluation functions based on the velocity of the robot and the geometrical placement of the cameras for the handoff method. The proposed handoff method improves the reliability of the entire handoff process and reduces the effect of the delay caused by the pan-tilt motion.

Finally, a self-localization and tracking system of a robot using the IR sensor and the IR tags is developed. Because the coverage of a single IR tag is limited, multiple IR tags are required to cover a large environment. The localization using the tags has three shortcomings: uncertainty problem as distance increases, discontinuity problem, and noise problem. The developed system deploys the IR tags and employs the adaptive Kalman filter to minimize those shortcomings.

This dissertation also presents operation methods among heterogeneous sensors. The static cameras and the dynamic cameras are installed to have different views per camera. The static camera system has high accuracy of localization. However, when the robot is occluded, the static camera system fails to localize and track the robot. The operation between the static camera system and the dynamic camera system can overcome the failure because the view of the systems is different. Therefore, continuous tracking is possible using this operation in the proposed iSpace. We defined commands for this operation and constructed an inter-process based communication method. Apart from that, the camera systems failed to localize and track the robot in a low-illumination environment. However, the IR sensor mounted on the robot

performs in the low-illumination environment although localization and tracking performance using the IR sensor is less accurate than the static camera system. Therefore, the proposed operation among robots and cameras can overcome this failure. We also defined commands for this operation and constructed a TCP-based communication method.

This dissertation deals with real-time operation and implementation of sensor systems in a proposed iSpace and an operation method for interworking among heterogeneous sensor systems. Thus, this dissertation does not suggest a robot control method. However, if the part of robot control is added, the proposed system could be applied to the research experiments on multi-robot cooperation or coordination. Therefore, it is necessary to include a robot control with the proposed system in our future research.

In the future, we will extend the target from robots to other entities such as humans, objects, and pets. In order to interact with human and other objects, we may need to obtain face recognition, gesture recognition, and pet recognition. In addition, we need to adopt other sensors such as range sensors, microphones, and RGB-D sensors to extend the operation among the sensors. Table 7.1 is an outline of the study highlights.

Table 7.1 Summary of Study Proposal

Summary of Study Proposal
● An iSpace using multiple static cameras, multiple dynamic cameras, and robots mounted with the IR sensor is proposed and implemented. ● Localization and tracking method using multiple static cameras with the re-projection based handoff method is proposed. ● Localization and tracking method using multiple dynamic cameras with the predictive handoff method is proposed. ● Localization and tracking method using the IR sensor and IR tags is developed. ● Operation methods among heterogeneous sensors are proposed.

Bibliography

[1] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianc, A. Klaptocz, S. e. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a Robot Designed for Education in Engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Castelo Branco, Portugal, pp. 59-65, 2009.

[2] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, "Image and animation display with multiple mobile robots," *The International Journal of Robotics Research,* vol. 31, no. 6, pp. 753-773, 2012.

[3] J. H. Lee, J. S. Choi, B. H. Lee, and K. W. Lee, "Complete coverage path planning for cleaning task using multiple robots," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Taxas, USA, pp. 3618-3622, 2009.

[4] P. Santana, J. Barata, H. Cruz, A. Mestre, J. Lisboa, and L. Flores, “A multi-robot system for landmine detection,” in *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, pp. 720-728, 2005.

[5] Y.-S. Jung, K.-W. Lee, S.-Y. Lee, M. Choi, and B.-H. Lee, “An efficient underwater coverage method for multi-AUV with sea current disturbances,” *International Journal of Control, Automation and Systems,* vol. 7, no. 4, pp. 615-629, 2009.

[6] D. Portugal, and R. P. Rocha, “Multi-robot patrolling algorithms: examining performance and scalability,” *Advanced Robotics,* vol. 27, no. 5, pp. 325-336, 2013.

[7] P. R. Wurman, R. D'Andrea, and M. Mountz, “Coordinating hundreds of cooperative, autonomous vehicles in warehouses,” in *Proceedings of the 19th national conference on Innovative applications of artificial intelligence*, Vancouver, British Columbia, Canada, pp. 1752-1759, 2007.

[8] J. Kim, and B.-H. Lee, “Non-Oscillatory Multi-Robot Motion for Stable Target Capture,” *International Journal of Robotics and Automation,* vol. 26, no. 4, pp. 441-449, 2011.

[9] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli,

"Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems," *Robotics, IEEE Transactions on,* vol. 23, no. 6, pp. 1170-1183, 2007.

