Intelligent Robotic Systems in Support of a Declining Birthrate and an Aging Population

by

YANG Guang

Student ID Number: 1218004

A dissertation submitted to the
Engineering Course, Department of Engineering,
Graduate School of Engineering,
Kochi University of Technology,
Kochi, Japan

Assessment Committee:
Supervisor: Prof. Wang, Shuoyu
Co-Supervisor: Prof. Yang, Junyou, Shenyang University of Technology
Co-Supervisor: Associate Prof. Shibata, Kyoko
Prof. Oka, Koichi
Associate Prof. Hoshino, Yukinobu

September 2020

Abstract

According to the Japanese National Institute of Population and Social Security Research, the productive age population (between 15 and 64 years old) will continue to decrease. And in 2060, one out of every 2.5 people will be 65 years old or older. The challenge can be summarized as a declining birthrate and an aging population, with two consequences: (i) the increase of caring cost and (ii) the shortage of young labor.

Confronting these challenges: (i) on one hand, we tried to achieve high-level cognitive intelligence on personal care robot (understanding people) so that caring tasks can be conducted as required where no care givers are available; (ii) on the other hand, we focus on the intelligent of understanding the environments, that enables autonomous material transportation in construction sites, so that the effect cased by the lack of young labor in construction sites can be effectively decreased.

All the presented approaches have been evaluated in a real household environment and construction sites respectively.

Contents

List of Figures

List of Tables

Chapter 1

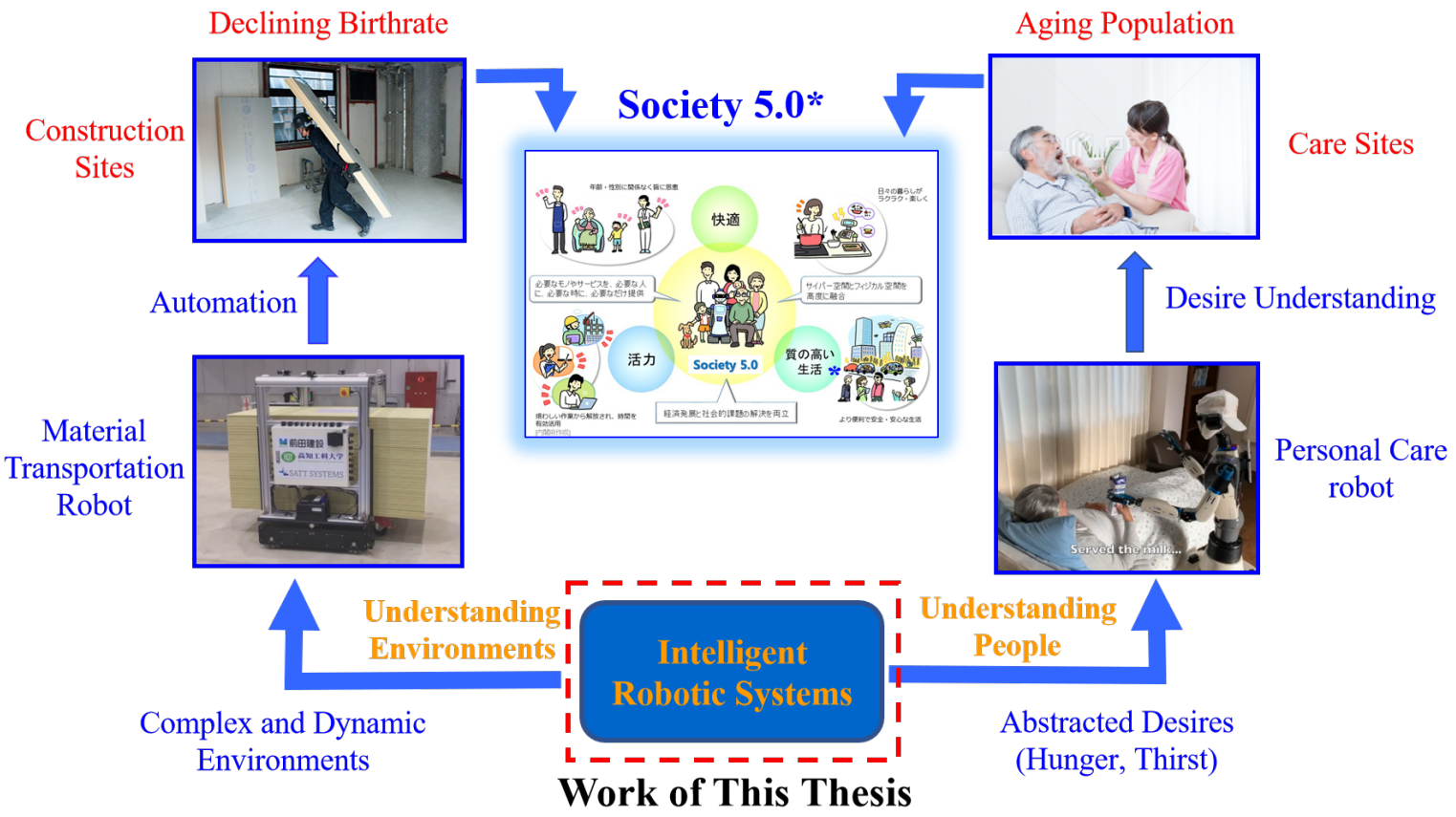
Outline

The challenge of a declining birthrate and an aging population comes with two consequences: (i) the increase of caring cost; and (ii) the shortage of young labor.

Confronting these challenges, we have conducted two novel studies (Figure 1-1) accordingly: (i) personal care robots, to help reducing the caring cost of human care givers, and (ii) autonomous material transportation robots, to deal with the time-consuming and laborious on-site handling.

In the rest of this thesis, (i) in Chapter 2, we introduce a series of approaches and methods to achieve high-level intelligence (understanding people) on personal care robot KUT-PCR so that caring tasks can be conducted as required where no care givers are available. As a result, the quality of life of the care recipients can be improved while the caring cost will be decreased; (ii) in Chapter 3, we discuss how the automatic material transportation can be achieved through "understanding environments", so that the effect caused by the lack of young labor in construction sites can be effectively decreased.

All the presented approaches have been evaluated in a real household environment and construction sites respectively.



*** Society 5.0 - 科学技術政策 - 内閣府 https://www8.cao.go.jp/cstp/society5_0/**

Figure 1-1: Outline of the research.

Chapter 2

Personal Care for the Bedridden Elderly

Many countries including China, Japan and Germany are facing an ever-increasing and aging population and a shortage of caring force. As Figure 2-1 shows, the shortage of caregivers in Japan is estimated to increase up to 790,000 in 2035. Caregivers who

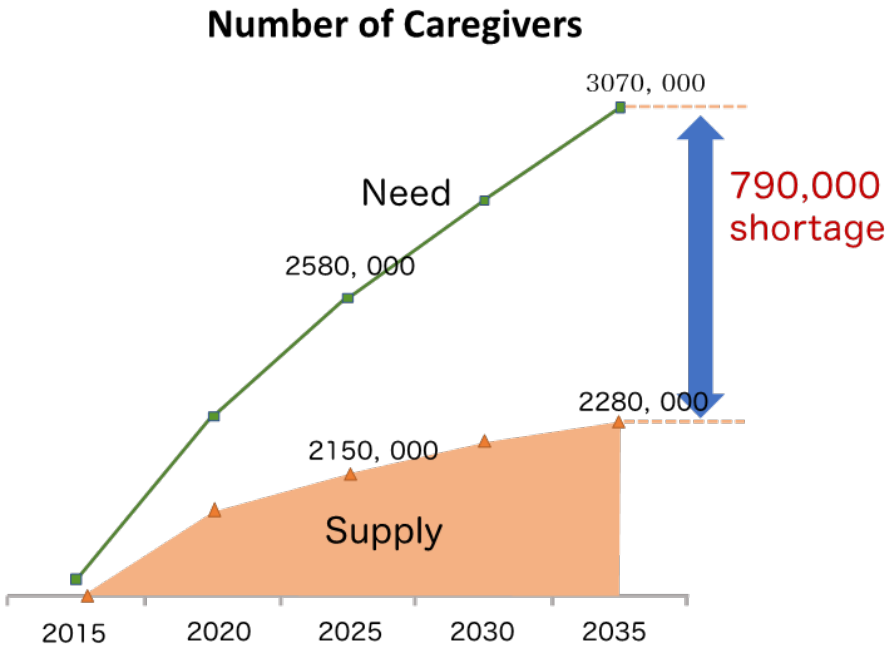


Figure 2-1: Shortage of caregivers.

look after people that are bedridden, due to aging, illness, or an accident, need not

only physical strength but also a proper understanding of their patients' physiological needs. Professional caregivers are required to carry out such tasks in nursing homes and hospitals, while visiting caregivers are needed in individuals' homes [1]. However, in order to deal with the nursing shortage [2][3] and improve care efficiency, a small number of caregivers are unable to focus on specific care recipients but instead must check in on them periodically.

This means that, the bedridden individuals sometimes have no-one to take care of their needs (Figure 2-2), which leads to inconvenient lives.

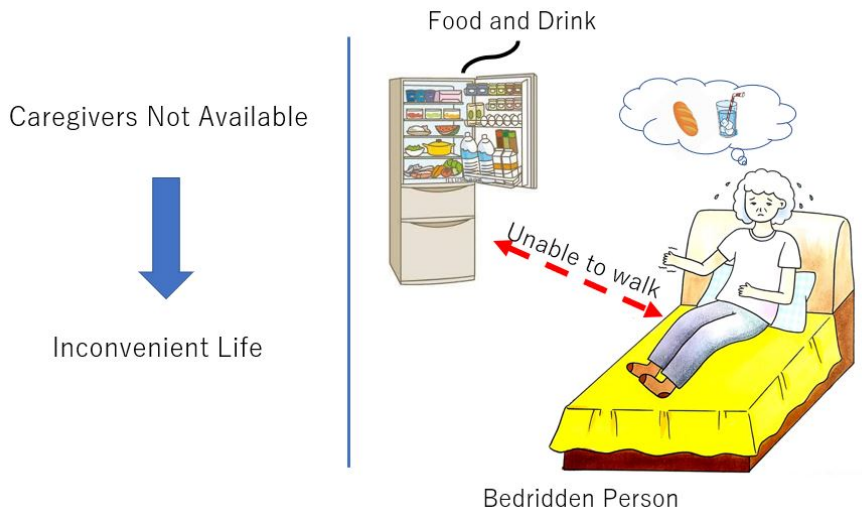


Figure 2-2: Inconvenient life.

In this case, if we could develop personal care robots capable of conducting simple care tasks (e.g., delivering drinks/food, adjusting the temperature or lighting), the burden on the caregivers can be reduced dramatically (Figure 2-3). As a result, the quality of life for many people could hopefully be increased.

In this work, we argue that intelligence of the personal care robots can be classified into three aspects: (i) intelligence of perception; (ii) intelligence of motion; and (iii) intelligence of reasoning. Accordingly, in the following sections, after we introduce the basic robotic system, we will present how these three types of intelligence are realized considering the following research: (i) pose estimation of daily containers; (ii) context-aware motion control; and (iii)desire-driven reasoning.

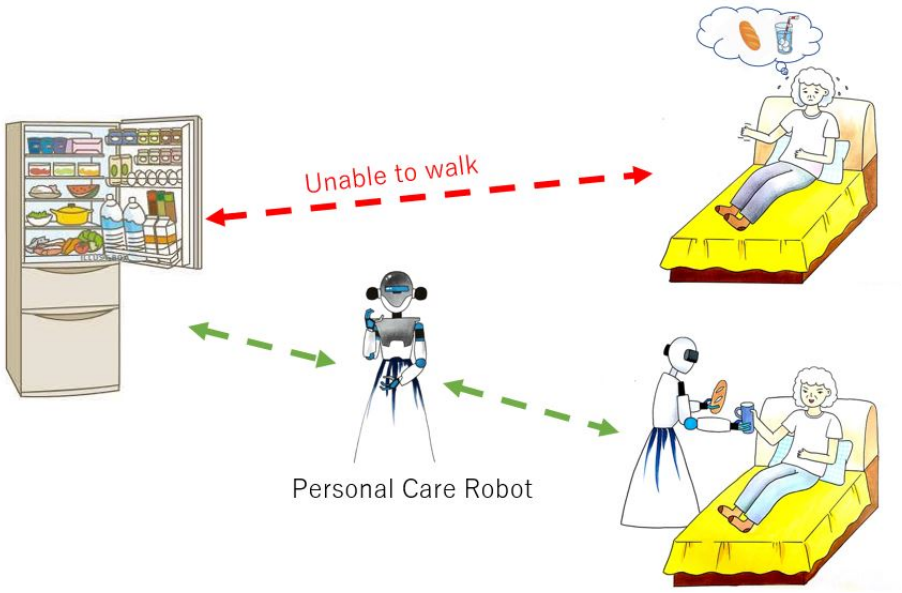


Figure 2-3: Robotic assistance.

2.1 System Overview

2.1.1 Hardware and Sensors

Figure 2-4 is our newly developed personal care robot named KUT-PCR. KUT-PCR is a mobile humanoid robot. The humanoid upper body, omnidirectional mobile platform, and various types of sensors allow KUT-PCR to conduct caring tasks similar to human caregivers.

The humanoid upper body is designed so that tasks requiring object manipulation can be achieved. The head has three degrees of freedom including roll, pitch, and tilt. Each of the two arms has seven degrees of freedom as well as an end effector with one degree of freedom. Also, the waist has three degrees of freedom so that it can bow and bend like human beings, which effectively extends the operation.

The robot's upper body also includes a range of sensors. It uses two RGB-depth (RGB-D) cameras, mounted on the head and chest, to perceive its environment using RGB images and point clouds. Force sensors attached to the wrists provide information about the objects it holds, while microphones and speakers allow natural language communication and multimedia applications.

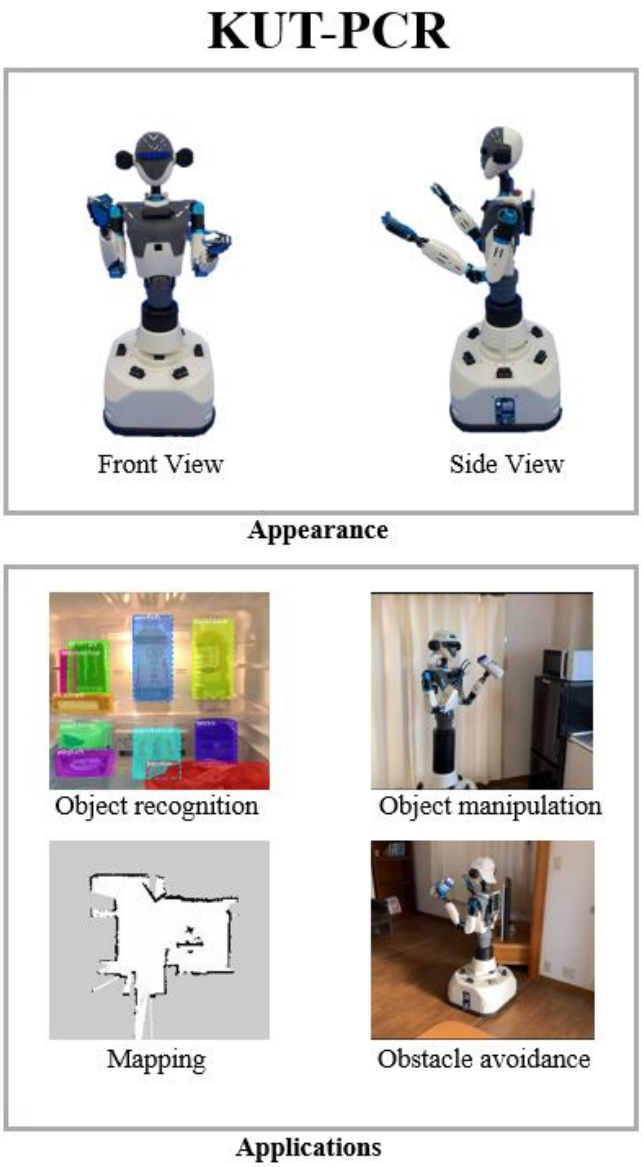


Figure 2-4: Personal care robot: KUT-PCR.

