Doctoral Thesis reviewed by Ritsumeikan University

Reconfigurable Intelligent Space for the sustainable high-quality service provision

（持続的な高品質なサービス提供のための再構成可能な知能化空間に関する研究）

September 2016
2016年 09月

Doctoral Program in Integrated Science and Engineering
Graduate School of Science and Engineering
Ritsumeikan University

立命館大学大学院理工学研究科
総合理工学専攻博士課程後期課程

PARK JongSeung
パク ジョンスン

Supervisor : Professor LEE Joo-Ho
研究指導教員 ： 李 周浩 教授

Reconfigurable Intelligent Space for the sustainable high-quality service provision

立命館大学
理工学部
総合理工学専攻

学生証番号 6130110034-4
PARK JongSeung

2016 年 9 月

Abstract

This thesis introduces reconfigurable intelligent space (R+iSpace) that can solve the device's spatial constraint problem in conventional intelligent space (iSpace). iSpace provides many kinds of services to user in it. For achieving its goal, many kinds and numbers of devices are installed in the space. Through the input devices, iSpace recognizes the spatial situation and users' demands. Then, the agent robots and the output devices provide physical and non-physical service to the users. For the successful service provision, the target user should be placed in the coverage of device and she/he should face appropriate direction. However, the spatial situation can be changed frequently. As a result, to solve this problem, all kinds of devices should be installed to everywhere for the consistent high-quality service provision. However, it is an unfeasible solution.

R+iSpace can solve the problem by rearranging the devices. Each device is mounted on the wall climbing robot, called mobile module (MoMo), and MoMo can rearrange itself according to the spatial situation. This thesis discusses three related essential research topics for achieving R+iSpace. First, the suitable mechanical structures of MoMo are discussed. There are some requirements of R+iSpace, and these requirements can be satisfied by the suitable mechanical structure of MoMo. This thesis introduce the mechanical structure of three prototypes of MoMo. The experiments for verifying MoMo's mobility were performed. Second, this thesis discusses an appropriate location for device. The successful service provision can be achieved when the device is located in the appropriate area according to situation. This thesis proposes the algorithm for the best effort location of device, and it is verified by simulation experiments. Third, the method of path generation without collisions and deadlock situations is discussed. Since there are more than one MoMo in the same field, in order to avoid collisions and deadlock situations, the robust path generation method is required. A multiple layer structured path generation algorithm is proposed. The proposed method is verified by simulation experiments.

Contents

List of Figures

List of Tables

Chapter 1

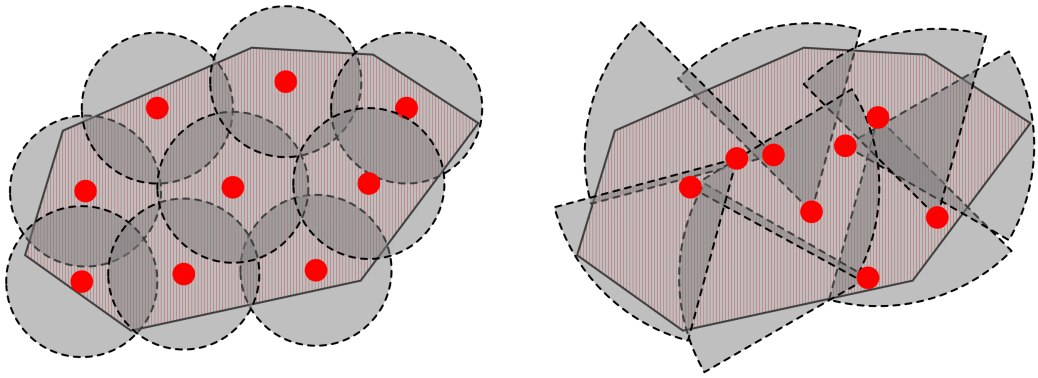
Introduction

"Where is the best position of device in varying environment?"

This is a traditional question that has been discussed in research fields which provide service by using devices. All devices have their usable distance and angle of view, and they makes a spatial constraint. A device is useless when a target of the device located in outside of the device's coverage. Therefore, before using a device, the device or its target should be rearranged to suitable place where the target is placed into the coverage of the device. However, in some research fields (e.g., Sensor Networks, Ubiquitous Computing, Intelligent Environment), the devices are fixed to their initial positions, and the targets can be moved freely in the space. Accordingly, the importance of the suitable device deployment has been more emphasized in these research fields.

Actually, there are many approaches to solve this problem. One of the famous approaches is Art Gallery Problem (AGP) which asked for the minimum number of guards placed inside of a polygon that are sufficient to cover the entire polygon [2,3]. However, this method assumed that a guard can watch any point as long as a line-of-sight exists. Therefore, it is not suitable for the device deployment method in real world since an actual sensor has its sensible range.

In other research, the device is modeled to have a limited coverage, and the studies proposed a method to cover the target area with minimum number of devices [4–6]. The proposed algorithm erates the arrangement of device as shown in Fig. 1.1. The devices in Fig. 1.1(a) are omni-directional devices, such as microphones and speakers. Therefore, the coverage of device was modeled as a form of sphere. The coverage



(a) Omni-directional devices deployment (b) Directional devices deployment

Fig. 1.1 Target Area and Device Deployment

of device in Fig. 1.1(b) was modeled as a form of a sector because it has a viewing direction and the angle of view. As shown in Fig. 1.1, the coverage of devices covers the entire target area with minimum number of devices. This methods can provide a good solution in cases of some applications, such as a fire alarm application and surveillance system. However, it cannot be a solution in cases of some applications, such as a face recognition and an emotion recognition. This method does not consider a direction of target, and the applications require the suitable direction of target. The detailed description of this problem will be explained in Section 1.2. As a result, this deployment of devices is not suitable in lots of applications. This thesis handles this problem and proposes a new approach to solve the problem.

1.1 Research background

As mentioned avobe, this thesis proposes a new approach to solve the device deployment. In this section, the research background of this thesis will be described in detail before explanation of the new approach. First, the overview of the research fields that place emphasis on the device deployment method will be described (e.g., Sensor Network, Ubiquitous Computing, Intelligent Environment). Then, their proposed device deployment method will be explained.

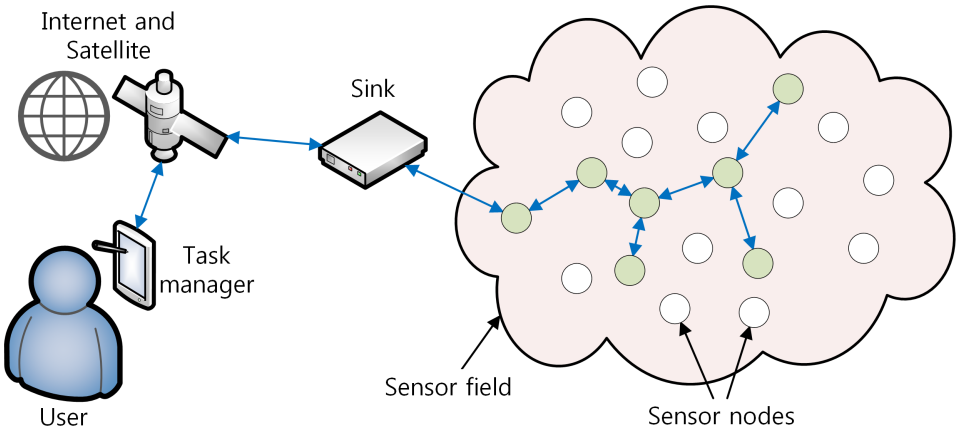


Fig. 1.2 Communication Architecture of Wireless Sensor Network

Sensor Network comprises lots of sensor devices that can communicate with each other through the communication networks and aim at performing tasks, such as exploration, surveillance, monitoring, and tracking target over a specific region [7, 8]. Research on sensor networks was originally motivated by military applications. However, the availability of low-cost sensors and communication networks has resulted in the development of many other potential applications. Nowadays, it is often referred to as a Wireless Sensor Network (WSN) according to the development of the wireless communication technologies and the increment of its importance. WSN has been enabled by the availability, particularly in recent years, of sensors that are smaller, cheaper, and intelligent [9]. Sensor nodes are places on which the sensors are located, and they are scattered in a sensor field as shown in Fig. 1.2. Sensor nodes collect data about environment and route data back to the sink. The sink communicates with a user's task manager node through Internet or satellite.

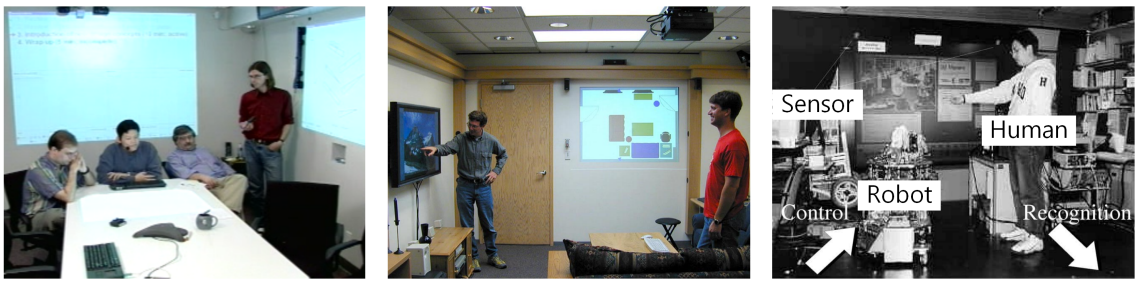
A concept of Ubiquitous Computing (UbiComp) was proposed by W. Mark, and UbiComp enhances computer use by making many computers available throughout the physical environment, while making them effectively invisible to the user [10, 11]. In the long term, UbiComp can pervade all spheres of life, and it will be able to change human life. For example, it will increase comfort in the private home area with improved energy efficiency; intelligent vehicles may make roads safer; adaptive

personal assistance systems could raise work productivity in the office; and in the medical field, implantable sensors and micro-computers monitor the health of the user. UbiComp has several characters as follows [12–14].

- Decentralization or modularity of the systems and their comprehensive networking
- Embedding of the computer hardware and software in other equipment and objects of daily use
- Mobile support for the user through information services anywhere and any time
- Context awareness and adaptation of the system to current information requirements
- Automatic recognition and autonomous processing of repetitive tasks without user intervention

Currently, there are many and various applications and research of UbiComp, such as, retailing [15,16], industrial production and material management [17,18], personal identification and authentication [19], health care [20,21], and mobility and transport [22,23].

Intelligent Environment (IE) is an interactive space with a user by embedded systems and information & communication technologies, and IE brings computation into the physical world and enhances occupants experiences [32]. The development of IEs



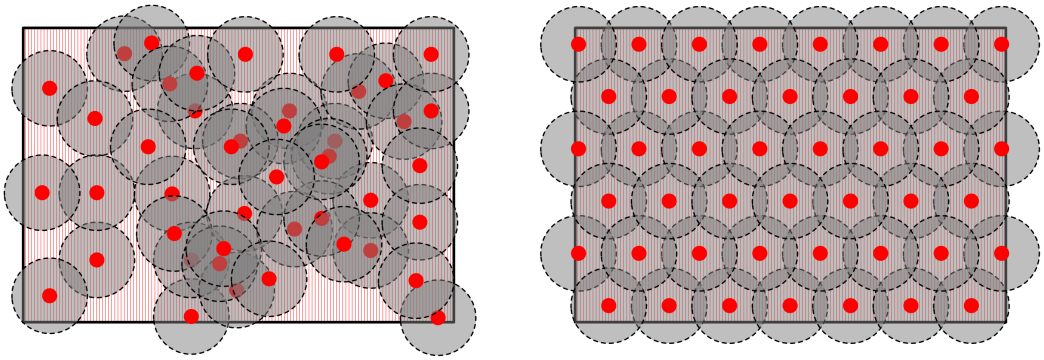
(a) Oxygen project [24–26] (b) EasyLiving project [27–29] (c) Intelligent Space [1,30,31]

Fig. 1.3 Examples of Intelligent Environments

is affected by the relative maturity of several areas of Computer Science, such as Artificial Intelligent, Human-Computer Interaction, Sensors & Actuators, Networks & Middleware, and Ubiquitous Computing. The studies on IEs have been performed with various names according to research group (e.g., Oxygen project [24–26], EasyLiving project [27–29], Intelligent Space [1,30,31]), and they are shown in Fig. 1.3.

Studies, as mentioned above, have the common characteristics as follows; countless small sensor devices, inter-communication networks, distributed system, and service provision by the cooperation of devices. These common characteristics cause the same problem that is related with the arrangement of devices. All devices have a limited usable coverage. Therefore, deploying devices in consideration of the devices' usable coverage is important for the successful service provision. In the previous research, various approaches for appropriate deployment of devices were proposed.

The previous studies can be divided by various conditions. First, as it is shown in Fig. 1.1, they can divided into two categories by the form of the device's modeling. In [33–36], the device has a circular coverage, and this indicates that the device has an omni-directional view. However, in real world, most devices have a viewing direction and an angle of view. Therefore, in research of the other category, the coverage of device is modeled as a form of a sector [4,5,37].



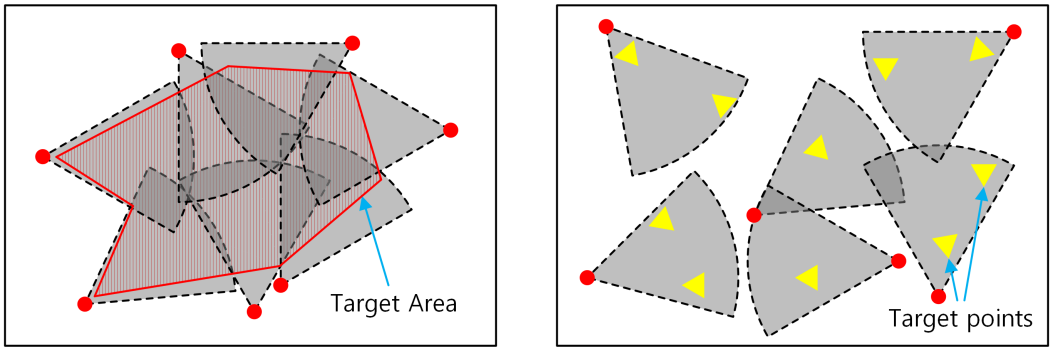
(a) Initial random deployment (b) After rearrangement

Fig. 1.4 Examples of Initial and Rearranged Deployment using Movement-Assisted Sensors

The previous research can be divided into two categories by the device's mobility. In typical research, the devices are fixed on the space, then they have fixed coverage. In the other studies, movable devices, called movement-assisted sensors, are used. These studies are associated with the research field on swarm robotics. In these research, the devices are randomly scattered on the space initially, then they moves their positions to be arranged evenly [34, 38–41]. Fig. 1.4 shows examples of the initial situation and after rearrangement situation in these research.

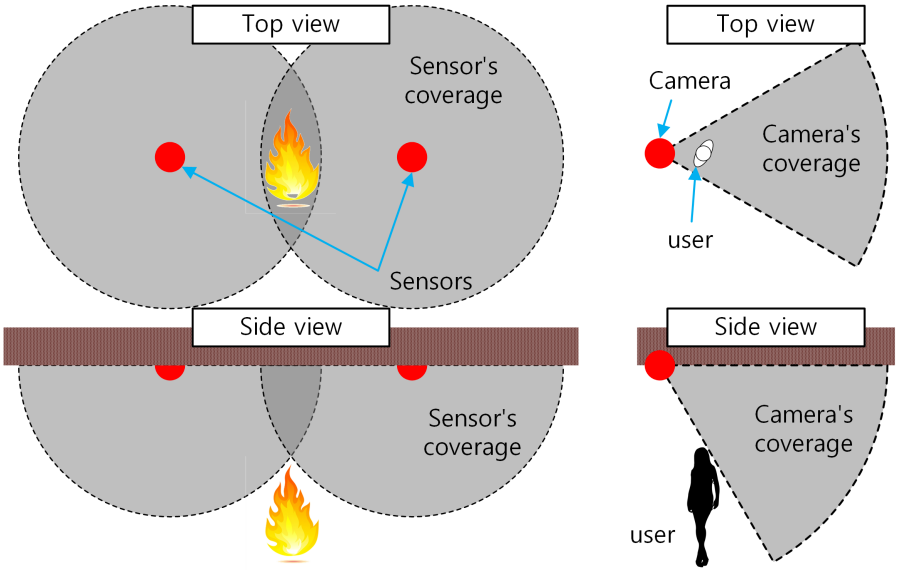
The modeling of the target is also significant criterion of classification. Fig. 1.5 shows the arrangement results of two types of research. The target in Fig. 1.5(a) is modeled as an area, and the coverages of all devices should cover whole area of target [6, 42, 43]. This methods are useful to arrange sensors in some applications, such as a fire alarm system, a surveillance system of art gallery, and a target tracking system. In this systems, usually, the target's position cannot be designated. Therefore, the devices should cover whole area for providing perfect service. In Fig. 1.5(b), the target is modeled as a point, thus, the devices are arranged to cover only the target points [5, 35, 44]. In these research, there are lots of targets and they are fixed to the space. This method gives more efficient solution when the targets are not moving.

Next, the previous research also can be classified as two categories according to device's fixable location. In some research, the device can be installed to any place in



(a) Arrangement for covering target area (b) Arrangement for covering target points

Fig. 1.5 Examples of Device Arrangements according to the types of Target



(a) Blind spot in fire alarm system

(b) Blind spot in surveillance camera system

Fig. 1.6 Examples of blind spot in three dimensional space

the space without a spatial limitation. Typically, the approaches based on the AGP solution are classified as this [3, 45]. The devices are placed to their best position to cover the targets. In the other research, the device's location is constrained as a placement site [5, 33, 46]. Through this method, the computation time to find a best arrangement can be reduced. In [33, 46], the placement sites are arranged as a form of grid. On the other hand, the placement sites in [5] are located according to the status of actual environment.

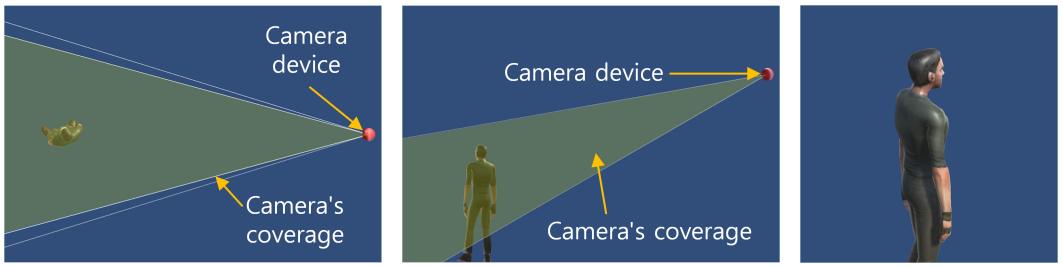
Lastly, the previous studies can be classified according to the dimension of space in the algorithm. In many research, the space is modeled as two dimensional map in the top view, such as floor plan, and the sensor deployment algorithm of these research finds the arrangement of devices that the devices' coverage can cover the floor plan. However, this method can cause blind spots in actual cases. Fig. 1.6 shows examples of blind spot. To solve this problem, some research modeled the

environment as three dimensional space, and device's coverage was also modeled in three dimension [43, 47, 48].

1.2 Problem statement

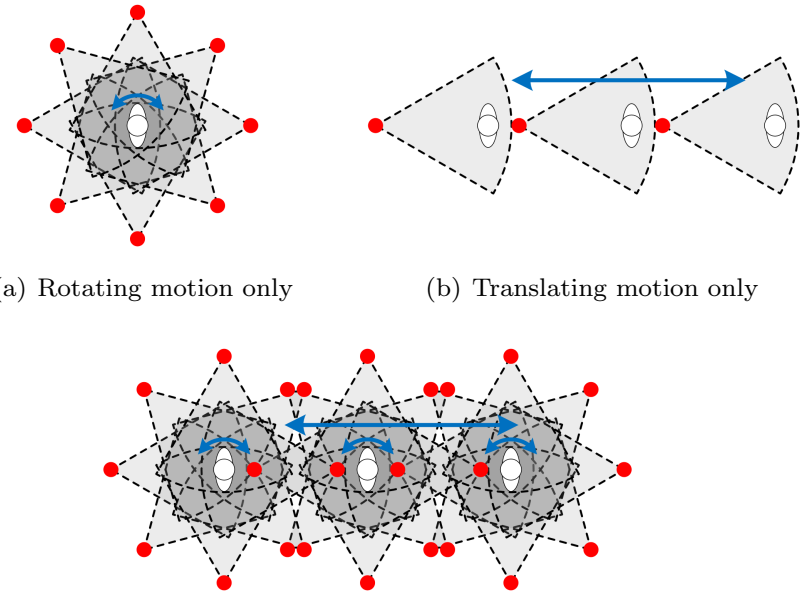
The previous section introduced the brief outline of related research fields and the approaches to find suitable arrangement of devices in previous studies. Most previous approaches aim to find the device deployment for simple applications, such as a fire alarm system, a surveillance camera system, and target tracking system. However, nowadays, users request more higher-level services at Sensor Network, UbiComp, and IEs. The personalized information provision, health monitoring for elderly people, life log, etc. are representative requested services. The more detailed information than former services is required when these services are provided. For example, the user tracking system can make accurate results, when the cameras are placed where enables to cover the whole area by their field of view. However, the success of the face recognition for the personalized service provision cannot be achieved by the arrangement proposed in previous research. The simple and obvious failure example is shown in Fig. 1.7. The user in Fig. 1.7 is located inside of coverage of camera, but the face recognition is failed. This result is caused by the target's modeling that does not contain the information of target's viewing direction.

In addition, there is another problem in the determination of the suitable arrange-



(a) Top view of space (b) Side view of space (c) Captured image from camera device

Fig. 1.7 Example of failure situation of face recognition



(a) Rotating motion only

(b) Translating motion only

(c) Rotating and translating motion

Fig. 1.8 Camera deployment examples for face recognition in three cases of target's motion

ment of devices. As shown in Fig. 1.7, the suitable location of device is decided by not only the target's position but also the target's direction. In real-life situation, the target can be moved to any place in the space freely. For example, in the case of Fig. 1.7, the user can move freely in the room. This indicates that the countless numbers of devices are required to cover even though it is a small room. Fig. 1.8 shows the examples of a device deployment in three status of target's motion.

Even if the devices are deployed with consideration for the above-mentioned problem, the obtained results from the devices may not always be the best. The best results are achieved when the devices are located at the optimal position. This indicates that devices should be installed everywhere; however, this is not feasible. In this thesis, this problem is called as device's spatial constraint problem.

The many conventional studies assumed a specific situation to evade this problem.

Then, according to the assumed situation, the devices are arranged to appropriate location. For example, when the face recognition method is developed, the researchers assumed that the user located in front of the camera with a certain direction. These assumptions result in improvements of applications. However, it also causes inconvenience when the application are used in the real-life situation.

This thesis proposes a new approach to solve the device's spatial constraint problem. The proposed approach is based on the mobility of devices. In the proposed environment, all devices have mobility. Thus, the devices can be rearranged to their best locations according to the changed spatial situation.

Chapter 2

Reconfigurable Intelligent Space

In this thesis, Reconfigurable Intelligent Space (R+iSpace) is proposed to solve the device's spatial constraint problem. The research on R+iSpace is based on Intelligent Space (iSpace) that is one of the famous IE. This chapter discusses iSpace briefly, then the concept of R+iSpace will be introduced. Lastly, the essential research topics to implement R+iSpace is described.

2.1 iSpace and DIND

iSpace was first proposed by the Hashimoto Laboratory at the University of Tokyo since 1996 [1, 30, 31]. iSpace is a system that provides appropriate services to users in the space by using various sensor devices and agents. Fig. 2.1 illustrates a conceptual diagram of iSpace. As shown in Fig. 2.1, lots of Distributed Intelligent Networked Devices (DINDs) are installed on the ceilings and walls of iSpace. DIND will be explained in detail after the explanation of iSpace. DINDs receive the spatial information through the connected sensor, then they make the higher-level information by recognition processing. Then, it provides a service to the user in it informationally and physically by the control of agents. iSpace is a platform to which desultory technologies are installed. The employment of the desultory technologies in iSpace led to reduction of effort for completing the system when a new hardware or software is developed. Moreover, since the same platform is adopted, performance comparisons between different hardware or software module become possible. Hashimoto Laboratory defined iSpace's several properties as follows; Modular, Scalable, Integration,

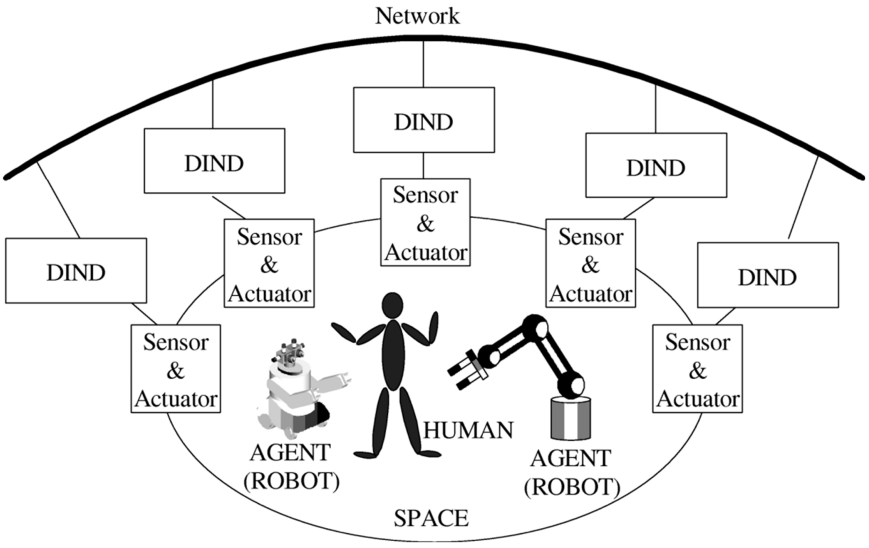


Fig. 2.1 Conceptual diagram of Intelligent Space [1]

Realizable, Low cost, and Easy configuration.

