Best Effort Location for a Device in Reconfigurable Environment

Jong Seung Park
Department of Science and Engineering, Ritsumeikan University, Kusatsu-city, Japan
Email: gr0119hk@ed.ritsumei.ac.jp

Joo-Ho Lee
Department of Information Science and Engineering, Ritsumeikan University, Kusatsu-city, Japan
Email: leejooho@is.ritsumei.ac.jp

***Abstract*—Reconfigurable intelligent space (R+iSpace) is an intelligent environment proposed to solve the location restrictions of distributed intelligent networked devices (DINDs) installed on walls and ceilings. Such DINDs in R+iSpace are mounted on mobile modules (MoMo), and a MoMo can rearrange its own location. Through the R+iSpace structure, an application that uses such DINDs can accomplish service provision successfully and consistently by rearranging the DINDs to suitable locations. This paper proposes a method to determine suitable DIND locations according to a given environment. The proposed algorithm is verified through simulation experiments.**

***Index Terms*—intelligent space, the wall climbing robot, the best effort location of device, the reconfigurable system**

I. Introduction

In recent years, many studies on intelligent environments (IE), which are extensive environmental systems that provide physical and nonphysical service to users, have been conducted. Studies on IEs have been conducted under various names, such as smart rooms, ubiquitous computing environments, and intelligent space (iSpace) [1-3]. iSpace is a representative IE. Many distributed intelligent networked devices (DINDs) are installed on the walls and ceilings of the iSpace [1]. The iSpace process is summarized as follows. First, sensing applications recognize the spatial situation and user demands using input DINDs. Then, all DINDs share the latest information about the spatial situation and user demands. Proper services are then provided through the output DINDs and agent robots. The user can obtain a required service through the above process.

However, iSpace has an unsolved problem, i.e., the spatial restriction of the DINDs. It is evident that recognition results from a blind spot of a camera DIND are poor. Even if the recognition target is located in the field of vision of the camera DIND, the results may not always be the best. The best results are achieved when the DIND is located at a best effort location that yields the highest recognition performance. However, the best effort location changes frequently according to changing situations. This suggests that DINDs should be installed everywhere; however, this is not feasible. In addition to camera DINDs, most other devices encounter the same problem when used in iSpace. Therefore, conventional research typically assumes a specific situation, and then DINDs are installed at appropriate locations for the target situation. However, in real-life situations, spatial situations change frequently, and the rearrangement of DINDs according to any given situation is a burden. To address this problem, the reconfigurable intelligent space (R+iSpace) was proposed [4]. R+iSpace can reconfigure the space according to the spatial situation, which is achieved by the rearrangement of DINDs. A specially designed mobile module called MoMo was proposed to rearrange the DINDs. MoMo is a mobile robot that can move freely on modified walls and ceilings, hereafter referred to as fields. A DIND is mounted onto a MoMo, and can be rearranged anywhere within the field. The latest MoMo prototype and experimental field are shown in Fig. 1. The latest MoMo prototype employs a pin-lock mechanism to attach itself to the field. The field is composed of many stations combined with MoMo's legs. The structure of the latest MoMo prototype is briefly explained in the next section. DINDs can be rearranged on the field freely using the MoMo. However, this does not mean that R+iSpace can guarantee the best performance. As mentioned previously, DINDs should be placed at their best effort location; however, developing a decision algorithm for best effort locations for all DINDs is complicated because many factors influence the application's performance. The optimal arrangements of devices have been examined in many previous studies [5]-[9]. Optimal arrangements in previous studies can be divided into two categories. The target application in the first category cannot be changed easily. The optimal arrangement problem of security cameras belongs to this category. The entrances and windows of buildings and valuables can be the target of security cameras, and such things do not change frequently. This indicates that the optimal location of a security camera will also not change;

Manuscript received November 18, 2015; revised March 1, 2016.

doi: 10.12720/jait.7.3.146-155

therefore, the cameras do not need to be rearranged after they are installed at the optimal location. However, these are not universal situations and for some cases rearrangement is required. The optimal arrangement of the second category is determined after long-term observation of the target or modeling of the target's behavior. The optimal location determined by this method is the most efficient position from a long-term perspective. However, in real-life situations, the prediction of human behavior is not easy, and the determined location is not always optimal for every situation.

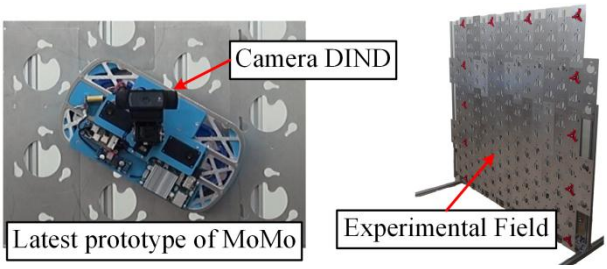


Figure 1. Latest prototype of MoMo and experimental field of R+iSpace

A method to decide the DINDs' destination was discussed in previous works [11, 12]. In the previous works, the algorithm considers only the relative position between the sensor and target. In this paper, the improved algorithm is proposed that gives the best effort location of DINDs at the given situation with the consideration of the travel time to the destination and the adaptability to the small changes of situation.