2.1.2 Mathematical Model

The parameters configuration as shown in Figure 2-5 is as follows:

$\Sigma(x, o, y)$: world coordinate;

$\Sigma(x', o', y')$: robot coordinate;

v: robot speed;

α: angle between x' and the direction of v;

f_i: force of each wheel;

v_i: speed of each wheel;

o': geometric center and center of gravity of the robot;

L: distance from o' to each wheel;

θ: angle between x' and the wheel numbered 1.

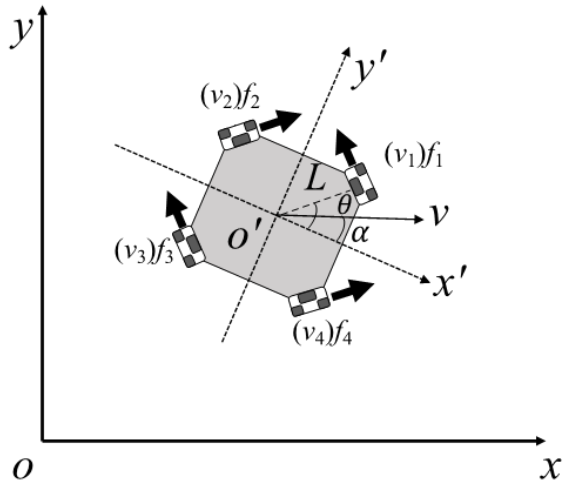


Figure 2-5: Model of the mobile platform.

Kinematics

The kinematic model of the robot is:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -sin\theta & sin(\frac{\pi}{2} - \theta) & L \\ cos\theta & cos(\frac{\pi}{2} - \theta) & -L \\ -sin\theta & sin(\frac{\pi}{2} - \theta) & -L \\ cos\theta & cos(\frac{\pi}{2} - \theta) & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \dot{\theta} \end{bmatrix} \tag{2.1}$$

where v_x and v_y are the subcomponents of the robot's velocity, respectively.

Also, the following relations should be fulfilled:

$$v_1 + v_2 = v_3 + v_4 \tag{2.2}$$

Dynamics

The dynamics model of the robot is given as:

$$M_0 \ddot{X} = BF \tag{2.3}$$

where B, M_0, F and $\ddot{X}$ are defined as:

$$B = \begin{bmatrix} -sin\theta & cos\theta & -sin\theta & cos\theta \\ cos\theta & sin\theta & cos\theta & sin\theta \\ L & -L & -L & L \end{bmatrix} \tag{2.4}$$

$$M_0 = \begin{bmatrix} M + m & 0 & 0 \\ 0 & M + m & 0 \\ 0 & 0 & I \end{bmatrix} \tag{2.5}$$

$$F = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \tag{2.6}$$

$$\ddot{X} = \begin{bmatrix} v_x & v_y & \ddot{\theta} \end{bmatrix}^T \tag{2.7}$$

2.2 Pose Estimation of Daily Containers

Considerable research effort has been devoted to object fetching. In addition to academic research, many methods have been proposed and verified in competitions such as the Amazon Picking Challenge. One of the most effective approaches is based on a pipeline with two stages, namely (i) object detection based on a convolutional neural network (CNN) and (ii) point-cloud registration based on the iterative closest-point (ICP) algorithm [4]. Our research benefits from this pipeline while limiting the scope of objects to daily containers, which, to the best of our knowledge, has not been discussed previously.

By daily containers, we mean the boxes and bottles that normally contain drink, food, or medicine. We focus on daily containers because fetching and serving them properly would fulfill the majority of the daily needs of a person who is elderly, including eating and drinking. To realize the pipeline successfully, we confront two main challenges.

1) **Highly personalized containers**: the daily containers used by different families vary considerably regarding brand, color, and shape. For instance, "tea" for user A may imply a bottled drink whereas the ideal service for user B would be hot fresh tea in a mug. This type of variable (sometimes opposite) definition of classes makes it almost impossible to meet every need with just one trained model. A more feasible solution is to retrain the model while considering various users or changes in preferences. Therefore, the complexity of the model retraining should also be considered seriously.

2) **Self- and partial occlusions**: because of self- and partial occlusions, the scanned point clouds may not contain all the information about a given container. To perform highly accurate registration with the ICP algorithm, we must address properly problems such as initial alignment and model cloud processing.

Herein, we present a pose-estimation pipeline for daily containers. We investigated several CNN models with the aim of maintaining a balance between performance and

retraining complexity. Additionally, with the good initial alignment and suitable model clouds provided by our proposed methods, accurate pose estimation could be obtained in daily environments considering partial occlusions. Consequently, the life-support robot KUT-PCR can fetch the target container successfully with the proposed pipeline.

2.2.1 Related Work

Object Detection

As a classical computer-vision problem, object detection has drawn increasing attention in recent years with the application of deep learning. The classical approaches to this challenge are based on algorithms such as shape matching [5] and histogram back projection [6]. However, these methods usually suffer from low recognition accuracy and limited tolerance for unstructured environments.

Deep learning for image classification was implemented successfully in 2012 [7] and was modified soon after to solve problems such as object detection. Delicately designed models such as MobileNets [8] and Faster R-CNN [9] allowed accurate rectangular masks (RMs) to be generated containing the target objects. Moreover, the use of deep learning has increased the performance of semantic segmentation dramatically, making highly accurate pixel-level segmentation available [10].

The choice among such approaches depends primarily on the needs of the given task. Traditional methods do not perform reliably in cluttered daily environments. Semantic segmentation requires large numbers of images labeled at pixel level as training data, but that is impractical because it is common to have to retrain the model for different users or newly included containers when supporting the lives of people who are elderly. Therefore, object-detection approaches resulting in RMs are chosen given their low retraining complexity and steady performance in daily scenarios.

Pose Estimation

Because object detection can identify and localize daily containers in two-dimensional (2D) red–green–blue (RGB) images, pose estimation is supposed to provide six-dimensional (6D) pose estimation for each recognized container based on the object cloud extracted from the scanned scene cloud.

There are several widely used approaches to pose estimation. Local descriptors such as SIFT [11] have been applied successfully for objects with sufficient texture. As for the texture-less objects that are common in daily life, three-dimensional (3D) template-matching-based methods including LINEMOD [12] prove effective, combining depth and color information. Nevertheless, this type of method normally performs less well than desired when confronted with an unstructured environment.

Moreover, ICP-based registration methods [13] can be used to align 3D point clouds, thereby producing accurate pose estimation. Because ICP is an iterative local optimizer, we must address problems such as initial alignment and point-cloud pre-processing to guarantee performance.

2.2.2 Method

Figure 2-6 illustrates the structure of our proposed pipeline. The original point cloud acquired from a Kinect 2 motion-sensing device is regarded as the "scene cloud" containing the raw sensor information and can be registered with the RGB image via accurate camera calibration. With a fine-tuned CNN model in the object-detection phase, accurate RMs of the target containers can be obtained from the RGB images. Then, by projecting the RMs from the RGB-image frame to the point-cloud frame, the point cloud of the target container can be extracted from the scene cloud, the former being described as the "object cloud." Nevertheless, because the extraction is based on rectangles, there will be some background outliers.

Meanwhile, models of various containers are drawn with CAD software or scanned using a depth camera. A suitable model is retrieved from the model library based on the recognized container label. The "model cloud" results from multiple processing

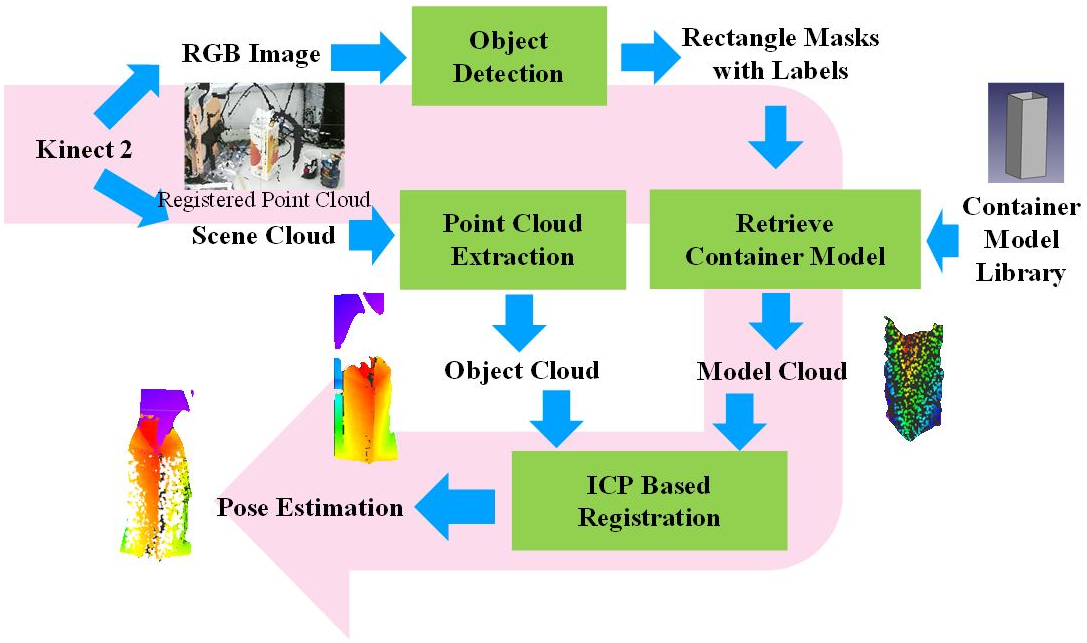


Figure 2-6: Pose-estimation pipeline.

of container models such as down sampling and rotation. The ICP algorithm is then used to register the model cloud with the object cloud. Upon successful registration, a 6D pose can finally be estimated.

To select a CNN model that could keep accuracy, speed, and retraining complexity in ideal balance, we explored several models from the Tensorflow detection-model zoo. Ultimately, we chose the ssd_mobilenet_v1_coco model, which was pre-trained on the Microsoft COCO dataset [14]. Originally, the dataset consisted of only general container classes such as "bottle". By taking advantage of transfer learning, we fine-tuned the model with 200 labeled images (which were finished easily with software such as DarkLabel) for three containers and less than 10 h of training on a PC (Intel i7 processor with no graphics-processing unit). Recognizing more containers in various scenarios requires more effort, but the overall process is relatively easy and simple, making it practical for daily use.

The ICP algorithm was designed initially to register two point clouds with the same or similar number of points and close initial poses. In our case, good initial alignment and suitable model clouds are crucial for acceptable performance.

Regarding pose initialization, without good initial alignment, the ICP algorithm can easily converge to an incorrect local minimum. We have observed experimentally that the initial orientation of a container does not influence the result dramatically because containers are usually placed vertically on platforms such as a desk. In previous work, we proposed an approach based on prior knowledge to address the initial position $p_m(x, y, z)$.

With a finely tuned CNN model, we can assume that the generated RMs surround the target container properly. In other words, the container should be approximately centered in the corresponding RM (Figure 2-7).

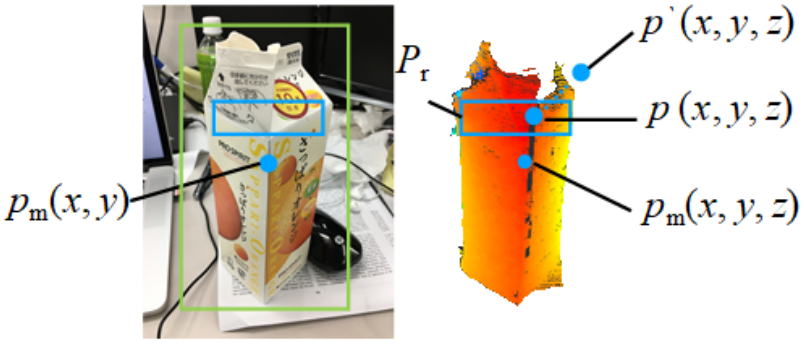


Figure 2-7: Pose initialization.

We then simply set the median point of the mask as $p_m(x, y)$ in 2D RGB space. Projecting the point into the point-cloud frame allows $p_m(x, y, z)$ to be obtained in 3D space. It should lie on the surface of the container, which is an initial pose close to the target cloud. Nevertheless, it often fails for two reasons: (i) the sensor data are noisy, meaning that the value of z could be either empty or inaccurate; (ii) if the median point in the RGB image is covered by other objects, $p_m(x, y, z)$ could be set close to an incorrect object.

Considering these factors, we modify the approach as follows. A blue rectangular area is placed on the RM as shown in Figure 2-7; P_r is the extracted point cloud with regard to this rectangle. The *CheckPointCloud* function in Equation 2.8 iterates all the points in P_r, eliminates empty values, and reorganizes the remaining points, returning the median point along the z axis as $p(x, y, z)$ in the 3D point-cloud space.

$$p(x, y, z) = CheckPointCloud(P_r) \tag{2.8}$$

Furthermore, we push the point back along the z axis by a distance d, which is half the z dimension of the model. Eventually, the initial-position guess $p(x, y, z)$ can be calculated. In practice, $p'(x, y, z)$ would be compensated further considering the container frame position.

This newly introduced approach allows good initial poses to be calculated with higher accuracy and better tolerance to sensor noise and object occlusions.

Regarding model cloud processing, limited camera positions mean that only one side of the container can be scanned. Providing a full model of the container directly to the registration would result in poor performance because the external points could act as outliers.

Additionally, some containers such as the juice box shown in Figure 2-8 have movable parts that can influence the registration. Herein, we use only the most stable and reliable parts of the container when constructing the mesh model.

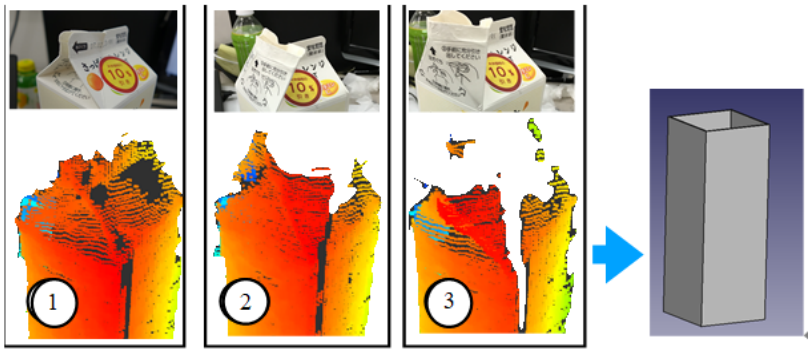


Figure 2-8: Valuable parts of a container.

In [15], the self-occlusion issue was solved using a multi-hypothesis approach. Because most daily containers have relatively simple shapes (mainly cubes and cylinders), we can decrease the algorithmic complexity by having far fewer candidate crops (in [15], 30 crops were required for each object). As shown in Figure 2-9, because the object clouds from bottles scanned from different angles are similar, only one candidate crop is needed in that case.

The situation becomes slightly more complicated when cubic containers are involved. Two types of target cloud are possible, namely the "half" type and the "one face" type. Two candidate model clouds that comprise only those parts labeled with green lines are provided to the ICP registration, and for pose estimation we choose the one with the lower registration error.

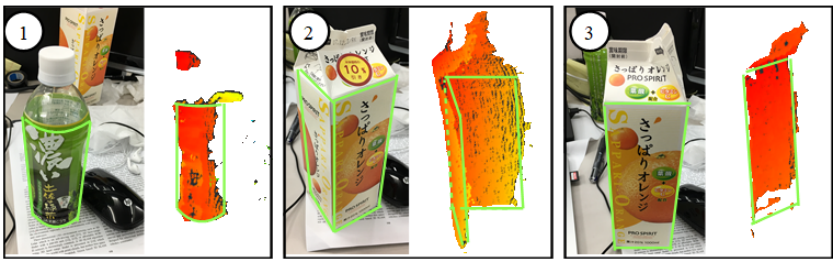


Figure 2-9: Self-occlusion examples.