In order to implement iSpace, lots of sensors are installed to the space to achieve the spatial information. Moreover, the processors (computers) are needed to process the achieved raw data from sensors. Then, the agent devices and agent robots can be operated remotely based on the recognized information by the processing. This indicates that a device for the communication with agents is required. DIND comprises these three fundamental elements. Fig. 2.2 shows the fundamental structure of a DIND. By installing this DIND into the whole space, it can recognize objects inside the space easily, and it can perform information exchange and sharing by mutual communication through the network. Furthermore, since such functions are obtained by just attaching DINDs, the space where people live is altered into an intelligent environment without a great deal of effort. Nowadays, according to the development of single board computer, such as Raspberry Pi and wireless communication technology, DINDs can be minimized and iSpace can be realized.

iSpace has a device's spatial constraint problem similarly to the other IEs. The reason for the occurrence of the problem is that the device is fixed in the location that device was installed initially. If devices can rearrange to their optimal locations

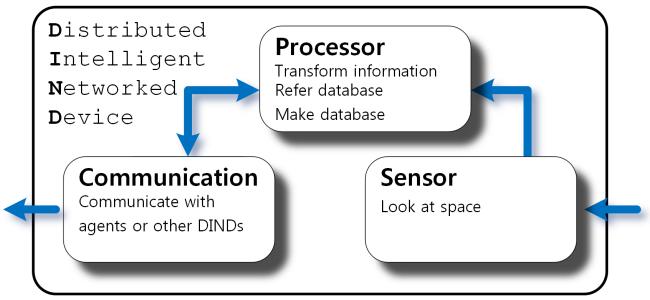


Fig. 2.2 Fundamental structure of DIND [1]

according to the current spatial situation, iSpace can provide appropriate services successfully. J.-H. Lee and his research group proposed Ubiquitous Display (UD) that is a mobile robot on which a projector is mounted [49–51]. This method can be a solution for the problem. UD can provide visual information through the projector, and it can rearrange to its best location according to a surrounding situation. Fig. 2.3 shows the detailed structure of UD. Through this mechanism, UD can provide service successfully, even though user moves freely.

However, there are problems to apply this method to all devices in new iSpace. First, iSpace comprises lots of devices, and most devices should be mounted on a mobile robot to provide service successfully. In addition, typically, sensor devices in iSpace are installed on high places of walls or a ceiling because devices installed on the places can have more extended viewing area without occlusions. This indicates that the size of the mobile robot should be relatively large for its stability. As a result, lots of relatively large mobile robot are placed in iSpace, and they should be movable freely in the space. Obviously, it makes human in same space uncomfortably. The movements of human in iSpace will be disturbed by mobile robots.

Using the mobile robots has the other problem. A lot of application requires the target's location with high accuracy to provide service without failures. To obtain the accurate target's position, the accurate locations of sensor devices are an essential prerequisite because the target's position is obtained based on the positions of sensor devices. All devices in new iSpace are mounted on mobile robots, therefore the devices' locations also are changed according to the movements of the mobile

robots. The mobile robots should compute an algorithm, such as Simultaneous Localization And Map-building (SLAM) that requires high computing performance, to obtain their current positions. Moreover, the algorithm also requires additional sensors, e.g., odometry sensors, sonar sensors, laser range finders, and cameras. As a result, the economic problem will occur when new iSpace that comprises the movable devices is implemented using mobile robots.

2.2 R+iSpace and DIND's mobility

As it is shown in the case of UD, the mobility of device can solve the device's spatial constraint problem. Proposed new iSpace in this thesis, Reconfigurable Intelligent Space (R+iSpace), can be achieved by the movable DINDs.

Before the detailed explanation of R+iSpace, changed concept of DIND is described. In the beginning of iSpace, DIND indicates a sensor device with a processor and a communication function, as it is shown in Fig. 2.2. DINDs controls agent devices or agent robots to provide appropriate services. In other words, the agents, that does not have a processor to decide own behavior, are subordinative devices of DINDs. However, according to the development of various service, an information achieved from a DIND are used in various services at the same time. Moreover, according to the development of recognition technology, the computational complexity that a DIND should proceed was increased. To reduce the computational complexity, the agent devices equip a processor to decide own behavior. As a result, the concept of DIND

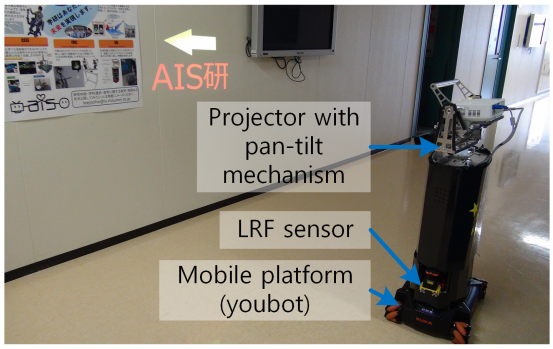
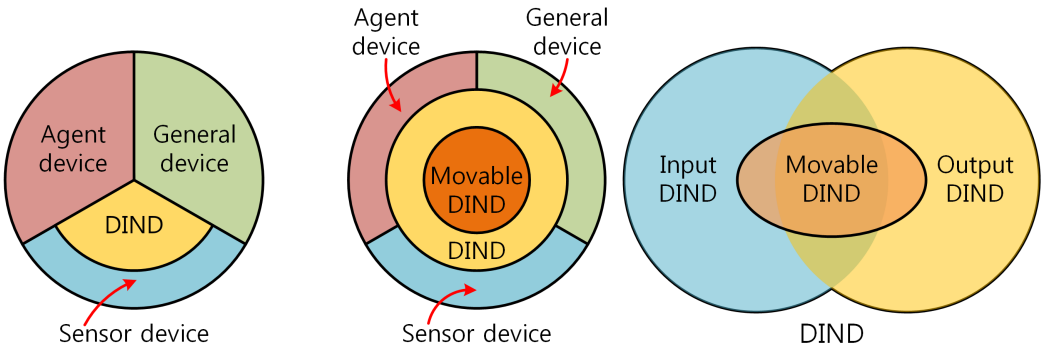


Fig. 2.3 Ubiquitous Display



(a) In iSpace (b) In R+iSpace

Fig. 2.4 Categories of devices and DIND's area in iSpace and R+iSpace

was extended. Fig. 2.4 shows the brief classifications of devices in conventional iSpace and R+iSpace. As it was explained before, the agent devices in Fig. 2.4 indicate the dependent devices, such as an alarm of fire detection system. The general devices indicate devices, such as electronics, illuminations, embellishments, furniture, etc. As it is shown in Fig. 2.4(a), DINDs in iSpace belong to the sensor category, because DINDs are made by appending a processor that has communication function on sensor device. However, in case of R+iSpace, all kinds of devices not only agent devices but also general devices can be DINDs, and DINDs can be categorized into two parts according to the role of connected device; input device and output device. Typically, the sensor devices are categorized as input devices, and agent devices (e.g., projector, speaker) will be categorized as output devices. Most of general devices, such as interior designs also will be categorized as output devices because they are placed in the space to be shown. In some specific cases, for example, a DIND performs the higher level recognizing process based on the other DINDs' recognition results, the DIND does not have any device. The fundamental structure if DIND in R+iSpace is shown in Fig. 2.5. The blocks in Fig. 2.5 that have dashed border can be removed according to DIND's objective. Especially, the mobile module block that has green color dashed border is the main factor to divide into DIND and movable DIND. The green color task lists in processor block are performed in movable DIND.

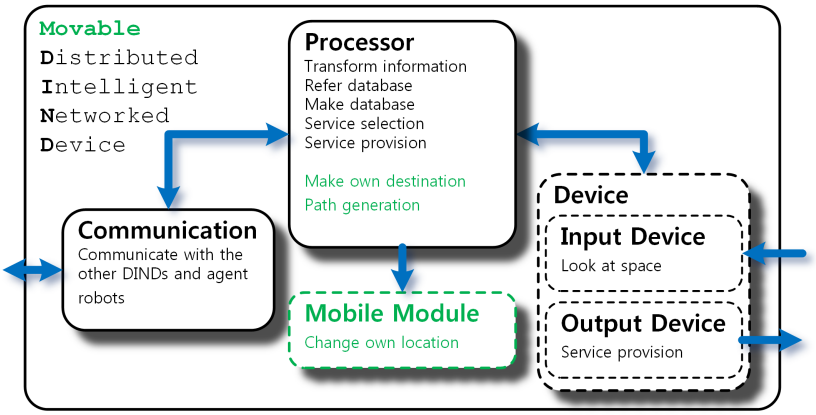


Fig. 2.5 Fundamental structure of DIND in R+iSpace

As it was mentioned above, supplement the mobility to DINDs is an excellent solution to solve the device's spatial constraint problem. To avoid the drawbacks that occur in UD, several requirements for R+iSpace were defined as follows.

- Movable DINDs in R+iSpace can automatically rearrange their position according to a spatial situation.
- Movable DINDs should avoid interference with the movement of human users.
- The estimation or measurement of rearranged positions of Movable DIND should be obtained easily and with high accuracy.
- Movable DIND should not require additional energy to maintain its current state.
- Movable DIND's mobility and adhering method should be stable and robust against external environments include a disturbance.
- To change from iSpace to R+iSpace should be simple, and it should be reasonable in economic aspect.

The first requirement is defined to solve the device's spatial constraint problem that the conventional iSpace has had. The other requirements are suggested to increase the usefulness and feasibility of R+iSpace. If a user feel uncomfortable, the new iSpace is meaningless. To avoid the problems that occurs when using typical mobile robot, such as UD, the second and third requirements are defined. DINDs should

maintain current location when they are not used, and this situation often occurs in real-life. To increase the energy efficiency of R+iSpace, it is important to reduce an energy consumption while DINDs maintain current location. Therefore, the fourth requirement is defined. The fifth requirement is defined for the stability of the moving and adhering method of movable DIND, and the last requirement is defined for the cost problem.

Fig. 2.6 shows the conceptual diagram of R+iSpace. The conceptual diagram of R+iSpace is similar with iSpace's it, because R+iSpace is based on iSpace. The major differences from iSpace are movable DINDs and fields. A field indicates the modified wall or ceiling for the movement of movable DINDs. As shown in Fig. 2.5, movable DIND has a mechanical structure for its mobility, called Mobile Module (MoMo), and MoMo has a form of wall climbing robot. Typically, the moving area of a wall climbing robot is separated from the human moving area. Therefore, the existence of movable DINDs does not interfere with the movement of human user. Thus, the second requirement of R+iSpace can be satisfied easily. The fields indicate modified walls and ceilings for the movement and adhesion of MoMo. Typically, a wall climbing robot can move on a specific plane that is different according to the adhering method

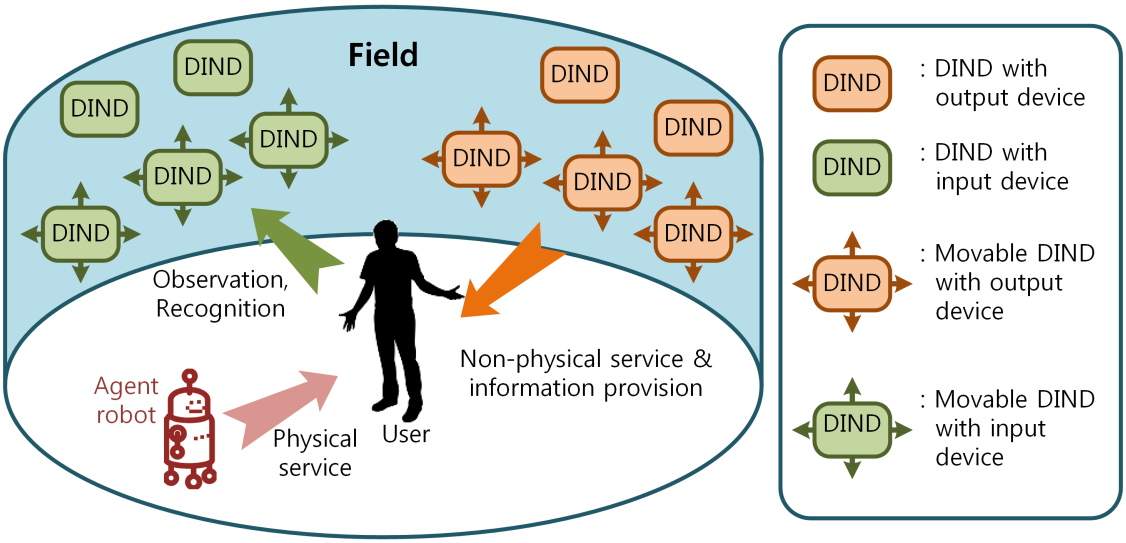


Fig. 2.6 Conceptual diagram of R+iSpace

Fig. 2.7 An example situation of R+iSpace

of the wall climbing robot. In other words, the structure of field depends on the structure of MoMo.

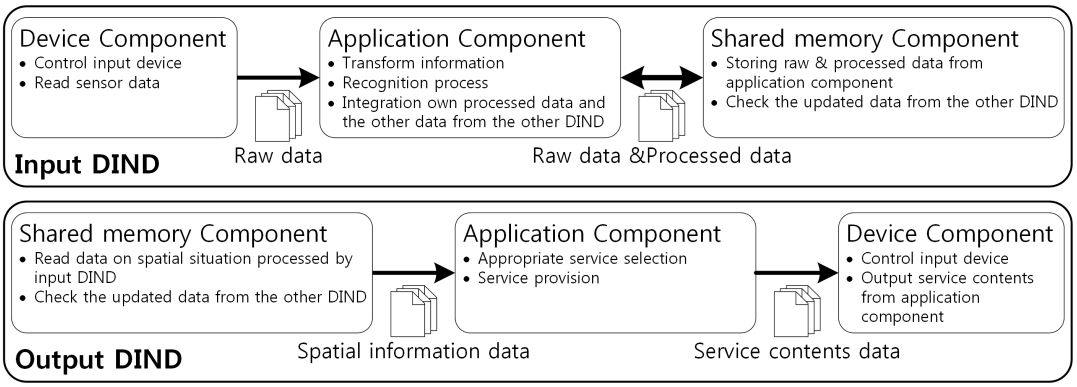
Fig. 2.7 shows an example image of R+iSpace. As it was mentioned, the electronics, illuminations, and interior designs can be DINDs in R+iSpace. In this example, the television, wall clock, and illuminations became movable DINDs. To be general devices to movable DINDs can help not only the change of the interior decoration scheme but also the increment of the performance of provided service. For example, there is an interesting research project, IllumiRoom of Microsoft that augments the area surrounding a television with projected visualizations to enhance traditional gaming experiences [52]. In this research, the projector extend the field of view that is limited by the display size of a television. Through the spatial augmented reality, the system can provides enhanced gaming experiences. However, if there are many colorful objects around the television, the effect of the spatial augmented reality will be reduced. If the objects are mounted on MoMo, they can be rearranged when the spatial augmented reality is started. As a result, the user can obtain the enhanced gaming experiences successfully.

2.3 Process flow of R+iSpace

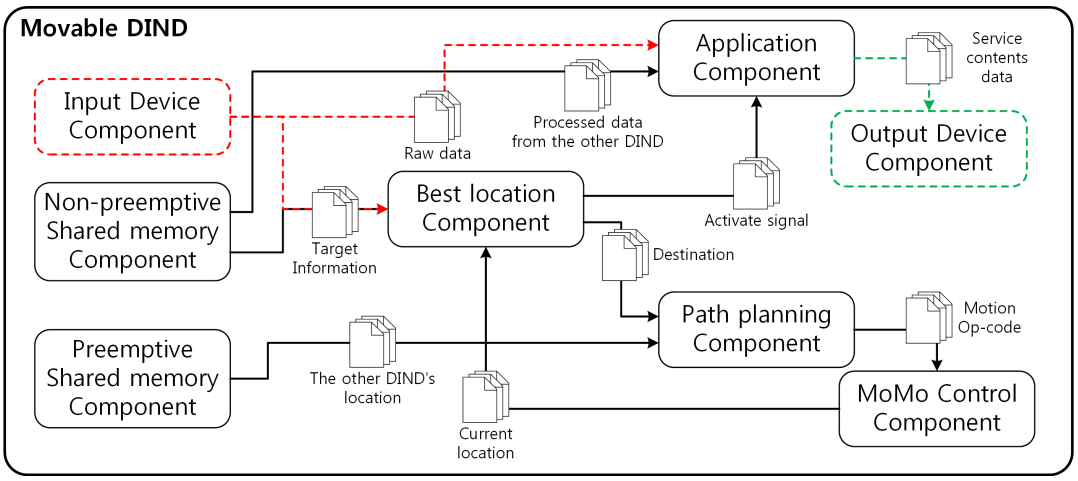
R+iSpace can be achieved by movable DINDs. However, as above-mentioned, movable DINDs should be able to rearrange to suitable location according to spatial situation

to improve the overall performance of R+iSpace than conventional iSpace. This indicates that the additional processes are required for movable DINDs. Fig. 2.8 show the process flow of DIND and movable DIND in R+iSpace. DINDs with input device, hereafter referred to as input DINDs, receive the spatial information from connected input device and the other DINDs, then received information is processed by the processor. Lastly, the results of processing are shared to the other DINDs and agent robots. DINDs with output device, hereafter referred to as output DINDs, receive the spatial information from the other input DINDs. Then, it provides appropriate service according to the spatial situation. As it is shown in above description, the process flow of DIND is simple; it is composed of three components, i.e., Shared Memory component, Device component, and Application component. Device component controls the connected device; receive raw data from the input device, provide information to user through output device. In the application component, the processor recognizes spatial situation or service provision. Face recognition, gesture recognition, target tracking, and visual auditory information provision are representative programs that are executed in application component. Shared memory component stores the latest information of spatial situation from all DINDs. When the new information is achieved from input devices, the shared memory component informs the latest information to the other DINDs and the agent robots.

As it is shown in Fig. 2.8, there are several additional components in movable DINDs, e.g., best location component, path planning component, MoMo control component, and preemptive shared memory component. The process flow of movable DIND is as follow. At the initial state, best location component calculates or obtains the target's location and direction. Then, the processor calculates the best location of device for the service provision. The detailed information on the definition of best location in this thesis and determination method is described in Chapter 4. After the calculation of the best location, if current location of movable DIND is same to the best location, processor performs the same process with the DIND's process flow; executes sensor component, (nonpreemptive) shared memory component, and processing (application) component. If the connected device is not a sensor device, the sensor component can be omitted. If it is different, the information of the destination, which is a suitable location for service provision, is transferred to path planning component. In the path planning component, the fastest path to reach the destination is calculated. To avoid the deadlock situation and collision between movable DINDs, the preemptive shared memory component is used. The preemptive shared memory



(a) Process flow of DIND



(b) Process flow of movable DIND

Fig. 2.8 Process flows of DIND and movable DIND in R+iSpace

component contains the information on all movable DIND's position and destination. The detailed path generation algorithm will be discussed in Chapter 5. After the path generation, the operation code of motion to follow the generated path is generated and transferred to the MoMo control component. In the MoMo control component, the actual motion of movable DIND is generated and the current location of movable DIND is updated. The updated movable DIND's location is stored to the preemptive shared memory.

In this chapter, the concept, requirements, and process flow of R+iSpace was discussed. R+iSpace is proposed to solve the device's constraint problem, and the proposed approach is the device's mobility. This thesis defines several requirements of R+iSpace for its usefulness and feasibility, and it can be achieved by the appropriate mechanical structure of MoMo. However, R+iSpace cannot be implemented only the appropriate mechanical structure of MoMo. There are two essential research topics in software parts to complete the implementation of R+iSpace. The first one is the algorithm on the determination of movable DIND's suitable location, and the other one is the algorithm on the conflict free path generation to reach to the suitable location without deadlock situation. These three topics are essential for the implementation of R+iSpace. This thesis addresses in order for these three topics.

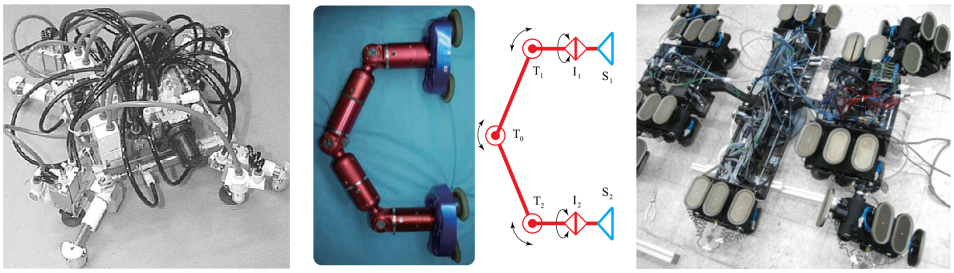
Chapter 3

Mechanical Structure of MoMo and field

In this chapter, the first research topic, the mechanical structure of MoMo and field, is addressed. As it was mentioned above, the mechanical structure of MoMo and field should be designed to satisfy the requirements of R+iSpace. First, in Section 3.1, the survey results of related works is introduced to find a suitable mechanical structure. There are various wall climbing robots in previous research, and they have various adhering methods. The merits and demerits of adhering methods that are used in conventional wall climbing robot will be described. After the description on related works, the employed adhering method in prototypes of MoMo are introduced in detail. Then, the detailed mechanical structures and movement schemes of prototypes of MoMo are illustrated. To verify that the latest prototype of MoMo satisfies the requirements of R+iSpace, three experiments were performed. The experimental methods and experiment results are described in Section 3.4. Lastly, this chapter will conclude with the discussion of the experiment results.

3.1 Related works

There are various adhesion methods for conventional wall climbing robots: suction force, sticky materials, magnetic force, a mechanical combination, etc. Fig. 3.1, Fig. 3.2, Fig. 3.3, Fig. 3.4, and Fig. 3.5 shows representative examples of conven-



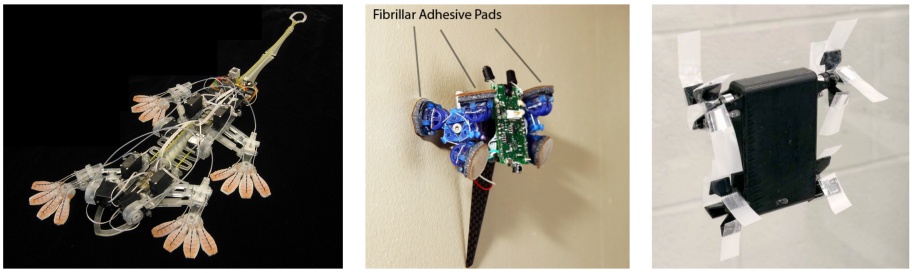
(a) Planar Walker [53] (b) W-Climbot [54] (c) MultiTrack [55]

Fig. 3.1 Conventional wall climbing robots using a suction force

tional wall climbing robots.

The wall climbing robot of Fig. 3.1(a) is named as 'Planar Walker' and it was proposed by I.-M. Cheng et al [53, 56]. It used four gripper units to adhere to a wall and four pneumatic pistons for its gait locomotion. A gripper unit is mounted on the bottom of sliding unit and the piston is located between two sliding units. A gripper unit comprises a vacuum pad, an air suction filter, and a lifter cylinder to provide the vacuum suction force to the surface. The lifter cylinder lifts up the vacuum pad when the gripper unit is moved. Planar Walker's gait locomotion is designed based on the normal inchworm locomotion. Planar Walker can move or rotate by changing the position of gripper unit in regular sequence.

Fig. 3.1(b) shows the biped wall climbing robot named as 'W-Climbot', and it was proposed by H. Zhang et al [54, 57]. As it is shown in Fig. 3.1(b), it has two suction modules on its feet. The form of suction module is similar to equilateral triangle, and three suction pads are located on the edges of the triangle. A suction module has three ultrasonic sensors to detect the incline angle and the distance between the suction cups and the target surface. As it is shown in the kinematic model of W-Climbot, this robot has five actuators for its movement. This robot has three types of movement schemes, i.e., inchworm gait, swinging-around gait, and flipping-over gait. The inchworm gait is composed of contraction and extension of its body by using the middle three joints. The swinging-around gait is generated by the rotating motion of adjacent joints to suction modules. The flipping-over gait is designed to overcome the obstacles on the surface.



(a) Stickybot [58] (b) Waalbot [59, 60] (c) Robot by [61]

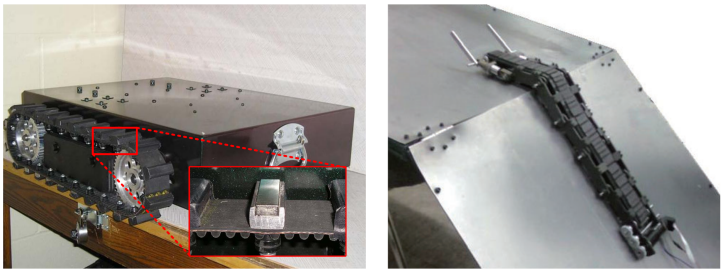
Fig. 3.2 Conventional wall climbing robots using a sticky material

The last example using a vacuum suction force is shown in Fig. 3.1(c). This robot is named as 'MultiTrack', and it was proposed by G. Lee et al [55]. MultiTrack has specially designed ten caterpillar tracks, and six suction units, that produce the adhesion force, are installed on a caterpillar track. This robot is composed of three parts, and this mechanism enables the transition from floor to vertical surface and the climb over a thin wall. The movement scheme using caterpillar tracks can increase movement speed than using leg type's locomotions.

The wall climbing robots using suction force usually can move freely on wall surfaces, and heavy devices can be mounted on the robot when the suction force is sufficiently strong. However, this mechanism has several drawbacks for MoMo. Robots that employ this mechanism require a continuous energy supply to maintain the robot's adhered status. Moreover, estimation of a robot's position is not easy; without additional sensors, this estimation includes errors, and these errors accumulate according to increments in moving distance.

The Fig. 3.2 shows conventional wall climbing robots that uses a sticky materials. First, the 'Stickybot' (Fig. 3.2(a)) is a bio-inspired climbing robot, and it was proposed by S. Kim et al [58]. Stickybot has adhesive pads on its feet, and the adhesive pads have an anisotropic structure that permits controllable adhesion. Therefore, the Sticky Bot can move freely on the smooth surfaces of walls. The locomotion of Stickybot is very similar to the gecko. Two legs of diagonal position are rearranged into next position simultaneously.

The 'Waalbot' that is shown in Fig. 3.2(b) was proposed M.P. Murphy et al [59,60].



(a) Crawler [62] (b) Robot by [63]

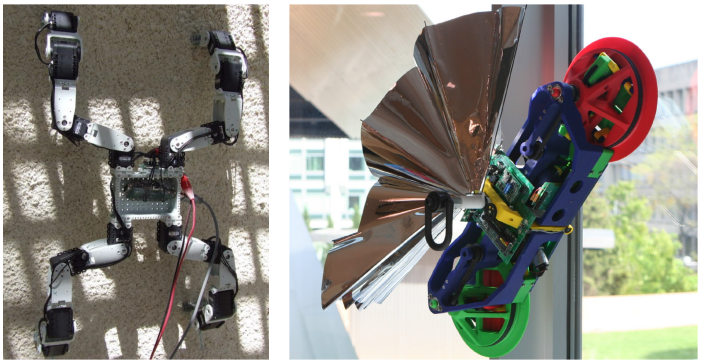
Fig. 3.3 Conventional wall climbing robots using a magnetic force

Waalbot is actuated by two motors, each controlling a three-footed wheel that is named Tri-Foot. Each foot is connected to wheel plane by a ankle joint, and it has a foot pad on the bottom. The foot pad is composed of fibrillar adhesive material. Waalbot can move on the wall surface by rotating the Tri-foot, its movement scheme is similar to usual mobile robot.

K. Daltorio et al., proposed a wall climbing robot that is made by the modification of the Mini-Whegs^{TM} [61]. This robot has four wheel-legs that are made by the attachment of the Scotch tapes into four directions. This robots locomotion is also similar to usual mobile robot.