The remainder of this paper is organized as follows. In Section II, the mechanical structure of R+iSpace and MoMo's wall-attachment method are explained briefly. The DIND's stable location and adjacent locations are also described. In Section III, the proposed method, which determines the DIND's best effort location, is introduced. In addition, a detailed explanation of the method is described through an example of a camera device frequently used in iSpace. To verify the proposed method, simulation experiments are described in Section IV. Conclusions and suggestions for future work are presented in Section V.

II. Structure of the MoMo Prototype

A. Mechanical Structure of MoMo and the Field

As shown in Fig. 1, the MoMo prototype is composed of two legs, a main plate, and DIND parts. The distance between two legs is 140mm, and this is the step distance of MoMo. A leg consists of two actuators, a pin plate, and a leg housing. The detailed structure of the leg and station are shown in Fig. 2. Three pins are mounted on the pin plate, and a nut is placed at the center of the pin plate. The nut on the pin plate is combined with a screw that is connected to the pinning actuator. The pin plate has translating motion in a direction that is perpendicular to the field by the screw's rotation, and this is controlled by the pinning actuator. The panning actuator rotates the connected leg when the leg is unlocked. In case of a locked leg, the panning actuator rotates MoMo's whole body. The field is composed of numerous stations that can be combined with MoMo's legs. As shown in Fig. 2, the station has three holes. A hole is composed of an entering hole, a locking hole, and a bridging path between the entering and locking holes. The leg is locked through the following sequence: (1) placing pins through the entering holes of a station; (2) rotating the leg until the pins are positioned on the locking holes; and (3) pulling the pins until the pinhead and leg housing fix the leg firmly.

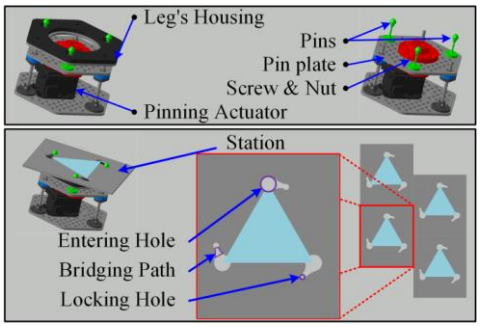


Figure 2. Detailed structure of a MoMo leg and the station

B. DIND's Location and Adjacent Locations

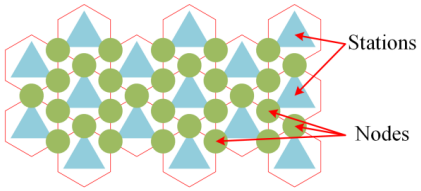


Figure 3. Arrangement of stations and nodes

The distance between the two adjacent stations is the same as the distance between two legs of MoMo. Thus, the DIND location, when MoMo is fixed, is positioned at the center between two adjacent stations. Here, the DIND location is denoted as a 'Node.' The arrangement of the nodes is shown in Fig. 3. In R+iSpace, it is recommended that all DINDs are used only when they are placed on nodes for the following reasons. First, significant noise is generated in the DIND results by the vibration or movement of MoMo when MoMo is not firmly fixed. This reduces R+iSpace performance. Next, the estimation of the DIND's location is easy when MoMo is firmly fixed. The DIND's position is very important information when a DIND is used. The DIND's position is obtained easily by pre-calculated node positions when the DIND is used only during MoMo's locked status. Last, restriction of the DIND to nodes is advantageous in terms of energy efficiency. A large load occurs in the panning actuator of the fixed leg when MoMo is not fixed securely. Therefore, to use a DIND on other positions requires more energy to maintain the current position. This causes energy consumption and can cause damage to MoMo, such as damage to the actuator by overheating. Therefore, the DIND location is restricted to nodes.

An adjacent node is defined as a node that is reachable by one or two movements from the current node, 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. As can be seen, ${}_1N_{Adj}$ and ${}_2N_{Adj}$ are 10 and 30, respectively.

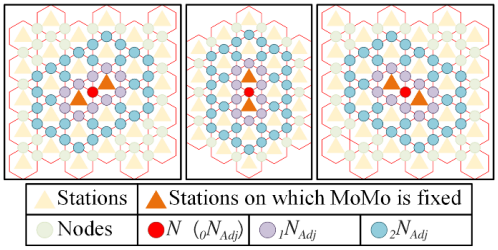


Figure 4. Arrangement of current node ***N*** and adjacent nodes

III. The Proposed Algorithm

Before explaining the proposed method, the goal of the proposed method is described briefly. The goal of the proposed algorithm is to determine the destination of the DIND. The destination is a suitable node that can be used to accomplish a target application in the current situation. The proposed algorithm considers the application's success rate and response time until the application is executable. 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 DIND, the relative position of the DIND from the target, the environmental structure, etc. Among these factors, a DIND in R+iSpace can resolve two factors by rearranging its location, i.e., the relative position of the DIND 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 the DIND's destination. The evaluation factors are as follows.