Partial occlusions occur because the target container is partially hidden by other objects, making the scanned container surfaces incomplete. Herein, we consider only partial occlusions caused by other objects on the same surface with the target container.

The multi-hypothesis method itself offers a certain degree of tolerance of partial occlusions. In Figure 2-10-1, the coffee can is causing an occlusion: two surfaces of the juice box are scanned, one incompletely so. In this case, the half-type model cloud would perform poor registration whereas the one-face model cloud would register the complete surface accurately. Therefore, no extra effort is required to deal with the occlusion in this situation.

However, if both scanned surfaces are incomplete because of the coffee can (Figure 2-10-2), the pose is estimated inaccurately. In this situation, a suitable model cloud could be provided by cropping the model upward from the bottom. In previous work, we cropped the model with a constant step length until the registration error ε decreased below a given threshold ε_t.

However, it is difficult to choose the step length because doing so involves a trade-off between speed and accuracy. Herein, we propose an approach involving a model cropped with a variable step size. During the experiment, we noticed that the ICP

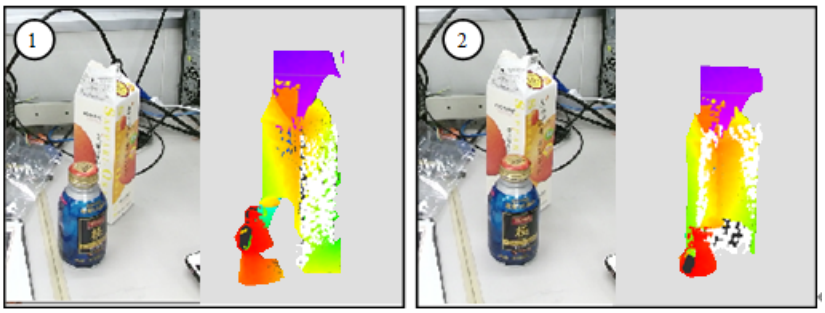


Figure 2-10: Examples of partial occlusions.

registration error ε could also be a metric for the degree of partial occlusion. Therefore, we calculate the model-cropping step length l as:

$$l = \mu\varepsilon \tag{2.9}$$

where μ is an empirical value. Similar to the principle of a classical P controller, l can be calculated based on the degree of partial occlusion. Two successful registration scenarios are demonstrated in Figure 2-11.

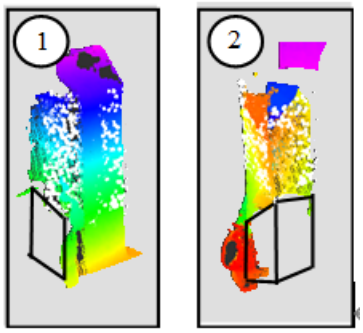


Figure 2-11: Successful registration scenarios with partial occlusions.

2.2.3 Experiments

Evaluation of Effectiveness and Accuracy

In the first experiment, we used a Kinect 2 RGB-D camera as the data source. We placed several daily containers on a crowded desk and used a maker board to provide

the ground truth of the target-container poses (Figure 2-12).

We began with pose estimation with no partial occlusion, taking a juice box as the target container (Figure 2-12-1). We conducted 20 estimations of various poses relative to the camera. The position error was less than 2 mm and the orientation error was less than 2 degrees.

Next, we conducted 20 estimations of various poses with the juice box partially hidden (Figure 2-12-2). We note that the error bounds increased to 1 cm in position and 7 degrees in orientation considering 18 successful registrations. The remaining two trials ended with failed information because ε did not decease below ε_t after multiple model cropping. Nevertheless, the performance fulfills the requirements of container picking.

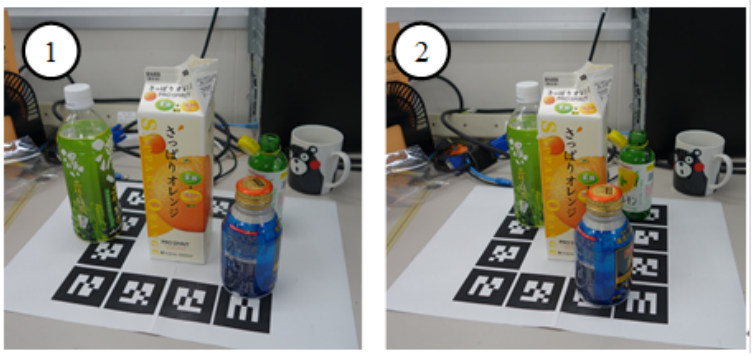


Figure 2-12: Scenarios used to evaluate accuracy.

Container Fetching with KUT-PCR

The second experiment was to demonstrate various fetching tasks performed with a KUT-PCR robot. The scenario involved a working desk that was excluded from the model training phase to challenge the pipeline's generalization ability.

We placed several daily containers on the desk along with some tools. Figure 2-13-1 to Figure 2-13-4 show one of 10 fetching tasks conducted with the right arm, and Figure 2-13-5 to Figure 2-13-8 show the same task recorded from another angle.

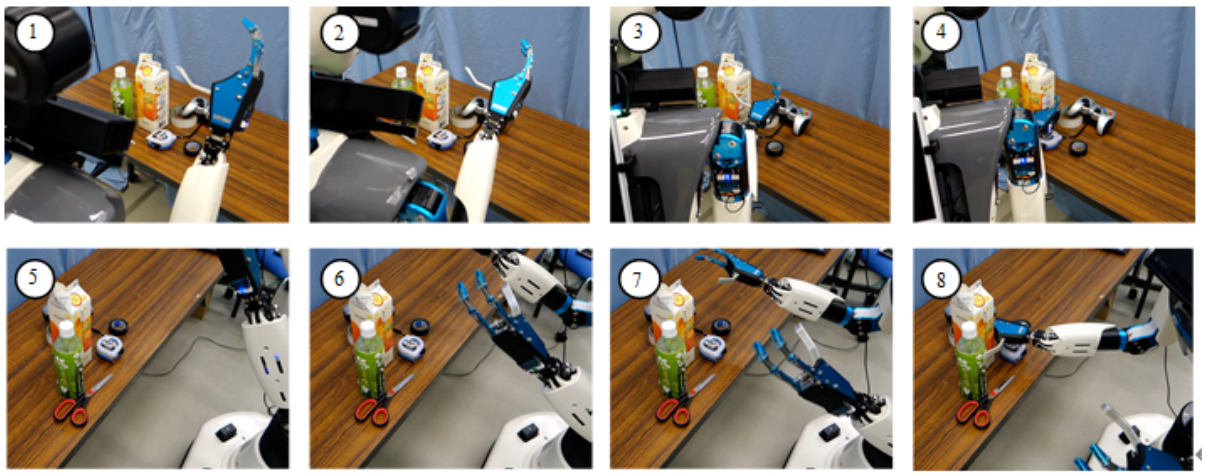


Figure 2-13: Container-fetching experiment with KUT-PCR.

2.3 Context-Aware Local Planning

Personal care robots can help with the everyday activities of people who suffer from illness, disease, or injury. A fine-tuned local planner is essential for a robot to follow a planned path while avoiding obstacles. Within this well-studied domain, our area of interest is a local planner that can change its configurations actively considering the environmental context.

In the field of robotics, context awareness refers to the ability of a robot to sense and react according to its environment [16]. A semantic-aware robotic system typically comprises two functions, namely (i) recognizing different environmental semantics and (ii) reacting accordingly with reasonable actions. In the present work, we focus on the latter function.

Previous research has introduced context-aware global planning methods that allow a robot to plan a global path without invading human workspaces [17]. However, in scenarios involving homecare in household environments, interacting closely with people is inevitable. Because avoiding certain locations is not an option, we argue that a context-aware local planner is required for the robot to deal with diverse scenarios.

The need for a context-aware local planner can be explained with a simple caring scenario involving a human caregiver. While caring for a bedridden patient, the caregiver walks relatively quickly through doorways and empty rooms (i.e., where

there are no other people). However, upon entering the bedroom, the caregiver slows down so as not to disturb the patient. Furthermore, when conducting activities near the patient or other people, the caregiver pays extra attention so as to avoid accidents such as colliding with people or dropping objects. Put simply, a human caregiver performs homecare tasks with different behavioral characteristics depending on where she/he is and who/what is around.

Herein, we first present a dynamic re-configurable local planner so that suitable behavioral characteristics can be triggered with pre-defined semantic labels. We then introduce a complete system describing how a robot can take advantage of the proposed planner. Finally, through extensive experiments in an actual household domain, we establish that the proposed approach is effective.

2.3.1 Related Work

Semantic information can provide considerable impetus for advances in robotics applications, especially when considering human environments. Researchers have previously built a system based on a state-of-the-art convolutional network, generating semantic labels for indoor (e.g., kitchen, office, corridor) and outdoor (e.g., parking garage, food court, café) environments [18].

With the means to sense or predict semantic information in an operating environment, considerable corresponding efforts have been made to enable robot motion that is more reasonable and effective, the main direction being semantic-aware path planning. Human motion patterns have been learned based on sampled hidden Markov models and used in a path-planning algorithm based on a probabilistic roadmap [17]. This was done to minimize social distractions such as going through someone's working space. From another perspective, avoiding the need to observe people, an affordance map has been proposed that describes the environment considering geometric features, whereupon a global planning method A* could be conducted aimed at a semantic-aware global path [16].

The aforementioned approaches are focused on minimizing the extent to which robots affect people by preventing the former from entering the working spaces of the

latter. However, to the best of our knowledge, re-configuring a local planner [6] in run time to achieve better performance is yet to be discussed.

2.3.2 Method

In this section, we introduce a robot controller based on a dynamic window approach (DWA). We also discuss how we modified the original approach to be able to adjust the robot motion given a pre-defined semantic label.

Dynamic Footprint

The DWA generates actuator commands so that the robot can follow the global path without running into obstacles or violating the dynamic limitations of the actuators. An initial step is to define the footprint of the robot with a polygon, which is the basis for collision checking in the following procedures. In most cases, the footprint is hardcoded and remains unchanged unless there is a mechanical re-design (2-14).

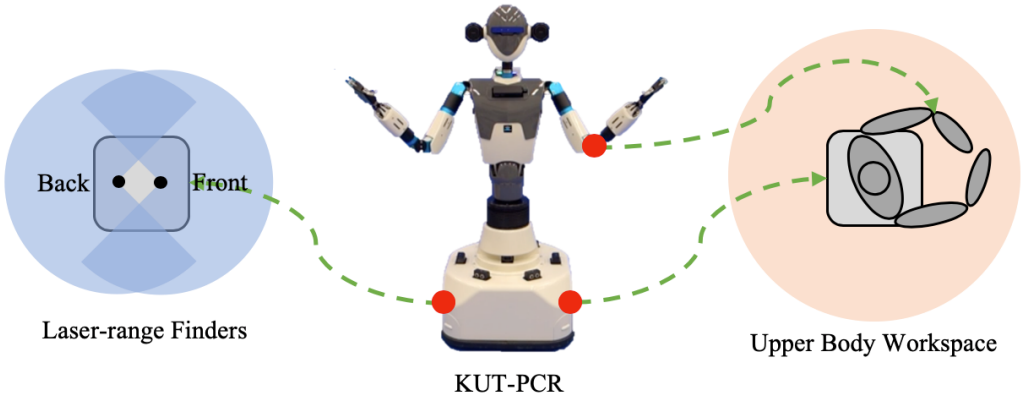


Figure 2-14: Perception range and work space.

In our implementation, the robot footprint is assumed to be dynamic with two considerations, namely that (i) different levels of safety should be achievable by adjusting the size of the footprint and (ii) for a mobile humanoid robot, the configurations of the upper body should also be considered for controlling the motion of the mobile platform.

Figure 2-15-1 shows the upper body remaining within the mobile platform; the light-blue area is referred to as the inflation footprint, obtained by extending the robot shape in four directions with the inflation lengths η_f, η_l, η_b, and η_r. The original footprint considering only the mobile platform without inflation is a square of side $2R$, while an actual footprint is described by the distances from the origin to each side. Therefore, the expanded footprint in 2-15-1 is denoted as $F = R + \eta_f, R + \eta_l, R + \eta_b, R + \eta_r$. The inflation provides a "safe zone" between obstacles and the robot. A larger inflation body provides better capability to deal with possible collisions, such as those with fast-moving obstacles.

In 2-15-2, the forearms of the upper body reach beyond the mobile platform, therefore requiring a method for expanding the footprint considering the upper-body configuration. Evaluation points e_{le}, e_{lw}, e_{re}, and e_{rw} represent the coordinates mapped to the ground of the robot's left elbow joint, left wrist, right elbow joint, and right wrist, respectively.

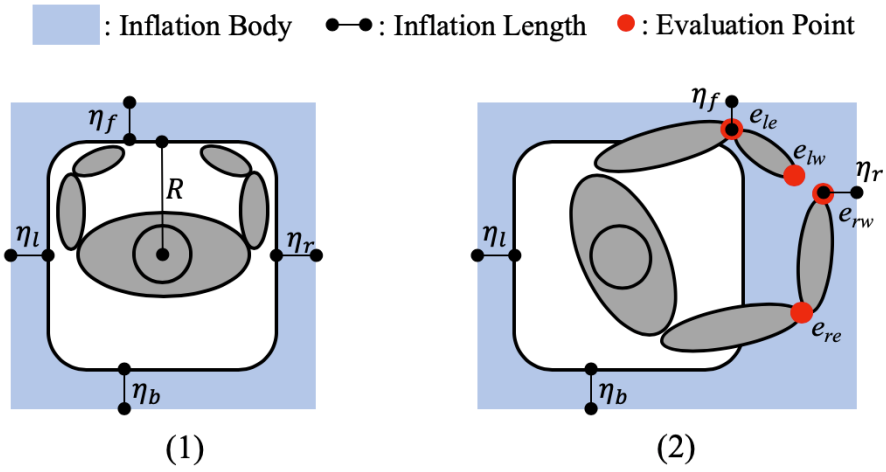


Figure 2-15: Dynamic footprint.

Table 2.1 presents a footprint inflation algorithm that calculates the expanded footprint on the fly with respect to the evaluation points on the upper body. First, a footprint F considering only the mobile platform is initialized. The evaluation points are then passed to the function $CALC\_RECTANGLE$ that calculates the minimum bounding rectangle REC of the input point sets by using the gift-wrapping

Table 2.1: Footprint Inflation Algorithm

Algorithm: INFLATION

Input: Evaluation points e_{le}, e_{lw}, e_{re}, and e_{rw}
Output: Expanded footprint $\boldsymbol{F}$
// Initialize a footprint.
$$\boldsymbol{F} = \{R + \eta_f, R + \eta_l, R + \eta_b, R + \eta_r\}$$
// Calculate the minimum bounding rectangle from the evaluation points.
$$\boldsymbol{REC} = CALC\_RECTANGLE(e_{le}, e_{lw}, e_{re}, e_{rw})$$
// Evaluate each distance in the generated rectangle.
for $i = 1$ to 4 **do**
    if $\boldsymbol{REC}[i] > R$ **then**
        $\boldsymbol{F}[i] = \boldsymbol{REC}[i]$
    end if
end for
return $\boldsymbol{F}$

method. We then evaluate each distance (origin to side distance) in REC with R; if the distance exceeds R, then the corresponding distance in F is replaced. Eventually, the resulting footprint is returned.

Additional Reduction of Velocity Search Space

KUT-PCR sits on an omnidirectional mobile platform: transitional/rotational motion is generated by four omnidirectional wheels and is denoted as $v(\dot{x}, \dot{y}, \dot{\theta})$. Normally, the search space of the possible velocities is reduced as follows:

1) **Circular Trajectories**: the DWA considers only circular trajectories, which are determined by a given velocity comprising the two-dimensional translational speed $(\dot{x}, \dot{y})$ and the one-dimensional rotational speed $\dot{\theta}$. This leads eventually to a three-dimensional velocity search space. For clarity, we discuss the problem in a two-dimensional space considering only $\dot{x}$ and $\dot{\theta}$. Initially, the space V_s is determined by the actuator limitations (maximum and minimum speed) (Fig 2-16).