The wall climbing robots using sticky material also can move freely on wall surfaces, but this mechanism also has similar drawbacks to wall climbing robot using suction force. First, this mechanism also has a difficulty to estimate the robot's current position, and these robots requires a smooth surface without any dust. Usually, the adhesive strength of the adhesive pads is vulnerable to dust and humidity. Moreover, the adhesive strength will be decreased gradually while it moves.

The wall climbing robots in Fig. 3.3 can adhere to the wall by magnetic force. The 'Crawler' that is shown in Fig. 3.3(a), was designed for the movement on the oil tank [62]. Crawler employs permanent magnets to adhere to the wall, and the magnets are attached on the caterpillar for robot's movement. Through this simple structure, Crawler can move on the wall freely. The robot that was proposed by G. Lee et al., (Fig. 3.3(b)) has long track-wheel with segmented polymer magnet [63]. This robot has five compliant rotational joint, and they enables the transition between



(a) Clibo [64] (b) Shady [65]

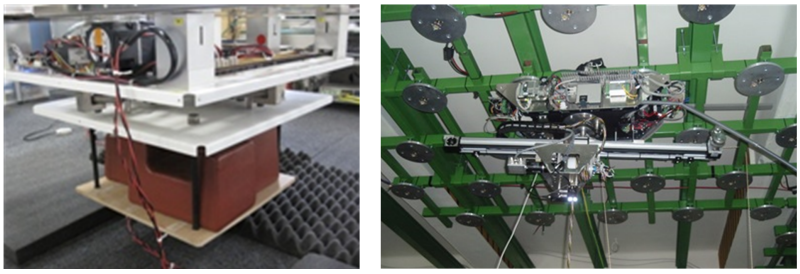
Fig. 3.4 Conventional wall climbing robots using claws or grippers

two walls.

The wall climbing robots using permanent magnets do not require a continuous energy supply to maintain the robot's adhered status, and the permanent magnets can ensure the robot's adhered status for extended periods. However, this mechanism also has the position estimation problem and the field's material is limited to ferromagnetic substance material. Therefore, this mechanism is not suitable for a MoMo.

Fig. 3.4 shows the conventional wall climbing robots that employed the claws or gripper mechanism. The 'Clibo' (Fig. 3.4(a)), that was proposed by A. Sintov et al., has specially designed claws on the tip of its feet [64]. Clibo is designed for movement on the rough surfaces, such as stucco walls. The robot's kinematics and motion, is a combination between mimicking a technique commonly used in rock-climbing using four limbs to climb and a method used by cats to climb on trees with their claws. It uses four legs, each with four-degrees-of-freedom and specially designed claws attached to each leg that enable it to maneuver itself up the wall and to move in any direction.

The 'Shady' that is shown in Fig. 3.4(b) was proposed by M. Vona [65]. Shady is a bilaterally-symmetric mechanism with two rotating grippers. Shady's locomotion comprises as a series of grip-rotate-grip steps. Shady's grippers are symmetric 6-bar linkage mechanisms situated in rotating barrels. When the gripper is gripping the window bar completely, the two four-bar linkages in the gripper are both in



(a) HangBot [66, 67] (b) Acroboter [68]

Fig. 3.5 Conventional wall climbing robots using mechanical equipments

singularity.

The biggest problem of claws is the reliability of adhesion status of robot. This mechanism is affected by the external force or disturbance easily. Moreover, typically, the available payload is small. The grip-based adhesion is very safe since even a loss of power does not necessarily lead to a drop-off of the system. On the other hand, gripper type wall climbing robots are not very fast. Furthermore, their payload is low compared to robots using magnetic or pneumatic attraction.

Fig. 3.5 shows the wall climbing robots that employ the mechanical equipment to adhere to the walls of ceilings. The 'HangBot' (Fig. 3.5(a)), that was proposed by R. Fukui et al., requires a special wall that has many holes aligned to be used in combination with the robot [66,67]. The 'Acroboter' (Fig. 3.5(b)), that was proposed by G. Stepan et al., requires the installation of numerous anchor points in a ceiling [68]. Such use of mechanical equipment has two merits: a wall climbing robot using this method does not require additional energy to maintain its current state. The current position of the robot is determined by pre-installed positions of mechanical equipment, and it allows the current position of the robot to be easily obtained without additional sensors. Using this method, a MoMo can readily satisfy the necessary conditions.

In this section, the adhesion method and movement scheme of the conventional wall climbing robots were introduced. Table 3.1 shows the characteristic according to the adhesion methods. As it is shown in Table 3.1, to use the mechanical equipment is most suitable adhesion method for MoMo among described methods. Among the described movement schemes, mechanism using a wheel or a caterpillar is not suitable

Table 3.1 Comparison of characteristics according to the adhesion methods

	Suction force	Magnetic force	Sticky material	Claws and Gripper	Mechanical equipment
Surface roughness	-	+	-	+	+
Energy efficiency	-	+	+	+	+
Robustness on disturbance	+	+	-	-	+
Available payload	+	+	-	-	+
Simplicity of estimating position	-	-	-	-	+

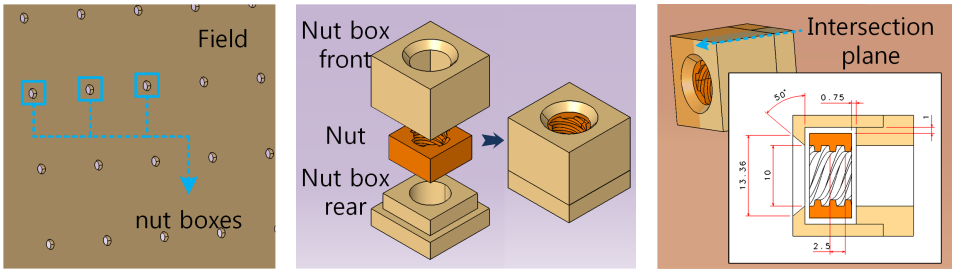
for MoMo because there is a difficulty when estimates robot's current position.

3.2 Adhering mechanism

As it is mentioned in previous section, the proper adhesion method for MoMo is using mechanical equipment. In this section, the adhesion methods that were used in prototypes of MoMo are described; Screw-Nut mechanism, Pin-Lock mechanism, and Advanced Pin-Lock mechanism. The common features of these methods are as follows; High-reliability of adhesion status, Zero energy consumption during adhesion status, Easily obtainable robot's position with high accuracy. The adhesion method of the Pin-Lock mechanism and Advanced Pin-Lock mechanism is basically same. The Advanced Pin-Lock mechanism is designed to solve the problem that occurs in attaching situation.

3.2.1 Screw-Nut mechanism

The screw nails and nuts are usually used in every area to fix an object to the other object. Typically, it is highly stable system because the loosening of the tightened screw nail and nut does not occur in natural situation. Therefore, this mechanism is



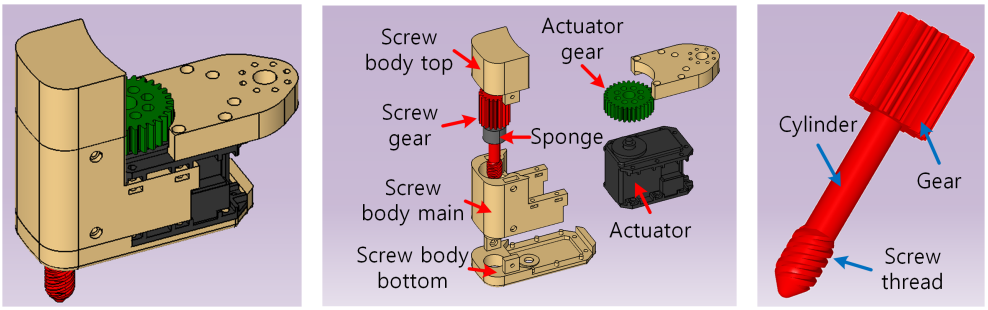
(a) Arrangement example of nut boxes (b) Exploded view of a nut box (c) Cross-sectional view of a nut box

Fig. 3.6 Arrangement example of nut boxes and 3D CAD drawings of a nut box

useful for MoMo.

In the case of wall climbing robot, the screw-nut mechanism can be applied as follow structure. First, a wall climbing robot has screw parts that are controllable, and the nuts are inserted into the walls. Through this mechanism, the wall climbing robot can be adhered to the wall, and this method does not require any additional energy to maintain adhesion status. Moreover, the robot's position is restricted by the combined nut's position. Therefore, the calculation of robot's current position and direction can be achieved easily by the adding a determined offset value to the nut's position. Moreover, its accuracy is high, reliable and robust. The determined offset value is also pre-calculated value by the solving the kinematics of robot. For easy estimation of robot's position, the nut are installed at uniform distance. The wall climbing robot is able to move by changing the fixed nut in a certain order. This leads to the position error of MoMo less than a certain boundary. An example of the arrangement of nuts and the mechanical structure of nut are shown in Fig. 3.6.

As above-mentioned, the screw should be controllable. This indicate that the actuator is needed for rotating the screw. A screw module is the parts that include a controllable screw and actuator. The example of a screw module is shown in Fig. 3.7. As it is shown in Fig. 3.7(b), the actuator gear is attached on the output horn part of actuator. Through the actuator gear, the screw gear can be rotated. The screw body consists of the body top, the body bottom, and the body main. The body top part restricts the translational movement of the screw gear to the upward direction.



(a) Isometric view of a screw module (b) Exploded view of a screw module (c) Detailed view of a screw gear

Fig. 3.7 3D CAD drawings of screw module and screw gear

The body bottom part is in contact with the wall. The role of body main part is to connect the screw gear and the actuator and it has four threads of 10mm pitch. These threads make the translational motion of screw gear from the rotational motion of screw gear. The detailed view of the screw gear is shown in Fig. 3.7(c). As it is shown in Fig. 3.7(c), the screw gear consists of the thread part, the gear part, and the cylinder part. The thread part also has four threads of 10 mm pitch. Through this part, a screw can move out to nut hole and can tighten and loosen with the nut. The tip of the screw is in a sharp shape with a slope of 80 degrees due to ignore the small errors between the axis of screw and the axis of nut hole. As above-mentioned, the gear part is connected with the actuator gear, and the screw gear can receive the rotational torque from actuator gear via this part. The cylinder part is designed to overcome a gap between the robot and the wall. When the robot moves on the wall, some place has the gap. It may make the MoMo unstable. The cylinder part can reduce the gap by the sequence that is shown in Fig. 3.8. Fig. 3.8 shows the screw tightening sequence when there is the gap. As it shown Fig. 3.8, after the screw gear enters into a nut hole, the thread of screw body cannot affect the motion of screw. If the screw gear enters more deeply according to nut's thread, the gap will be removed. The tension of sponge, which is located inside the screw module, assists interlocking of the threads, when the screw is loosened.

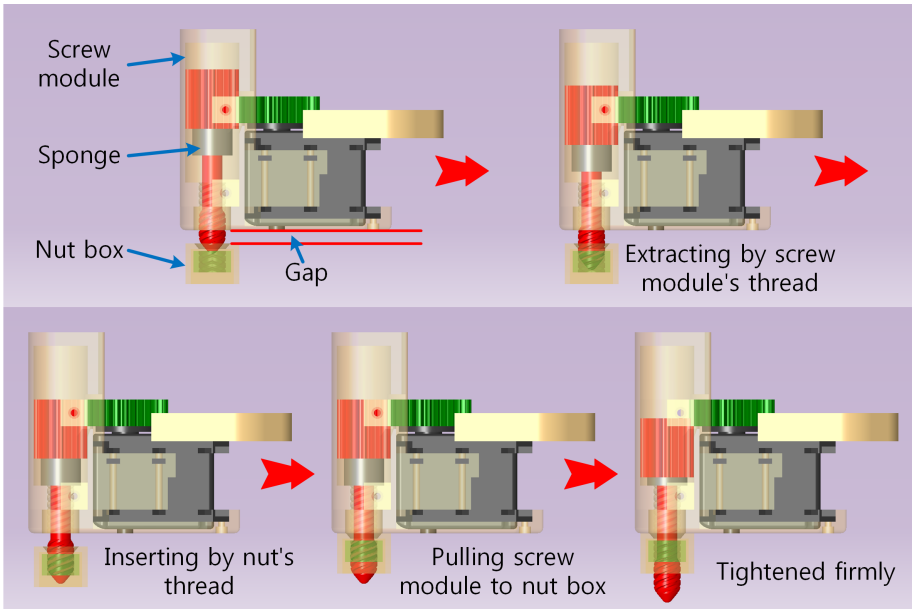
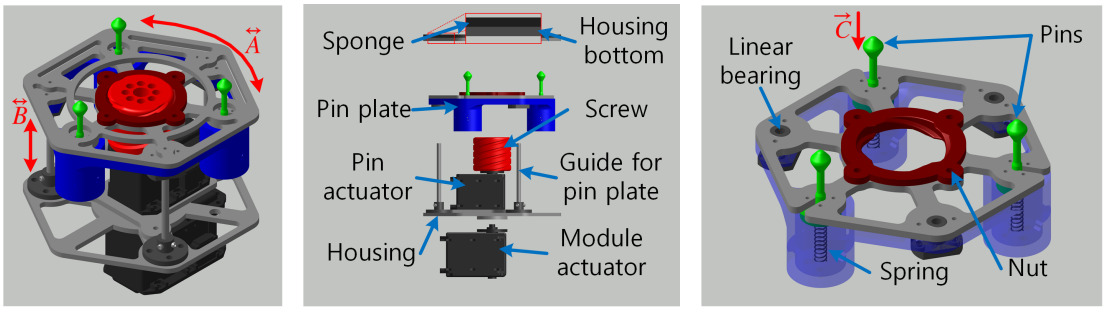


Fig. 3.8 Tightening sequence of screw module with a gap

3.2.2 Pin-Lock mechanism

Sometimes, the abrasion on the screw gears and nuts occurred during the combining when the screw gear is not in alignment with nut. The abrasion makes incomplete combination, and the incomplete combination results a gap between the wall climbing robot and field. This gap occasionally makes a critical position error and a failure of locomotion. To solve this problem, the additional sensors are needed to detect the nut's location. However, it is not good solution economically.

The pin-lock mechanism can solve the problem without additional sensors, and it has more simpler mechanical structure than screw-nut mechanism. The wall climbing robot using the pin-lock mechanism has a several pins, and the field for the robot has a lot of distinctive holes. The detailed mechanical structure of locking module is shown in Fig. 3.9. As it is shown in Fig. 3.9, the screw and nut are moved inside locking module, and the screw is not loosened fully. Therefore, there is no abrasion by the



(a) Isometric view of locking module

(b) Side view and exploded view of locking module

(c) Detailed view of pin plate and pin

Fig. 3.9 3D CAD drawing of locking module

misalignment. A locking module is composed of a module actuator, a pin actuator, a housing, and a pin plate. The module actuator rotates the locking module in the direction of $\overleftrightarrow{A}$ in Fig. 3.9(a). The pin actuator makes the translational movement of the pin plate in the direction of $\overleftrightarrow{B}$ by rotating the connected screw. The housing part functions as not only case of leg, but also the guiding of pin plate. The shafts function as a guide for the pin plate's motion. There is a sponge on the bottom side of the housing part. The sponge is appended to solve a combining error by an uneven plane of the field.

As it is shown in Fig. 3.9(c), three pins are included in a pin plate, and they are located on the non-adjacent three corners of each hexagonal pin plate. The pins are contacted to the pin plate by the spring in the pins' housing. The pins are pushed back in the direction of of $\overrightarrow{C}$ in Fig. 3.9(c) by an external force of the direction. When the external force is eliminated, the pins are returned default status that the pins are contacted to the pin plate. This structure can reduce the abrasion of pin, and the misalignment of pin can be overcame with an oscillating motion. The detailed contents of the oscillating motion is described in the next section. The pin consists of three parts: a pinhead, a locking area, and a thin bar. The form of the pinhead is a rounded cone. The locking area is cylindrical form, and its diameter is smaller than the pinhead's maximum diameter. The diameter of the thin bar that is cylindrical form is smallest diameter in the pin's structure.

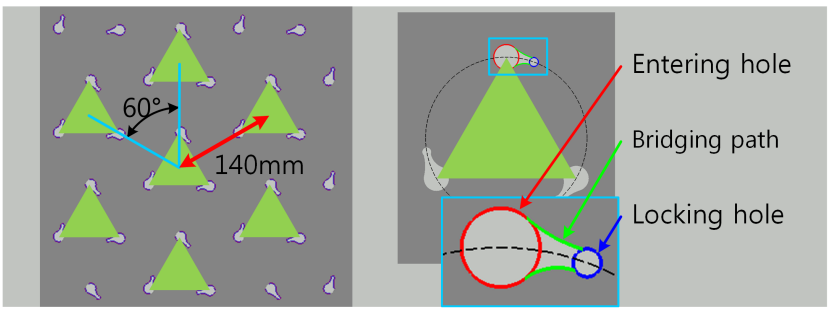
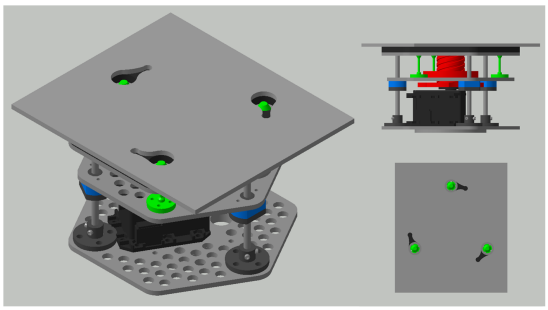
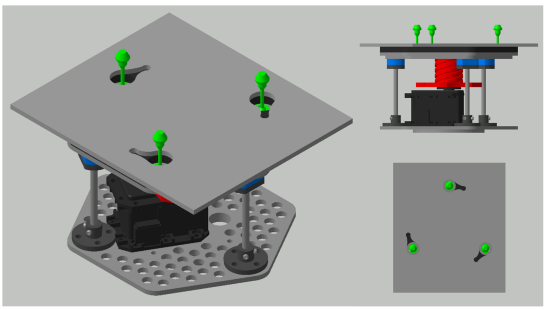


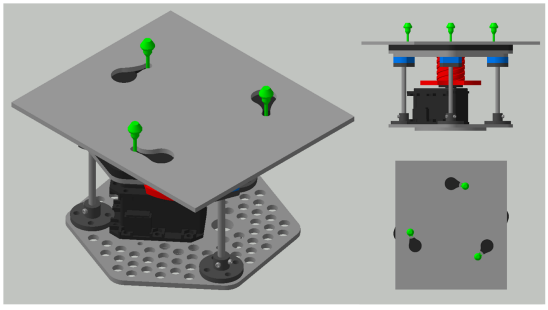
Fig. 3.10 Arrangement of field's hole and its shape



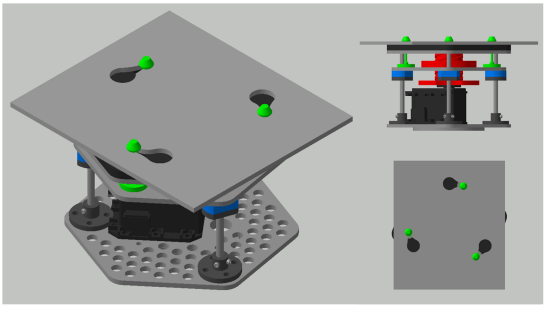
(a) Unlocked state



(b) Pushed motion by pin actuator



(c) Rotated motion by pin actuator



(d) Locked state

Fig. 3.11 Sequence of the locking motion of locking module

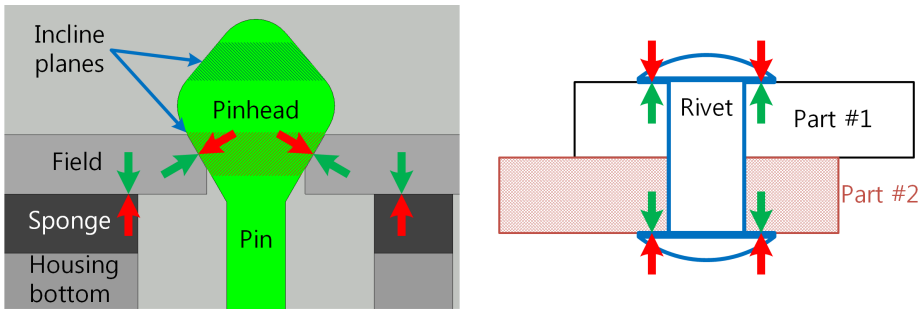
As above-mentioned, the field has distinctive holes for combining with a locking module. The placement and shape of the holes are illustrated in Fig. 3.10. A hole comprises three parts: an entering hole, a locking hole, and a bridging path. The locking motion of the module is achieved by the combination of the motions of the pin and module actuators and is illustrated in Fig. 3.11. A description of the figure is as follows.

1. Pin plate is pushed by the pin actuator (Fig. 3.11(a) → Fig. 3.11(b)).
2. Pin plate is rotated by the module actuator until the pins are located in their locking positions (Fig. 3.11(b) → Fig. 3.11(c)).
3. Pin plate is pulled by the pin actuator (Fig. 3.11(c) → Fig. 3.11(d)).

Fig. 3.12 shows a cross-sectional view of a pin and field when the pin is locked. The filed is gripped by the bottom plane of locking module and pinhead part. This is similar to a fastened rivet that is shown in Fig. 3.12(b). As it is shown in Fig. 3.12, there is a gap between the pin and locking position of hole, and there is two incline planes in the pin. These allow locking module to combine with the field smoothly.

3.2.3 Oscillating motion for the Pin-Lock mechanism

The misalignment between pins and entering holes can occur by various reasons, such as assembly tolerances, a gear backlash, etc, and the robot does not equip sensors



(a) Locked module and station (b) Fastened rivet

Fig. 3.12 Cross-sectional view of locked module and fastened rivet

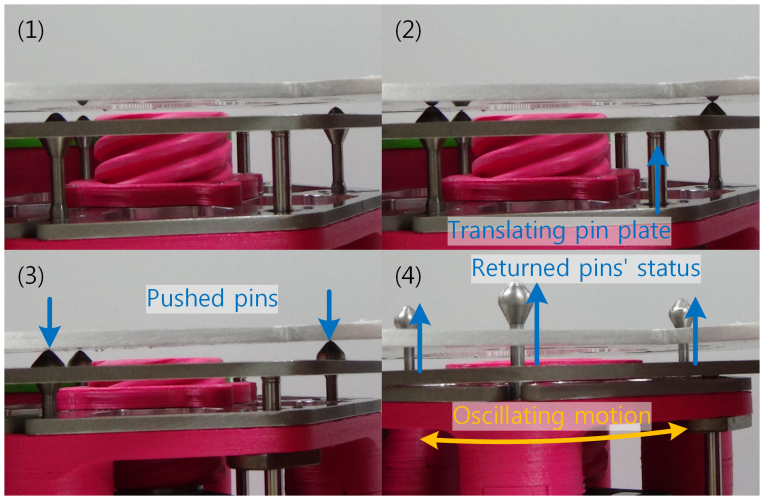


Fig. 3.13 The pins' passive movement during the misalignment

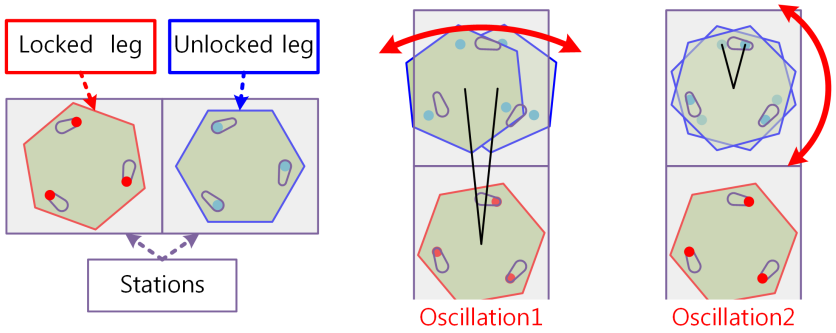
to check the status of pins. Besides, it is not a straightforward task to put multiple pins into the multiple holes without sensors simultaneously. The task of the inserting pins is same situation with above-mentioned situation. The task will be failed and the locking module will be damaged, even though only one pin is not aligned with its allowable area. To solves this problem, the oscillating motion of locking module is proposed. In this section, the form of wall climbing robot is assumed that the robot uses two locking modules as its legs. The locomotion of this robot comprises as a series of locking-rotate-locking steps that is similar to the Shady's locomotion.

As it was mentioned above, the pin can be pushed back by the external force. It indicates that the pin will be pushed back when the pin is misaligned, and the MoMo3 will not be damaged in this case. Besides, the pin is returned back when the pin is aligned with allowable area. The pin's movement when the pin is misaligned is illustrated in Fig. 3.13. The locking module uses two types of oscillating motions as the Fig. 3.14, and these oscillating motions are defined as following equation.

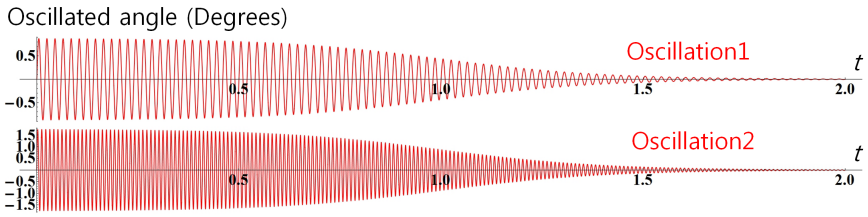
$$F_1(t) = \mathrm{Mag}_1 \cdot \sin(2\pi \cdot \mathrm{Freq}_1 \cdot t) \cdot D(t) \tag{3.1}$$

$$F_2(t) = \mathrm{Mag}_2 \cdot \sin(2\pi \cdot \mathrm{Freq}_2 \cdot t) \cdot D(t) \tag{3.2}$$

Mag_1, Mag_2, Freq_1 and Freq_2 were determined as 0.88°, 1.77°, 50 Hz and 100 Hz



(a) Two types of oscillating motion



(b) Time plots of the oscillating angles

Fig. 3.14 Oscillating motions of wall climbing robot

empirically. In (3.1) and (3.2), a $D(t)$ is a negative sigmoid function and it is added to reduce a magnitude of oscillating motion according to the increase of t. $D(t)$ is defined as follows. According to the (3.3), the magnitude of oscillating motion is reduced under 10 % of initial magnitude after 1.44 s. Fig. 3.14(b) shows the graphs of (3.1) and (3.2).

$$D(t) = \frac{e^{-5(t-1)}}{1 + e^{-5(t-1)}} \tag{3.3}$$

To verify the effectiveness of the oscillating motion, an experiment was performed on the adhering motion. In this experiment, it was supposed that the two module actuators have 3° error. In this situation, the maximum position error between a pin and an entering hole is 10.58 mm As shown in Fig. 3.15, the wall climbing robot can

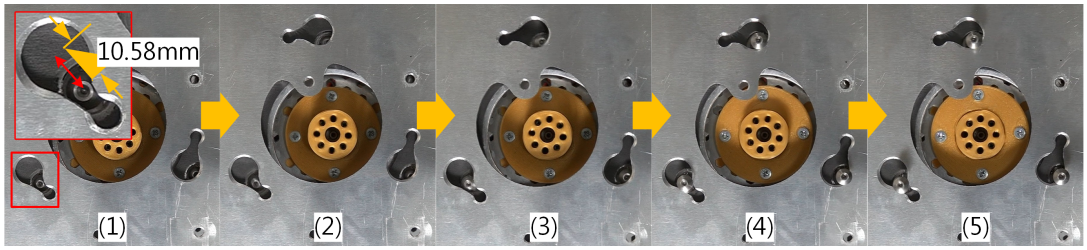


Fig. 3.15 The experiment for the verification of effectiveness of oscillating motion

overcome the misalignment and lock its legs even if the gap or gear backlash exist.