- Distance from the target
- Angle between the target's direction and the DIND'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.

A. Standardization of DIND Target

Here, before explaining the factors, the target is discussed first. Obviously, the target differs according to the types of DINDs and the applications. To make a standardized evaluation function, the various targets should also be standardized. The proposed algorithm uses information about 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 DIND 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.

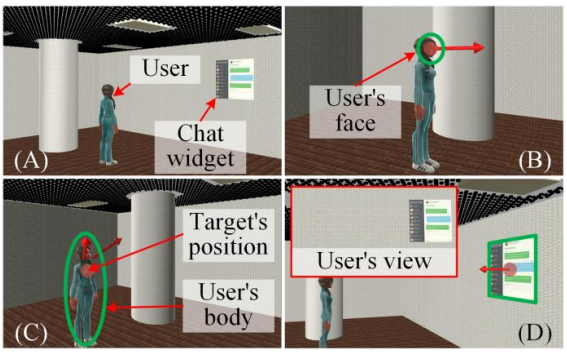


Figure 5. Examples of target according to application; (A)Whole view of R+iSpace, (B)Target of face detection, (C)Target of gesture recognition, and (D)Target of chat widget application

Fig. 5 shows examples of targets according to DINDs and applications. As shown in Fig. 5(A), a user is standing in the R+iSpace. The target in the Face Detection (FD) example application is shown in Fig. 5(B). The target of the FD application 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. 5(B).

Another example application is shown in Fig. 5(C). In this case, the camera is used for a gesture recognizing (GR) application. The target of the GR application is the user's whole body. In this example, the optimal location of the GR application is along the frontal upper diagonal direction of the user. This indicates that the GR application has two optimal locations, and the target also has two directions.

The last example is the target of a projector for a chat widget application. 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. 5(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 applications 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.

B. Evaluation Functions of Factors

Here, this section gives a detailed explanation of the factors and describe their evaluation functions. 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 DINDs, the functions should be simplified. Therefore, the evaluation functions for the four basic factors are determined 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 DIND. Besides, this function is simple and applicable to various DINDs 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 the DIND by considering the travel time. A detailed explanation and the calculation methods are discussed below.

1) Distance from target

Most DINDs demonstrate different performance according to the distance between the DIND and its target. For example, when the camera for the GR application is too close to the user, the GR application 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 DINDs.

The evaluation function of the distance is defined as follows.

$$\mathrm{EV}_1(\mathrm{N}) := \begin{cases} 0 & D_N \le D_m \\ \dfrac{D_N - D_m}{\alpha_1} & D_m < D_N \le D_m + \alpha_1 \\ 1 & D_m + \alpha_1 < D_N \le D_M - \alpha_1 \\ \dfrac{D_M - D_N}{\alpha_1} & D_M - \alpha_1 < D_N \le D_M \\ 0 & D_M \le D_N \end{cases} \quad (1)$$

In (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 DIND. D_m and D_M are obtained when the DIND 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 (1) for all available nodes. The parameters are determined by the specifications of the DIND and the application. Fig. 6 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. 6, the performance of the application changes according to changes in distance. The application worked well when the distance between the DIND and the target was smaller than a certain distance (e.g., D_M) and greater than another certain distance (e.g., D_m).

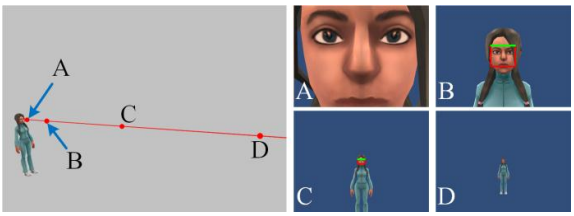


Figure 6. Change of FD application performance according to changing distance from target

2) Included angle between DIND viewing direction and target direction

The included angle between the DIND'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.

$$\mathrm{EV}_2(\mathrm{N}) = \max_{i \in k} \mathrm{EV}_2^i(\mathrm{N}) \quad (2)$$

$$\mathrm{EV}_2^i(\mathrm{N}) := \begin{cases} 1, & \mathrm{IA}^i \le \mathrm{IA}_M - \alpha_2 \\ \dfrac{\mathrm{IA}_M - \mathrm{IA}^i}{\alpha_2}, & \mathrm{IA}_M - \alpha_2 < \mathrm{IA}^i \le \mathrm{IA}_M \\ 0, & \mathrm{IA}_M < \mathrm{IA}^i \end{cases} \quad (3)$$

In (3), IA^i is the included angle between the DIND's viewing direction and the target's i th direction. IA_M denotes the available maximum included angle; this parameter is also determined by the specifications of the device and application.