2) **Admissible Velocities**: avoiding obstacles is the top priority for a robot. Any generated circular trajectory that could result in colliding with an obstacle is considered a failed trajectory and the corresponding velocity is regarded as inadmissible. All the safe trajectories in the initial space establish a space V_a.

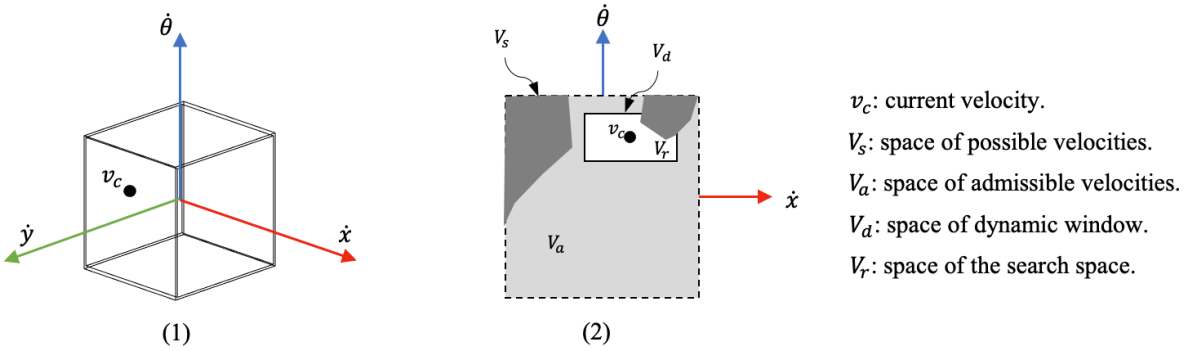


Figure 2-16: Velocity search space.

3) **Dynamic Window**: originally, the dynamic window V_d contains the velocities that are reachable in the next time interval, considering the limited accelerations of the actuators. Thus, the final velocity search space V_r is calculated by $V_r = V_s \cap V_a \cap V_d$. In our application, we assume that reducing the dynamic window and initial velocity space further results in different motion patterns fitting certain requirements.

4) **Reduction of** V_s: usually, robots tend to exhaust the power of the actuators to obtain the highest performance (greatest efficiency). However, in personal care scenarios, there can be additional restriction on the actuators. In Figure 2-17, we reduce the rotational speed equally in both directions to avoid high-speed turning actions. The speed in the forward direction is similarly limited, whereas the limitation on the backward direction is obviously larger. This indicates that although the backward speed is required to avoid dynamic obstacles, high-speed backward operation in a human-centered environment is dangerous. The reduction ζ_s is denoted as:

$$\zeta_s = \{\dot{x}_{max}, \dot{x}_{min}, \dot{y}_{max}, \dot{y}_{min}, \dot{\theta}_{max}, \dot{\theta}_{min}\} \tag{2.10}$$

where $\dot{x}_{max}$,$\dot{x}_{min}$ are the velocity limitations in the positive and negative directions, respectively, on the x axis, and $\dot{y}_{max}$, $\dot{y}_{min}$, $\dot{\theta}_{max}$, $\dot{\theta}_{min}$ are the restrictions on the y and θ axes, respectively.

5) **Reduction of V_d**: while V_s defines the operation bounds, the dynamic performance considering acceleration/breaking is determined by V_d. The original dynamic window V_d is typically a rectangle located symmetrically around the current speed, but we add an additional limitation for further reduction. For instance, although high-speed backward operation must be prohibited, we expect backward motion while avoiding a dynamic obstacle to be efficient (i.e., with high acceleration). The reduction of V_d is described with ζ_d, namely:

$$\zeta_d = \{\ddot{x}_{max}, \ddot{x}_{min}, \ddot{y}_{max}, \ddot{y}_{min}, \ddot{\theta}_{max}, \ddot{\theta}_{min}\} \tag{2.11}$$

where $\ddot{x}_{max}$ and $\ddot{x}_{min}$ are the acceleration limits in the positive and negative directions, respectively, of accelerations $\ddot{x}$, $\ddot{y}_{max}$, $\ddot{y}_{min}$, $\ddot{\theta}_{max}$, $\ddot{\theta}_{min}$ are the limitations of accelerations $\ddot{y}$ and $\ddot{\theta}$, respectively, and T is the time interval. By tuning parameters η_f, η_b, η_l, η_r, ζ_s, and ζ_d, different behavioral patterns of the robot can be defined freely.

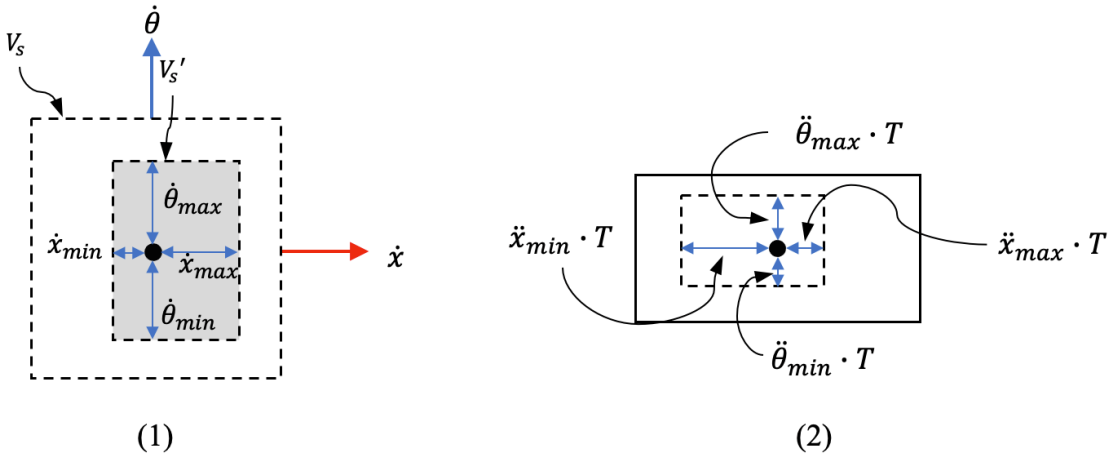


Figure 2-17: Additional limitation on velocity search space.

2.3.3 System Configuration

We now introduce a complete system (Figure 2-18) that enables a personal care robot to reconfigure the local planner dynamically considering environmental semantics.

The dark blocks refer to the modules introduced herein, while the light ones are robot-specific software packages. Block 1 comprises navigation and object-detection functions; the module perceives the environment continuously and passes semantic labels to block 3 if certain locations are reached or objects are detected. Block 3 keeps a list of all the pre-defined patterns; given a semantic label, it simply goes through all the patterns and chooses the configuration corresponding to the given semantic label. Meanwhile, the checking points sent from block 2 are evaluated by block 4, deciding how the footprint should be adjusted. Blocks 3 and 4 both produce planner parameters that are then sent to block 5. In the system, the communication between blocks 5 and 6 takes place at 10 Hz, while the rest are event-driven with no fixed updating frequency.

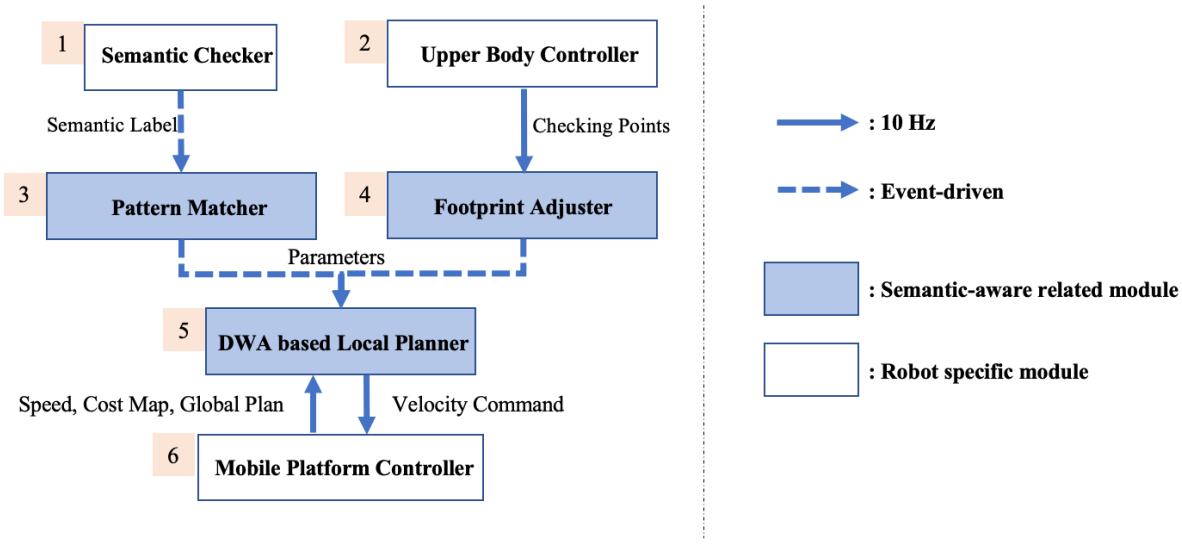


Figure 2-18: Control system diagram.

2.3.4 Experiments

To evaluate the proposed method, we conducted experiments in an actual household environment. In the first experiment, we examined how the dynamic footprint guaranteed safety during operations of a humanoid mobile platform. In the scenario, KUT-PCR was expected to fetch objects from the refrigerator and deliver them to the next-door room. In trial 1, the robot did not pick up anything. In trial 2, the

robot picked up a bottle of drink; it held the bottle with two arms stretched out.

As Figure 2-19 shows, in trial 1 the upper body stayed within the range of the mobile platform, thus the robot finished the task having strictly followed the global path. In trial 2, with the arms stretched outside the mobile platform, the dynamic footprint method kept the robot away from the wall with a path that departed slightly from the planned global path.

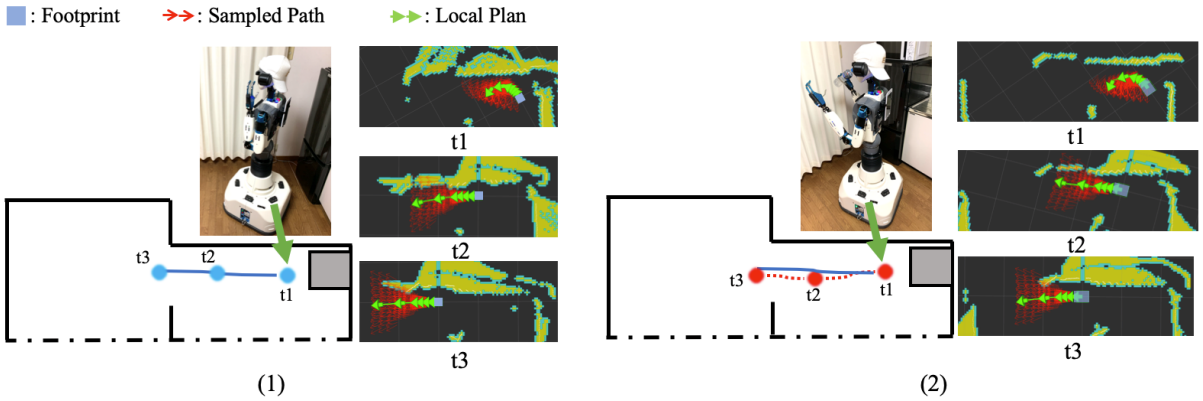


Figure 2-19: Experimental trials without and with dynamic footprint.

In the second experiment, we conducted a fetch-and-serve task commonly seen during caring services, and we examined how the behavioral patterns of the robot changed when faced with different semantic situations. Figure 2-20 demonstrates the parameter configurations of four behavioral patterns including "Kitchen", "Bedroom", "Doorway", and "Near Patient". Parameters considering footprint size, velocity and acceleration are described with five levels: very small, small, middle, big, and very big. The parameters served the purpose of our applications and can be adjusted freely to meet diverse requirements.

Pattern ID	Name	Footprint Size	vel_x forward	vel_x backward	vel_y	vel_w	acc_x	acc_y	acc_w
1	Kitchen	●●●	●●●●	●	●●●	●●●	●●●●	●●●	●●●●
2	Bedroom	●●●●	●●●	●	●●●	●●	●●●●	●●●	●●●●
3	Doorway	●	●●	●	●●	●	●●●	●●●	●●
4	Near Patient	●●	●	●	●	●	●●●	●●●	●●●

●: Very Small ●●: Small ●●●: Middle ●●●●: Big ●●●●●: Very Big

Figure 2-20: Behavioral pattern configurations.

As shown in Figure 2-21, at t1, t2, and t3 the robot detected semantic locations ("Doorway", "Bedroom", and "Kitchen"); at t4 the object-dependent semantic label "Near patient" was triggered. The robot succeeded in switching among four predefined behavioral patterns dealing with different semantic scenarios.

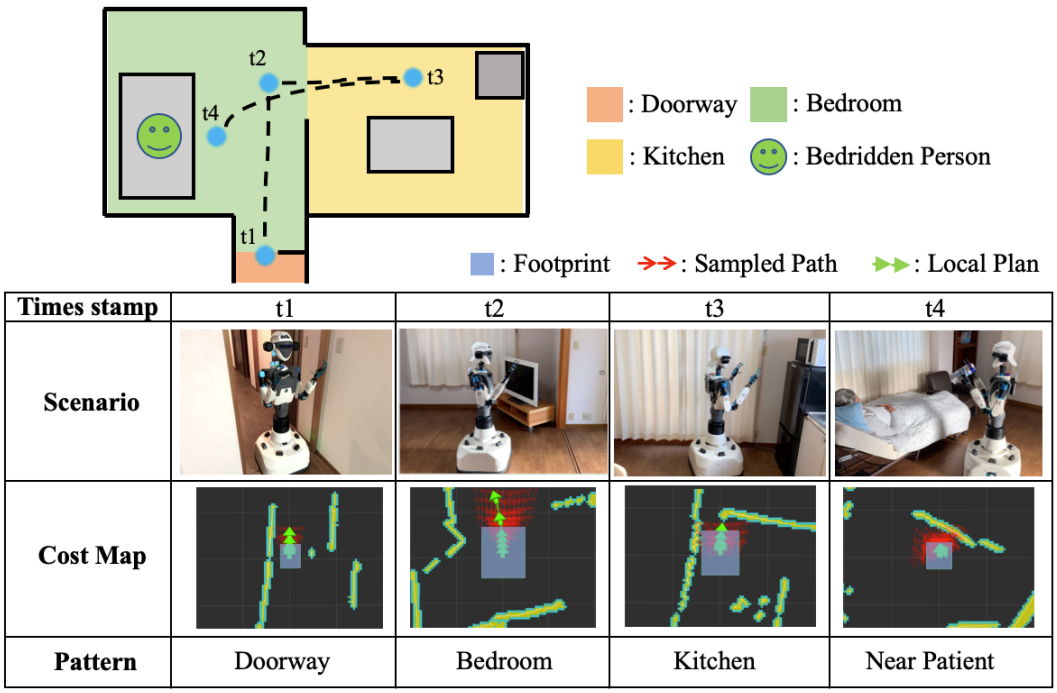


Figure 2-21: Personal care trial considering environmental context.

2.4 Desire-Driven Reasoning

Several types of robots have been developed to provide daily care services. For instance, there are robots that can provide mechanical assistance to caregivers or care recipients: a nursing-care assistance robot called RIBA [19] is able to transport a patient between their bed and a wheelchair; exoskeletons can amplify the strength of caregivers [20]; and transfer robots from TOYOTA can carry out tasks when operated locally by a caregiver. In recent years, non-contact robots have also been developed that can carry objects or open and close curtains, such as TOYOTA's HSR [21].

Since the first type of robot operates directly on patients who are bedridden, caregivers who understand their operations must also be present. By contrast, since

robots of the latter type do not make direct contact with patients, caregivers are not required to be present. However, these robots must still be given clear commands, such as "fetch a bottle of tea" or "set the room temperature to 25 degrees," and patients may struggle to remember whether there is any tea in the refrigerator or decide what exact temperature they want to set for the room. Patients generally find it much easier to express their physiological needs, such as hunger, thirst, being too hot or cold, needing brighter or dimmer light, or wanting fresh air. If personal care robots could understand such physiological needs and take appropriate actions to satisfy them, a higher intelligence level would have been reached, and these robots could eventually contribute in real caring scenarios.