Through the oscillating motion, the pins can be realigned with allowable area, and then the task of the inserting pin is accomplished successfully. Through the pin-lock mechanism and oscillating motion, the abrasion problem of the screw-nut mechanism can be solved.

3.3 Mechanical structure and movement scheme of prototypes of MoMo

In this section, the detailed mechanical structure of produced prototypes of MoMo is introduced. Three prototypes were tested to develop the suitable wall climbing robot for the implementation of R+iSpace. First and second prototypes used the screw-nut mechanism, and third prototype of MoMo employed the pin-lock mechanism. The movement scheme of first prototype of MoMo is the gait locomotion using four legs, and the movement scheme of the other prototypes is similar to the Shady's locomotion. The major specifications of the prototypes of MoMo is shown in Table 3.2.

3.3.1 First prototype of MoMo

As it was mentioned above, the first prototype of MoMo (MoMo1) employed the screw-nut mechanism and gait locomotion using four legs. The detailed 3D CAD drawing of MoMo1 is illustrated in Fig. 3.16. As it is shown in Fig. 3.16, MoMo1 uses four screw modules, and they function as MoMo1's legs. As a result, MoMo1 has eight actuators for its locomotion; four actuators are located in the screw modules

Table 3.2 Specifications of the prototypes of MoMo

	First prototype	Second prototype	Third prototype
Weight [kg]	1.45	2.55	1.60
Size [mm]	190×255×110	210×381×129	200×280×129
No. of Actuator	8 Pieces	5 Pieces	4 Pieces
Moving distance for one step [mm]	150	196	140
Moving speed [cm/s]	0.33	2.05	2.80

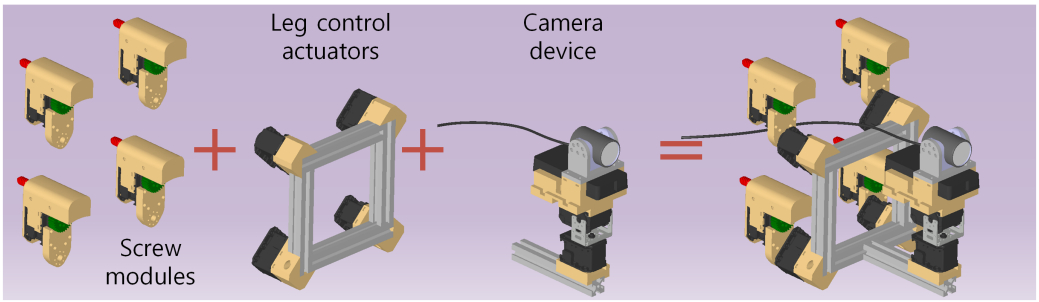


Fig. 3.16 3D CAD drawing of MoMo1

and remained four actuators are used for rotating the screw modules.

The gait sequence of the MoMo1 is designed with emphasis on stability. According to the proposed gait sequence, at least three screw modules are fixed with the nuts in field during its locomotion. The gait sequence is shown in Fig. 3.17 that shows an example of the MoMo1's movement to adjacent position. MoMo1 in first or last images in Fig. 3.17 holds a stable posture. At this posture, the additional energy is not required for maintaining current posture, because all joints are singular state against the gravity. Therefore, when the gait of the MoMo1 is ended, the MoMo1 change its posture to this posture to reduce the energy consumption.

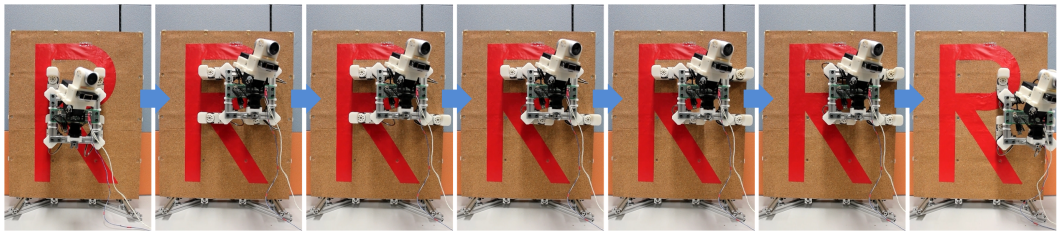
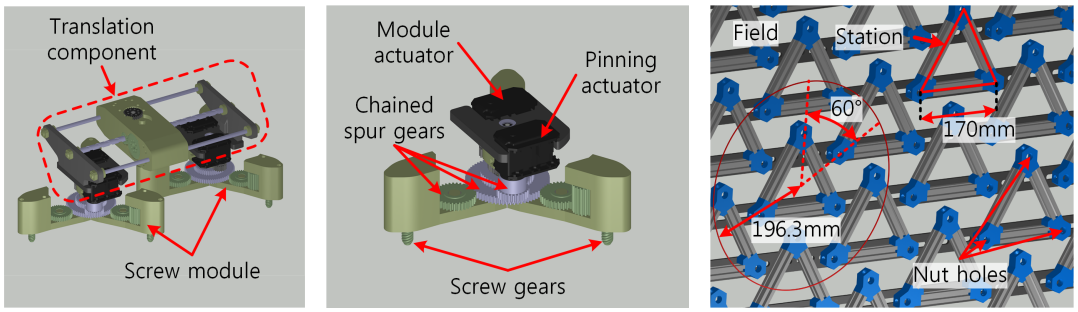


Fig. 3.17 Gait sequence of MoMo1



(a) Isometric view of MoMo2

(b) Detailed view of screw module of MoMo2

(c) Isometric view of field

Fig. 3.18 3D CAD drawing of MoMo2 and field

3.3.2 Second prototype of MoMo

There are major two problems in the mechanical structure of MoMo1; the movement of MoMo1 is slow and the number of used actuators in MoMo1 is too many. The low speed of MoMo results the time delay of service provision, and the number of used actuators is directly related to economical cost problem. To solve this problem, the second prototype of MoMo (MoMo2) was designed. The main cause of low speed of MoMo1 is the that it spent large amounts of time loosening and tightening the screws.

3D CAD drawings of MoMo2 and its experimental field are shown in Fig. 3.18. It comprises two legs and a translation component. This structure requires only five actuators for producing MoMo2. The translation component is moved in a straight

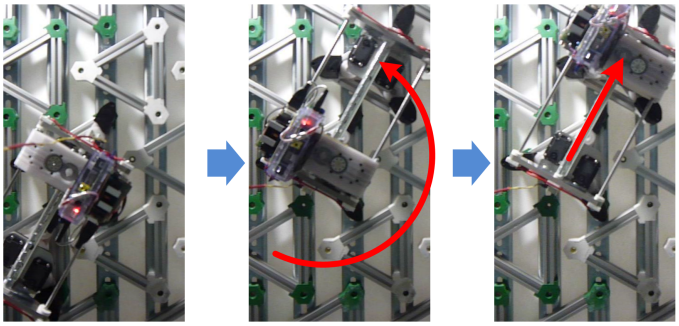


Fig. 3.19 Gait sequence of MoMo2

line between the two legs. This motion is used to reduce MoMo2's moment of inertia during a whole body rotating motion. The leg is an expanded screw module designed based on MoMo1's screw module, it has three screw gears that is controllable by a pinning actuator and chained spur gears. Thus, the screw gears are controlled simultaneously. This mechanism can increase the moving speed by the reduction of spent time loosening and tightening the screws. Moreover, it also help to reduce the number of steps in the gait sequence. For MoMo1, movement between two positions comprised 14 steps. However, for MoMo2 only four steps are required. In detail, MoMo2 moves to adjacent position by following sequence. First, MoMo2 loosens screws of a screw module on which the translation component is not placed. Then, MoMo2 rotates the screw module to the desired adjacent station. When the screw module is placed on desired station, MoMo2 fastens screws of the screw module. Last, the translation component moves to the screw module. Through this sequence, the device mounted on MoMo2 can be replaced to any station of field. Actual movement sequence is illustrated in Fig. 3.19.

The arrangement of nuts is adjusted according to changes of the mechanical structure of screw module. In the new field, three nuts compose one station and they are allocated in the form of an equilateral triangle. These stations are allocated at uniform distance and this is the same distance as between the two legs of MoMo2. This indicates that the two legs are fixed at two adjacent stations. As it is shown in Table 3.2, the no-load speed of MoMo2 is more than six times faster than that of MoMo1.

Through the simultaneous control of three screw gears and reduction of steps of

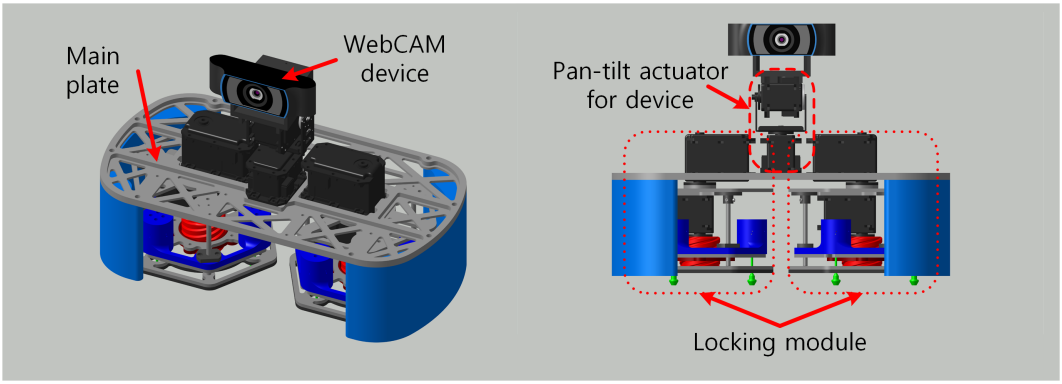


Fig. 3.20 3D CAD drawing of MoMo3

gait sequence, MoMo2 achieved more higher moving speed than MoMo1 with low cost. However, the abrasion problem of the screw-nut mechanism, that is discussed in previous section, occurs in MoMo2 frequently.

3.3.3 Third prototype of MoMo

To solve the MoMo2's problem, third prototype of MoMo (MoMo3) employed the pin-lock mechanism. The basic form of MoMo3 is similar to MoMo2, and MoMo3 comprises two locking modules and the main body. The locking modules function as MoMo3's legs, and the main body connects two legs. The 3D CAD drawing of MoMo3 is illustrated in Fig. 3.20.

There are two major differences in mechanical structure between MoMo2 and MoMo3. The first difference is the employed adhesion mechanism that were mentioned above. The other one is the existence of the translation component. The translation component is eliminated to increase the moving speed. Through the elimination of the translation component, the gait sequence of MoMo3 is simplified and can be summarized as follows.

Step 1 Unlock leg #1.
Step 2 Rotate the whole body of MoMo3 using the module actuator of leg #2 until the leg #1 is located at desired station.
Step 3 Lock leg #1.

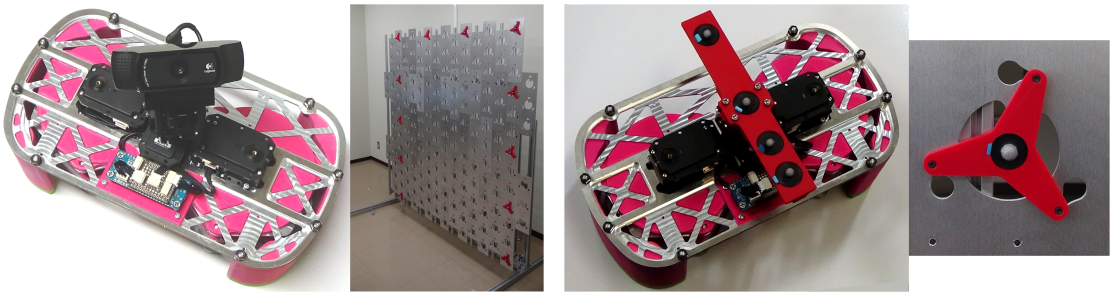
The elimination of the translation component has a drawback. As aforementioned, this component is designed to reduce the moment of inertia during the whole body rotating motion. To solve the problem of the increased moment of inertia due to the elimination of the translation component, MoMo3 employs a module actuator with higher torque than that of MoMo2.

3.4 Experiments

As it is shown in Table 3.2, MoMo3 has highest performance for using in R+iSpace among three prototypes. Therefore, the experiments in this section is performed on MoMo3. In this thesis, two types of experiments were performed. First one is designed to verify the mobility of MoMo. As aforementioned, MoMo should satisfy the requirements of R+iSpace. All requirements except third requirement are satisfied by the mechanical structure of MoMo, and it is expected that MoMo can satisfy the third requirement. To verify this expectation, the mobility experiment was performed. In this experiment, the MoMo is moved on the experimental field. In the interim, the position error between device's estimated positions and actual positions were calculated and confirmed. The accurate of motion repeatability is also a critical factor in the MoMo's mobility, because the device does not perform an initialization after rearrangement. The accuracy of repeatability is presented as the position error after rearrangement at the same location. The next experiment was performed to verify R+iSpace's effectiveness. As it was mentioned in Chapter 2, it is expected that the performance of provided service in iSpace will be increased when the device's spatial constraint problem is solved. In this experiment, the performance of a same service in iSpace and R+iSpace are compared.

3.4.1 Verification of MoMo3's mobility

The experiments for the verification of MoMo3's mobility was performed using the motion capture system. The motion capture system, 'Raptor-E' by 'Motion Analysis' was used [69]. six cameras, which can be operated up to 500 fps at a full resolution of 1280×1024 pixels, was used in this experiment. MoMo3 and its experimental field are shown in Fig. 3.21(a). As shown in Fig. 3.21(b), markers for the motion capture were mounted on the field and on MoMo3. The marker mounted on the



(a) MoMo3 with WebCAM device and experimental field

(b) Markers on MoMo3 and the field

Fig. 3.21 MoMo3 and its experimental field

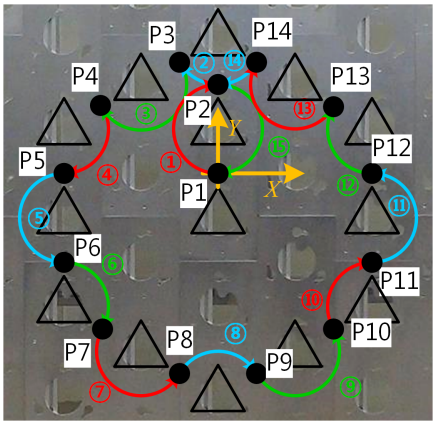


Fig. 3.22 Moving path of MoMo3 in the mobility verification experiment

center of MoMo3 indicates the device's position. The motion capture system records the device's current position during the experiment.

The error of the estimated position and the accuracy of repeatability of MoMo3 are confirmed in following experiment. MoMo3 moved along a path shown in Fig. 3.22; the distance for one lap is approximately 186.62 cm. MoMo3 repeats the gait motion 16 times to complete one lap and moves along the path seven times. The positions are illustrated in Fig. 3.22 as $[P_1, P_2, \cdots, P_{14}]$, and the device passed the positions

at least seven times during the experiment. P_n^k indicates the kth recorded positional data on the P_n. The maximum value of k (Max_k) is 14 when $n = 1$ or $n = 2$, and in other cases is 7. The average data on the P_n is denoted as MP_n, which is used as a reference position. The accuracy of repeatability at P_n (RE_n) is calculated as follows.

$$\mathrm{RE}_n = \frac{1}{7} \sum_{i=1}^{\mathrm{Max}_k} \left(Dist \left(P_n^i, \mathrm{MP}_n \right) \right) \tag{3.4}$$

In the above equation, $Dist(P_a, P_b)$ indicates a distance between two points P_a and P_b. RE_n indicates the average distance between the recorded positions and MP_n at the same location.

The ideal position of a device that takes only the mechanical structure of the field into consideration can be calculated directly, and the calculated device's ideal position is the estimated position. In this experiment, the estimated positions (EP_n) are determined as in Table 3.3. The error of the estimated positions (EE_n) are calculated as follows.

$$\mathrm{EE}_n = \frac{1}{7} \sum_{i=1}^{7} \left(P_n^i - \mathrm{EP}_n \right) \tag{3.5}$$

EE_n is a crucial factor to satisfy the fourth necessary condition of R+iSpace, i.e., the error of the estimated position should be within a certain range, and it should not accumulate.

The results of the accuracy of repeatability and accuracy of estimated position experiments are shown in Table 3.4. The average position error at the same position is 0.21 mm and has a maximum of 0.49 mm. This result shows that MoMo3 can reach the same position within a maximum error distance of 0.49 mm; this is acceptable error in the applications of IE. In R+iSpace, the position of the device is determined as the estimated position. Therefore, the error between the estimated position and actual position should be small and within a certain range. As shown by the results, the maximum error is approximately 2.2 mm, which is an acceptable error for R+iSpace. From Fig. 3.23, the estimated positional error is not accumulated with the continued movement.

Table 3.3 Estimated device's positions on the path [mm]

EP_1	MP_1	EP_8	EP_1+(-60.622,-315,0)
EP_2	EP_1+(0,140,0)	EP_9	EP_1+(60.622,-315,0)
EP_3	EP_1+(-60.622,175,0)	EP_{10}	EP_1+(181.865,-245,0)
EP_4	EP_1+(-181.865,105,0)	EP_{11}	EP_1+(242.487,-140,0)
EP_5	EP_1+(-242.487,0,0)	EP_{12}	EP_1+(242.487,0,0)
EP_6	EP_1+(-242.487,-140,0)	EP_{13}	EP_1+(181.865,105,0)
EP_7	EP_1+(-181.865,-245,0)	EP_{14}	EP_1+(60.622,175,0)

Table 3.4 Results of the repeatability accuracy and the estimated position error experiments

	Repeatability Accuracy [mm]	Estimated position Error [mm]
P_1	0.262	0.376
P_2	0.492	1.607
P_3	0.087	0.944
P_4	0.148	0.879
P_5	0.209	1.442
P_6	0.095	1.109
P_7	0.144	1.131
P_8	0.162	2.217
P_9	0.164	0.572
P_{10}	0.136	1.564
P_{11}	0.113	1.060
P_{12}	0.347	1.749
P_{13}	0.132	0.686
P_{14}	0.137	1.848
Average	0.212	1.197
σ	0.130	0.566

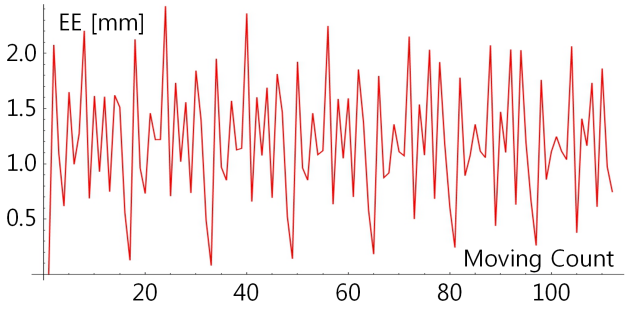
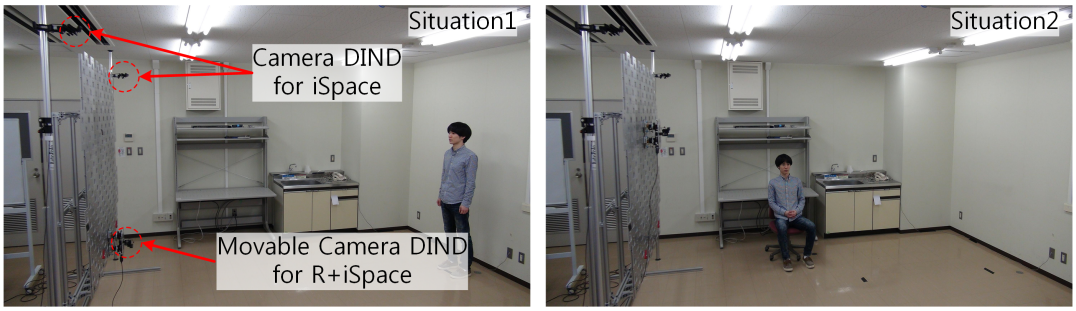


Fig. 3.23 Estimated error graph according to the moving count

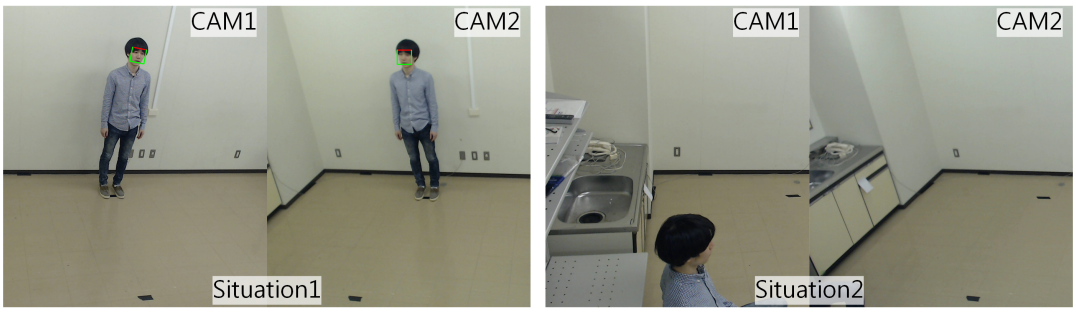
3.4.2 Verification of effectiveness of R+iSpace

As aforementioned, R+iSpace is designed to solve the device's spatial constraint problem. Therefore, the effectiveness of R+iSpace is verified when it solves this problem. In this experiment, there are three cameras for face detection in the space. A camera is mounted on MoMo3, and two cameras are fixed on the field. The cameras that are fixed on the field represent iSpace, and the camera mounted on MoMo3 represents R+iSpace. These cameras capture the user's face, and the captured images are processed by the 'OKAO Vision', a face detection program by 'Omron' [70, 71].

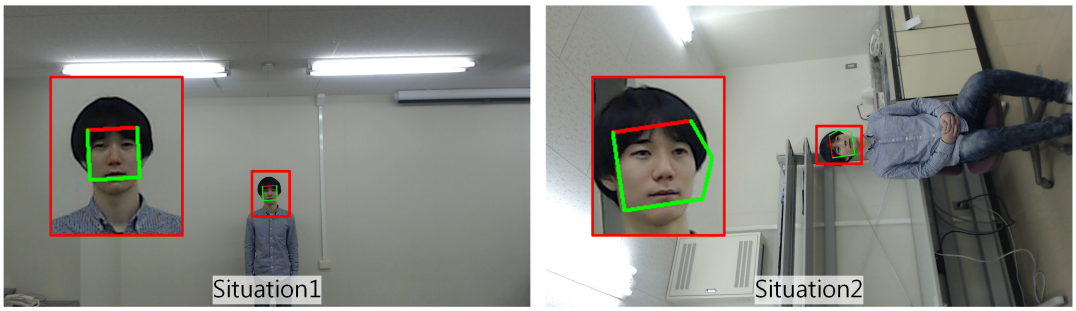
Fig. 3.24 shows two predetermined situations. In the initial state, the user stood as in Situation1 in Fig. 3.24, and then sat on the chair as in Situation2. MoMo3 rearranges the device to the proper location for face detection according to the target's position. In this experiment, the location of the target was given beforehand and the MoMo's destination and the camera's viewing direction were also calculated beforehand. Fig. 3.24(a) shows a whole view of the two situations. The processed results of captured images via iSpace cameras are shown in Fig. 3.24(b), and Fig. 3.24(c) is the processed results of captured images via R+iSpace camera. As can be seen, the face detection via the fixed iSpace cameras failed when the situation changed, but the face detection via the camera that mounted on MoMo3 succeeded because the camera is rearranged according to the target's position. This indicates that the device's spatial constraint problem can be solved in R+iSpace, and the performance of provided service in R+iSpace is higher than it in iSpace.



(a) Whole view of two situations



(b) Results of captured images via iSpace cameras



(c) Results of captured images via R+iSpace camera

Fig. 3.24 Experiment to verify effectiveness of R+iSpace

3.5 Discussion and conclusion

In this chapter, suitable mechanical structure of MoMo and field was discussed. The implementation of R+iSpace is enabled by the proposed MoMo's mechanical structure. To find suitable mechanical structure of MoMo, the adhesion methods and movement schemes of conventional wall climbing robots were surveyed and analyzed. Then, this thesis proposed two adhesion method, i.e., a screw-nut mechanism and a pin-lock mechanism. The mechanical structures and movement schemes of the produced prototypes of MoMo were also introduced. In Section 3.4, the objectives, processes and results of two experiments was described. MoMo's mobility was confirmed by the first experiment, where, the accuracy of repeatability and the error of the estimated position are measured and analyzed. In the second experiment, it was confirmed that rearrangement of device yielded an improvement in the application's performance. As a result, R+iSpace could solve the device's spatial constraint problem, and it results the increment of performance of provided service. In addition, MoMo3 is found to be an excellent solution for the implementation of R+iSpace.

Chapter 4

Best Effort Location of Devices for a Successful Service Provision

In this chapter, this thesis discusses the suitable location of devices. As it was mentioned in Chapter 2 and Chapter 3, the performance of provided will be increased when the device is rearranged to its suitable location. This is a basic concept of R+iSpace, and to realize the rearrangement function of devices the suitable mechanical structure of MoMo was proposed in Chapter 3. However, the first question of this thesis, 'Where is the best position of device?', did not solved yet. This is difficult problem due to diversity of applications and used devices. To address this problem, this thesis standardizes the modeling of targets of application. In addition, several factors, that significantly affect on the performance of service provision, are determined, and their evaluation functions are also determined. The results of evaluation functions are integrated by proposed algorithm, and the best effort location of device for successful service provision is determined based on the integrated results. The proposed algorithm is verified by simulation experiment.

4.1 Basic concept of proposed algorithm for detemining device's location

Before explaining the proposed algorithm, the objective of the proposed algorithm is described briefly. The objective of the proposed algorithm is to determine the

destination of movable DIND that employs MoMo3 for its mobility. The destination is a suitable node that can be used to accomplish a target application in the current situation. The 'node' that is referred in the preceding sentence indicates the device's location at stable status. The node's location is limited by the structure of field. It will be described later in detail. The proposed algorithm considers the application's success rate and response time until the application is executable. This approach is effective because these metrics are important measures when a user evaluates iSpace performance.