[10] J. McLurkin, "Experiment Design for Large Multi-Robot Systems," in *Proceedings of Robotics Science and Systems Workshop: Good Experimental Methodology in Robotics*, Seattle, WA, USA, 2009.

[11] T. Fukuda, and T. Shibata, "Hierarchical control system in intelligent robotics and mechatronics," in *Proceedings of International Conference on Industrial Electronics, Control and Instrumentation*, Maui, HI, pp. 33-38, 1993.

[12] N. Streitz, J. Geißler, and T. Holmer, "Roomware for Cooperative Buildings: Integrated Design of Architectural Spaces and Information Spaces," *Cooperative Buildings: Integrating Information, Organization, and Architecture*, Lecture Notes in Computer Science N. Streitz, S. i. Konomi and H.-J. Burkhardt, eds., pp. 4-21: Springer Berlin Heidelberg, 1998.

[13] K. C. Ng, H. Ishiguro, M. Trivedi, and T. Sogo, "Monitoring dynamically changing environments by ubiquitous vision system," in *Proceedings of Visual Surveillance, 1999. Second IEEE Workshop on, (VS'99)*, pp. 67-73, 1999.

[14] P. Castro, and R. Munz, “Managing context data for smart spaces,” *Personal Communications, IEEE,* vol. 7, no. 5, pp. 44-46, 2000.

[15] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "EasyLiving: Technologies for Intelligent Environments," *Handheld and Ubiquitous Computing*, Lecture Notes in Computer Science P. Thomas and H.-W. Gellersen, eds., pp. 12-29: Springer Berlin Heidelberg, 2000.

[16] M. Lucente, G.-J. Zwart, and A. George, “Visualization space: a testbed for deviceless multimodal user interface,” in *Proceedings of AAAI Intelligent Environments Symposium*, CA, USA, pp. 87–92, 1998.

[17] B. Johanson, A. Fox, and T. Winograd, “The Interactive Workspaces project: experiences with ubiquitous computing rooms,” *IEEE Pervasive Computing,* vol. 1, no. 2, pp. 67-74, 2002.

[18] G. D. Abowd, “Classroom 2000: An experiment with the instrumentation of a living educational environment,” *IBM Systems Journal,* vol. 38, no. 4, pp. 508-530, 1999.

[19] A. F. Bobick, S. S. Intille, J. W. Davis, F. Baird, C. S. Pinhanez, L. W. Campbell, Y. A. Ivanov, A. Sch\, \#252, tte, and A. Wilson, “The KidsRoom: A Perceptually-Based Interactive and Immersive Story

Environment," *Presence: Teleoperators and Virtual Environments,* vol. 8, no. 4, pp. 369-393, 1999.

[20] L. Rudolph, "Project Oxygen: Pervasive, Human-Centric Computing – An Initial Experience," *Advanced Information Systems Engineering*, Lecture Notes in Computer Science K. Dittrich, A. Geppert and M. Norrie, eds., pp. 1-12: Springer Berlin Heidelberg, 2001.

[21] T. Sato, Y. Nishida, and H. Mizoguchi, "Robotic room: Symbiosis with human through behavior media," *Robotics and Autonomous Systems,* vol. 18, no. 1–2, pp. 185-194, 1996.

[22] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong, "ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing," *Computer,* vol. 37, no. 10, pp. 73-81, 2004.

[23] X. Zhao, Z. He, S. Zhang, S. Kaneko, and Y. Satoh, "Robust face recognition using the GAP feature," *Pattern Recognition,* vol. 46, no. 10, pp. 2647-2657, 2013.

[24] S. Mitra, and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews,* vol. 37, no. 3, pp. 311-324, 2007.

[25] J. Holmes, and W. Holmes, *Speech Synthesis and Recognition*, 2nd

ed.: CRC press, 2001.

[26] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, “Robust automatic speech recognition with missing and unreliable acoustic data,” *Speech Communication,* vol. 34, no. 3, pp. 267-285, 2001.

[27] J.-H. Lee, and H. Hashimoto, “Intelligent Space — concept and contents,” *Advanced Robotics,* vol. 16, no. 3, pp. 265-280, 2002.

[28] D. Herrero, and H. Martínez, “Fuzzy Mobile-Robot Positioning in Intelligent Spaces Using Wireless Sensor Networks,” *Sensors,* vol. 11, no. 11, pp. 10820-10839, 2011.