For personal care robots to perform actions that satisfy given physiological needs when caregivers or home helpers are unavailable, they must at least have the following functions: (i) estimate their own location and those of the target objects; (ii) reach the desired destination while avoiding obstacles; (iii) recognize the presence and state of objects; (iv) manipulate objects; (v) identify patients' physiological needs; and (vi) reason about the objects and operations required to satisfy these physiological needs.

Functions (i) and (ii) have long been fundamental to research into autonomous mobile robots. In recent years, simultaneous localization and mapping (SLAM) [22] proved to be highly successful and has been applied in various fields [23][24]. Function (iv) is critical for industrial robots, for which several different mechanical designs have been proposed, along with methods of controlling the end effectors [25]. Great progress has also been made toward function (iii), including the development of image sensors (high-speed industrial cameras [26] and depth cameras [27]) and advanced algorithms (deep learning [28]). In particular, dramatic advances have been made by combining work on functions (iii) and (iv), as demonstrated by multiple global competitions, such as the DARPA Robotics Challenge (DRC) [29] and the Amazon Picking Challenge (APC) [30].

These techniques can also be applied to personal care robots. For instance, given the command to serve a bottle of water, HSR can pick up the bottle and serve it to the care recipient [21], while Dora [31] can navigate between rooms to search for

a book given its name. Such achievements have gained much attention in robotics research and shown convincing results. However, to the best of our knowledge, there has been little discussion on how to recognize physiological needs (such as thirst) and infer how they can be satisfied by particular objects or actions.

Our previous studies on personal care robots presented results related to functions (iii) and (v). In this part, we instead concentrate on function (vi) and propose a method of reasoning about the objects and operations needed to satisfy a given physiological need. In addition, we also evaluate the proposed approach in real household scenarios. Implementing function (vi), however, presents two main difficulties.

First, there is a range of possible options: since more than one object or action may be able to satisfy a given need, we need a reasoning method that can determine which one to choose. For example, there may be several instances of a particular object class in multiple locations (e.g., a house can have two or more windows and doors) or a single action may involve different states (e.g., an electrical switch can be on or off). Second, operations can be uncertain; for example, a robot may reach the target location only to find that an object does not exist or an action is impossible to perform.

2.4.1 Related Work

In recent years, developments in mechanical design [32], actuator performance [33], and sensor properties [34] have resulted in increasing numbers of robots being deployed in various fields. Robots can now navigate complex environments [35][36], interact with people [37], and manipulate different types of objects [38]. However, in order for them to perform such tasks more intelligently, further research into knowledge representation and reasoning will be required.

Various approaches to this problem have been proposed, all focusing on generating a series of robot actions given a clear command such as "fetch a bottle of cola." For such human–robot interaction (HRI) tasks, the BC action language can be used to formalize both sensing and physical actions, enabling service robots to behave intelligently while dealing with incomplete information, underspecified goals, and dynamic

changes [39].

Answer set programming (ASP) is suitable for representing and reasoning with commonsense knowledge. Partially observable Markov decision processes (POMDPs) provide a mathematical framework that enables autonomous robots to solve motion-planning problems in uncertain and dynamic environments. ASP and POMDPs can also be combined to automatically tailor sensor input processing and navigation methods for robots deployed in partial domains [40]. In addition, PDDL [41] is a domain definition language for specifying deterministic planning domains and problems. When combined with heuristic search methods, such as the fast downward planning system [42], it can address many planning or even control problems.

Researchers also attempted to build higher-level knowledge systems that are not limited to one or two representations but can instead handle different tasks by taking advantage of different techniques. Integrating various methods (such as probabilistic graphs, PDDL, or POMDP solvers) into one framework can enable robots to plan in the face of uncertain and incomplete information [31], and this idea has been implemented in a mobile robot platform.

The ontology-based unified robot knowledge (OUR-K) framework [43] has also been introduced for service robots, and it includes both knowledge descriptions and associations. Other researchers also discussed how to structure knowledgebases by combining different knowledge areas [44], going on to propose the KNOWROB framework, which introduces representational structures and a common vocabulary for representing knowledge.

Although these frameworks have successfully addressed problems in various fields, most researchers have concentrated on solving the problems caused by vague or incomplete information when given task-oriented instructions; as far as we are aware, few have considered situations where there are no instructions in the first place. In scenarios involving caring for patients who are bedridden, there is a gap between their abstract physiological needs (e.g., “hunger”) and the corresponding instructions (e.g., “fetch a pack of biscuits”).

2.4.2 Household Environment Description

Human caregivers can provide appropriate service because they have two types of knowledge: commonsense knowledge that describes how various objects could contribute to satisfying a need and instance knowledge that describes properties of objects in the household environment, such as their locations and weights.

Commonsense Knowledge

It is generally assumed that cognitive activities, such as reasoning and decision making, presuppose the existence of a conceptual system in the person's memory. For example, a caregiver may give someone a bottle of tea if that person feels thirsty because their understanding of "tea" includes the idea that "tea can satisfy thirst." For a robot to do likewise, it would also need a thorough understanding of concepts related to household environments, which we call commonsense knowledge.

Table 2.2 gives ten desires and ten objects commonly seen in personal care scenarios, listing the contribution of each object to satisfying each desire on a scale from 0.0 to 1.0, where 0.0 indicates that the object makes no contribution. For instance, milk makes contributions of 0.6 and 0.3 to satisfying thirst and hunger, respectively, while bread helps more with hunger (0.9) and juice helps more with thirst (0.8). Here, we only list some of the commonsense knowledge K_c related to one individual who is bedridden; the detailed values will vary between patients depending on their personal preferences and may also change during the personal care process. In this study, we do not focus on the acquisition and updating of commonsense knowledge so we consider K_c to be fixed for a given patient.

Instance Knowledge

Unlike commonsense knowledge, instance knowledge is dynamic, as it describes the objects' properties; here, this means their names, spatial properties, characteristics, and electrical states.

1) **Name**: an object's name identifies its type.

Table 2.2: Commonsense Knowledge

	HUNGER	THIRST	FRESH AIR	SILENCE	HIGHER TEMPERATURE	LOWER TEMPERA-TURE	HIGHER HUMIDITY	LOWER HUMIDITY	HIGHER ILUMILAT-ION	LOWER ILLUMI-NATION
MILK	0.3	0.6	0	0	0	0	0	0	0	0
JUICE	0	0.8	0	0	0	0	0	0	0	0
BISCUIT	0.9	0	0	0	0	0	0	0	0	0
BREAD	0.9	0	0	0	0	0	0	0	0	0
WINDOW	0	0	0.8	0.9	0.8	0.8	0.6	0.6	0.3	0.3
DOOR	0	0	0.6	0.9	0	0	0	0	0.8	0.8
TV	0	0	0	0.8	0	0	0	0	0	0
AIR CONDITI-ONER	0	0	0	0	0.9	0.9	0.8	0.8	0	0
HEATER	0	0	0	0	0.9	0	0	0	0	0

2) **Spatial properties**: the position of an object in the world can be defined using three-dimensional coordinates (x, y, z). For example, the positions of door d_1 and window $w_1$could be represented as $(x_{d_1}, y_{d_1}, z_{d_1})$ and $(x_{w_1}, y_{w_1}, z_{w_1})$, respectively (Figure 2-22). In addition, for objects such as doors, windows, and refrigerators, the positions of their movable parts can substantially affect their functional attributes.

   Since there are various different types of mechanical structures, we define a parameter ζ, called the "opening degree," to describe an object's spatial state. For example, the opening degree of a sliding window is described by $\zeta = 2\frac{d}{D}$, while $\zeta = \frac{\theta}{\theta_m}$ describes a push-pull door. The specific algorithms used for perception and to calculate ζ are delegated to the robot controller, and only ζ is stored in the spatial description. Thus, the spatial properties of the window in Figure 2-22 are completely described by $P_s = \{(x, y, z), \zeta\} = \{(0.0, 3.0, 4.0), 0.7\}$.

3) **Characteristics**: an object's characteristics describe its physical properties, namely its weight, volume, and state. For example, a bottle of milk may be defined as $P_c = \{w, v, s\} = \{2.5, 0.3, Liquid\}$.

4) **Electrical state**: An appliance may have multiple different functional states, which can dramatically affect its operation; we capture this in the electrical state ϑ. For example, an air-conditioner is described by:

   $\vartheta \in \{"OFF, Heating, Cooling, Ventilation"\}$ and $size(\vartheta) = 1$

   meaning that it has four operational states but can only be in one of them at any given time.

   Table 2.3 lists all the instance knowledge about the objects in the room shown in Figure 2-22. .

Now that we have defined both commonsense and instance knowledge, we can introduce the complete description for an object O. Since commonsense knowledge is used to describe the functions of an object class, each instance inherits the common-

Table 2.3: Instance Knowledge

	Name	Spatial Properties	Characteristics	Electrical State
w_1	"Window"	{(0.0, 3.0, 4.0),0.7}	–	–
d_1	"Door"	{(0.0, 7.0, 3.0),0.2}	–	–
m_1	"Water"	{(2.0, 0.3, 2.0), ""}	{2.5, 0.3,"*Liquid*"}	–
a_1	"Air-conditioner"	{(0.2, 1.5, 6.0), ""}	""	"*Heating*"

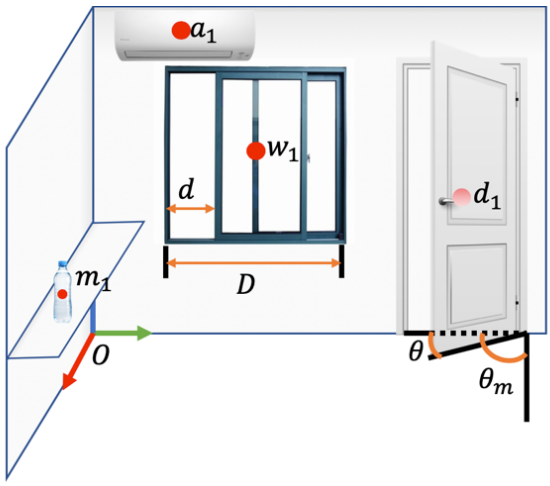


Figure 2-22: Instance knowledge for the objects in one room.

sense knowledge of its class. Specifically, an object O can be described as:

$$O = (n, K, P_s, P_c, P_e) \tag{2.12}$$

where n is the object's name. Here K is the commonsense knowledge retrieved from K_c based on n, namely:

$$K = K_c(n) = \{C_0, ..., C_k\} \tag{2.13}$$

where C_0,C_k are the contributions to the k desire types.

In addition, P_s, P_c, and P_e represent the object's spatial properties, characteristics, and electrical state, respectively:

$$P_s = \{(x, y, z), \zeta\} \tag{2.14}$$

$$P_c = \{w, v, s\} \tag{2.15}$$

$$P_e = \{\vartheta\} \tag{2.16}$$

Knowledge Integration

Now we have a way to describe objects in household environments, the other fundamental challenge is how to build and maintain a personal care knowledge base involving the objects in personal care scenarios. The robot should have a certain degree of prior knowledge when initially activated and then integrate new knowledge while interacting with the environment. Knowledge integration is the task of identifying how new and prior knowledge interact with each other and incorporating additional information into an existing knowledge base.

We divide the knowledge integration task into two basic operations: initialization and updating.

1) **Initializing the knowledge base**: the initialization step provides basic prior

knowledge about two types of objects: static and dynamic objects.

We define the knowledge base as KB. Static objects, such as doors and windows, are added directly to KB, while dynamic objects, such as food and drinks whose quantities and locations are unknown, are initialized as dummy objects based on commonsense understanding.

For example, in Figure 2-23, each type of object maintains a list of objects of that type in the current environment. Here, the house is assumed to have two doors, three windows, and one bed. The figure shows KB's initial state; the locations of the static objects are fixed and will not be further verified by the robot system, while the initial dummy objects for juice, bread, and biscuits (based on commonsense understanding) will be verified and updated as the robot searches the environment.

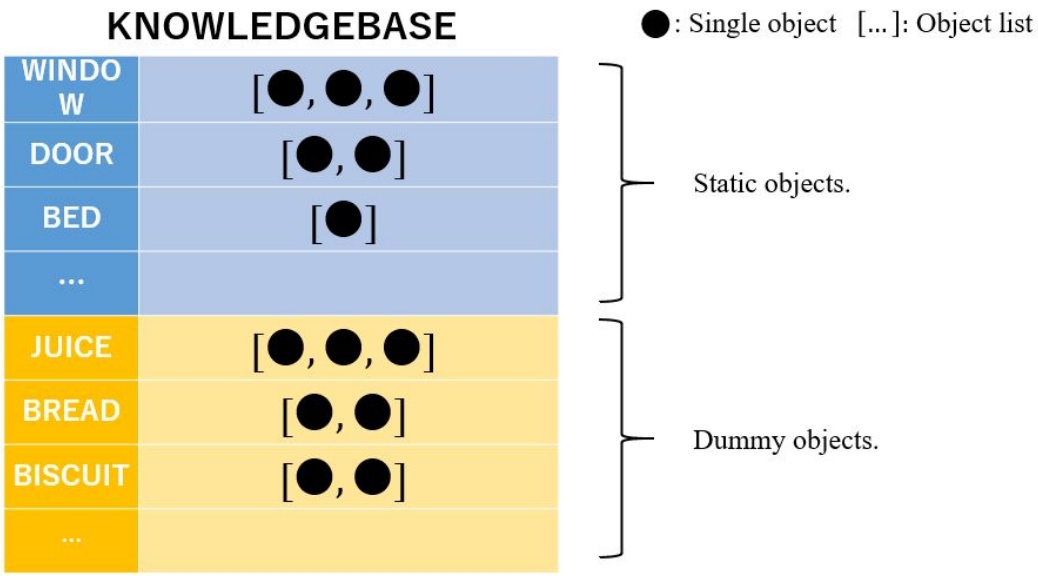


Figure 2-23: Example of initializing a knowledge base.

2) **Updating the knowledge base**: the update step proceeds based on the robot's perceptions. The result of a valid perception is denoted as $P = (f, KB_p)$, where f describes the spatial field that the robot has perceived and KB_p is a small knowledge base containing the objects identified.

Table 2.4: Knowledge Base Update Algorithm

Algorithm: UPDATE (*KB, P*)

Input: Knowledgebase *KB*, Perception $P = (f, KB_p)$

Output: Updated knowledgebase *KB*

// Fetch all objects from KB that are located in the field P.f.

$$KB_s \leftarrow KB.find(P.f)$$

// Calculate objects to add to the knowledgebase.

$$KB_{add} \leftarrow \{o | o \notin KB_s \text{ and } o \in P.KB_p\}$$

// Calculate objects to delete from the knowledgebase.

$$KB_{del} \leftarrow \{o | o \in KB_s \text{ and } o \notin P.KB_p\}$$

// Perform the addition and deletion operations.

$$KB \leftarrow KB.add(KB_{add})$$

$$KB \leftarrow KB.delete(KB_{del})$$

RETURN *KB*

As Figure 2-24 shows, for a given perception $P = (f, KB_p)$, the objects in KB located within the perceived field f are first fetched and then used to build a sub-knowledge base KB_s. The intersection between KB_s and KB_p consists of the objects verified by the perception P. Objects that are in KB_s but not KB_p should be deleted from KB, since they cannot be identified in their recorded locations, while objects in KB_p but not KB_s should be added to KB. Table 2.4 shows the algorithm used to update the knowledge base.

An object o can be considered to belong to the knowledge base KB if there is an object o_k in KB that is equal to o. The definition of object equality is as follows:

IF $o_k.name = o.name$ **AND** $DIS(o_k.P_s, o.P_s) < \gamma$, **THEN** $o = o_k$

where the DIS function calculates the spatial distance between the two objects and the threshold γ accounts for factors such as localization and perception error.