Generally, an application's success rate is affected by the performance of the application, the specifications of the device, the relative position of the device from the target, the environmental structure, etc. Among these factors, a device in R+iSpace can solve two factors by rearranging its location, i.e., the relative position of the device from the target and the environmental structure. However, rearrangement of a DIND generates delay relative to its use in the given application. Therefore, the proposed algorithm considers two additional factors, i.e., adaptability in small changes of the location/posture of the target and travel time to the destination. All nodes are evaluated by the evaluation functions for each factor. Then, the proposed method determines a movable DIND's destination. The evaluation factors are as follows.

- Distance from the target
- Angle between the target's direction and the device's viewing direction
- Occlusion by an obstacle
- Obstacles on the field
- Results of adjacent nodes
- Necessary travel time from the current node

The first two factors are related to the relative position from the target, and the next two factors are related to the environmental structure. These four factors are the basic factors that influence the application performance. The fifth factor is included for rapid response to the small changes of the location/posture of the target. The last factor makes it possible for R+iSpace to provide service rapidly.

4.1.1 Node and adjacent nodes

As it was mentioned above, the node indicates the device's location at stable status, and it is assumed that DIND employs MoMo3 for its mobility. As it is shown

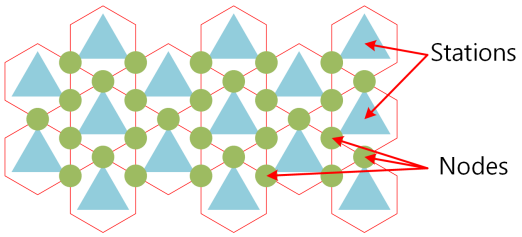


Fig. 4.1 Arrangement of stations and nodes

in Fig. 3.21(b), the device's location is the center of MoMo3, and the stable status indicates that the MoMo3's two locking modules are locked state. As a result, the node indicates the center position between two stations of field. Fig. 4.1 shows the arrangement of stations and nodes. In R+iSpace, it is recommended that all devices are used only when they are placed on nodes for the following reasons. First, significant noise is generated in the device's results by the vibration or movement of MoMo3 when MoMo3 is not firmly fixed. This reduces R+iSpace performance. Next, the estimation of the device's location is easy when MoMo3 is firmly fixed. The device's position is very important information when a device is used. The device's position is obtained easily by pre-calculated node positions when the device is used only during MoMo's locked status. Last, restriction of the device to nodes is advantageous in terms of energy efficiency. A large load occurs in the module actuator of the fixed leg when MoMo3 is not fixed securely. Therefore, to use a device on other positions requires more energy to maintain the current position. This causes energy consumption and can cause damage to MoMo3, such as damage to the actuator by overheating. Therefore, the device's location is restricted to nodes.

Adjacent nodes are defined as nodes that are reachable by one step or two steps movements from the current node $N({}_0N_{Adj})$, and it is denoted by ${}_iN_{Adj}$. Here, i indicates the movement counter number to reach ${}_iN_{Adj}$ from the current node N. Nodes ${}_1N_{Adj}$ and ${}_2N_{Adj}$ are illustrated in Fig. 4.2. As can be seen, ${}_1N_{Adj}$ and ${}_2N_{Adj}$ are 10 and 30, respectively.

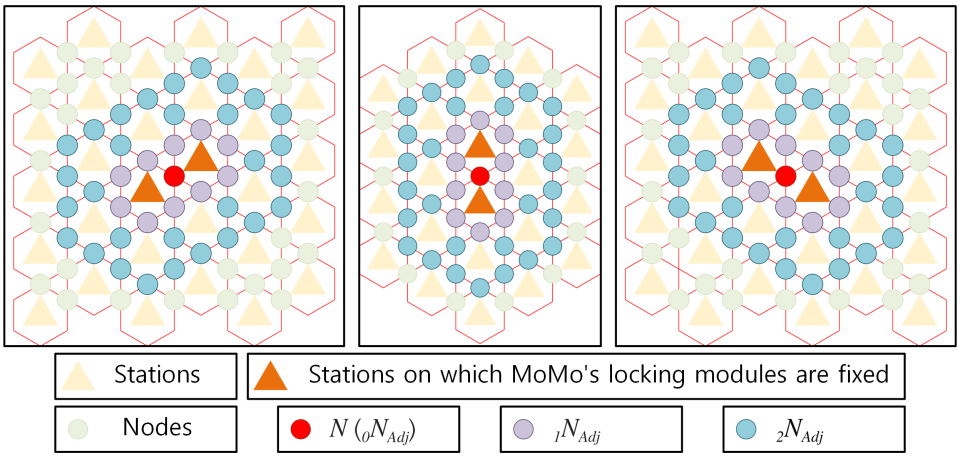
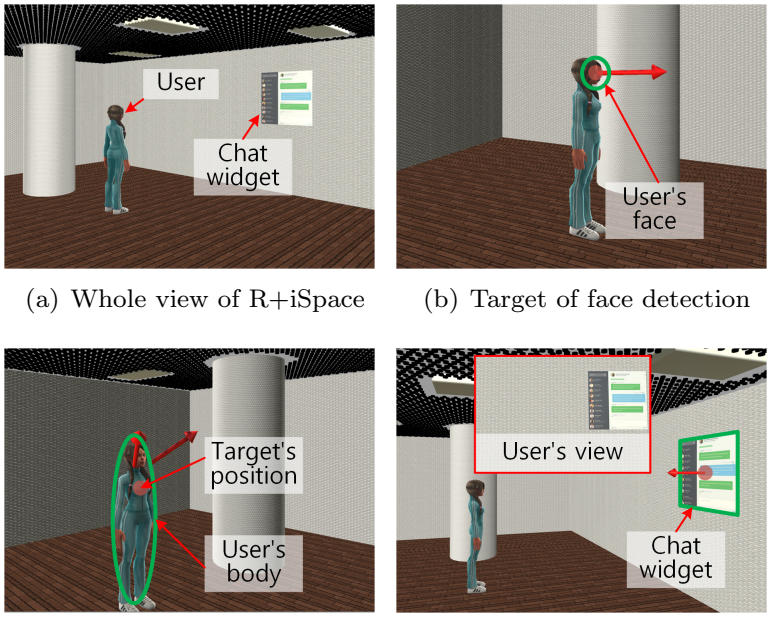


Fig. 4.2 Arrangement of current node $N({}_0N_{Adj})$ and adjacent nodes ${}_iN_{Adj}$

4.1.2 Standardization of the target of application

Here, before explaining the factors, the target is discussed in this section. Obviously, the target differs according to the types of devices and the applications. To make a standardized evaluation function, the various targets should also be standardized. The proposed algorithm uses information about the target position and direction to simplify target modeling. The target's position is determined as the center position of the target, and the target's direction is determined as the direction to the optimal location of the device in empty space. Note that some applications have symmetrical characteristics. Therefore, the target can have multiple optimal locations. In such cases, the target has multiple directions. In some applications (e.g., fire detecting system), the target (e.g., flame of fire) does not have a specific direction. In this cases, the target's direction is defined as direction towards the device.

Fig. 4.3 shows examples of targets according to devices and applications. As shown in Fig. 4.3(a), a user is standing in the R+iSpace. Fig. 4.3(b) shows an example in cases of the face detection. The target of the face detection is the user's face, and the optimal location should be placed in front of the user's face. Therefore, the target's position and direction are determined as shown in Fig. 4.3(b). Another example is



(a) Whole view of R+iSpace (b) Target of face detection

(c) Target of gesture recognition (d) Target of chat widget application

Fig. 4.3 Examples of target according to application

shown in Fig. 4.3(c). In this example, the camera is used for a gesture recognition that has a best performance when the camera is placed on the frontal upper diagonal direction of user. The target of the gesture recognition is the user's whole body, and the target has two directions as it is shown in Fig. 4.3(c). The last example is a chat widget application that is shown by a projector. The projector's target is a projection image whose position is determined by the user's viewing direction. For this application, the target is placed to the top right position of the user's view. This is shown in Fig. 4.3(d). When the projector can be modeled as a pinhole projector, the target's position is set to the center of the projection image, and the target's direction is set to the normal direction of the projection plane.

Through the standardization of targets and the determination of evaluation factors, the proposed algorithm achieves extensive usability. In iSpace, many types of applica-

tions and devices are used, and their characteristics differ. Therefore, all applications and devices have evaluation functions with different forms. However, the proposed method provides the same form for evaluation functions through simplification of the evaluation factors and target. The proposed algorithm can be applied to most applications and devices by changing the parameters of the evaluation function according to the specifications of the application and device.

4.2 Implementation of proposed algorithm for detemining device's location

In this section, a detailed explanation of the factors and their evaluation functions are discussed. The results of the evaluation functions should show the suitability of the nodes according to each factor. However, to apply an evaluation function for various devices, the functions should be simplified. Therefore, we determine the evaluation functions for the four basic factors as a trapezoidal-shaped function, which is often used in fuzzy algorithms. A gradient of each function is determined by parameter α_i, which makes the evaluation functions robust. These functions do not give exact suitability, but it can provide stable ranges for using the device. Besides, this function is simple and applicable to various devices and applications.

The results are integrated after calculation of the evaluation functions for the four basic factors. Essentially, the integrated result is obtained by the weighted average method. The obtained integrated results are entered to a filter designed for the fifth factor. Finally, the algorithm determines the destination of a movable DIND by considering the travel time. A detailed explanation and the calculation methods are discussed below.

4.2.1 Basic factor 1: Distance

Most devices demonstrate different performance according to the distance between the device and its target. For example, when the camera for the gesture recognition is too close to the user, the gesture recognition result is poor because the camera can only partially capture the user. In addition, when the camera is too far from the target, the target's size in the captured image is too small. Thus, the recognition result is also poor. This suggests that the camera can only function accurately within

an appropriate distance range from the target. Note that this characteristic is evident in most applications and devices.

The evaluation function of the distance is defined as follows.

$$\mathrm{EV}_1(N) \coloneqq \begin{cases} 0 & D_N \leq D_m \\ \dfrac{D_N - D_m}{\alpha_1} & D_m < D_N \leq D_m + \alpha_1 \\ 1 & D_m + \alpha_1 < D_N \leq D_M - \alpha_1 \\ \dfrac{D_M - D_N}{\alpha_1} & D_M - \alpha_1 < D_N \leq D_M \\ 0 & D_M < D_N \end{cases} \tag{4.1}$$

In (4.1), N denotes node, and D_N denotes the straight distance between N and the target. There are three parameter variables, i.e., D_m, D_M, and α_1. D_m and D_M denote the minimum and maximum distances for reliable use of the device. D_m and D_M are obtained when the device is placed on a line that represents the target's direction. The evaluation values inside the range between D_m and D_M are defined as 1, and the evaluation values outside the range decrease gradually. The algorithm calculates (4.1) for all available nodes. The parameters are determined by the specifications of the device and the application. Fig. 4.4 shows an example of $\mathrm{EV}_1(N)$ in the FD application. In this example, the red line is the optimal direction from the target. As shown in Fig. 4.4, the performance of the application changes according to changes in distance. The application worked well when the distance between the device and the target was smaller than a certain distance (e.g., D_M) and greater than another certain distance (e.g., D_m).

4.2.2 Basic factor 2: Angle

The included angle between the device's viewing direction and the target's direction is the second factor that affects application performance. A greater included angle results in poorer application performance. This is evident because the target direction is the optimal location's direction from the target. The evaluation function for a target with k directions is as follows.

$$\mathrm{EV}_2(N) \;=\; \max_{i \in k} \mathrm{EV}_2^i(N) \tag{4.2}$$

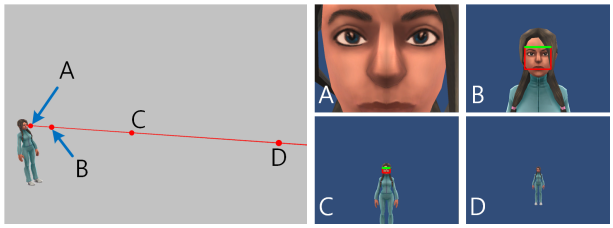


Fig. 4.4 Change of face detection result according to changing distance from target

$$\mathrm{EV}_2^i(N) := \begin{cases} 1 & \mathrm{IA}^i \le \mathrm{IA}_M - \alpha_2 \\ \dfrac{\mathrm{IA}_M - \mathrm{IA}^i}{\alpha_2} & \mathrm{IA}_M - \alpha_2 < \mathrm{IA}^i \le \mathrm{IA}_M \\ 0 & \mathrm{IA}_M < \mathrm{IA}^i \end{cases} \tag{4.3}$$

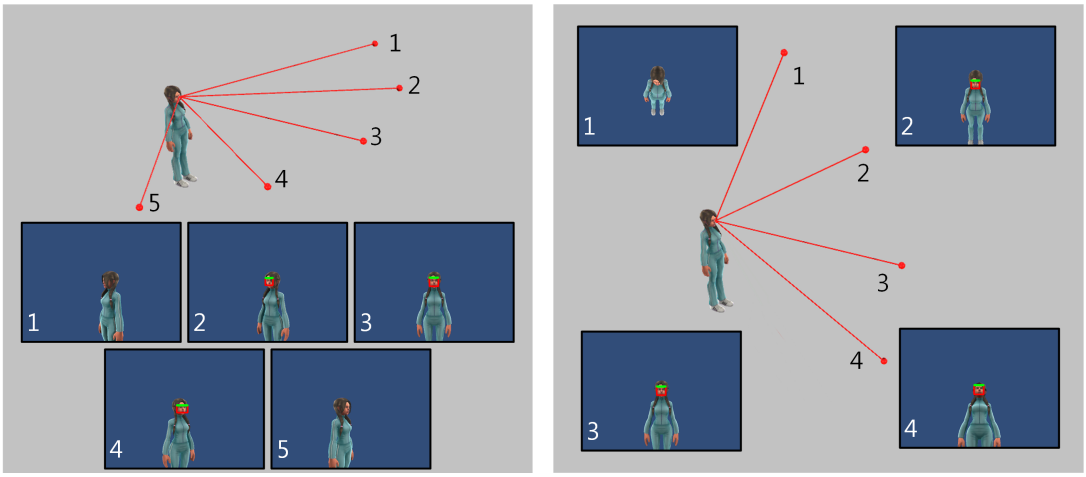
In (4.3), IA^i is the included angle between the device's viewing direction and the target's ith direction. IA_M denotes the available maximum included angle; this parameter is also determined by the specifications of the device and application.

(4.2) can be adopted as an evaluation function in most devices and applications. However, some applications or devices have an extreme difference in performance according to the direction of the included angle.

$$\mathrm{EV}_2(N) = \max_{i \in k} \left({}_h\mathrm{EV}_2^i(N) \cdot {}_v\mathrm{EV}_2^i(N) \right) \tag{4.4}$$

$${}_h\mathrm{EV}_2^i(N) := \begin{cases} 1 & {}_h\mathrm{IA}^i \le {}_h\mathrm{IA}_M - {}_h\alpha_2 \\ \dfrac{{}_h\mathrm{IA}_M - {}_h\mathrm{IA}^i}{{}_h\alpha_2} & {}_h\mathrm{IA}_M - {}_h\alpha_2 < {}_h\mathrm{IA}^i \le {}_h\mathrm{IA}_M \\ 0 & {}_h\mathrm{IA}_M < {}_h\mathrm{IA}^i \end{cases} \tag{4.5}$$

$${}_v\mathrm{EV}_2^i(N) := \begin{cases} 1 & {}_v\mathrm{IA}^i \le {}_v\mathrm{IA}_M - {}_v\alpha_2 \\ \dfrac{{}_v\mathrm{IA}_M - {}_v\mathrm{IA}^i}{{}_v\alpha_2} & {}_v\mathrm{IA}_M - {}_v\alpha_2 < {}_v\mathrm{IA}^i \le {}_v\mathrm{IA}_M \\ 0 & {}_v\mathrm{IA}_M < {}_v\mathrm{IA}^i \end{cases} \tag{4.6}$$



(a) Horizontal included angle (b) Vertical included angle

Fig. 4.5 Change of face detection performance according to change of included angle

(4.4) represents application performance in detail due to the split calculation according to the direction of the included angle. Fig. 4.5 shows an example of $\mathrm{EV}_2(N)$ in the face detection. As mentioned previously, performance declined according to increased included angle. As shown in Fig. 4.5, the application could not detect the user's face when the included angle was greater than a certain value.

4.2.3 Basic factor 3: Occlusion

Most devices cannot work well when their targets are occluded. To avoid this situation, the proposed algorithm should consider the influence from the environment. This factor is designed for the influence of the environment. The evaluation function is defined as follows:

$$\mathrm{EV}_3(N) := \begin{cases} 1 & \text{There is no occlusion.} \\ 0 & \text{There is an occlusion.} \end{cases} \tag{4.7}$$

(4.7) is usable with most devices. However, for some devices, such as speakers and microphones, obstruction caused by an obstacle does not have a significant effect on the performance of the application. In other words, this factor is meaningless for such devices. Therefore, such devices do not calculate this evaluation function, and they set the result to 1.

4.2.4 Basic factor 4: Obstacle

Typically, many objects can be installed on the field. Such objects are a hindrance to MoMo's movement. In particular, when MoMo is placed near such an object, its movement is restricted. Therefore, the algorithm should not select nodes adjacent to such objects as destination. Through the following evaluation function, the algorithm can avoid the selection of nodes that are adjacent to such objects.

$$\mathrm{EV}_4(N) := \begin{cases} 0 & D(N) \leq {}_oD_m \\ \dfrac{D(N) - {}_oD_m}{\alpha_4} & {}_oD_m < D(N) \leq {}_oD_m + \alpha_4 \\ 1 & {}_oD_m + \alpha_4 < D(N) \end{cases} \tag{4.8}$$

Where,

$$D(N) := \min_{\forall \mathrm{Obj}} \mathrm{Dist}\,(N, \mathrm{Obj}) \tag{4.9}$$

In (4.8), ${}_oD_m$ denotes the available minimum straight distance. $\mathrm{Dist}\,(N, \mathrm{Obj})$ indicates the straight distance between node N and the boundary of the object Obj, and $D(N)$ is the minimum $\mathrm{Dist}\,(N, \mathrm{Obj})$ for all Obj.

4.2.5 Integration of basic four factors

The basic four factors are designed to take application performance into account. The remained two factors relate to the efficiency of the overall system. In other words, the evaluation values of the application's performance are sufficient for the four evaluation functions described above.

The calculation of the fifth factor is based on the integration results of basic four factors. Therefore, the integration process should be performed prior to the calcula-

tion of the fifth factor. The integration is performed as follows.

$$\mathrm{EV}_{int}(N) \coloneqq \begin{cases} 0 & \text{If } \prod_{k=1}^{4} \mathrm{EV}_k(N) = 0 \\ \sum_{k=1}^{4} \omega_k \cdot \mathrm{EV}_k(N) & \text{Else} \end{cases} \tag{4.10}$$

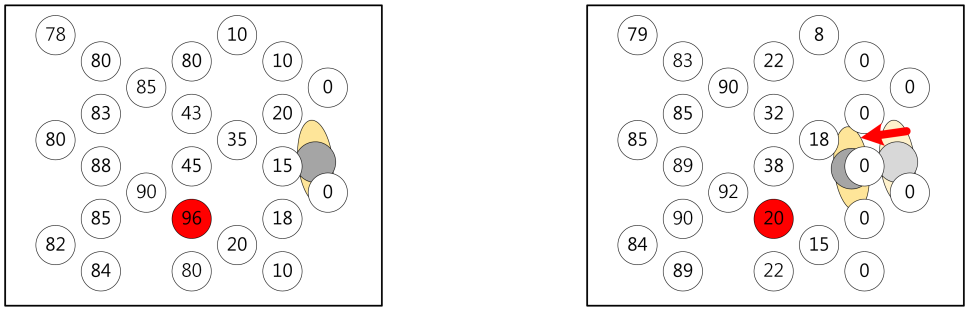
Where,

$$\sum_{k=1}^{4} \omega_k = 1 \quad \wedge \quad 0 \leq \omega_k < 1 \tag{4.11}$$

If a node has an evaluation result of 0 for a factor, this indicates that the device is not usable with the node in terms of the given factor. Therefore, the integration result of the node should be 0. In other cases, the results are integrated through the weighted average (WA) filter.

4.2.6 Evaluation results of adjacent nodes

As mentioned above, this factor is related to the fast response to a slight movement of the target. In R+iSpace, the target's position can be changed freely, and the suitable locations of a device are changed according to the changes of target's position and direction. Here, the suitable location of a device indicates the device's location when the service provision is successful. The goal of this factor is to find suitable locations after the target's slight movement among current suitable locations. Fig. 4.6 shows an example situation of the target's slight movement. The values of EV_{int} are filled in the nodes according to the given situations. In Fig. 4.6(a), the red node has highest evaluation result, but this node has insufficient evaluation result after the target's slight movement. To solve this problem, the proposed algorithm considers the evaluation results of adjacent nodes. Basical idea of this factor is as follows. First, a node will have a similar evaluation result with current evaluation results of adjacent nodes after target's slight movement. Because, the evaluation result is closely related with the relative position from the target. Therefore, the nodes which has highly evaluated adjacent nodes will achieve the high evaluation result.



(a) Evaluation results before target's slight movement

(b) Evaluation results after target's slight movement

Fig. 4.6 An example situation of target's slight movement

The calculation of this factor is performed by the WA filter with adjacent nodes. To prevent cost generation of nodes that have an evaluation result of 0, the calculation of the WA filter is performed only when the node has an evaluation result greater than 0. The updated results are calculated as follows.

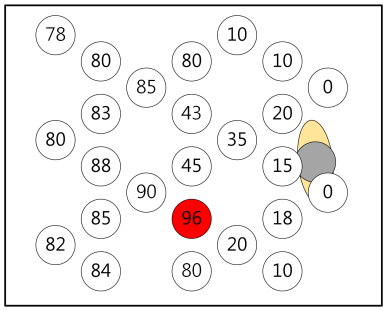
$$\mathrm{EV}_5(N) := \begin{cases} \displaystyle\sum_{i=0}^{2} \omega_5^i \cdot \overline{\mathrm{EV}_{int}({}_iN_{Adj})} & \text{If } \mathrm{EV}_{int}(N) \neq 0 \\ 0 & \text{Else} \end{cases} \tag{4.12}$$

Where,

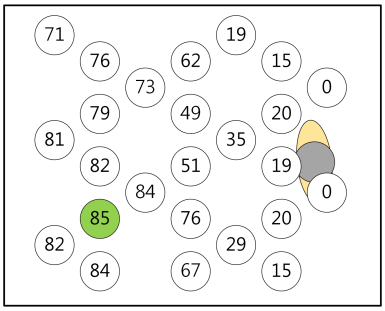
$$\omega_5^0 + \omega_5^1 + \omega_5^2 = 1 \quad \wedge \quad 0 \leq \omega_5^2 \leq \omega_5^1 \leq \omega_5^0 < 1 \tag{4.13}$$

In (4.12), $\overline{\mathrm{EV}_{int}({}_iN_{Adj})}$ denotes the average value of all adjacent nodes ${}_iN_{Adj}$. Through this process, the updated evaluation value of a location includes the adjacent nodes' evaluation values.

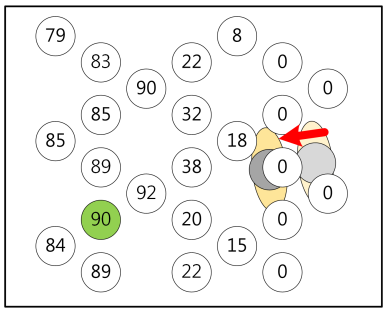
MoMo selects the node with an adjacent node with a high evaluation result as the destination using the WA filter. Fig. 4.7 shows an example that illustrates the effectiveness of this process. In Fig. 4.7(a), node red node has the highest evaluation result. However, in Fig. 4.7(b), node green node achieves the highest evaluation result.



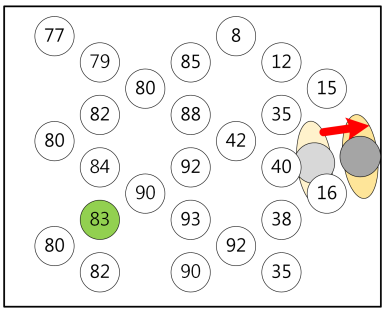
(a) EV(N) before target's slight movement



(b) $EV_5(N)$ before target's slight movement



(c) EV(N) after target's slight movement1



(d) EV(N) after target's slight movement2

Fig. 4.7 The effect of the weighted average calculation with adjacent nodes

As it is shown in Fig. 4.7(c) and Fig. 4.7(d), this node has sufficiently high evaluation result after target's slight movement.

4.2.7 Travel time and final decision

Rapid response is one of the significant factors of R+iSpace's performance. Therefore, travel time should be considered when a destination is decided. However, calculating actual travel time for all nodes requires significant computation time, because, a field comprises many nodes. To reduce this computation time, the proposed algorithm divides the process into two steps. In the first step, the algorithm calculates the pseudo travel time for all nodes. The pseudo travel time is defined based on MoMo's

average moving speed and the straight distance from the current node. Then, the pseudo travel time is integrated with the results of the fifth factor. These processes are calculated as follows.

$$ {}_{pseudo}T(N) \coloneqq \frac{\mathrm{Dist}\,(N, N_C)}{V_{MoMo}} \tag{4.14} $$

$$ \mathrm{EV}_6^{pseudo}(N) \coloneqq (1-\omega_6)\cdot \mathrm{EV}_5(N) + \frac{\omega_6}{{}_{pseudo}T(N)+1} \tag{4.15} $$

In (4.14), N_C and V_{MoMo} denote the current node of the device and MoMo's average velocity, respectively. The pseudo travel time is nearly proportional to the actual travel time; however, it is not an exact result. Therefore, the algorithm performs the second step.

In the second step, the algorithm selects several nodes that have the top ranked results among all nodes as destination candidates. The algorithm then computes the actual travel times to the candidates. Then, the algorithm integrates the actual travel times with the results of the evaluation. The destination is then decided as the node with the highest evaluation result among all candidates. The calculation of this step is as follows.