[29] J.-H. Lee, K. Morioka, N. Ando, and H. Hashimoto, “Cooperation of distributed intelligent sensors in intelligent environment,” *IEEE/ASME Transactions on Mechatronics,* vol. 9, no. 3, pp. 535-543, 2004.

[30] K. Morioka, J.-H. Lee, and H. Hashimoto, "Intelligent Space for Human Centered Robotics," *Advances in Service Robotics*, H. S. Ahn, ed., InTech, 2008.

[31] C. Losada, M. Mazo, S. Palazuelos, D. Pizarro, and M. Marrón, “Multi-Camera Sensor System for 3D Segmentation and Localization of Multiple Mobile Robots,” *Sensors,* vol. 10, no. 4, pp. 3261-3279, 2010.

[32] A. Canedo-Rodriguez, R. Iglesias, C. Regueiro, V. Alvarez-Santos, and X. Pardo, “Self-Organized Multi-Camera Network for a Fast and Easy Deployment of Ubiquitous Robots in Unknown Environments,” *Sensors,* vol. 13, no. 1, pp. 426-454, 2012.

[33] D. Brŝĉić, and H. Hashimoto, “Sensing in intelligent spaces: Joint use of distributed and onboard sensors,” *Automatika,* vol. 50, no. 3-4, pp. 205-214, 2009.

[34] D. Pizarro, M. Mazo, E. Santiso, M. Marron, D. Jimenez, S. Cobreces, and C. Losada, “Localization of Mobile Robots Using Odometry and an External Vision Sensor,” *Sensors,* vol. 10, no. 4, pp. 3655-3680, 2010.

[35] F. Hashikawa, K. Morioka, and N. Ando, "Mobile Robot SLAM Interacting with Networked Small Intelligent Sensors Distributed in Indoor Environments," *Simulation, Modeling, and Programming for Autonomous Robots*, Lecture Notes in Computer Science I. Noda, N. Ando, D. Brugali and J. Kuffner, eds., pp. 275-286: Springer Berlin Heidelberg, 2012.

[36] I. Fernández, M. Mazo, J. Lázaro, D. Pizarro, E. Santiso, P. Martín, and C. Losada, “Guidance of a mobile robot using an array of static cameras located in the environment,” *Autonomous Robots,* vol. 23, no.

4, pp. 305-324, 2007.

[37] K.-W. Lee, J.-B. Park, and B.-H. Lee, “Dynamic localization with hybrid trilateration for mobile robots in intelligent space,” *Intelligent Service Robotics,* vol. 1, no. 3, pp. 221-235, 2008.

[38] T. Tanikawa, K. Ohara, and K. Ohba, “Structural environmental informatization using sensor network with heterogeneous sensors,” *Intelligent Service Robotics,* vol. 3, no. 1, pp. 21-28, 2010.

[39] J. S. Park, and J.-H. Lee, “Reconfigurable intelligent space, R+iSpace, and mobile module, MoMo,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3865-3866, 2012.

[40] J.-E. Lee, J. Park, G.-S. Kim, J.-H. Lee, and M.-H. Kim, “Interactive Multi-Resolution Display Using a Projector Mounted Mobile Robot in Intelligent Space,” *International Journal of Advanced Robotic Systems,* vol. 1, no. 9, 2012.

[41] S. Gruenwedel, V. Jelaca, J. O. Nino-Castaneda, P. v. Hese, D. v. Cauwelaert, D. v. Haerenborgh, P. Veelaert, and W. Philips, “Low-complexity scalable distributed multicamera tracking of humans,” *ACM Transactions on Senor Networks,* vol. 10, no. 2, pp. 1-32, 2014.

[42] W. Limprasert, A. Wallace, and G. Michaelson, “Real-Time People

Tracking in a Camera Network," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 3, no. 2, pp. 263-271, 2013.

[43] A. S. Jalal, and V. Singh, "The State-of-the-Art in Visual Object Tracking," *Informatica,* vol. 36, no. 3, 2012.

[44] Hagisonic, "User's Guide Localization System StarGazer™ for Intelligent Robots," 2008, http://www.hagisonic.com/, accessd: 10 May 2014.

[45] J. L. Fernández, C. Watkins, D. P. Losada, and M. D.-C. Medina, "Evaluating Different Landmark Positioning Systems Within the RIDE Architecture," *Journal of Physical Agents,* vol. 7, no. 1, pp. 3-11, 2013.

[46] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots," *IEEE Transactions on Robotics and Automation,* vol. 15, no. 2, pp. 219-229, 1999.

[47] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-577, 2003.