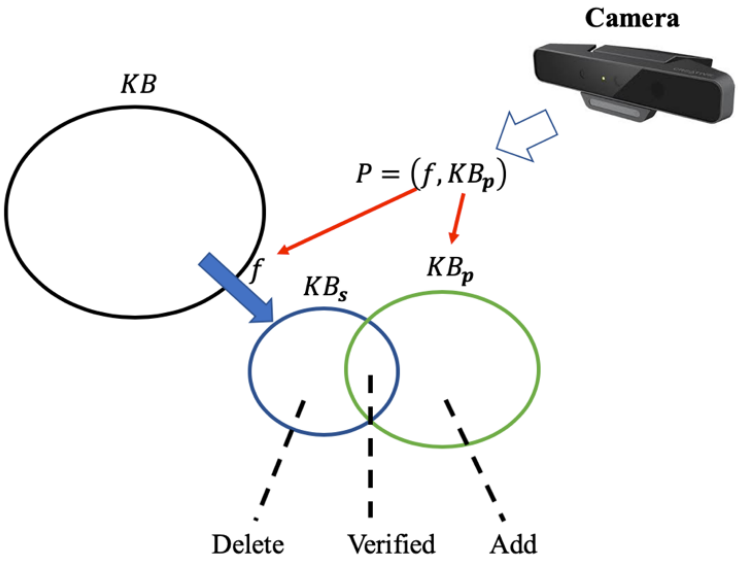


Figure 2-24: Updating the knowledge base.

2.4.3 Desire-Driven Reasoning Approach

Desire-driven reasoning is defined as reasoning via a sequence of steps with the aim of meeting given desires (in this case of people who are bedridden).

Desire-Driven Reasoning System

Figure 2-25 shows a flowchart of the proposed desire-driven reasoning (DDR) system. First, the knowledge base KB is initialized. When the robot is activated by a particular patient desire d, this is passed to the DDR module. This then reasons about suitable goals, considering d and KB. The resulting goal is then sent to the planner, which generates an action list for the robot to execute.

Two main loops define the robot's behavior, including execution, perception, and knowledge updating. The robot controller executes a loop consisting of a motion execution command followed by a perception query command. If nothing is identified during a given iteration, the loop continues until the action list is confirmed exhausted, indicating the task is complete. If, during this process, the robot acquires a valid perception (either recognizing objects or the locations of objects in KB), KB is updated using the knowledge integration method, and a new action list is calculated

with the same d but the newly updated KB. Then, the controller begins executing the newly generated action list.

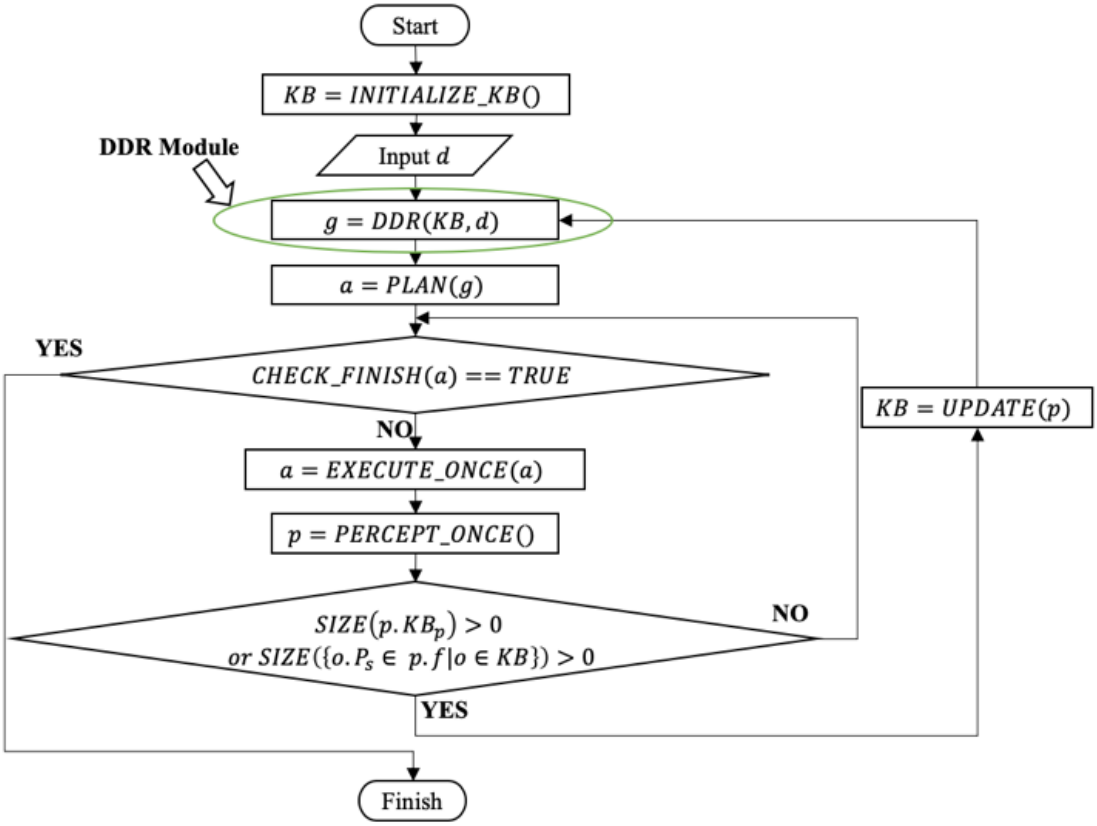


Figure 2-25: Flowchart for the desire-driven reasoning system.

Desire-Driven Reasoning Module

The system's core component is the DDR module, which reasons as follows (Figure 2-26): (i) fetch candidates from the knowledge base that can contribute to meeting the given need; (ii) evaluate the candidates; (iii) select the highest-rated candidate as the goal.

Table 2.5 describes the DDR algorithm. First, the FIND function fetches all objects that make contributions greater than 0 to the given desire d from the knowledge base KB. Then, the EVALUATE function ranks the objects. Finally, the SELECT function selects the highest-scoring object, which is returned to the task planner as the goal for further planning.

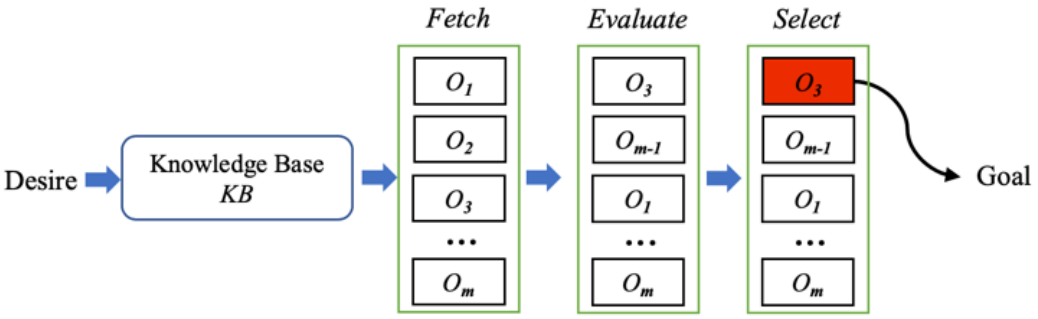


Figure 2-26: Workflow of the DDR module.

Table 2.5: DDR Algorithm

Algorithm: DDR(KB, d)

Input: Knowledgebase KB, Desire d

Output: Goal G

// Fetch objects that can contribute to fulfilling the given desire.

$$\boldsymbol{O} \leftarrow KB.FIND(d)$$

// Evaluate the objects and create a list of object–rank pairs.

$$\boldsymbol{O}_{rank} \leftarrow EVALUATE(\boldsymbol{O}, d)$$

// Select the highest-scoring object as the goal.

$$G \leftarrow SELECT(\boldsymbol{O}_{rank})$$

RETURN G

Table 2.6: Object Ranking Algorithm

Algorithm: $RANK(o, d)$

Input: Object o, Desire d

Output: Rank R

// Fetch the object's contribution to meeting the given need.

$$R_{contribution} \leftarrow o.K.FIND(d)$$

// Calculate the cost of transportation and manipulation.

$$R_{cost} \leftarrow COST(o.P_c, o.P_s, o.P_e, d)$$

// Calculate the overall rank.

$$R = \alpha R_{contribution} + \frac{\beta}{R_{cost}}$$

RETURN R

The key element of the EVALUATE function is the object ranking method, which considers two aspects: the object's contribution to fulfilling the need and the operation cost. Table 2.6 describes how the object ranks are calculated. First, the $FIND$ function retrieves the contribution $R_{contribution}$ made by the object o to fulfilling the desire d from the commonsense knowledge base K. Next, the $COST$ function calculates the operation cost R_{cost}, based on the given desire and the object's characteristics, spatial properties, and electrical state. Finally, the overall rank R is calculated as a weighted sum of $R_{contribution}$ and R_{cost}. α and β are selected so that the importance of desire fulfilling contribution and task conduction cost can be reflected.

The $COST$ function depends on the object type, and the detailed implementation requires knowledge of navigation and vision systems that is beyond the scope of this paper. In short, it evaluates the transportation distance and the manipulation complexity based on the robot and object states and the given desire. The higher R_{cost} is, the more difficult it is for the robot to meet the given desire with the specified object.

Long Short-Term Care Preference

A desire-driven reasoning system (DDRS) bridges the gap between abstracted desire d and specific action a in consideration of care preference $\hat{o}$ over the available options. As shown in Figure 2-27-1, a constant LTCP knowledge base "LTCP KB" was used as the resource of $\hat{o}$. It simply fetches related preference data considering d and provides these data to the DDRS for further evaluation.

As shown in Figure 2-27-2, rather than a fixed knowledge base, a LSTCPM is used to generate $\hat{o}$ dynamically each time a request is made. The LSTCPM generates $\hat{o}$ in consideration of LTCP, given desire d, and data from the previous reasoning circle (including reference function ψ, request timestamp t, and executed action a). As a result, the sequential influence of service actions is modeled.

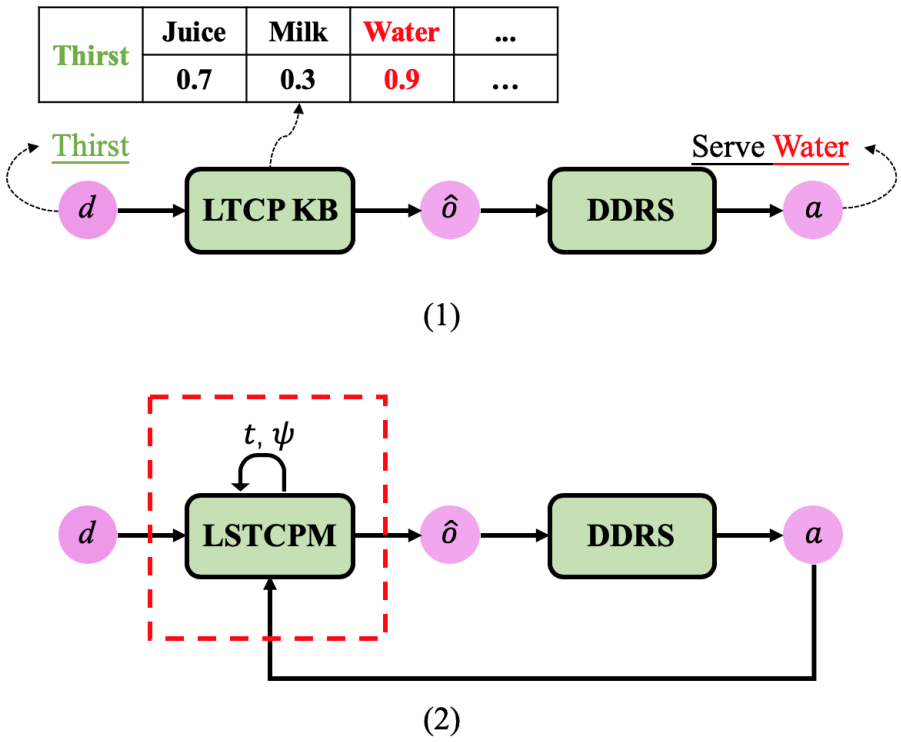


Figure 2-27: Flowchart of DDRS cooperating with LSTCPM.

In practice, requests can be considered as a time-aware sequence with uneven time intervals. Here, k is the index of the request, and t_k is the timestamp indicating when the request occurs. Figure 2-28 shows how the system works given a request list from

d_0 to d_k.

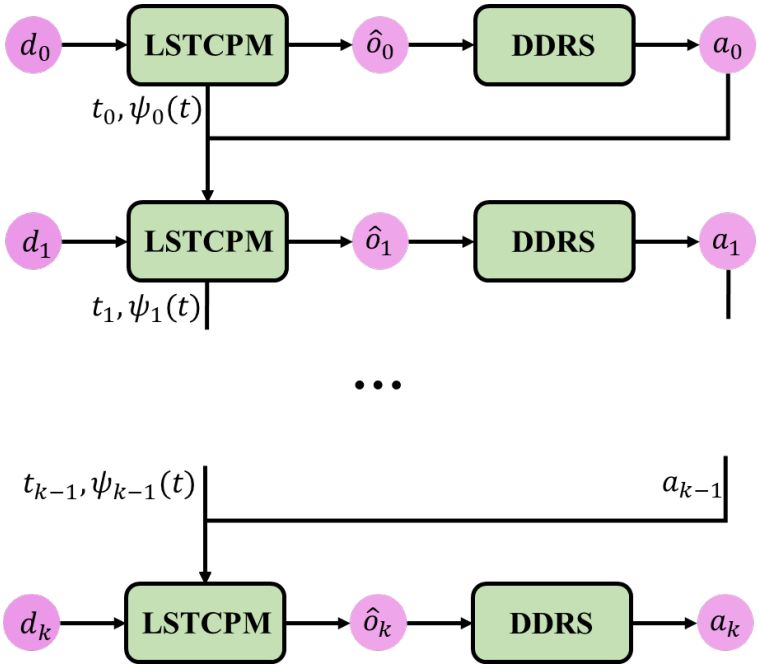


Figure 2-28: Workflow considering t_0 requests.

In addition to obtaining trusted feedback, another important step is updating the LTCP in consideration of the obtained feedback, which allows the LTCP to be optimized during interactions. The EKF [45] algorithm is well known, and we describe them to convey the implementation details of how the EKF can be adopted to estimate LTCP during personal care over time.

2.4.4 Experiments

To evaluate our KUT-PCR personal care robot, we conducted experiments in a real household domain. Here, the aim was to evaluate whether the proposed DDR method could enable the robot to carry out appropriate actions when given only a person's physiological needs.

Experimental Setup

The experimental domain (Figure 2-29) consisted of two rooms, namely a bedroom and a kitchen. There were three static objects: a bed (bedroom), dining table

(kitchen), and refrigerator (kitchen). The two rooms were connected by a sliding door. Figure 2-30 shows photographs of the domain, taken from the bedroom (left) and kitchen (right).

In addition, a patient who was bedridden lay on the bed, and KUT-PCR was initially at position S (Figure 2-31). When the robot was activated by the patient's desire, it began to perform the operations generated by the proposed DDR system.

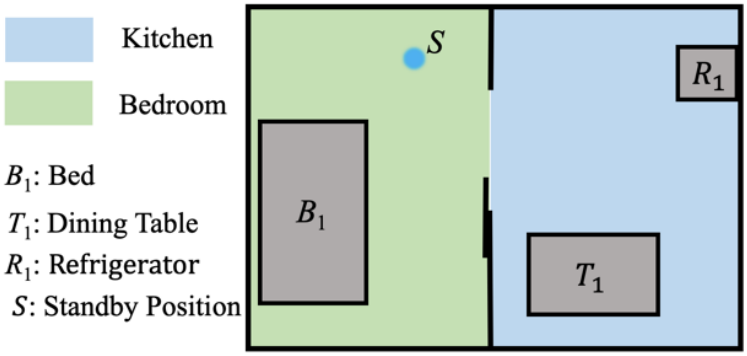


Figure 2-29: Experimental domain.

Figure 2-30: Photographs of the experimental care domain.

Results

In order to validate different aspects of our proposed reasoning system, we conducted three trials, each based on the patient feeling hungry but with different object configurations.