Equation (2) can be adopted as an evaluation function in most DINDs and applications. However, some applications or DINDs have an extreme difference in performance according to the direction of the included angle.

$$\mathrm{EV}_2(\mathrm{N}) = \max_{i \in k} \left({}_h\mathrm{EV}_2^i(\mathrm{N}) \cdot {}_v\mathrm{EV}_2^i(\mathrm{N}) \right) \quad (4)$$

$${}_h\mathrm{EV}_2^i(\mathrm{N}) := \begin{cases} 1 & {}_h\mathrm{IA}^i \le {}_h\mathrm{IA}_M - {}_h\alpha_2 \\ \dfrac{{}_h\mathrm{IA}_M - {}_h\mathrm{IA}^i}{{}_h\alpha_2} & {}_h\mathrm{IA}_M - {}_h\alpha_2 < {}_h\mathrm{IA}^i \le {}_h\mathrm{IA}_M \\ 0 & {}_h\mathrm{IA}_M < {}_h\mathrm{IA}^i \end{cases} \quad (5)$$

$${}_v\mathrm{EV}_2^i(\mathrm{N}) := \begin{cases} 1 & {}_v\mathrm{IA}^i \le {}_v\mathrm{IA}_M - {}_v\alpha_2 \\ \dfrac{{}_v\mathrm{IA}_M - {}_v\mathrm{IA}^i}{{}_v\alpha_2} & {}_v\mathrm{IA}_M - {}_v\alpha_2 < {}_v\mathrm{IA}^i \le {}_v\mathrm{IA}_M \\ 0 & {}_v\mathrm{IA}_M < {}_v\mathrm{IA}^i \end{cases} \quad (6)$$

Equation (4) represents application performance in detail due to the split calculation according to the

direction of the included angle. Fig. 7 shows an example of $\mathrm{EV}_2(\mathrm{N})$ in the FD application. As mentioned previously, performance declined according to increased included angle. As shown in Fig. 7, the application could not detect the user's face when the included angle was greater than a certain value.

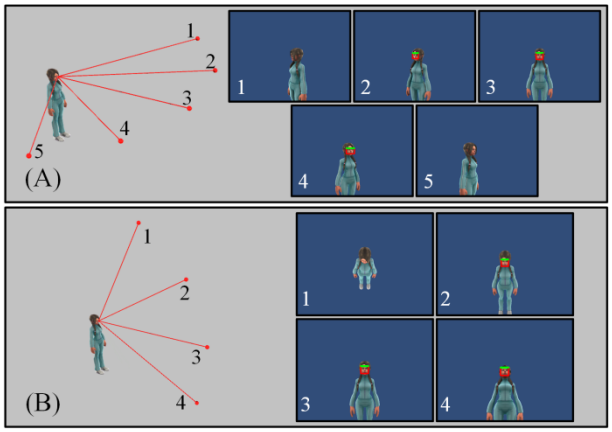


Figure 7. Change of FD application performance according to change of included angle; (A)Horizontal and (B)Vertical

3) Occlusion by obstacle

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

$$\mathrm{EV}_3(\mathrm{N}) \coloneqq \begin{cases} 1 & \text{There is no obstacle} \\ 0 & \text{There is an obstacle} \end{cases} \tag{7}$$

Equation (7) is usable with most DINDs. However, for some DINDs, 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 DINDs. Therefore, such DINDs do not calculate this evaluation function, and they set the result to 1.

4) Obstacles on the field

Typically, many objects are 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 the DIND's destination. Through the following evaluation function, the algorithm can avoid the selection of nodes that are adjacent to such objects.

$$\mathrm{EV}_4(\mathrm{N}) \coloneqq \begin{cases} 0 & D(N) \le {}_oD_m \\ \dfrac{D(N) - {}_oD_m}{\alpha_4} & {}_oD_m < D(N) \le {}_oD_m + \alpha_4 \\ 1 & {}_oD_m + \alpha_4 < D(N) \end{cases} \tag{8}$$

where,

$$D(\mathrm{N}) \coloneqq \min_{\forall \mathrm{Obj}} \mathrm{Dist}(N,\mathrm{Obj}) \tag{9}$$

In (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 $D(\mathrm{N})$ is the minimum $\mathrm{Dist}\,(N,\mathrm{Obj})$ for all Obj.

5) Integratin of the basic four factors' results

The basic four factors are designed to take application performance or MoMo's next motion into account. The next 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 fifth factor is calculated based on the integrated results. Therefore, the integration process should be performed prior to the calculation of the fifth factor. The integration is performed as follows.

$$\mathrm{EV}_{int}(\mathrm{N}) \coloneqq \begin{cases} 0 & If \ \prod_{k=1}^{4} \mathrm{EV}_k(\mathrm{N}) = 0 \\ \sum_{k=1}^{4} \omega_k \cdot \mathrm{EV}_k(\mathrm{N}) & Else \end{cases} \tag{10}$$

where,

$$\sum_{k=1}^{4} \omega_k = 1 \quad \wedge \quad 0 \le \omega_k \le 1 \tag{11}$$

If a node has an evaluation result of 0 for a factor, this indicates that the DIND 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.