[48] R. Simmons, "The curvature-velocity method for local obstacle

avoidance," in *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, pp. 3375-3382 vol. 4, 1996.

[49] K. H. Jeong, J. S. Choi, and B. H. Lee, "Reliability-Based Camera Handoff for Cooperative Tracking with Multiple Pan-Tilt Cameras," *International Journal of Control, Automation and Systems,* vol. 11, no. 4, pp. 815-825, 2013.

[50] Wikipedia. "Bayer Filter," Nov. 26, 2013; http://en.wikipedia.org/wiki/Bayer_filter.

[51] R. Jean, "Demosaicing with the bayer pattern," *Red,* vol. 33, no. R53, pp. 2, 2010.

[52] J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1066-1077, 2000.

[53] J. Canny, "A Computational Approach To Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, 1986.

[54] Matrox Electronic Systems, Ltd., *Matrox Imaging Library 8 User Guide*, Canada: Matrox Electronic Systems, Ltd., 2005.

[55] G. Bradski, and A. Kaehler, *Learning OpenCV*, CA, USA: O'Reilly

Media, Inc., 2008.

[56] N. Dalal, and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 886-893, 2005.

[57] L. Xuefeng, T. Tomizawa, D. Hyun Min, K. Yong-Shik, K. Ohara, K. Bong Keun, T. Tanikawa, and K. Ohba, “Multiple robots localization using large planar camera array for automated guided vehicle system,” in *Proceedings of International Conference on Information and Automation*, Changsha, pp. 984-990, 2008.

초 록

본 논문에서는 다수의 이종 비전 센서들과 로봇에 장착된 IR 센서를 이용하는 지능형 공간에서 다중 센서 운용에 대한 새로운 접근이 제시되었다. 지능형 공간은 본래 로봇의 작업공간에 존재하면서 로봇의 미션을 도와주고 특정한 상황에서는 로봇을 제어하는 시스템을 말한다. 기존 대부분의 연구들에서의 지능형 공간은 정적 카메라들로 구성되어왔다. 하지만 지능형 공간의 기능을 확장하기 위해서는 다수의 이종 센서들의 탑재와 이를 위한 운용 기법이 필요하다.

우선, 본 논문은 제안된 지능형 공간 내의 각 센서 그룹들을 위한 하위시스템들을 제안하였다. 비전 센서들은 정적 (고정) 카메라와 동적 (팬-틸트) 카메라로 나뉜다. 각 비전 센서들을 이용하는 하위 시스템들은 각기 다른 방법으로 로봇을 탐지하고 추적한다. 다수의 정적 카메라를 이용하는 하위시스템은 높은 정확도로 로봇의 위치를 추정한다. 그리고 다수의 정적 카메라들의 협업을 위한 월드좌표와 픽셀좌표간의 변환을 이용한 핸드오프 방법이 제안되었다. 다수의 동적 카메라를 이용하는 하위시스템은 로봇을 놓치지 않기 위해서 다양한 시야를 갖도록 고안되었다. 그리고 다수의 동적 카메라들의 협업을 위해 예측된 로봇 위치와 카메라 간의 관계, 그리고

로봇과 카메라 간의 관계를 이용하는 핸드오프 기법이 제안되었다. 로봇을 위한 하위시스템은 로봇에 장착된 IR 센서와 지능형 공간에 부착된 IR 태그들을 이용해서 로봇의 위치를 추정한다. 각 IR 태그들은 오차를 최소화하도록 설치되었으며 IR 센서는 저조도 환경에서도 로봇의 위치 추정이 가능하다는 특징을 갖는다.

본 논문에서는 작업공간의 다양한 환경변화에서도 강인한 추적을 위해 각 센서의 장점을 활용하는 센서 선택 방법을 제안하였다. 이를 위해 각 하위시스템 간 인터페이스 규약과 센서 우선순위, 그리고 센서 선택 조건들이 정의되었다. 제안된 센서 선택법은 기존의 센서 융합보다 수행속도가 빠르다는 이점을 지니기 때문에 실시간 시스템 구성에 적합하다.

각 하위시스템들의 성능은 다양한 실험을 통해서 검증되었다. 그리고 제안된 센서선택법을 이용한 다중 센서의 운용은 로봇이 정적 카메라의 시야에서 가려지는 상황과 저조도 환경이 존재하는 작업공간에서 로봇이 주행하는 상황에서 검증되었다.

주요어: intelligent space, 센서 선택법, 이종 센서, 카메라 핸드오프, 다중 카메라 추적

학 번: 2009-30911