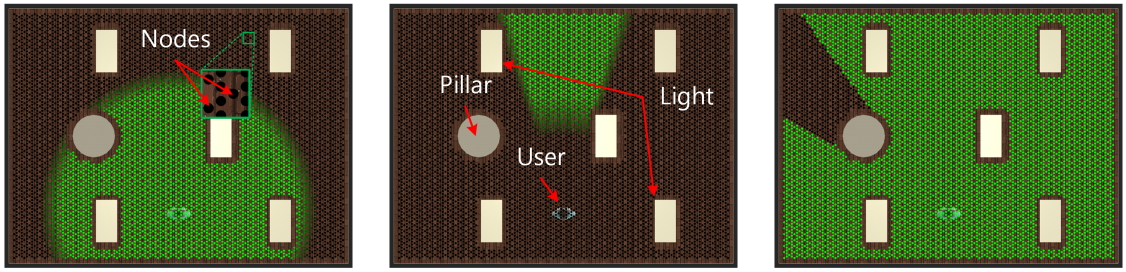
$$ \mathrm{EV}_6(N) \coloneqq (1-\omega_6)\cdot \mathrm{EV}_5({}_sN) + \frac{\omega_6}{{}_{actual}T({}_sN)+1} \tag{4.16} $$

$$ \mathit{Destination\ Node} \coloneqq \underset{\forall_s N}{\arg\max}\,(\mathrm{EV}_6({}_sN)) \tag{4.17} $$

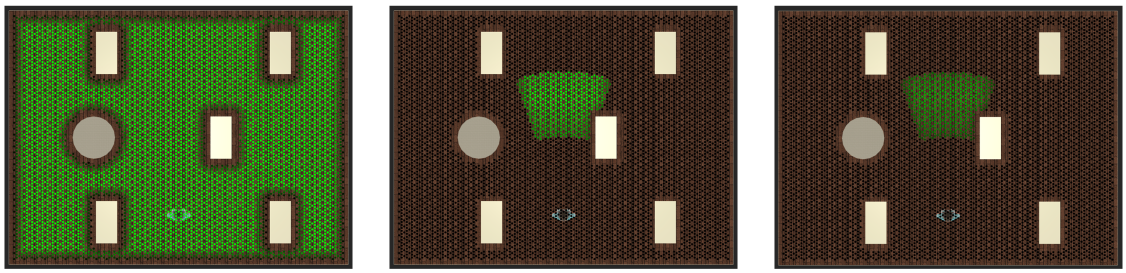
In (4.16) and (4.17), ${}_sN$ denotes candidate nodes with higher evaluation results than the other nodes in (4.15).

4.2.8 Application example of the proposed algorithm

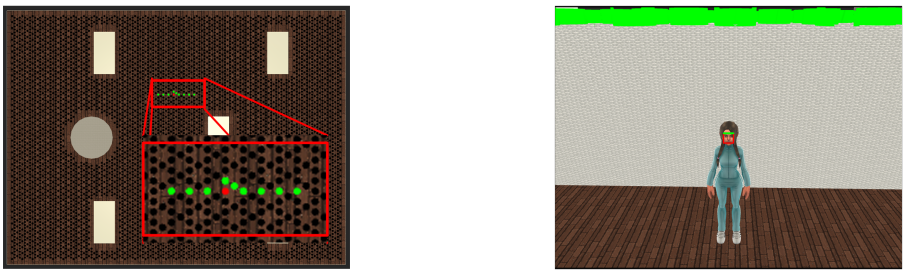
In this section, the proposed algorithm is explained with a concrete example in a simulation environment which was programmed using the Unity 3D game engine. The target application in the example is the OKAO Vision which can support face detection [70,71]. A Logitech C920 WebCAM was modeled for the target application. In the simulation, the camera had a diagonal field of view of 78° and captured with



(a) Distance (b) Included angle (c) Occlusion



(d) Obstacles (e) Integration results (f) After WA filter



(g) Candidates and destination (h) Result of face detection at destination

Fig. 4.8 The evaluation results of each process and the result of application

a resolution of 480 × 360. The experimental room was 8.0 m × 6.0 m × 2.8 m, and the field was installed on the ceiling. In the room, there was a pillar, and five lights were fixed to the ceiling, as shown in Fig. 4.8. Fig. 4.8(a) and Fig. 4.8(b) show the evaluation results obtained using evaluation functions of the distance factor and the included angle factor, respectively. The light green nodes indicate that the nodes obtained a high evaluation value. According to a decline in the evaluation value, the nodes have a dark green color. Fig. 4.8(c) and Fig. 4.8(d) show the results for the third and fourth factors, and Fig. 4.8(e) shows the integrated results of the basic influential four factors. The integrated results change according to the WA filter (Fig. 4.8(f)). The destination candidates are selected by (4.15) as shown in Fig. 4.8(g). Finally, the destination is determined by (4.17). The determined destination is the red node in Fig. 4.8(g), and the result of the face detection at the destination is illustrated in Fig. 4.8(h).

4.3 Determination of parameters in the proposed algorithm

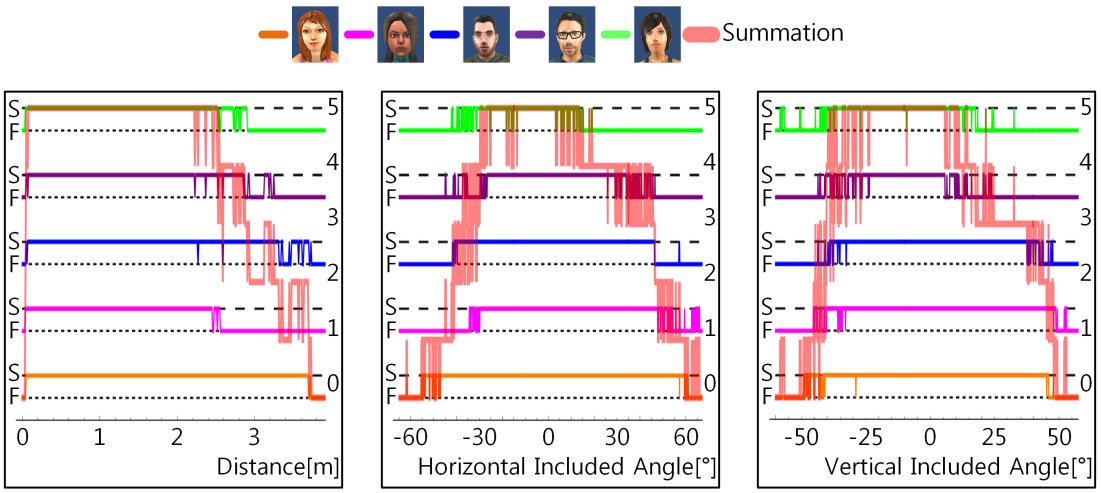
In the proposed algorithm, various parameters are decided by the application and device. To determine the parameters related to the distance and included angle factors, the following experiments was performed. The camera object captured the target while varying the distance from the target and rotating the horizontal and vertical directions. The captured images were processed by the face detection application. To obtain the proper parameters, we performed this experiment for five targets. The experimental results are shown in Fig. 4.9. The vertical axis of the graphs in Fig. 4.9 show the success or failure of the face detection, and the translucent red line indicates the number of successful targets. The parameters related to the object factor on the field were determined empirically based on the minimum straight distance between two nodes. The weight factors for integration and the fifth factor were also decided empirically. These parameters are shown in Table 4.1.

To find the proper parameters for ω_6 and the number of candidates (NoC), the following experiment was performed. This experiment was performed in four different environments with different complexity but equal size. The environments are shown in Fig. 4.10. Here, The first environment is empty space, and the second environment has moderate complexity that consists of a pillar and five lights. Third environment has higher complexity than the second environment. There are two pillars and thir-

Table 4.1 Parameters used in the evaluation function

Parameters	Decided values
D_m, D_M, α_1	0.5 m, 2.6 m, 0.5 m
${}_h\mathrm{IA}_M$, ${}_h\alpha_2$, ${}_v\mathrm{IA}_M$, ${}_v\alpha_2$	30.0°, 20.0°,20.0°, 5.0°
${}_oD_m$, α_4	0.1 m, 0.3 m
ω_1, ω_2, ω_3, ω_4	0.152, 0.758, 0.015, 0.075
ω_5^1, ω_5^2, ω_5^3	0.500, 0.300, 0.200

teen lights in third environment. Fourth environment has the highest complexity among the given environments. Note that the shape of the fourth environment is not a simple rectangle. There are walls inside the space that obstruct face detection and the movement of MoMo. This experiment was performed in the following sequence. There is a user in the environment, and the user's position and direction are

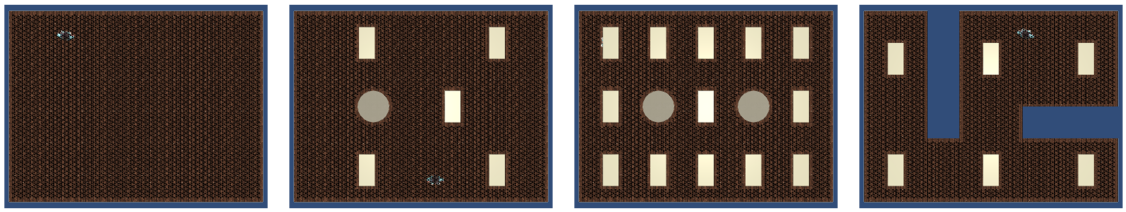


(a) Face detection results according to distance from target

(b) Face detection results according to horizontal included angle

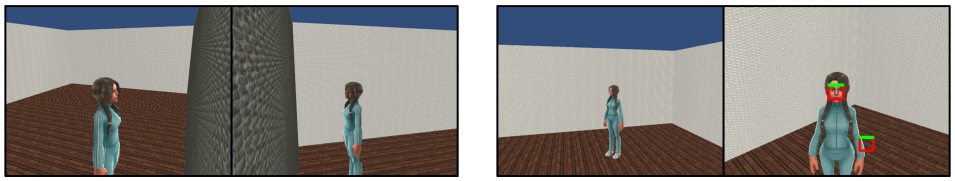
(c) Face detection results according to vertical included angle

Fig. 4.9 Face detection results according to distances and included angles



(a) Map1 (b) Map2 (c) Map3 (d) Map4

Fig. 4.10 Environments used in the simulation experiment

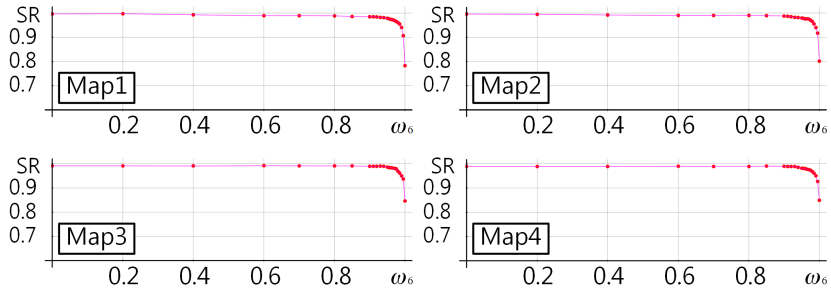


(a) Exception cases (b) Failure cases

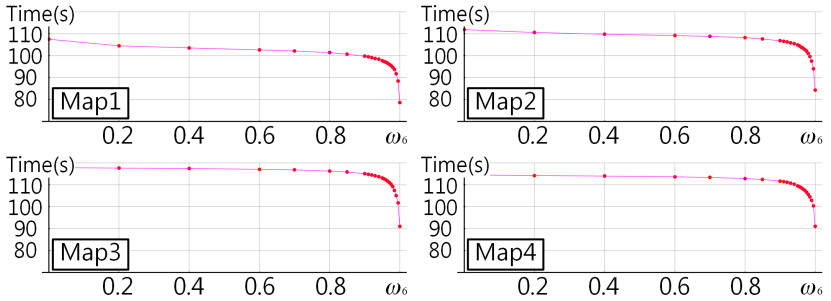
Fig. 4.11 Examples of exception and failure cases

changed in a randomly generated predetermined order. Movable DIND moves to the destination obtained by the proposed algorithm. The connected camera device to the movable DIND captures target images. The captured images are processed by the face detection software.

The data used for the analysis of the experimental results were restricted to cases wherein the proposed algorithm could find the destination of movable DIND. In some cases, the evaluation values of all nodes are 0, and the proposed algorithm cannot determine the destination. For example, when a user is placed nearby and toward the wall, there is no suitable location for face detection. In the analysis, the failure cases for face detection included simple failures and incorrect faulty detection cases. Examples of exception and failure situations are shown in Fig. 4.11. Fig. 4.11(a) shows cases in which there was no destination. The left image in Fig. 4.11(b) shows a case in which movable DIND was not located at a suitable location, and the right image shows an example of the incorrect detection.



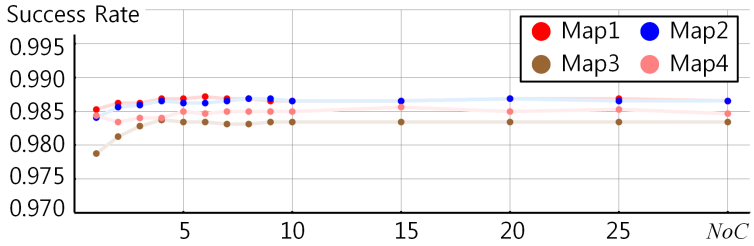
(a) Success rate according to to transition of ω_6



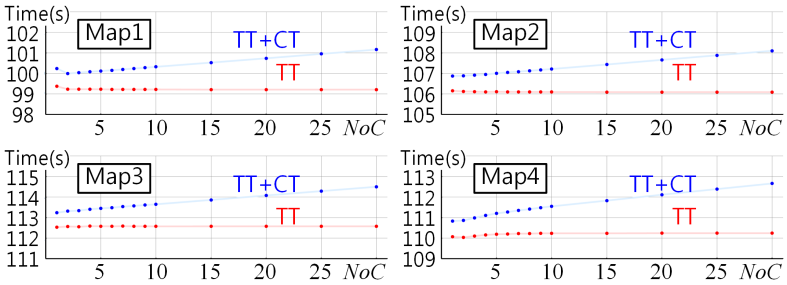
(b) Travel Time according to to transition of ω_6

Fig. 4.12 Experimental results for parameter ω_6

In next experiments, the performance of face detection according to transition of ω_6 and NoC were compared. First, experiment to determine the proper value of ω_6 was performed. In this experiment, the value of NoC was fixed at 6 that is sufficiently large not to affect the success rate of face detection. Then, an experiment for deciding NoC was performed. In this experiment, the ω_6 is determined as decided value in the previous experiment. The resultant graphs are shown in Fig. 4.12. The graphs in Fig. 4.12(a) show the relationship between ω_6 and the success rate of face detection, and the graphs in Fig. 4.12(b) show the relationship between ω_6 and travel time. The proper value of ω_6 was determined based on the success rate and travel time. Between these two conditions, the success rate showed higher importance than travel time. We determined a lower limitation for success rate (98.5 %). Among the nodes that satisfy



(a) Success rate according to to transition of NoC



(b) Travel Time and Computation Time according to to transition of NoC

Fig. 4.13 Results of the experiment for NoC parameter

the first condition, the value of ω_6 with the shortest travel time was selected. As a result, the decided values of ω_6 differed according to the given environments. Next, we performed an experiment for NoC with the same process. In this experiment, the travel time and computation time according to NoC were compared. The compared results are shown in Fig. 4.13. As shown in Fig. 4.13(a), there was no significant difference in success rate according to changing NoC. Therefore, we decided NoC based on the travel time and computation time shown in Fig. 4.13(b). The values of ω_6 and NoC according to the environment are given in Table 4.2.

Table 4.2 Parameters of ω_6 and NoC

Parameters	Decided values			
	Map1	Map2	Map3	Map4
ω_6	0.91	0.91	0.955	0.94
NoC	2	1	1	1

4.4 Simulation experiment to verify the proposed algorithm

Here, the simulation experiments conducted to verify the performance of the proposed algorithm is described. As mentioned above, the proposed algorithm gives the best effort locations of movable DINDs according to the current spatial situation. This indicates that the previous situation cannot affect to the output of the proposed algorithm. Therefore, the simulation experiment was performed as follow sequence. At first, the camera device that is mounted on movable DIND is placed on their predecided initial location, and the target's position and direction is generated randomly. Then, the algorithm calculates the best effort location of movable DIND for face detection mentioned in the previous section. In this experiment, it is assumed that the target's direction was the front direction of the user's face.

The experiments consisted of two phases. The first phase of the experiment was performed to verify the appropriateness of the evaluation functions for the performance of provided service. In this experiment, movable DIND with camera device moved to all available nodes, and then camera device captured the target. The results of this experiment are illustrated in Fig. 4.14. The magenta area in the figures indicates that the application was successful in that area. The blue area indicates that the results of the integration results of basic influential four factors. The color of this area becomes lighter according to the decline in the evaluated result. The red point is the decided destination obtained by proposed method. As shown by this result, most of the blue area is included in the magenta area. Note that, the dark blue area is included in the successful area. Table 4.3 shows the success rates according to the evaluation results. As can be seen in Table 4.3, the high scoring nodes obtained by proposed algorithm achieved a high application success rate regardless of types of environment. As a result, it was verified that the evaluation functions of the proposed

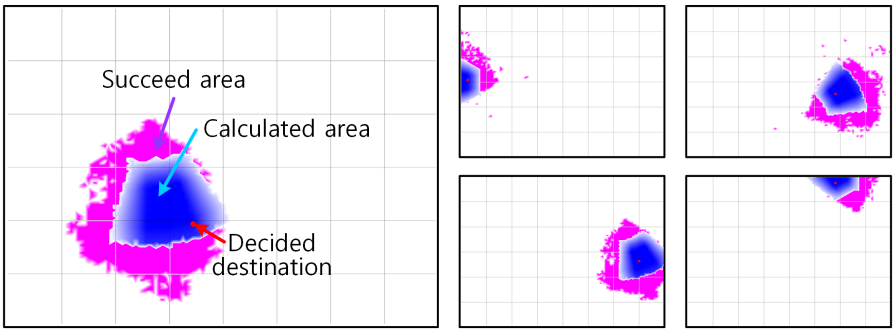


Fig. 4.14 Floor plans of successful location and evaluation results.

Table 4.3 Success rate according to the results of the evaluation functions

Evaluation Result	Environment			
	Map1	Map2	Map3	Map4
$0.0 < \mathrm{EV}_{int}(N) \leq 0.1$	0.6	0.56	0.42	0.46
$0.1 < \mathrm{EV}_{int}(N) \leq 0.2$	0.7	0.69	0.68	0.77
$0.2 < \mathrm{EV}_{int}(N) \leq 0.3$	0.81	0.82	0.78	0.81
$0.3 < \mathrm{EV}_{int}(N) \leq 0.4$	0.81	0.86	0.79	0.81
$0.4 < \mathrm{EV}_{int}(N) \leq 0.5$	0.85	0.87	0.82	0.85
$0.5 < \mathrm{EV}_{int}(N) \leq 0.6$	0.88	0.91	0.88	0.88
$0.6 < \mathrm{EV}_{int}(N) \leq 0.7$	0.94	0.94	0.95	0.94
$0.7 < \mathrm{EV}_{int}(N) \leq 0.8$	0.98	0.98	0.97	0.96
$0.8 < \mathrm{EV}_{int}(N) \leq 0.9$	0.99	0.97	0.99	0.98
$0.9 < \mathrm{EV}_{int}(N) \leq 1.0$	1	0.99	1	0.99

method are valid for determining the best effort location of movable DIND.

The next phase of the experiment was performed to compare application performance between conventional iSpace and R+iSpace with the proposed method. There were seven cameras in the iSpace environments, which were used for the face detection. In this experiment, it was determined that the face detection was successful when the face was detected by one or more cameras. The experimental results are shown in Table 4.4 and Table 4.5. Table 4.4 shows the results of all situations, and Table 4.5

Table 4.4 Success rate comparison results between conventional iSpace and R+iSpace for all situations

	iSpace	R+iSpace
Map1	$0.365_{(3647/10000)}$	$0.916_{(9158/10000)}$
Map2	$0.356_{(3556/10000)}$	$0.903_{(9024/10000)}$
Map3	$0.352_{(3516/10000)}$	$0.889_{(8886/10000)}$
Map4	$0.319_{(3185/10000)}$	$0.764_{(7637/10000)}$

Table 4.5 Success rate comparison results between conventional iSpace and R+iSpace without the exception cases

	iSpace	R+iSpace
Map1	$0.375_{(3491/9301)}$	$0.985_{(9158/9301)}$
Map2	$0.368_{(3389/9208)}$	$0.980_{(9024/9208)}$
Map3	$0.365_{(3306/9051)}$	$0.982_{(8886/9051)}$
Map4	$0.354_{(2764/7800)}$	$0.979_{(7634/7800)}$

shows the situation in which the destination exists. The destination of device does not exist when the target user is located on close position to walls and is facing to the walls. Among the failure cases of R+iSpace, when the destination of the movable DIND exists, 10.64 % of cases failed due to incorrect detection. In most failure cases, with the exception of the incorrect detection cases, the target was located close to the walls. This indicates that the evaluation result for the destination node was low in such cases. The average evaluation results for these cases is approximately 0.38. The results of these cases can be improved by changing the parameters. As demonstrated by these results, the success rate for R+iSpace obtained with the proposed algorithm increased by 62.0 % over conventional iSpace.

4.5 Discussion and conclusion

In this chapter, a method to determine the best effort location of movable DIND in R+iSpace was proposed. The proposed method considers the relationship between a target and a device. The target's position and direction are determined by the

application in the proposed method, and the proposed method decides the appropriate location's range according to the specification of the application and the device. The proposed method also considers the travel time to the destination. This chapter has verified the proposed method with simulation experiments conducted using the face detection. Through the proposed method, best effort locations of devices to provide service according to the given current spatial situations can be obtained. The weight factors for weighted average is determined empirically. According to the simulation experiment result, the provided service can be guaranteed its consistent high-performance.

Chapter 5

Path Generation without Collision and Deadlock Situation

Typically, lots of MoMos adhere to the same field, and the presence of the other MoMos interferes in own movement. If a MoMo moves without any consideration of the other MoMos, the collision will occur. Even though MoMo decides own path that can avoid a collision with the other MoMos, the deadlock situation can occur. Therefore, MoMo should equip a path generation method that considered the collision and deadlock situation. This chapter introduces the path generation algorithm that can avoid collision and deadlock situation. The proposed algorithm is designed to target MoMo2, but basic concept of this algorithm can be applied to MoMo3. The proposed algorithm comprises six layers, and the next position of MoMo is achieved by integration of the results of six layers. This chapter is organized as follows. In the first section of this chapter, the concept and coordinate system of the field for MoMo2 is introduced. Then, the detailed description and calculation method is introduced in Section 5.2. The proposed algorithm is verified by simulation experiment. The three different field that has different complexity was used in the simulation experiment.

5.1 Background

This section introduces the problem in path generation. The shortest path of typical mobile robot is achieved by A* algorithm easily. However, in MoMo cases, there are

many problems that are caused due to iSpace's property. As above-mentioned, iSpace employed distributed system architecture for its reliability and scalability. However, the absence of the supervisor results in the difficulty to design a path generation algorithm. In this section, the problems situation are discussed with detailed examples, and then, the coordinate system and nodes for the proposed algorithm are introduced.

5.1.1 Problem statement

As it was mentioned above, numerous MoMos are installed on a single field. Therefore, the path planning algorithm should consider both collision and deadlock situations. The collision situation can be avoided easily by the communication. MoMos can achieve the other MoMo's location via shared memory that contains all MoMo's current location. However, even though MoMo decides own path with consideration of the current locations of the other MoMos, the collision can sometimes occur. Fig. 5.1 shows an example of this situation. It occurred due to the shared memory is accessible by two or more MoMos at the same time. To solve this problem, the preemptive shared memory was proposed. In this scheme, only one MoMo can access to the

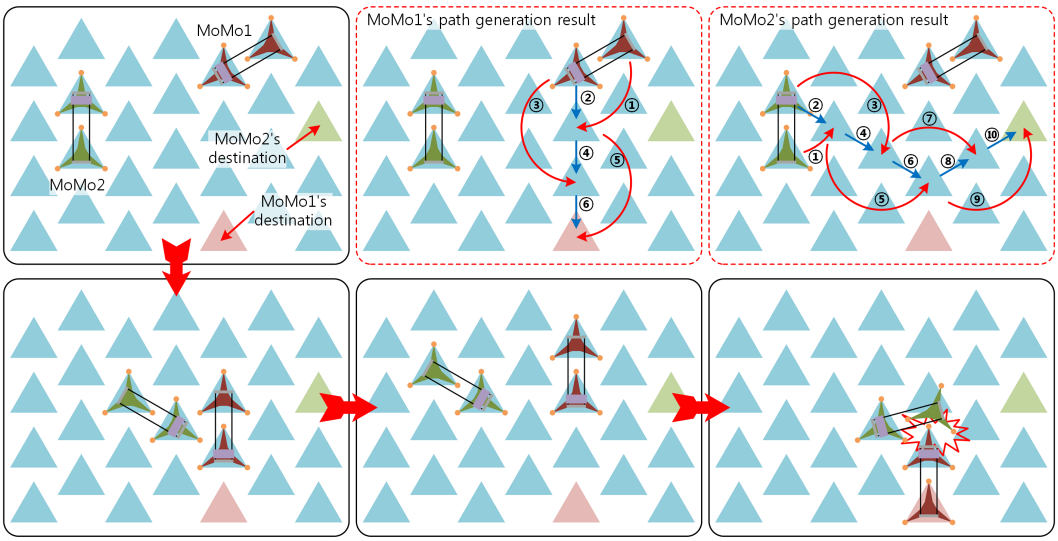


Fig. 5.1 Collision situation example

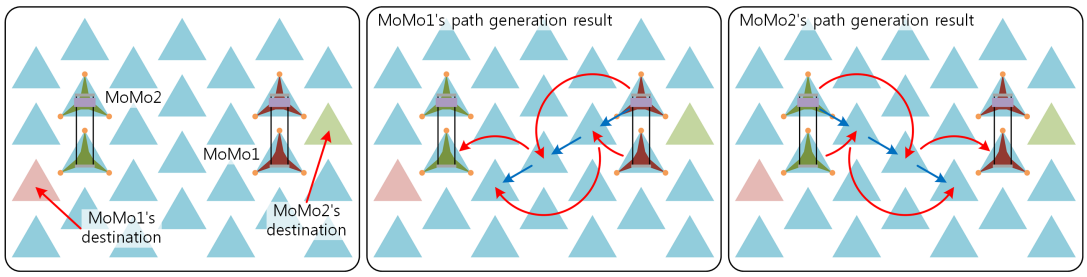


Fig. 5.2 Deadlock situation example

preemptive shared memory, and the memory is held by the accessed MoMo. The memory is released when MoMo complete the calculation of path generation, and at the same time, MoMo's next position is stored. Through this process, MoMos can achieve conflict-free path generation. However, this algorithm cannot avoid the deadlock situation, the paths to the goals of MoMos are occupied by the other MoMos. An example of deadlock preventing two MoMos from finding their goal is shown in Fig. 5.2.

A lot of research on multi robot system proposed a path generation method with good performance [72, 73]. However, these methods have a big problem to be employed in MoMo. MoMo is controlled by low-cost single board computer, such as a Raspberry Pi. The proposed algorithms are too heavy to be used in MoMo in real time. Additionally, the destination of MoMos is changed frequently according to situation, and it requires new path generation.