6) Evaluation results of adjacent nodes

As mentioned above, this factor is related to the fast response to a slight movement of the target. 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.

$$\mathrm{EV}_5(\mathrm{N}) \coloneqq \begin{cases} \sum_{i=0}^{2} \omega_5^i \cdot \overline{\mathrm{EV}_{int}({}_iN_{Adj})} & If \ \mathrm{EV}_{int}(\mathrm{N}) \neq 0 \\ 0 & Else \end{cases} \tag{12}$$

where,

$$\sum_{i=0}^{2} \omega_5^i = 0 \quad \wedge \quad 0 \le \omega_5^0 \le \omega_5^1 \le \omega_5^2 < 1 \tag{13}$$

In (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. 8 shows an example that illustrates the effectiveness of this process. In Fig. 8(A), node N_A has the highest evaluation result. However, after processing, node N_B achieves the highest evaluation result. In this situation, the evaluation values of N_A and N_B change by the *WA* filter from 1.0 and 0.71 to 0.63 and 0.74, respectively. This result shows that MoMo can cope with a changing situation faster when it is placed on node N_B.

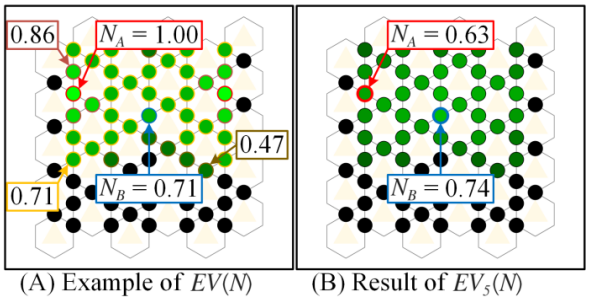


Figure 8. The result of the weighted average filter

7) Necessary travle time from the current node

Rapid response is one of the significant factors of R+iSpace's performance. Therefore, travel time should be considered when a DIND's destination is decided. However, calculating actual travel time for all nodes requires significant computation time. 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.

$$_{pseudo}T(N) \coloneqq \frac{\mathrm{Dist}(N, N_C)}{V_{MoMo}} \tag{14}$$

$$\mathrm{EV}_6^{pseudo}(\mathrm{N}) \coloneqq (1-\omega_6)\cdot \mathrm{EV}_5(\mathrm{N}) + \frac{\omega_6}{_{pseudo}T(N)+1} \tag{15}$$

In (14), N_C and V_{MoMo} denote the current node of the DIND 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.

$$\mathrm{EV}_6({}_sN) \coloneqq (1-\omega_6)\cdot \mathrm{EV}_5({}_sN) + \frac{\omega_6}{_{actual}T({}_sN)+1} \tag{16}$$

$$Destination\ node \coloneqq \arg\max_{\forall {}_sN} \mathrm{EV}_6({}_sN) \tag{17}$$

In (16) and (17), ${}_sN$ denotes candidate nodes with higher evaluation results than the other nodes in (15).

C. Example Evaluation Function

This section explains the proposed algorithm 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 [10]. 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 resolution of 480 pixels × 360 pixels. 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. 9. Fig. 9(A) and Fig. 9(B) show the evaluation results obtained using the distance factor and the included angle factor. 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. 9(C) and Fig. 9(D) show the results for the third and fourth factors, and Fig. 9(E) shows the integrated results of the four basic factors. The integrated results change according to the *WA* filter (Fig. 9(F)). The destination candidates are selected by (15) as shown in Fig. 9(G). Finally, the destination is determined by (17). The determined destination is the red node in Fig. 9(G), and the result of the FD application for the destination is illustrated in Fig. 9(H).