1) **Trial 1**.

Figure 2-31: Personal care scenario.

In the first scenario, the food and drinks were placed in commonsense locations. Specifically, a loaf of bread and a packet of biscuits were placed on the kitchen table, while bottles of milk, juice, and cola were stored in the refrigerator.

The robot was activated by the "hunger" desire at time t_1. At that time, the robot initialized its knowledge base with the static objects, namely the bed (Bed1), dining table (Table1), and refrigerator (Refrigerator1), along with dummy objects for the biscuits (Biscuit1), bread (Bread1), milk (Milk1), juice (Juice1) and cola (Cola1).

Based on this information, the DDR algorithm ordered the robot to serve Biscuit1 to the patient so KUT-PCR turned and moved to the dining table, reaching it at t_2. When the robot perceived the food on the table, the knowledge base was updated. Since the presence of these objects agreed with the initial commonsense knowledge, only the positions of Biscuit1 and Bread1 were updated. After that, the robot fetched Biscuit1 and served it to the patient at t_3, during which time the planning module was paused. However, the moment that the robot handed over Biscuit1, the knowledge base was updated to change the position of Biscuit1 to match that of Bed1. Finally, at time t_4, KUT-PCR returned to the standby point. The route is shown in Figure 2-32, while Figure 2-33 shows photographs taken at times t_1–t_4. Here, the reasoning system worked as anticipated throughout, without any unexpected situations.

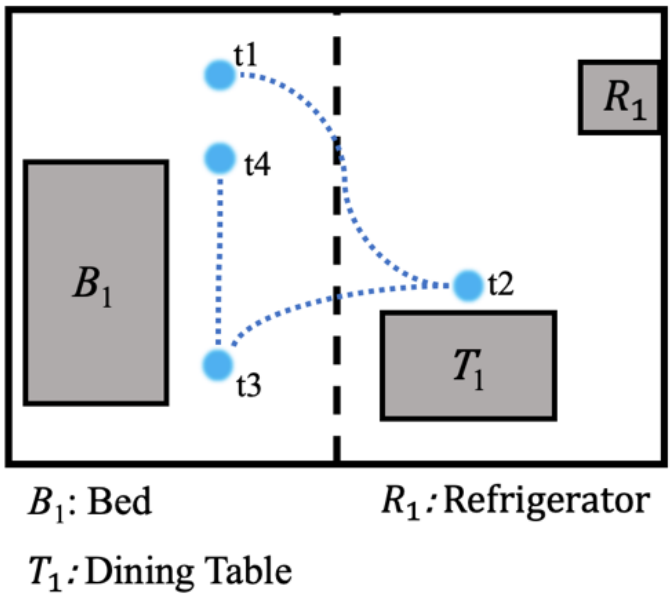


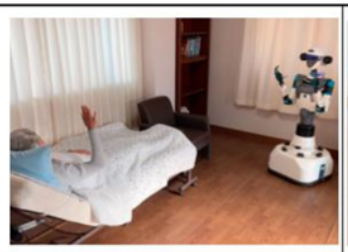
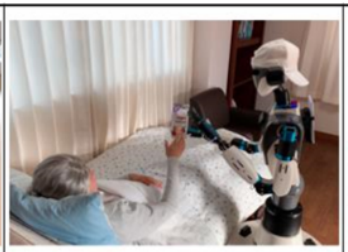
Figure 2-32: Route during Trial 1.

2) **Trial 2**.

In the second trial, no food was placed in the environment, although the drinks were stored in the refrigerator as usual. KUT-PCR's initialization and reasoning processes were as in Trial 1, and it again attempted to serve Biscuit1 to the patient. However, when the robot arrived at the dining table at t_2, it did not perceive any objects on the table and thus deleted the dummy objects Biscuit1 and Bread1 that were initially located there. This triggered the reasoning process again, and this time, the robot was instructed to serve the bottle of milk (Milk1) from the refrigerator. The robot then navigated to the refrigerator, picked up Milk1 at t_3, and was able to successfully deliver the milk to the patient at t_4 before returning to its standby position at t_5. Figure 2-34 and Figure 2-35 show the route and photographs taken at times t_1–t_5, respectively.

3) **Trial 3**.

In the third trial, both the food and drinks were placed as in Trial 1, but a new desk T2 was also placed in the bedroom, with another packet of biscuits on it. As in Trials 1 and 2, the robot initially began navigating toward the dining

Photograph				
Action	Startup with a request.	Fetch the biscuits.	Serve the biscuits.	Return to the standby position.
KB	***Bed*$_1$, *Table*$_1$, *Refrigerator*$_1$, *Bread*$_1$,*Biscuit*$_1$, *Milk*$_1$,*Cola*$_1$, *Juice*$_1$**	***Biscuit*$_1$**	***Biscuit*$_1$**	
Target Object	***Biscuit*$_1$**	***Biscuit*$_1$**	***Biscuit*$_1$**	

Green: added objects. **Red**: deleted objects. **Blue**: refreshed objects.

Figure 2-33: Photographs taken at times t_1–t_4 during Trial 1.

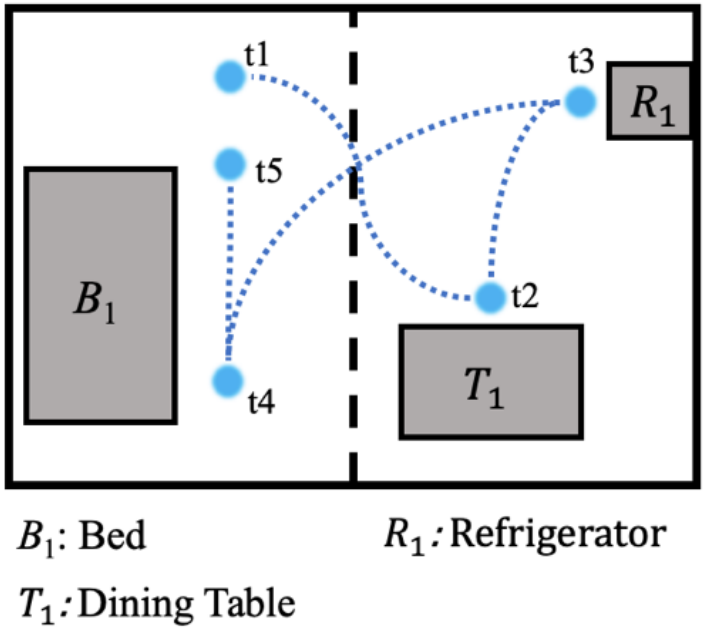
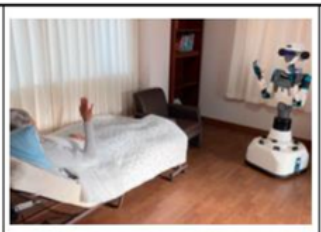
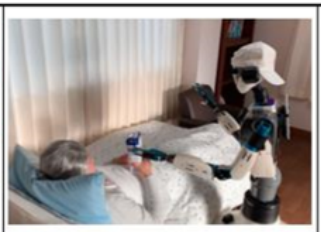


Figure 2-34: Route during Trial 2.

table. However, on the way, it perceived the biscuits on the new table at t_2 and updated its knowledge base with a new biscuit instance Biscuit2 located at T2. This triggered the reasoning process, leading the robot to select Biscuit2 as the target object to serve due to it being spatially closer. The robot then picked up Biscuit2 and served it to the patient at t_3 before returning to its standby position at t_4. Figure 2-36 and Figure 2-37 show the route and photographs taken at times t1–t4, respectively.

In summary, the first trial evaluated the DDR system when nothing unexpected occurred, while the second challenged it to deal with false instance knowledge, namely that an object was not at its expected location. Finally, the third trial tested whether the system could update itself to take advantage of dynamic knowledge. The proposed method enabled KUT-PCR to successfully complete all three trials.

Photograph					
Action	Startup with a request.	Check the table.	Fetch the milk.	Serve the milk.	Return to the standby position.
KB	***Bed*$_1$, *Table*$_1$, *Refrigerator*$_1$, *Bread*$_1$,*Biscuit*$_1$, *Milk*$_1$,*Cola*$_1$, *Juice*$_1$**	***Bread*$_1$, *Biscuit*$_1$**	***Milk*$_1$, *Cola*$_1$, *Juice*$_1$,**	***Milk*$_1$**	
Target Object	***Biscuit*$_1$**	***Biscuit*$_1$ -> *Milk*$_1$**	***Milk*$_1$**	***Milk*$_1$**	

Green: added objects. **Red**: deleted objects. **Blue**: refreshed objects.

Figure 2-35: Photographs taken at times t_1–t_5 during Trial 2.

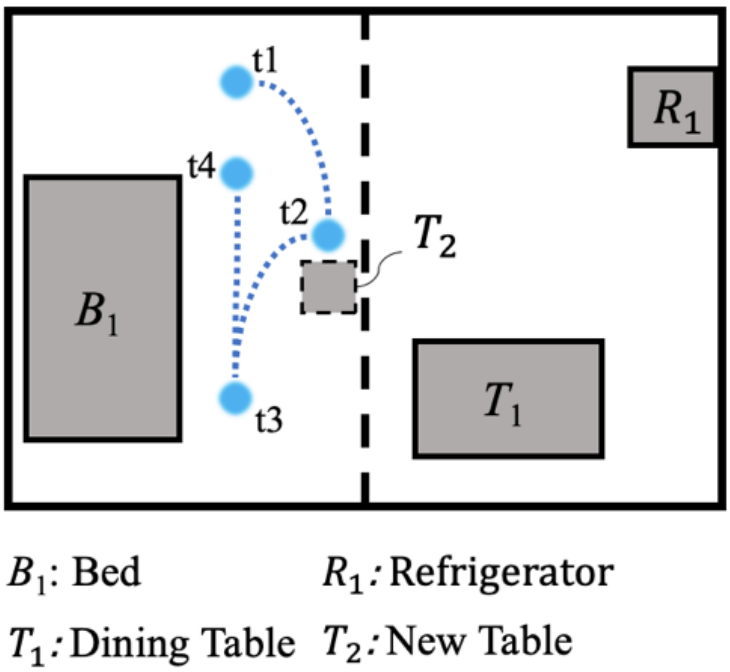


Figure 2-36: Route during Trial 3.

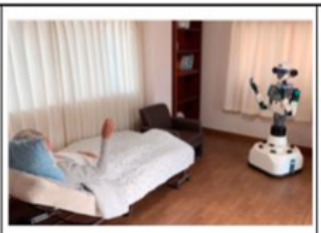
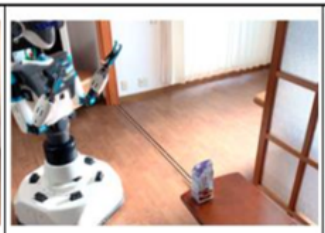
2.5 Conclusion

In this part of the thesis, we present an intelligent personal care system considering perception intelligence, motion intelligent, and reasoning intelligence.

We argue that personal care robots can assist patients similar to human caregivers, only if the robots can reason based on human's physiological desires rather than requiring direct instructions;

Achieving this final goal requires various types of intelligence of a care robot, including perception, motion, and reasoning. Accordingly, we presented a series of approaches and algorithms that allow a robot to "think", "see", and "manipulate".

Experiments with our newly developed KUT-PCR personal care robot in a real household domain were conducted, showing that the proposed method was able to successfully complete a range of trial scenarios. In future work, we plan to address more complex domains and consider a wider range of potential issues. We hope that with our personal care robot, bedridden people can live a more comfortable life, and the load of caregivers could be effectively reduced.

Photograph					
Action	Startup with a request.	Find a new pack of biscuits.	Fetch the biscuits.	Serve the biscuits.	Return to the standby position.
KB	***Bed*$_1$, *Table*$_1$, *Refrigerator*$_1$, *Bread*$_1$,*Biscuit*$_1$, *Milk*$_1$,*Cola*$_1$, *Juice*$_1$**	***Biscuit*$_2$**	***Biscuit*$_2$**	***Biscuit*$_2$**	
Target Object	***Biscuit*$_1$**	***Biscuit*$_1$ -> *Biscuit*$_2$**	***Biscuit*$_2$**	***Biscuit*$_2$**	

Green: added objects. **Red**: deleted objects. **Blue**: refreshed objects.

Figure 2-37: Photographs taken at times t_1–t_4 during Trial 3.

Chapter 3

Automatic Material Transportation in Construction Sites

Continued attempts have been conducted to improve the construction work efficiency with automatic material transportation robots, yet neither of them has been popularized. The reason is that navigating in highly unstructured and dynamic indoor environments remains a great challenge.

Therefore, we present a hallway exploration-inspired guidance approach (HEIGA), which guides an agent to its destination based on a series of directional guidance without the need for self-localization in a global frame.

A series of experiments were conducted, showing that HEIGA-based fully autonomous material transportation tasks can be achieved, without being influenced by the complexity or dynamics of the environment. We expect this new approach to:

1) free construction employees from heavy and repetitive material transportation tasks;

2) improve construction working efficiency.

3.1 Introduction

3.1.1 Motivation

Material transport [46][47][48] is an essential phase in construction processes that accounts for 15%-20% of the total project expenditure [49]. During the interior renovation process of a construction task, about 28 kinds of materials and goods unloaded from trucks need to be transported between working zones on different floors of the building.

Traditionally, transportation work requires a considerable amount of effort. For instance (Figure 3-1-1, a cart full of gypsum boards weighs around 1000 kg and typically requires the effort of three employees. Figure 3-1-2 shows how two workers push a cart through a narrow space. Nowadays, most construction sites are currently facing or are expected to experience the shortage of young workers in the future as the society is aging worldwide [50] [51] [52].

Figure 3-1: Human manipulation of cart.

Therefore, if construction robots [53][54] can be adopted in some transportation tasks in construction sites, employees will be freed up for more valuable tasks. As a result, working efficiency is estimated to increase by 20% (data from our cooperated construction company Maeda Corporation).

However, despite that autonomous robots have been applied successfully in various types of fields, including factories, warehouses [55], logistics centers [56], and hotels [57], enabling autonomous material handling in construction sites faces great

challenges.

Figure 3-2 presents how material handling tasks are conducted in construction sites. The building for demonstration has four floors connected by an elevator. In the assumed scenarios, autonomous material transportation begins as daily work finishes. Typically, a transportation task consists of two basic steps: (i) as the preparation step, all the building materials requiring transportation will be unloaded from trucks into carts, which will be located in D1-1; (ii) material transportation robots relocate carts from loading zones to the expected unloading zones (D2-1, D3-1, D4-1);

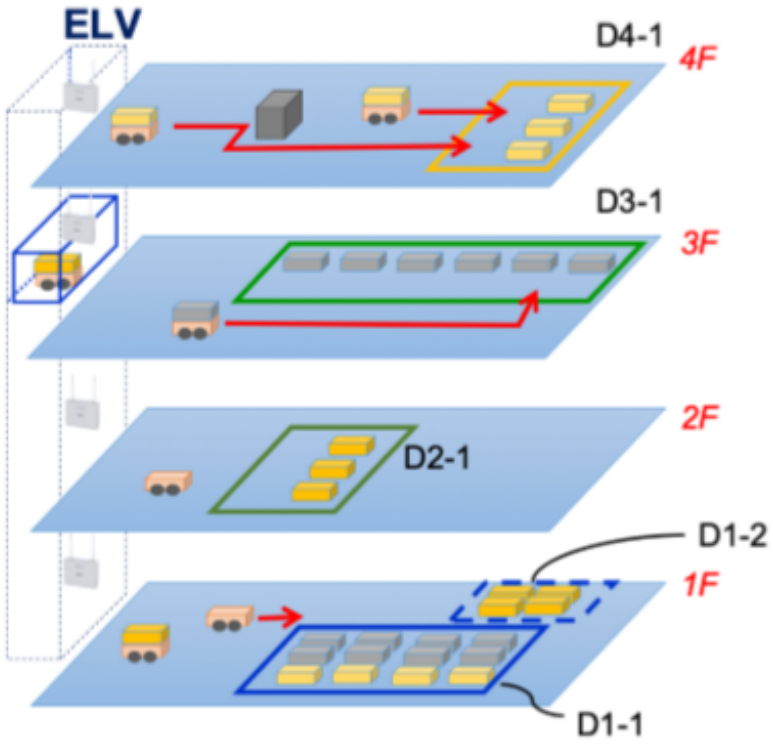


Figure 3-2: Demonstration of a material handling task.