5.1.2 Coordinate system and node setting for the proposed algorithm

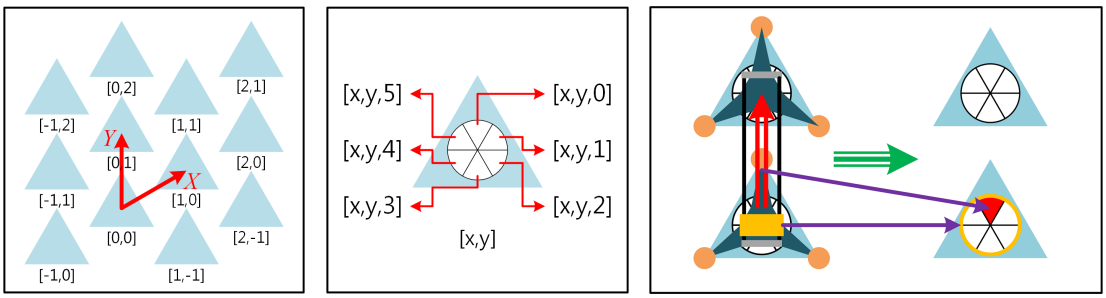
As it is shown in Fig. 3.18, the field for MoMo2 is composed of numerous stations, and its arrangement is distinctive. They are aligned in diagonal direction that is tilted 60°. Therefore, using the typical orthogonal coordinate system is not suitable for this field. The included angle of two axes of employed coordinate system is 60°, and the directions of the axes are parallel with the stations' aligned direction. The arrangement of stations and the coordinate system are illustrated in Fig. 5.3(a). As it was mentioned in Chapter 4, a node was defined as the device's stable location. In case of MoMo2, the node's location is different from MoMo3; the node's location is

equal to the station's location according to the definition of Chapter 4. However, if the node's position is defined as the same position of the station, a problem will occur. A node wall have six postures according to opposite screw module's location, and it will result in an error while the calculation of A*. Therefore, the proposed algorithm makes six node on a station, and it is illustrated in Fig. 5.3(b). The current node (N_c) of MoMo2 is presented in Fig. 5.3(c). Through the new node setting, the device position and MoMo's posture are easily determined from the current node information.

The adjacent nodes ($\mathrm{Ad}(N_c)$) is defined as the nodes that are reachable by a single motion (e.g., rotation, translation) from current node (N_c). It is illustrated in Fig. 5.4.

The following equations and notations are used throughout this paper. As above-mentioned, only adjacent nodes are involved in the calculations for each layer. $\mathrm{Ad}(N_c)$ indicates all nodes on the station S_{x_c,y_c} including N_c and $\mathrm{Tr}(N_c)$. $\mathrm{Tr}(N_c)$ is a node that can be reached by translation motion from N_c. It is calculated as follows.

$$\mathrm{Tr}\,(N_c) = [\mathrm{Tr}_{x_c,n_c}, \mathrm{Tr}_{y_c,n_c}, \mathrm{Tr}_{n_c}] \tag{5.1}$$



(a) Coordinate system and station arrangements (b) Nodes on station ($S_{x,y}$) (c) Relationship between the MoMo2 and current node (N_c)

Fig. 5.3 Field coordinate system and the arrangement of nodes and stations

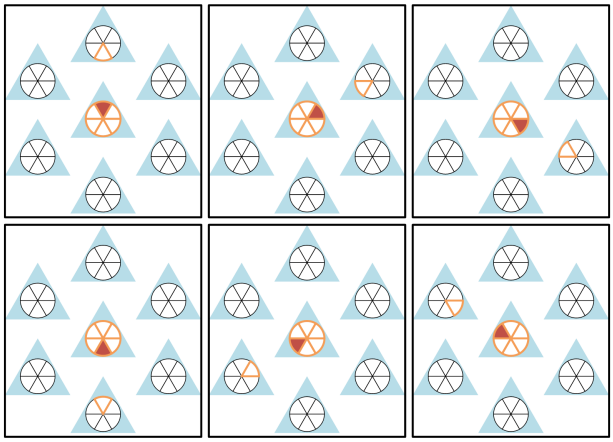


Fig. 5.4 Adjacent nodes ($\mathrm{Ad}(N_c)$) and current node (N_c)

Where,

$$\mathrm{Tr}_{x_c,n_c} = x_c + \frac{2}{\sqrt{3}} \cdot \sin(\frac{\pi}{6} \cdot n_c) \tag{5.2}$$

$$\mathrm{Tr}_{y_c,n_c} = y_c + \frac{2}{\sqrt{3}} \cdot \sin(\frac{\pi}{6} \cdot (n_c + 2)) \tag{5.3}$$

$$\mathrm{Tr}_{n_c} = \begin{cases} n_c + 3 & \text{If } n_c < 3 \\ n_c - 3 & \text{If } n_c \geq 3 \end{cases} \tag{5.4}$$

The function $\mathrm{Dist}\left((x_1, y_1), (x_2, y_2)\right)$ calculated by (5.5), gives the linear distance between (x_1, y_1) and (x_2, y_2) in field coordinate system.

$$\mathrm{Dist}\left((x_1, y_1), (x_2, y_2)\right) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (x_1 - x_2) \cdot (y_1 - y_2)} \tag{5.5}$$

5.2 Proposed algorithm of path generation

The proposed algorithm comprises six layers; movement cost to adjacent nodes, movement cost of valid path, movement cost to pseudo path, negotiation with the other

MoMos I & II, and appended cost for avoiding previous node. For each layer, MoMo calculates the cost of adjacent nodes according to the layer's criterion. At the last phase, the calculated costs of adjacent nodes in each layer are integrated. Finally, the node that has smallest cost value among adjacent nodes is decided as the next node. First three layer are designed to find fastest path to the destination, and remained layers are designed to avoid deadlock situation.

As it is shown above description, the proposed algorithm does not give a global complete solution. This method is close to the reactive path generation method that has high robustness to sudden environmental changes. This method has the other advantage on its computation time. Typically, the computation time of the reactive method is sufficiently fast to actualize real time systems [74].

5.2.1 Layer1: Movement cost to adjacent nodes

The criterion for first layer is the cost of reaching adjacent nodes. The cost calculation requires the cost of moving and the flag information of adjacent nodes. It was assumed that reserved stations by fixed obstacle and all MoMos are stored in the preemptive shared memory when R+iSpace is initialized. The flag information of the station, which is reserved by MoMos, is updated before MoMos move. The flag information of $N(=[x, y, n])$ is defined as follows:

$$\mathrm{Flag}_N := \begin{cases} 1 & \text{If } S_{x,y} \text{ or } S_{\mathrm{Tr}_{x,n},\mathrm{Tr}_{y,n}} \text{ is reserved.} \\ 0 & \text{Else} \end{cases} \tag{5.6}$$

MoMo retrieves these data from the shared memory, and calculates the costs of layer1. The resulting costs of N_c and $\mathrm{Tr}(N_c)$ are easily achieved by setting the cost of the current node to 0 and the resulting cost of $\mathrm{Tr}(N_c)$ to T_{tr}. The cost of the other adjacent node is the smaller of the two costs incurred by rotating clockwise or counterclockwise, calculated by (5.7) and (5.8), respectively. Here, the rotation angle

RA is restricted to five values by the structure of the field.

$$\mathrm{CW}_i = \begin{cases} 0 & \text{If } i = 0 \\ \infty & \text{If } \mathrm{Flag}_{N_{cw}(i)} = 1 \\ \mathrm{CW}_{i-1} + T_{ro} & \text{Else} \end{cases} \tag{5.7}$$

$$\mathrm{CCW}_i = \begin{cases} 0 & \text{If } i = 0 \\ \infty & \text{If } \mathrm{Flag}_{N_{ccw}(i)} = 1 \\ \mathrm{CCW}_{i-1} + T_{ro} & \text{Else} \end{cases} \tag{5.8}$$

Where,

$$\begin{bmatrix} i & = & RA/60^\circ \\ RA & = & 60^\circ, 120^\circ, \cdots, 300^\circ \end{bmatrix}$$

In (5.7) and (5.8), the subscripts $N_{cw}(i)$ and $N_{ccw}(i)$ denote the following:

$$N_{cw}(i) \coloneqq [x_c, y_c, (n_c + i) \bmod 6] \tag{5.9}$$

$$N_{ccw}(i) \coloneqq [x_c, y_c, (n_c - i) \bmod 6] \tag{5.10}$$

The final costs of layer1 are calculated as follows.

$${}_1L_{\mathrm{Ad}(N_c)} = \begin{cases} {}_1L_{\mathrm{Tr}(N_c)} \coloneqq \omega_1 \cdot T_{tr} \\ {}_1L_{[x_c,y_c,k]} \coloneqq \begin{cases} 0 & \text{If } k = n_c \\ \omega_1 \cdot \min(\mathrm{CW}_j, \mathrm{CCW}_{6-j}) + T_{sc} & \text{Else} \end{cases} \\ \text{Where,} k = 0, 1, \cdots, 5 \wedge j = (k - n) \bmod 6 \end{cases} \tag{5.11}$$

In (5.11), ω_1 is weight factor of layer1 that adjusts the scale of the results to those of the other layers. T_{sc} denotes the screw control time (the sum of T_{ls} to T_{ts}).

The calculation example is illustrated in Fig. 5.5. In this figure, ${}_1L_{[0,0,2]}$ and ${}_1L_{[0,0,4]}$ of MoMo_1 are ∞ by the other MoMos and the fixed obstacle. ${}_1L_{[0,0,3]}$, determined by (5.7), (5.8) and (5.11), is also ∞.

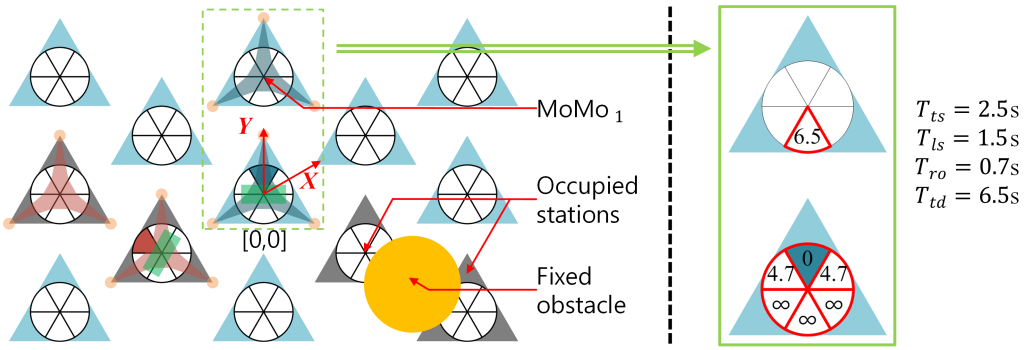


Fig. 5.5 The Example of the Calculation of layer1

5.2.2 Layer2: Movement cost of valid path

The criterion of layer2 is the cost of moving from the goal station S_{x_g,y_g} to $\mathrm{Ad}(N_c)$. The cost calculation is based on A*. The used map data in the calculation of A* contains both fixed obstacles and stations reserved by the other MoMos. The cost calculation proceeds as follows.

Step 1. Using A*, calculate the movement cost ($f_v(\mathrm{Ad}(N_c))$) from S_{x_g,y_g} to $\mathrm{Ad}(N_c)$.
Step 2. If no valid path exists between the goal station and the current node, all of $f_v(\mathrm{Ad}(N_c))$ are set to 0.
Step 3. Subtract $f_v(N_c)$ from all $f_v(\mathrm{Ad}(N_c))$.
Step 4. Multiply all $f_v(\mathrm{Ad}(N_c))$ by a weight factor ω_1.

To clarify the differential between the costs of layer2 and those of the other layers when integrating the results of all layers, the weight factor ω_2 should be larger than $2 \cdot \omega_1$. The cost of layer2 is then calculated as follows.

$$ {}_2L_{\mathrm{Ad}(N_c)} := \omega_1 \cdot (f_v(\mathrm{Ad}(N_c)) - f_v(N_c)) \tag{5.12} $$

If the integration procedure is restricted to layer1 and layer2, the algorithm has same result with the conventional path generating method. Therefore, this implementation is adopted when comparing the performances of the proposed algorithm and typical A* algorithm.

5.2.3 Layer3: Movement cost to pseudo path

The criterion of layer3, like that of layer2, is the movement cost ($f_p\left(\mathrm{Ad}(N_c)\right)$) from the goal station to the adjacent nodes. However, movement under layer3 is based on different map data. The used map data in this layer contains only the unmovable or uncontrollable objects, such as fixed obstacles. Hereafter, the valid and pseudo paths indicate the paths generated in layer2 and layer3, respectively. If there is no pseudo path, it indicate that the current destination is unreachable location. In this situation, the path planning component requests a new destination to the best location determination component. The final result of this layer is calculated as follows.

$$ {}_3L_{\mathrm{Ad}(N_c)} \coloneqq \omega_3 \cdot \left(f_p\left(\mathrm{Ad}(N_c)\right) - f_p\left(N_c\right)\right) \tag{5.13} $$

This layer allows MoMo to select an appropriate next node when there is no valid path. When a valid path exists, the next node should be determined from that path. This can be achieved by proper values of ω_1, ω_2, and ω_3. In the absence of a valid path, the algorithm selects the pseudo path by summation of the results of layer1, layer2 and layer3 in a common situation. By implementing this layer, MoMo can find a pseudo path when the valid paths are blocked by the other MoMos. However, since the positions of the obstructing MoMos are not altered, this only provides a temporary solution. To solve this situation, layer4 and layer5 are designed.

5.2.4 Layer4: Negotiation with the other MoMos I

In this section, a situation that a MoMo is placed on a valid path of the other MoMos is considered. Each MoMo is denoted sequentially by MoMo_A and MoMo_B. After complete to calculate costs of layer1, layer2 and layer3, all MoMos gradually increases and decreases ω_4 under the following conditions.

Increment conditions $\quad \omega_4^j = 0 \ (\ \forall j \ \wedge \ j \neq i \ ) \wedge \sum \left(\left| {}_2L^i_{\mathrm{Ad}(N_c)} \right| + \left| {}_3L^i_{\mathrm{Ad}(N_c)} \right| \right) = 0$

Decrement conditions $\quad \sum \left| {}_2L^i_{\mathrm{Ad}(N_c)} \right| \neq 0 \ \wedge \ \omega_3^i > 0$

Layer4 is deactivated when ω_4 of all other MoMos are set to 0. In the situation of this section, MoMo_B increases ω_4^B, and then layer4 of MoMo_A is activated. The primary

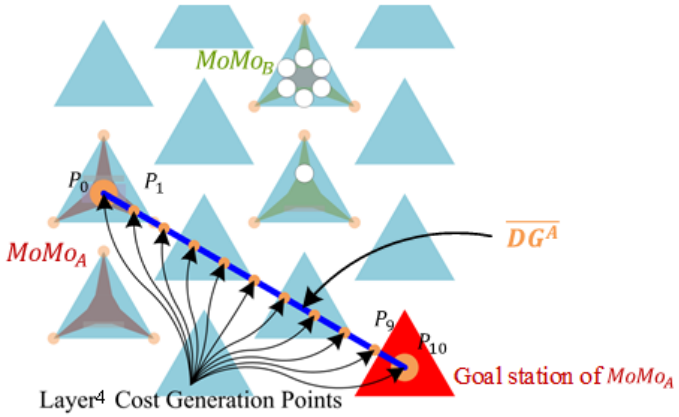


Fig. 5.6 An example situation that layer4 is activated

variables of layer4 are illustrated in Fig. 5.6 and defined as follows:

- $\overline{\mathrm{DG}^i}$: Linear line segment between device's location and goal station of MoMo_i.
- P_j^i : jth discrete point along the 10-segment interval $\overline{\mathrm{DG}^i}$.
- $\mathrm{FC}\,(N)$: The position of node N in field coordination.

Among above description of variables, $\mathrm{FC}\,([x, y, n])$ is defined as follows:

$$\mathrm{FC}\,([x, y, n]) := \left[x + \frac{2\sqrt{3}}{9} \cdot \sin(\frac{\pi}{6} \cdot n), y + \frac{2\sqrt{3}}{9} \cdot \sin(\frac{\pi}{6} \cdot (n+2)) \right] \tag{5.14}$$

This layer was imposed to avoid the situation that a MoMo is placed on the other MoMo's valid path. MoMo_B should move away from $\overline{\mathrm{DG}^A}$. We defines a cost function for this layer, generates offset cost of $\mathrm{Ad}(N_c^A)$ from P_j^A and ω_4^A. Among the adjacent nodes, the lowest-cost node is selected as the next node in the integrating layers sequence. Therefore, the cost function should satisfy following conditions.

- The cost function should be a strictly decreasing function for $\mathrm{Dist}(\mathrm{FC}\,(N)\,, P_j^k)$.
- The limit of the cost function as $\mathrm{Dist}(\mathrm{FC}\,(N)\,, P_j^k)$ approaches infinity should be 0.
- The difference between result cost of two nodes should be increased according

to the increase of ω_4^k.

- The result of the cost function should be 0 when ω_4^k is 0.

The above conditions are satisfied by the following cost function.

$$\mathrm{Cost}_1(N,k) := \sum_{j=0}^{10} \frac{\omega_4^k \cdot f(j)}{\left(\mathrm{Dist}\left(\mathrm{FC}(N), P_j^k\right)+1\right)^2} \tag{5.15}$$

Where,

$$f(j) = 0.40462 + \frac{1000}{(j+11.8869)^3} \tag{5.16}$$

Note that $f(j)$ is decreases with increasing j. The function was designed such that $f(0) \approx 1$ and $f(10) \approx 0.5$. The final cost of layer4 (plotted as a contour map in Fig. 5.7) is given by

$${}_4L^i_{\mathrm{Ad}(N_c^i)} := \sum_{\forall k \ \wedge \ k \neq i} \mathrm{Cost}_1\left(\mathrm{Ad}(N_c^i), k\right) \tag{5.17}$$

5.2.5 Layer5: Negotiation with the other MoMos II

Through layer4, MoMo can solve most situations which was considered in previous section. However, in special circumstances, such as that illustrated in Fig. 5.8, cannot

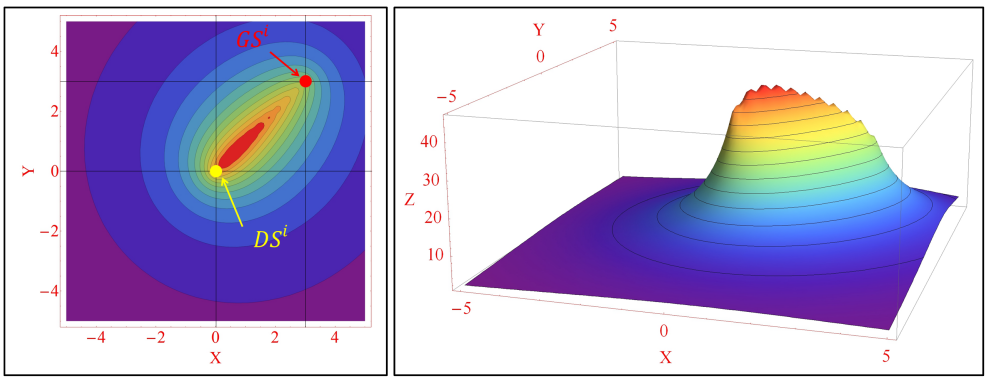


Fig. 5.7 Contour map of cost calculated by (5.17) ($\omega_4^i = 20$)

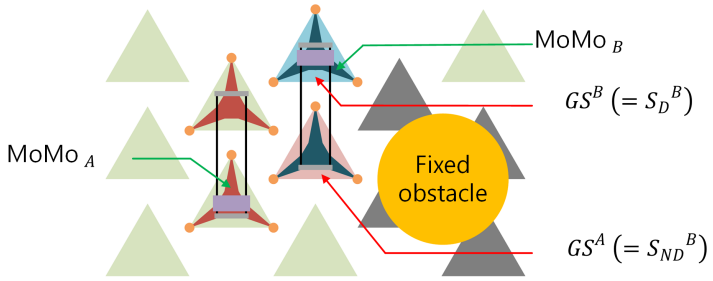


Fig. 5.8 An Example of an unsolved obstruct situation

solved by layer4. MoMo$_A$ cannot find valid path in Fig. 5.8, and, MoMo$_B$ had arrived on its goal station and ${S_{\mathrm{ND}}}^B$ is same with the goal station of MoMo$_A$. MoMo$_B$ will usually rotate its body to avoid occupying MoMo$_A$'s goal station by layer4. However, the movement of MoMo$_B$ is restricted by MoMo$_A$ and the fixed obstacle. Therefore, MoMo$_B$ cannot shift to other nodes, and MoMo$_A$ cannot reach its goal. To solve this special situation, layer5 is desinged. In this situation, ω_5 is increased. The detailed condition of increment of ω_5 is that all ${}_4L^i_{\mathrm{Ad}(N_c^i)}$ exceed a certain $\mathrm{Cost}_{threshold}$ due to increment of ω_4 of the other MoMos. The increased ω_5 influences the layer5 costs of the other MoMos. The offset cost of MoMo$_i$ generated by layer5 is given by

$$\mathrm{Cost}_2\left(N,k\right) := \frac{\omega_5}{\left(\mathrm{Dist}\left(\mathrm{FC}\left(N\right),{S_{\mathrm{D}}}^k\right)+1\right)^2} + \frac{\omega_5}{\left(\mathrm{Dist}\left(\mathrm{FC}\left(N\right),{S_{\mathrm{ND}}}^k\right)+1\right)^2} \tag{5.18}$$

This result causes MoMo$_A$ to select the farthest node from ${S_{\mathrm{D}}}^B$ and ${S_{\mathrm{ND}}}^B$. If the cost generated by layer5 releases the movement restriction of MoMo$_B$, MoMo$_B$ can avoid obstructing MoMo$_A$'s path. Consequently, MoMo$_A$ can find a valid path, and ω_4^A is decreased. Then, subsequent reduction in ω_5 occurs. The final cost of layer5 is calculated by (5.19).

$${}_5L^i_{\mathrm{Ad}(N_c^i)} := \sum_{\forall k \ \wedge \ k\neq i} \mathrm{Cost}_2\left(\mathrm{Ad}(N_c^i),k\right) \tag{5.19}$$

5.2.6 Layer6: Appended cost for avoiding previous node (N_p)

The costs of layer4 and layer5 largely solve the deadlock situation. However, the cost depends on the node's position. Unintentional repetitive movement between two nodes causes delays in whole system. Therefore, such motions are reduced by an additional cost to the previous node, which dissipates after a certain period of time. This additional cost is generated in layer6, and is given by

$$ {}_6L_{N_p} := \frac{\omega_6}{(t+1)^2} \tag{5.20} $$

where t is the time elapsed since the completion of the previous motion (in seconds).

5.2.7 Integration of all layer's results

The final procedure of proposed algorithm is integration of the cost results of all layers. The integration is calculated by (5.21). After integration algorithm selects the lowest cost node as MoMo's next node. Then, algorithm determines the motion for reaching the node. The algorithm calculates the stations that through passed during the motion, and stores the stations to preemptive shared memory as the reserved stations of the MoMo. Lastly, release the preemptive shared memory, then MoMo rearrange own position to the next node.

$$ L^i_{\mathrm{Ad}(N_c)} := \sum\nolimits_{l=0}^{5} {}_lL^i_{\mathrm{Ad}(N_c)} \tag{5.21} $$

5.3 Simulation experiment

The proposed algorithm was verified by simulation experiments. Two aspects of proposed algorithm were evaluated. First, the success rate of a specified task, i.e., that all devices arrive at their goal stations within a certain time, was compared between typical A* method and the proposed algorithm. Next, the actual movement costs were compared according to used layers. The experiment will be described in detail later. In the proposed algorithm has several important parameters, such as the weight factors and cost threshold. The value of these parameters affect to the

performance of algorithm. In this experiment, they are determined empirically. This indicates the determined values are not optimal values. Their optimal values can be achieved by various methods, such as genetic algorithm, bees algorithm, etc. The experimental parameters were set to the following values:

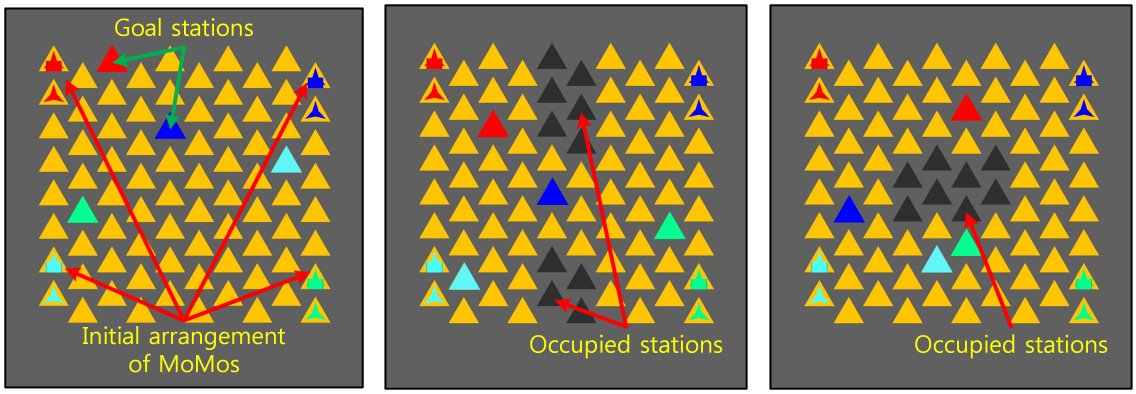
- $\omega_1 = 0.05$, $\omega_2 = 0.2$, $\omega_3 = 0.08$, $\max(\omega_4) = 200$, $\max(\omega_5) = 500$, $\omega_6 = 10$
- Rate of increase / decrease of ω_4 : 1.1 / 0.8
- Rate of increase / decrease of ω_5 : 1.05 / 0.4
- $\text{Cost}_{threshold} = 100$

5.3.1 Overview of experiment

Four MoMos are used in the experiment, and they are moving on three different fields. All fields consisted of 80 stations arranged in a 10×8 array. The first field contained no fixed obstacles; a few fixed obstacles occupied the second and third fields. The three fields are shown in Fig. 5.9. Initially, each MoMo was located at the edge of the field that is shown in Fig. 5.9(a), and was assigned a goal station by the simulator. A task was considered successful if all MoMos arrived at their goal stations within the specified time. Based on the actual movement of MoMos and the time required for device control, the task time was set to 10 minutes. Device control was assumed to be performed after every motion, including maintenance of the current node. Once the goal stations were assigned, the simulator calculated the reference costs for performance comparison. The reference cost, defined as the minimum cost of all situations, was determined up to $f_p(N_c)$. With the reference cost determined, all MoMos began moving toward their goal stations. During the movement, the simulator recorded the actual movement costs of all MoMos. After task completion, the positions and weight factors of the MoMos were reset to their initial values, and a new task was assigned to all MoMos. This process was repeated 2000 times. The task goals were predetermined at random.