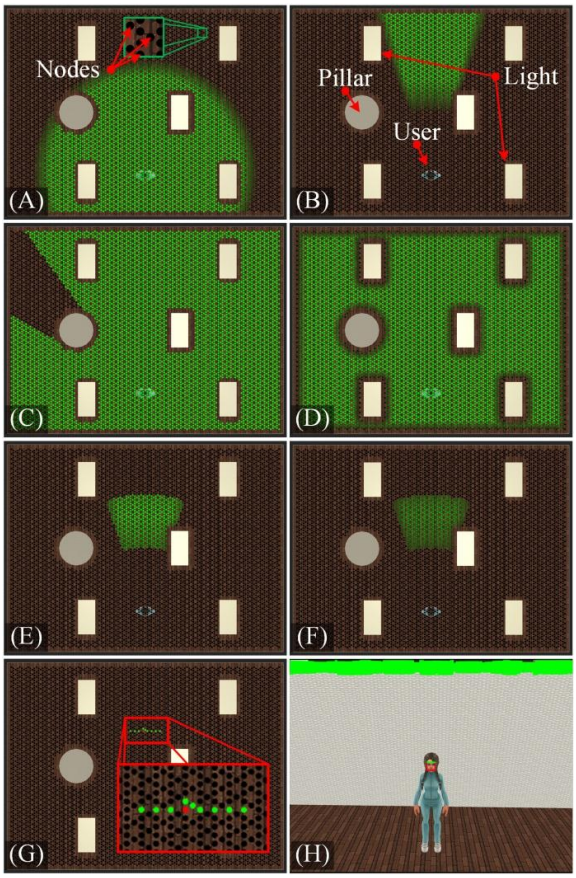


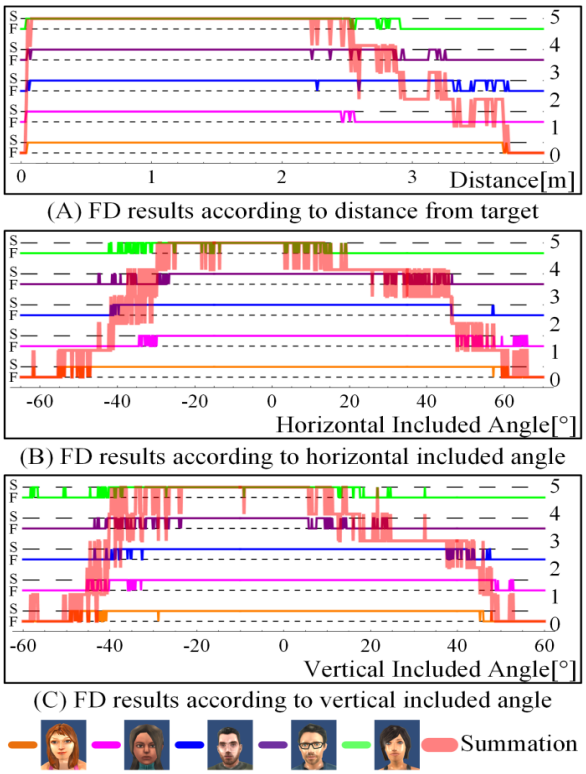
Figure 9. The evaluation results of each process and the result of application; (A) Distance, (B) Included angle, (C) Occlusion, (D) Obstacles, (E) Integration results, (F) After *WA* filter, (G) Candidates and destination, and (H) Result of face detection at destination

IV. Simulation Experiment

Here, we describe simulation experiments conducted to verify the performance of the proposed algorithm. As mentioned above, the proposed algorithm gives the best effort locations of DINDs according to the current spatial situation. This indicates that the previous situation cannot effect to the output of the proposed algorithm. Therefore,

the simulation experiment was performed as follow sequence. At first, the DIND is placed on their pre-decided initial location, and the target's position and direction is generated randomly. Then, the algorithm calculates the best effort location of DIND. The application was FD application 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 first half of this section describes the deciding process for the main parameters, then the results of the simulation experiments are illustrated.

A. Determining Main Parameters



(A) FD results according to distance from target

(B) FD results according to horizontal included angle

(C) FD results according to vertical included angle

Figure 10. FD application results according to distances and included angles

In the proposed algorithm, various parameters are decided by the application and DIND. To determine the parameters related to the distance and included angle factors, the following experiments were 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, this experiment was performed for five targets. The experimental results are shown in Fig. 10. The vertical axis of the graphs in Fig. 10 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 I.

TABLE I. PARAMETER USED IN THE EVALUATION FUNCTIONS

Parameters	Decided value
D_m, D_M, α_1	0.5m, 2.6m, 0.5m
${}_h\mathrm{IA}_M$, ${}_h\alpha_2$, ${}_v\mathrm{IA}_M$, ${}_v\alpha_2$	30.0 °, 20.0 °, 20.0 °, 5.0 °
${}_oD_m$, α_4	0.1m, 0.3m
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	0.5, 0.3, 0.2

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. 11. Here, The Map1 is empty space, and Map2 has moderate complexity that consists of a pillar and five lights. Map3 has higher complexity than Map2. There are two pillars and thirteen lights in Map3. Map4 has the highest complexity among the environments. Note that the shape of Map4 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 changed in a randomly generated predetermined order. MoMo moves to the destination obtained by the proposed algorithm. The camera object mounted on MoMo captures target images. The captured images are processed by the FD application.

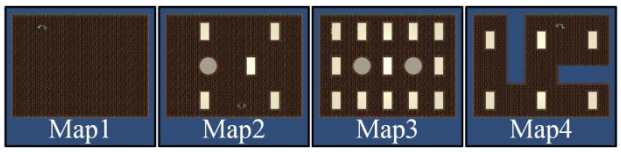
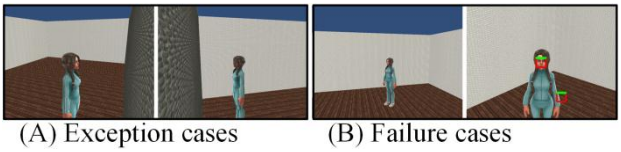


Figure 11. Environments used in the simulation experiment

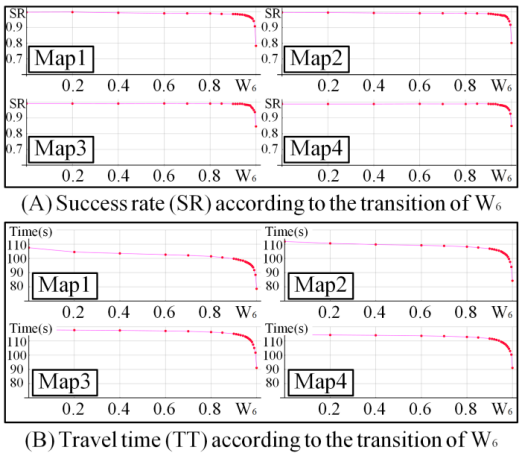
The data used for the analysis of the experimental results were restricted to cases wherein the proposed algorithm could find the destination of the DIND. In some cases, the evaluation values of all stations are 0, and the proposed algorithm cannot determine the destination of the DIND. For example, when a user is placed nearby and toward the wall, there is no usable location for the DIND. 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. 12. Fig. 12(A) shows cases in which there was no destination of DIND. The left image in Fig. 12(B) shows a case in which the DIND was not located at a suitable location, and the right image shows an example of the incorrect detection.



(A) Exception cases (B) Failure cases

Figure 12. Examples of exception and failure cases

We determined several values for candidates of ω_6 and NoC, and the results obtained according to parameter changes were compared. An experiment to determine the proper value of ω_6 was performed. Then an experiment for NoC was performed. The resultant graphs are shown in Fig. 13. The graphs in Fig. 13(A) show the relationship between ω_6 and the success rate of the application, and the graphs in Fig. 13(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 the first condition, the value of ω_6 with the shortest travel time was selected. In this experiment, the value of NoC was fixed at 6. As a result, the decided values of ω_6 differed according to the given environments.

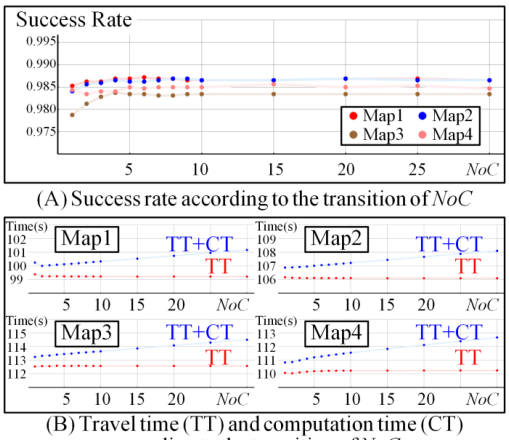


(A) Success rate (SR) according to the transition of W_6

(B) Travel time (TT) according to the transition of W_6

Figure 13. Experimental results for parameter ω_6

Next, an experiment for NoC was performed 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. 14. As shown in Fig. 14(A), there was no significant difference in success rate according to the change of NoC. Therefore, NoC was decided based on the travel time and computation time shown in Fig. 14(B). The values of ω_6 and NoC according to the environment are given in Table II.



(A) Success rate according to the transition of NoC

(B) Travel time (TT) and computation time (CT) according to the transition of NoC

Figure 14. Result of the experiment for NoC parameter

TABLE II. PARAMETER 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

B. Experiments to Verify the Proposed Algorithm

Here, we discuss experiments conducted to verify the proposed method's validity. The same application was used in these experiments. The experiments consisted of two phases. The first phase of the experiment was performed to verify the appropriateness of the evaluation functions for application performance. In this experiment, the DIND mounted on MoMo moved to all available nodes, and then the DIND captured the target. The results of this experiment are illustrated in Fig. 15. The magenta area in the figure indicates that the application was successful in that area. The blue area indicates that the results of the evaluation functions for the first two 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 III shows the success rates according to the evaluation results. As can be seen in Table III, the high scoring nodes obtained by proposed algorithm achieved a high application success rate regardless of map type. As a result, it was verified that the evaluation functions of the proposed method are valid for determining the best effort location of a DIND.

TABLE III. SUCCESS RATE ACCORDING TO THE RESULTS OF THE EVALUATION FUNCTIONS

$EV_{int}(N)$	Decided values			
	Map1	Map2	Map3	Map4
$0.0 < EV_{int}(N) \leq 0.1$	0.60	0.56	0.42	0.46
$0.1 < EV_{int}(N) \leq 0.2$	0.70	0.69	0.68	0.77
$0.2 < EV_{int}(N) \leq 0.3$	0.81	0.82	0.78	0.81
$0.3 < EV_{int}(N) \leq 0.4$	0.81	0.86	0.79	0.81
$0.4 < EV_{int}(N) \leq 0.5$	0.85	0.87	0.82	0.85
$0.5 < EV_{int}(N) \leq 0.6$	0.88	0.91	0.88	0.88
$0.6 < EV_{int}(N) \leq 0.7$	0.94	0.94	0.95	0.94
$0.7 < EV_{int}(N) \leq 0.8$	0.98	0.98	0.97	0.96
$0.8 < EV_{int}(N) \leq 0.9$	0.99	0.97	0.99	0.98
$0.9 < EV_{int}(N) \leq 1.0$	1.00	0.99	1.00	0.99

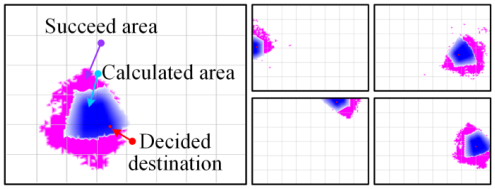


Figure 15. Maps of successful location and evaluation result

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 FD application. In this experiment, it was determined that the FD application was successful when the face was detected by one or more cameras. The experimental results are shown in Table IV. The results are divided into two sub-tables. The first sub-table of Table IV shows the results of all situations, and the second sub-table shows the situation in which the destination exists. Among the failure cases of R+iSpace, when the destination of the 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.