5.3.2 Result of experiment

The success rates of the proposed and conventional path generating algorithms are summarized in Table 5.1. Clearly, the success rate was higher in the proposed algorithm than in A* algorithm. This indicates that the proposed algorithm frequently



(a) $Field_A$ (b) $Field_B$ (c) $Field_C$

Fig. 5.9 Used fields in the experiment

Table 5.1 Success rate comparison between typical A* algorithm and the proposed algorithm

	A* Algorithm	The proposed algorithm
$Field_A$	$85.2\,\%_{(1704/2000)}$	$100\,\%_{(2000/2000)}$
$Field_B$	$53.2\,\%_{(1064/2000)}$	$99.5\,\%_{(1990/2000)}$
$Field_C$	$58.5\,\%_{(1169/2000)}$	$99.2\,\%_{(1983/2000)}$
Average	$65.6\,\%_{(3937/6000)}$	$99.6\,\%_{(5973/6000)}$

solves the deadlock situation. However, non-optimal values of the main parameters led to some failure cases in the proposed algorithm. For instance, if MoMo_A's path was blocked by MoMo_B and MoMo_A occupied adjacent stations of MoMo_B, ω_4^A and ω_5^B are increased. The increased ω_5^B forced MoMo_A to retreat, and then ω_4^A forced MoMo_B to retreat. MoMo_A then found a valid path and moved to its goal station, while MoMo_A decreased the ω_4^A. However, in some cases, before MoMo_A arrived, MoMo_B returned to its original position, and the interplay between MoMo_A and MoMo_B was repeated. This problem is solvable by adjusting $\max(\omega)$ and the increase or decrease rate of ω to proper values. The effectiveness of the layers was evaluated in the following experiment. The MoMos selected their next node under

Table 5.2 Success rate and cost ratio comparison according to used layers

		$Method_1$	$Method_2$	$Method_3$
$Field_A$	Success Rate	96.8 %	96.9 %	100 %
	Cost Ratio	1.29	1.27	1.19
$Field_B$	Success Rate	93.6 %	94.1 %	99.5 %
	Cost Ratio	1.58	1.59	1.38
$Field_C$	Success Rate	94.7 %	95.3 %	99.2 %
	Cost Ratio	1.53	1.54	1.31

different layer combinations. The simulator recorded the actual cost and whether the task had succeeded or failed. The effectiveness of each layer was verified by comparison with the recorded data. The simulator was the same as that used in the previous experiment. The results of this experiment are summarized in Table 5.2. In Method1, the path was generated using layer1, layer2, and layer4. In Method2, the path was generated using layer1, layer2, layer4, and layer5, and all layers were used in Method3. The cost ratio was calculated as follows.

$$\text{Cost Ratio} = \sum_{i=1}^{4} \text{AMC}^i \Big/ \sum_{i=1}^{4} f_p\left(N_c\right) \tag{5.22}$$

where, AMC^i indicates actual movement cost of MoMo_i from initial location to goal station. The cost ratio was calculated for successful tasks, and it always exceeded 1. As the cost ratio reduces, the performance of the algorithm improves. A cost ratio of 1 indicates that all MoMos follow their optimal paths.

As shown in Table 5.2, layer4 solved the deadlock situation in many cases. Layer5 solved the specific situation that was unsolved by layer4, but layer5 incurred additional actual cost. The actual cost was reduced by layer3 and layer6. Layer3 influenced MoMo's movement decision in the absence of a valid path. Guided by layer3, MoMo approached the other MoMo occupying its valid path. Since all P_j^i s approached the interruptive MoMo, the MoMo was rapidly affected by layer4. As mentioned in the previous section, layer6 reduced unnecessary motion. This indicates that layer2 and layer5 reduce the actual costs and increase the success rate.

5.4 Discussion and conclusion

In this chapter, a new path generating method was proposed. The proposed algorithm is validated by computer simulation experiment of various situations. In most cases, the proposed algorithm solved the deadlock situation within a specified period of time. As it was shown in Table 5.1, the success rate of the proposed algorithm was improved by more than 40 % on fields containing obstacles. The proposed algorithm gives a valid path to MoMo's goal station by layer3 when the other MoMos blocked the MoMo's valid path. Layer6 inhibits unnecessary repetitive motions caused by the confliction among layers. These result the decrease of MoMo's actual movement cost and increase of the success rate of path generation. It was shown in Table 5.2. According to the results of Method1 and Method2 in Table 5.2, layer5 incurs additional actual movement cost, but it increases the success rate of proposed algorithm. Through this algorithm, MoMos on the same field can select own motion that does not incur the collision situation and deadlock situation.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, the device's spatial constraint problem was discussed. Basically, the device's spatial constraint problem occurs by the device's specifications, such as usable distance and angle of view. The other main cause of the problem is the device's mobility. If the device can move to suitable location according to spatial situation, the device's spatial constrain problem will not occur. However, in most previous research, the devices are fixed on the space at the initial state, and their positions are chiefly unchanged. To solve the device's spatial constraint problem, the suitable deployment of devices are studied in previous research. The suitable deployment of devices that is proposed in previous research gives a good solution in some applications, e.g., a surveillance monitoring system, a fire alarm system, a target tracking system. On the other hand, it cannot be a sufficient solution in other applications, e.g., a facial emotion recognition system, a gesture recognition system.

Reconfigurable Intelligent Space (R+iSpace) was proposed in this thesis as a solution of the device's spatial constraint problem. R+iSpace can solve the problem by the automatic rearrangement of devices. As it was mentioned above, the problem will not occur when the device can move to suitable location according to spatial situation. For achieving the device's mobility, the device is mounted on specially designed wall climbing robot that is called Mobile Module (MoMo). In this thesis, three main research topics for the implementation of R+iSpace were addressed; suitable mechanical structure of MoMo and field, determination of device's suitable location,

and path generation without a collision and deadlock situation.

First, the suitable mechanical structure of MoMo and field was discussed in Chapter 3. The suitable mechanical structure indicates that the design of MoMo can satisfy the requirements of R+iSpace. Several requirements of R+iSpace were defined to increase the usefulness and feasibility of R+iSpace, and it can be achieved by the suitable mechanical structure of MoMo and field. As it is described in Chapter 3, three prototypes of MoMo were designed. MoMo1 and MoMo2 use screw-nut mechanism to adhere to the field. MoMo1 and MoMo2 have several screw gears in its screw modules, and the nuts are inserted into the field at the uniform distance. MoMo3 employs a pin-lock mechanism to adhere to the field. The pin-lock mechanism fixes the locking module to the field as follow simple sequence; inserting pins, rotating locking module, and pulling the pins. The fixed status of locking module is similar to the fastened rivet's status. The experiments for confirmation of satisfaction of the requirements were performed. The target mechanical structure was MoMo3 because it has better specifications than the other prototypes. As a result, MoMo3's mechanical structure can satisfy the requirements.

Next, the best effort location of device was discussed in Chapter 4. Through the development of MoMo3, the device can achieve its mobility. However, it is not sufficient to guarantee the application's consistent performance. It can be achieved when the device is placed on suitable location. In this section, the determination method of device's destination was proposed. The proposed method considers not only the application's successful implementation but also rapid response to the small changes of spatial situation. The proposed algorithm comprises basic four factors, adjacent nodes' result and travel time from current location. The basic four factors are as follows; distance from target, included angle with target's direction, occlusion by objects, and obstacles on the field. These factors influence on the application's performance directly. The proposed algorithm calculates the costs of each factors for all available nodes, then, integrates them by weighted average method. After calculation of integration, the algorithm calculates the influence from adjacent node. This calculation is designed to consider the rapid response to the small changes of spatial situation. The last factor is the travel time from current location. The application's performance is not determined by only the success rate of application. The rapid starting of application is also main factor to determinate the application's performance. Therefore the travel time is also main factor to decide the device's destination. After calculation of the influence by travel time, the destination of device is achieved. The proposed

algorithm was verified by computer simulation experiments. The success rate of application is increased when the device is placed on the destination that is calculated by proposed algorithm.

Last, the path generation method was proposed in Chapter 5. Basically, R+iSpace is a distributed system, therefore, each movable DINDs should generate own path, and the generated path does not occur a collision and deadlock situation. To avoid the collision between two movable DINDs, the preemptive shared memory was proposed. The preemptive shared memory contains the information on all movable DINDs' location and weight factors for negotiation. This memory can be accessed by one movable DIND, and the other movable DINDs wait until the memory is released. The path generation algorithm comprises six layers; cost to adjacent node, movement costs of valid path and pseudo path, negotiation with the other movable DINDs I and II, and appended cost on previous node. The algorithm calculates the costs of adjacent nodes, then select a node that has minimum cost as a next node. Proposed algorithm is verified by computer simulation experiment. According to the results of experiment, the proposed algorithm gives a conflict-free path, and the algorithm can solve the deadlock situation by negotiation layer. In detail, the success rate of the proposed algorithm, relative to the conventional algorithm, was improved by more than 40 % on fields containing obstacles.

6.2 Future works

In this thesis, only three essential topics for implementation of R+iSpace were discussed. Through this research, R+iSpace can be achieved, but, there are many remained studies. First, as it was mentioned in Chapter 5, the path generation algorithm is designed based on the MoMo2's movement. Therefore, the algorithm should be changed for MoMo3's movement scheme. Next, typically, in R+iSpace, there are several DINDs that have same type of device for the different purposes of use. In this situation, DINDs can switch the their purpose of use to each other according to the spatial situation, and it can lower the required movable DIND's moving speed. Besides, the response time of R+iSpace can be faster. The methods for the fulfillment of above concept will be discussed later, and the study on the optimal number of MoMo for the cooperation will be performed.

Bibliography

[1] J.-H. Lee and H. Hashimoto, "Intelligent space—concept and contents," *Advanced Robotics*, vol. 16, no. 3, pp. 265–280, 2002.

[2] J. O'rourke, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.

[3] A. Kröller, M. Moeini, and C. Schmidt, "A novel efficient approach for solving the art gallery problem," in *WALCOM: Algorithms and Computation*. Springer, 2013, pp. 5–16.

[4] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.

[5] Y. E. Osais, M. St-Hilaire, and R. Y. Fei, "Directional sensor placement with optimal sensing range, field of view and orientation," *Mobile Networks and Applications*, vol. 15, no. 2, pp. 216–225, 2010.

[6] F. Lin and P.-L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks," *Communications Letters, IEEE*, vol. 9, no. 1, pp. 43–45, 2005.

[7] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4, pp. 436–463, 2005.

[8] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[9] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[10] M. Weiser, "The computer for the 21st century," *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.

[11] M. Weiser, "Some computer science issues in ubiquitous computing," *Communications of the ACM*, vol. 36, no. 7, pp. 75–84, 1993.

[12] E. Aarts *et al.*, *Into ambient intelligence.* Springer, 2006.
[13] P. Gabriel, M. Bovenschulte, E. Hartmann, W. Groß, H. Strese, K. Bayarou, M. Haisch, M. Mattheß, C. Brune, H. Strauss *et al.*, "Pervasive computing: trends and impacts," *SecuMedia, Ingelheim*, 2006.
[14] M. Friedewald and O. Raabe, "Ubiquitous computing: An overview of technology impacts," *Telematics and Informatics*, vol. 28, no. 2, pp. 55–65, 2011.
[15] S. Chopra and M. S. Sodhi, "Looking for the bang from the rfid buck," *Supply Chain Management Review*, vol. 11, no. 4, 2007.
[16] S. Spiekermann, "Rfid and privacy: what consumers really want and fear," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 423–434, 2009.
[17] P. Schmitt, F. Thiesse, and E. Fleisch, "Adoption and diffusion of rfid technology in the automotive industry," in *Proceedings of the 15th European Conference on Information Systems*, 2007.
[18] E. Fleisch and C. Tellkamp, "The business value of ubiquitous computing technologies," in *Ubiquitous and pervasive commerce.* Springer, 2006, pp. 93–113.
[19] B. Gipp, J. Beel, and I. Rössling, "epassport: The world ' s new electronic passport," *A Report about the ePassport ' s Benefits, Risks and it ' s Security. CreateSpace*, 2007.
[20] W. Yao, C.-H. Chu, and Z. Li, "The adoption and implementation of rfid technologies in healthcare: a literature review," *Journal of medical systems*, vol. 36, no. 6, pp. 3507–3525, 2012.
[21] W. Yao, C.-H. Chu, and Z. Li, "The use of rfid in healthcare: Benefits and barriers," in *RFID-Technology and Applications (RFID-TA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 128–134.
[22] G. E. Burnett and J. M. Porter, "Ubiquitous computing within cars: designing controls for non-visual use," *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 521–531, 2001.
[23] M. Maekawa, "Its (intelligent transportation systems) solutions," *NEC Journal of Advanced Technology*, vol. 1, no. 3, p. 195, 2004.
[24] Mit project oxygen. [Online]. Available: http://oxygen.csail.mit.edu/Overview.html [Accessed: 2015-11-23]
[25] L. Rudolph, "Project oxygen: Pervasive, human-centric computing–an initial experience," in *Advanced Information Systems Engineering.* Springer, 2001, pp. 1–12.
[26] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st cen-

tury," *Computer*, vol. 36, no. 3, pp. 25–31, 2003.

[27] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," in *Handheld and ubiquitous computing.* Springer, 2000, pp. 12–29.

[28] S. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, and D. Robbins, "The new easyliving project at microsoft research," in *Proceedings of the 1998 DARPA/NIST Smart Spaces Workshop*, 1998, pp. 127–130.

[29] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on.* IEEE, 2000, pp. 3–10.

[30] H. Hashimoto, "Intelligent space-how to make spaces intelligent by using dind?" in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 14–19.

[31] J.-H. Lee, N. Ando, and H. Hashimoto, "Design policy of intelligent space," in *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 3. IEEE, 1999, pp. 1077–1082.

[32] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent environments: a manifesto," *Human-Centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–18, 2013.

[33] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *Computers, IEEE Transactions on*, vol. 51, no. 12, pp. 1448–1453, 2002.

[34] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 1, pp. 78–92, 2005.

[35] J. Wang and N. Zhong, "Efficient point coverage in wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 291–304, 2006.

[36] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 2, pp. 262–274, 2008.

[37] J. Adriaens, S. Megerian, and M. Potkonjak, "Optimal worst-case coverage of directional field-of-view sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1. IEEE, 2006, pp. 336–345.

[38] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in dis-

tributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, 2004.

[39] S. Li, C. Xu, W. Pan, and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," in *Information Fusion, 2005 8th International Conference on*, vol. 2. IEEE, 2005, pp. 7–pp.

[40] G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, pp. 640–652, 2006.

[41] S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 8, pp. 1108–1121, 2007.

[42] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," in *Wireless Internet, 2005. Proceedings. First International Conference on.* IEEE, 2005, pp. 114–121.

[43] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp. 133–147, 2007.

[44] D. G. Costa, L. A. Guedes, F. Vasques, and P. Portugal, "A routing mechanism based on the sensing relevancies of source nodes for time-critical applications in visual sensor networks," in *Wireless Days (WD), 2012 IFIP.* IEEE, 2012, pp. 1–6.

[45] A. Ganguli, J. Cortés, and F. Bullo, "Distributed deployment of asynchronous guards in art galleries," in *American Control Conference, 2006.* IEEE, 2006, pp. 6–pp.

[46] S. S. Dhillon and K. Chakrabarty, *Sensor placement for effective coverage and surveillance in distributed sensor networks.* IEEE, 2003, vol. 3.

[47] S. Hoseini, M. Dehghan, and H. Pedram, "Full angle coverage in visual sensor networks," in *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on.* IEEE, 2012, pp. 260–265.

[48] H. Zhang, L. Xia, F. Tian, P. Wang, J. Cui, C. Tang, N. Deng, and N. Ma, "An optimized placement algorithm for collaborative information processing at a wireless camera network," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on.* IEEE, 2013, pp. 1–6.

[49] J.-H. Lee, "Human centered ubiquitous display in intelligent space," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE.*

IEEE, 2007, pp. 22–27.

[50] J.-E. Lee, S. Miyashita, K. Azuma, J.-H. Lee, and G.-T. Park, "Anamorphosis projection by ubiquitous display in intelligent space," in *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer, 2009, pp. 209–217.

[51] J.-H. Ahn, J.-E. Lee, J.-H. Kim, S.-J. Kim, J.-H. Lee, and G.-T. Park, "Human position estimation in intelligent space for an active information display," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. IEEE, 2011, pp. 1497–1500.

[52] B. R. Jones, H. Benko, E. Ofek, and A. D. Wilson, "Illumiroom: peripheral projected illusions for interactive experiences," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 869–878.

[53] I.-M. Chen and S. H. Yeo, "Locomotion of a two-dimensional walking-climbing robot using a closed-loop mechanism: From gait generation to navigation," *The International Journal of Robotics Research*, vol. 22, no. 1, pp. 21–40, 2003.

[54] H. Zhu, Y. Guan, W. Wu, L. Zhang, X. Zhou, and H. Zhang, "Autonomous pose detection and alignment of suction modules of a biped wall-climbing robot," *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 2, pp. 653–662, 2015.

[55] G. Lee, H. Kim, K. Seo, J. Kim, and H. S. Kim, "Multitrack: A multi-linked track robot with suction adhesion for climbing and transition," *Robotics and Autonomous Systems*, vol. 72, pp. 207–216, 2015.

[56] I. Chen, S. H. Yeo *et al.*, "Locomotion and navigation of a planar walker based on binary actuation," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 329–334.

[57] Y. Guan, H. Zhu, W. Wu, X. Zhou, L. Jiang, C. Cai, L. Zhang, and H. Zhang, "A modular biped wall-climbing robot with high mobility and manipulating function," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1787–1798, 2013.

[58] S. Kim, M. Spenko, S. Trujillo, B. Heyneman, V. Mattoli, and M. R. Cutkosky, "Whole body adhesion: hierarchical, directional and distributed control of adhesive forces for a climbing robot," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1268–1273.

[59] B. Aksak, M. P. Murphy, and M. Sitti, "Gecko inspired micro-fibrillar adhesives for wall climbing robots on micro/nanoscale rough surfaces," in *Robotics and*

Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008, pp. 3058–3063.

[60] M. P. Murphy and M. Sitti, "Waalbot: An agile small-scale wall-climbing robot utilizing dry elastomer adhesives," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 3, pp. 330–338, 2007.

[61] K. Daltorio, A. D. Horchler, S. Gorb, R. E. Ritzmann, R. D. Quinn *et al.*, "A small wall-walking robot with compliant, adhesive feet," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on.* IEEE, 2005, pp. 3648–3653.

[62] L. P. Kalra, J. Gu, and M. Meng, "A wall climbing robot for oil tank inspection," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on.* IEEE, 2006, pp. 1523–1528.

[63] G. Lee, K. Seo, S. Lee, J. Park, H. Kim, J. Kim, and T. Seo, "Compliant track-wheeled climbing robot with transitioning ability and high-payload capacity," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on.* IEEE, 2011, pp. 2020–2024.

[64] A. Sintov, T. Avramovich, and A. Shapiro, "Design and motion planning of an autonomous climbing robot with claws," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 1008–1019, 2011.

[65] M. Vona, C. Detweiler, and D. Rus, "Shady: Robust truss climbing with mechanical compliances," in *Experimental Robotics.* Springer, 2008, pp. 431–440.

[66] R. Fukui, H. Morishita, T. Mori, and T. Sato, "Hangbot: A ceiling mobile robot with robust locomotion under a large payload (key mechanisms integration and performance experiments)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 4601–4607.

[67] R. Fukui, H. Morishita, T. Mori, and T. Sato, "Hangbot: A ceiling mobile robot with robust locomotion under a large payload," in *Experimental Robotics.* Springer, 2014, pp. 685–694.

[68] G. Stepan, A. Toth, L. Kovacs, G. Bolmsjo, G. Nikoleris, D. Surdilovic, A. Conrad, A. Gasteratos, N. Kyriakoulis, D. Chrysostomou *et al.*, "Acroboter: a ceiling based crawling, hoisting and swinging service robot platform," in *Beyond Gray Droids: Domestic Robot Design for the 21st Century Workshop at HCI*, vol. 2009, no. 3, 2009, p. 2.

[69] Raptor-e digital realtime system. [Online]. Available: http://www.motionanalysis.com/html/industrial/raptore.html [Accessed: 2015-12-13]

[70] Okao vision. [Online]. Available: http://www.omron.com/ecb/products/mobile/ [Accessed: 2015-12-13]

[71] S. Lao and M. Kawade, "Vision-based face understanding technologies and their applications," in *Advances in Biometric Person Authentication*. Springer, 2005, pp. 339–348.

[72] G. Wagner and H. Choset, "M*: A complete multirobot path planning algorithm with performance bounds," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3260–3267.

[73] A. W. ter Mors, "Conflict-free route planning in dynamic environments," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2166–2171.

[74] L. Breton, S. Maza, and P. Castagna, "A multi-agent based conflict-free routing approach of bi-directional automated guided vehicles," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[75] J. Park and J.-H. Lee, "Reconfigurable intelligent space, R+iSpace, and mobile module, momo," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3865–3866.

[76] J. Park and J.-H. Lee, "Design and evaluation of mobile module for reconfigurable intelligent space," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012.

[77] J. Park and J.-H. Lee, "A wall climbing mobile module based on screw and nut for stable and fixed error boundary sensing in intelligent space," in *System Integration (SII), 2012 IEEE/SICE International Symposium on*. IEEE, 2012, pp. 277–282.

[78] J. Park, T. Nunogaki, and J.-H. Lee, "The research on the algorithm for the optimal position and path for momo," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 7849–7854.

[79] J. Park, T. Nunogaki, and J.-H. Lee, "The optimal position of mobile modules in the reconfigurable intelligent space," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on*. IEEE, 2013, pp. 274–279.

[80] J. Park, T. Nunogaki, and J.-H. Lee, "The deadlock free path generation algorithm for multi-momo in R+iSpace," in *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*. IEEE, 2014, pp. 1030–1035.

[81] J. Park and J.-H. Lee, "The pin-lock based moving mechanism of wall climbing robot for the reconfigurable intelligent space," in *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on.* IEEE, 2015, pp. 85–90.

[82] J. Park, T. Nunogaki, and J.-H. Lee, "The mechanical structure of mobile module for new self-configurable intelligent environment," *ROBOMECH Journal*, vol. 2, no. 1, pp. 1–13, 2015.

Appendix A

Published Conference proceeding paper list

Research on R+iSpace and MoMo for Reconfigurable Intelligent Space

Author	○ JongSeung Park and Joo-Ho Lee
Conference	Robotics and Mechatronics Conference (ROBOMEC2012)
Date	2012.05
Page	1P1–B07

Reconfigurable intelligent space, R+iSpace, and mobile module, MoMo

Author	○ JongSeung Park and Joo-Ho Lee
Conference	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012)
Date	2012.10
Page	3865–3866

再構成可能な知能化空間とモバイルモジュールに関する研究

Author	○ JongSeung Park and Joo-Ho Lee
Conference	第 55 回自動制御連合講演会 (RENGO2012)
Date	2012.11
Page	ROMBUNNO.1K203

Design and evaluation of mobile module for Reconfigurable Intelligent Space

Author	○ JongSeung Park and Joo-Ho Lee
Conference	The 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI2012)
Date	2012.11
Page	220–224

Calibration Issues in FRC: Camera, Projector, Kinematics based Hybrid Approach

Author	○ Joo-Hang Lee, Kosuke Maegawa, JongSeung Park and Joo-Ho Lee
Conference	The 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI2012)
Date	2012.11
Page	218–219

A wall climbing mobile module based on screw and nut for stable and fixed error boundary sensing in intelligent space

Author	○ JongSeung Park and Joo-Ho Lee
Conference	IEEE/SICE International Symposium on System Integration (SII2012)
Date	2012.12
Page	277–282

人間-コンピュータ間の新たなインタラクションのための未来型ロボティクスコンピュータの開発

Author	○ Kosuke Maegawa, JongSeung Park, Kenji Iwamjoto, Joo-Hang Lee and Joo-Ho Lee
Conference	The 13th SICE System Integration Division Annual Conference (SI2012)
Date	2012.12
Page	68–71

FRC based augment reality for aiding cooperative activities

Author	○ Joo-Ho Lee, Kosuke Maegawa, Kenji Iwamjoto, JongSeung Park, Joo-Haeng Lee and Hyun Kim
Conference	The 22th IEEE International Symposium on Robot and Human

Interactive Communication (RO-MAN2013)
Date 2013.08
Page 294–295

The optimal position of mobile modules in the reconfigurable intelligent space

Author ○ JongSeung Park, Toshitake Nunogaki and Joo-Ho Lee
Conference The 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI2013)
Date 2013.10
Page 274–279

The research on the algorithm for the optimal position and path for MoMo

Author ○ JongSeung Park, Toshitake Nunogaki and Joo-Ho Lee
Conference The 39th Annual Conference of the IEEE Industrial Electronics Society (IECON2013)
Date 2013.11
Page 7841–7846

再構成可能な知能化空間における Mobile Module の移動時間を考慮した経路計画

Author ○ Toshitake Nunogaki, JongSeung Park and Joo-Ho Lee
Conference The 14th SICE System Integration Division Annual Conference (SI2013)
Date 2013.12
Page 1855–1859

The new path generation algorithm without deadlock situation for Multi-MoMo in R+iSpace

Author ○ JongSeung Park, Toshitake Nunogaki and Joo-Ho Lee
Conference IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2014)
Date 2014.07
Page 1030–1035

構成可能な知能化空間における mobile module のデッドロック回避を考慮した経路計画

Author ○ Toshitake Nunogaki, JongSeung Park and Joo-Ho Lee

Conference	The 32nd annual conference of the Robotics Society of Japan (RSJ2014)
Date	2014.09
Page	AC3C2-02

The deadlock free path planning algorithm for mobile module in r+ispace

Author	○ Toshitake Nunogaki, JongSeung Park and Joo-Ho Lee
Conference	The 10th joint workshop on machine perception and robotics (mpr2014)
Date	2014.10
Page	R-P-12

複数台の移動ロボットを対象とした経路計画法の検証用 RTC

Author	○ Toshitake Nunogaki, JongSeung Park and Joo-Ho Lee
Conference	The 15th SICE System Integration Division Annual Conference (SI2014)
Date	2014.12
Page	23–25

The pin-lock based moving mechanism of wall climbing robot for the Reconfigurable Intelligent Space

Author	○ JongSeung Park, Toshitake Nunogaki and Joo-Ho Lee
Conference	IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2015)
Date	2015.07
Page	23–25

Appendix B

Published Books and Journal paper list

Reconfigurable Intelligent Space and the Mobile Module for Flexible Smart Space

Author	○ JongSeung Park and Joo-Ho Lee
BookTitle	Sensor Networks for Sustainable Development
Publisher	CRC Press
Date	2014.06
Page	83–103

The mechanical structure of mobile module for new self-configurable intelligent environment

Author	○ JongSeung Park, Toshitake Nunogaki and Joo-Ho Lee
Journal	ROBOMECH Journal
Publisher	Springer
Date	2015.10
Volume(Number)	2(1)
Page	1–13

Best Effort Location for a Device in Reconfigurable Environment

Author	○ JongSeung Park and Joo-Ho Lee
Journal	Journal of Advances in Information Technology
Date	2016
Volume(Number)	7(3)
Page	1–10