TABLE IV. SUCCESS RATE COMPARISON RESULTS BETWEEN CONVENTIONAL ISPACE AND R+ISPACE

(A) For all situations

Environment	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	$0.319_{(3185/10000)}$	$0.764_{(7637/10000)}$

(B) The situations without the exception cases

Environment	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	$0.354_{(2764/7800)}$	$0.979_{(7634/7800)}$

V. CONCLUSION

This paper has proposed a method to determine a DIND's best effort location in R+iSpace. The proposed method considers the relationship between an application's target and a DIND. 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 DIND. The proposed method also considers the travel time to the destination. This paper has verified the proposed method with simulation experiments conducted using the FD application.

Through the proposed method, we can obtain best effort locations for DINDs to provide service according to the given current spatial situations. In addition, we have determined the weight factors for weighted average empirically. In future, we will discuss method for deciding optimized parameters in proposed algorithm.

In addition, we will introduce an improved method to locate a DIND's best effort location when multiple DINDs are installed on the same field.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 26330302 and MEXT-Supported Program for the Strategic Research Foundation at Private Universities.

REFERENCES

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

[2] B. Brumitt, et al., "Easyliving: Technologies for intelligent environments," in *Proc. Handheld and Ubiquitous Computing–Second International Conf. (HUC2000)*, Bristol, UK, Sept. 25–27, 2000, pp. 12–29.

[3] H. Yoon, et al., "A model of sharing based multi-agent to support adaptive service in ubiquitous environment," in *Proc. 2nd International Conf. Information Security and Assurance*, Busan, Korea, Apr. 24–26, 2008, pp. 332–337.

[4] J. Park and J. H. Lee, "Reconfigurable intelligent space, R+iSpace, and mobile module, MoMo," in *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 7–12, 2012, pp. 3865–3866.

[5] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE Wireless Communications and Networking Conf.*, New Orleans, USA, Mar. 16–20, 2003, pp. 1609–1614.

[6] 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, Apr. 2010, pp. 216–225.

[7] U. M. Erdem and S. Sclaroff, "Optimal placement of cameras in floorplans to satisfy task requirements and cost constraints," in *Proc. 5th Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, Prague 1, Czech Republic, 2004.

[8] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, no. 3, Aug. 2006, pp. 156–169.

[9] R. Bodor, et al., "Optimal camera placement for automated surveillance tasks," *Journal of Intelligent and Robotic Systems*, vol. 50, no. 3, Oct. 2007, pp. 257–295.

[10] S. Lao and M. Kawade, *Advances in Biometric Person Authentication: Part III Face Recognition–System and Application: Vision-Based Face Understanding Technologies and Their Applications*, Springer Berlin Heidelberg, 2005, pp. 339–348.

[11] J. Park, T. Nunogaki, and J. H. Lee, "The research on the algorithm for the optimal position and path for MoMo," in *Proc 39th Annual Conf. of the IEEE Industrial Electronics Society*, Vienna, Austria, Nov. 10–13, 2013, pp. 7849–7854.

[12] J. Park, T. Nunogaki, and J. H. Lee, "The optimal position of mobile modules in the reconfigurable intelligent space," in *Proc. 10th International Conf. on Ubiquitous Robots and Ambient Intelligence*, Jeju, Korea, Oct. 30–Nov. 2, 2013, pp. 274–279.

JongSeung Park is currently pursing Ph.D program in the Graduate school of Science and Engineering, Ritsumeikan University, Japan. He received B.E. degree in information control from Kwangwoon University, Korea in 2005. He received M.E. degree in control and instrumentation from Kwangwoon University, Korea in 2007. His research area includes mechanical structure design in robotics and intelligent space. Mr. Park is a student member of the Institute of Electrical and Electronics Engineers (IEEE).

Joo-Ho Lee is currently professor in Department of Information Science and Engineering, Ritsumeikan University, Japan. He received B.E. degree in 1993, M.E. degree in 1995 in electrical engineering from Korea University, Korea. He received Ph. D degree in electrical engineering and information systems from University of Tokyo, Japan in 1999. He worked as a JSPS researcher and post-doctoral researcher in University of Tokyo, Japan during 2000-2003. He worked as a research associate in Tokyo University of Science, Japan during 2003-2004. His research area includes electrical engineering, robotics, information engineering, system integration, and intelligent space. Prof. Lee is a member of the Institute of Electrical and Electronics Engineers (IEEE), Japan Society of Mechanical Engineering (JSME), Robotics Society of Japan (RSJ), Institute of Electronics, Information and Communication Engineering (IEICE), Society Instrument and Control Engineer (SICE), Institute of Electrical Engineers of Japan (IEEJ), and the Human Interface Society (HIS).