It is worth noticing that the preparation step is not specially required for automatic transportation. Unloading material from trucks to carts is a typical process in construction sites. The carts with wheels will ease the effort of relocating carts both for long-distance transportation and short-distance adjustments. Although a material transportation task seems clear and simple, automating the task has been a great challenge for decades, considering the following reasons.

1) **Undergoing interior renovation.** As Figure 3-3 demonstrates, the construction site is under interior renovation. The floor can be covered by cables or covers, walls are being printed, and power supply is limited to construction tools and elevators. Consequently, it is unreliable and, in some cases, impossible to apply fixed makers on the ground or use a series of routers to provide reference information.

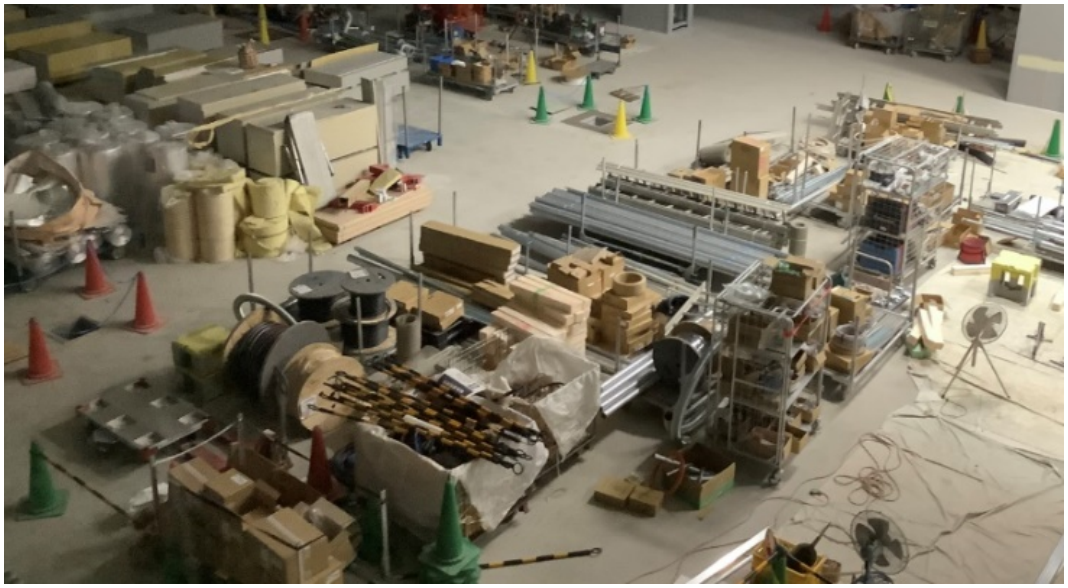


Figure 3-3: Undergoing interior renovation.

2) **Complex and dynamic environments.** One of the characteristics that a construction site is different from other fields (e.g., hotels, logistics centers, and offices) is that the sites change continuously. Available paths are not determined by the walls, but by various kinds of materials, carts, machines, which will be relocated frequently during the construction process. Although most of the novel localization techniques are robust within certain limits confronting dynamic environments, in most cases, changes as shown in Figure 3-4 and Figure 3-5 cannot be handled with expected performance.

3) **Dim and nonuniform illumination.** As Figure 3-6 shows, the illumination condition inside a construction side is typically dim, nonuniform, and even worse

Figure 3-4: Dynamic construction environment.

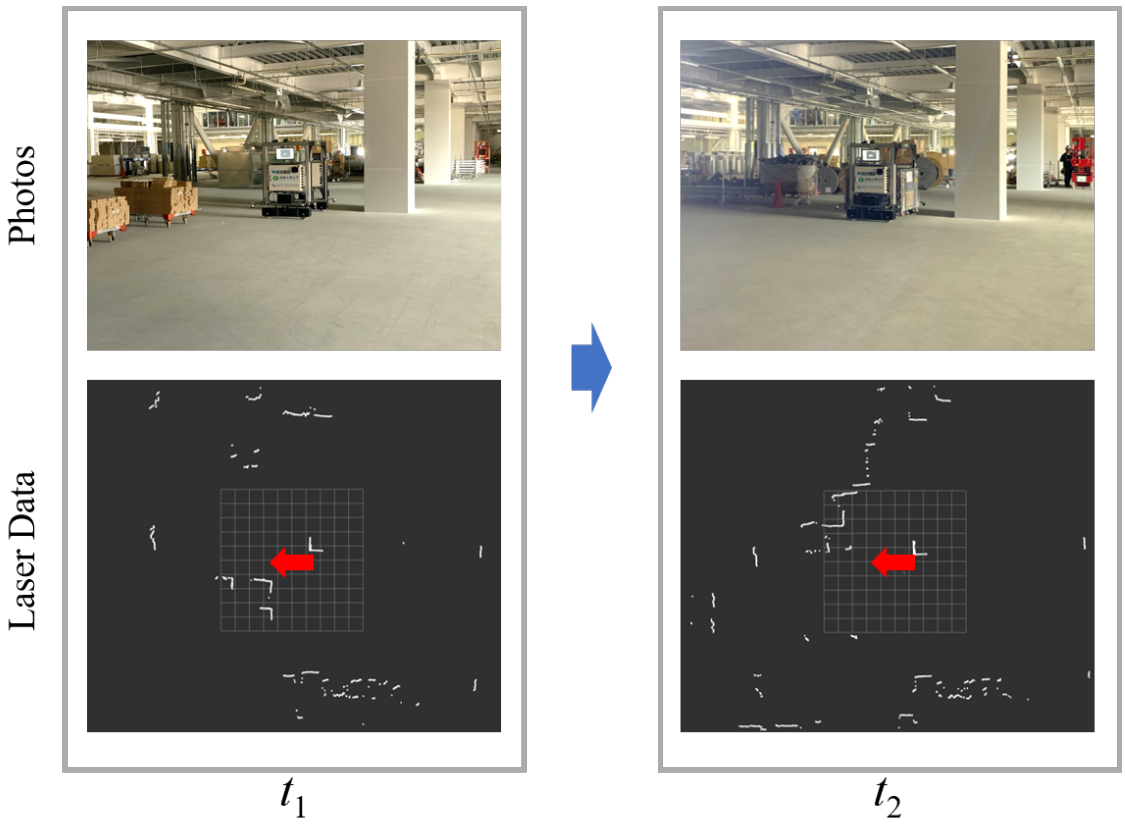


Figure 3-5: Dynamic environment concerning laser data.

during nighttime. The stability and performance of machine vision approaches are under great challenges.

Figure 3-6: Dim and nonuniform illustration.

Confronting these challenges, construction robots were initially designed in the beginning 1970s in Japan. And big Japanese construction companies have been investigating and developing robotized construction processes since the beginning of the 1980s, including transportation robots [58]. In that stage, traditional methods such as line-tracking [59] were used. However, construction environments need to be much more structured and controlled before construction robots can really start to take over[60]. Additionally, various methods (e.g., RFID tag [61], bar-code matrix [62], magnetic tapes/nails [63][64], GPS [65], Bluetooth beacons [66] and motion capture cameras [67]) were considered not suitable for dealing with the indoor construction sites.

In recent years, with the rapid development of sensor technology (e.g., laser rangefinders) and positioning/mapping algorithms (e.g., SLAM), another trend has emerged to realize automated handling in construction sites. The major contractors in Japan have launched their own handling robots and conducted experiments in their construction sites [68][69][70][71]. Approaches such as SLAM have been integrated so that the transportation systems are now much more robust handling unstructured and dynamic environments.

However, it is worth noting the navigation approaches mentioned above typically require real-time localization in a global frame, which has two features: (i) goals can

be described with precise coordinates based on a global frame and (ii) the shortest path can be followed.

These features are required in scenarios such as factories and warehouses. However, in many scenarios, there are no such requirements, including construction sites. The transport task in a construction site has the following features:

1) **Sufficient running time**. The nighttime usually spans 10 hours. As a result, there is no strict requirement for transportation speed. In other words, the path that robots follow does not have to be the shortest path of a global frame.

2) **Limited destinations**. The destinations belong to a discrete set whose elements are described as info points rather than precise coordinates in a global frame.

Based on the two assumptions above and the process, by which people find hotel rooms, we argue that the idea of the info point-based guidance approach is a more suitable solution.

3.1.2 Contribution

In this work, we:

1) present a newly developed gate-type robot that combines the advantages of the comely seen types. A gate-type robot can: (i) move carts of material stably and safely like a forklift-type robot [69][71]; (ii) has a compact structure, therefore can pass through narrow spaces as a unit load AGV [70]; (iii) transport heavy objects easily, as the weight of the cart is mainly carried by its own wheels similar to a trailer-type robot [68].

2) propose a hallway exploration-inspired guidance approach (HEIGA) capable of guiding an agent to its goal without the need for self-localization in a global frame. Therefore, the performance of the system can be decoupled from the complexity or the dynamics of the environment;

3) apply the proposed approach to solve the problem of automated transport tasks in construction sites and conducted experiments in three sites with our novel robot, Capper. The performance of the indoor guidance approach HEIGA is then evaluated.

3.2 Related Work

3.2.1 Navigation Approaches

Navigation is a fundamental problem in robotics. The traditional way of solving a navigation problem requires self-localization to plan and follow a path to the goal. Under this navigation mechanism, various techniques have been applied to enable self-localization. Electromagnetic guidance system is most popular and famous as an automated guidance method, which has been adopted by almost 90% of AGV [72]. [73] proposed a GPS-based path following control method for a car-like wheeled mobile robot considering skidding and slipping. An accurate method for localization of a mobile robot using Bluetooth beacons were introduced in [74], which is considered reliable and scalable. [59] discussed the line-tracking problem with AGVs and presented a robust line detection algorithm implemented on an embedded vision system. Laser reflectors as indistinguishable landmarks can also be used to address the global localization problem in a known environment [75]. Also, Amazon has introduced Kiva robots into their shipping facilities, which is a successful application of localization based on a series of markers [76]. And recently, simultaneous localization and mapping (SLAM) [77] has attracted significant attention as a method that requires no additional sensors or configurations in the environment. Additionally, improved algorithms (e.g., active SLAM [78] and hierarchical SLAM [79]) have allowed SLAM based navigation to be more robust confronting dynamic environments.

Meanwhile, efforts have been made to relax the requirement of continuous localization. More specifically, the idea of how people give and follow instructions (e.g., go forward, turn right) to reach the desired locations has inspired many applications.

[80] has the idea of providing limited instructions to guide people toward a goal. The main contribution of their work is to provide a mobile device-based solution generating a series of instructions. However, mapping and localization are required at the back end during the whole period. With the rapid development of machine vision, [81] presented a vision-language navigation approach with enforced cross-modal matching and self-supervised limitation learning. The instructions are activity-based (turn right and head toward the kitchen). However, the application requires highly accurate recognition of the environment. Thus, it is difficult to deploy it in the environment with complex illumination conditions and textureless background. Moreover, [82] presented a novel design of an augmented reality (AR) interface with activity-based instructions. They described how indoor navigation could be completed with sparse localization. Besides, they optimized the actual usage of the map to a graph rather than a full-sized map. However, since the application is HCI-based, the instructions can be confusing, and ambiguous to machines or robots. Furthermore, a full-sized map is the center of the application, although a graph is used to save, and calculate paths rather than an actual map. Still, the instruction system alone is not treated as an independent approach capable of leading the agent to its destination.

3.2.2 Autonomous Material Transportation Robots

In recent years, various types of material transportation robots have been proposed (Figure 3-7).

In this work, we presented a gate-type robot that combines the advantages of other types of robots, a gate-type robot can: (i) move carts of material stably and safely like a forklift-type robot; (ii) has a compact structure, therefore can pass through narrow spaces similar to a unit load robot ; (iii) transport heavy objects easily, because like a trailer-type robot, the weight of the cart is mainly carried by its own wheels.

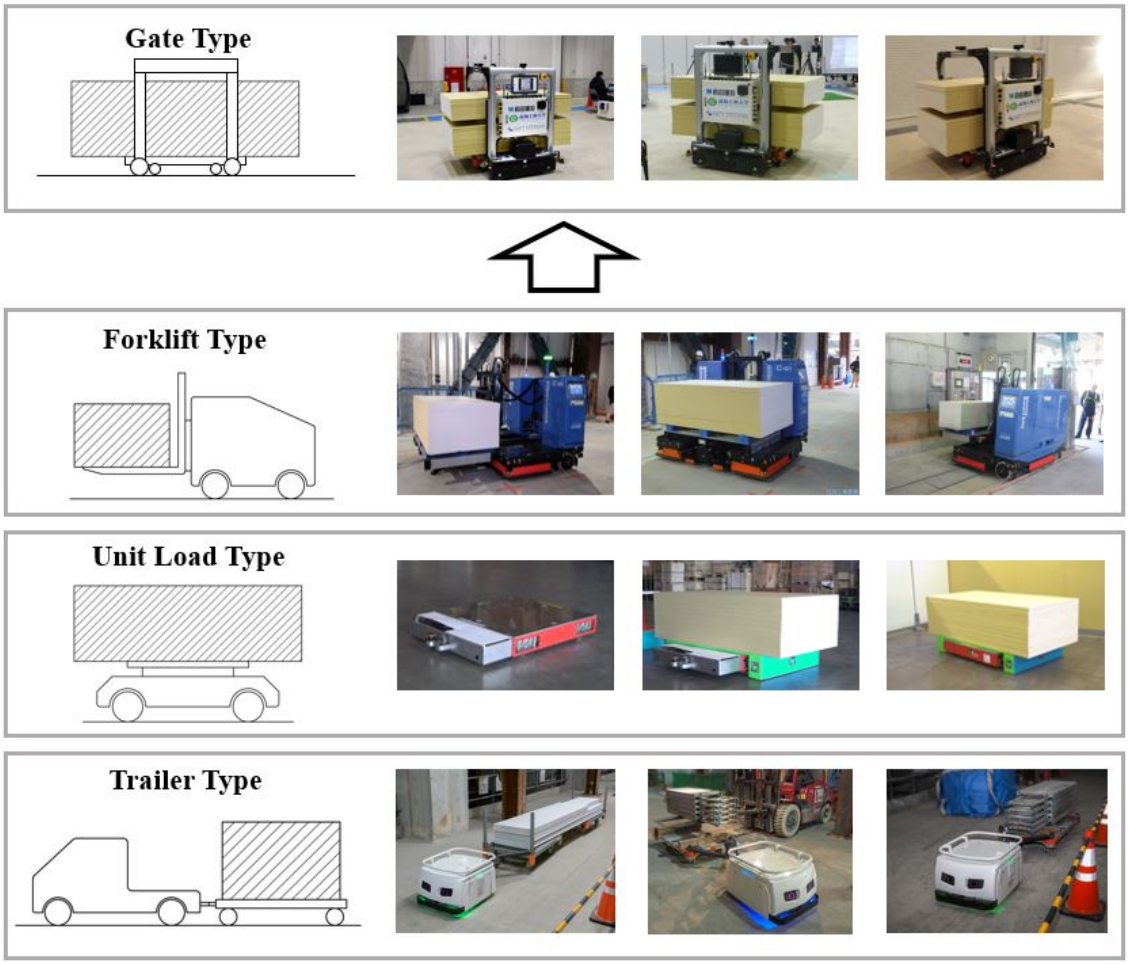


Figure 3-7: Different types of robots for material handling.

3.3 Gate-Type Material Transportation Robot

Given the challenges of transporting heavy carts in complex and unstructured construction sites, we developed a novel autonomous material transportation robot named Capper.

3.3.1 Mechanical Design

Capper is a gate-type robot (Figure 3-8), whose center of gravity is close to that of a cart. The omnidirectional base comprises four mecanum wheels which allow the robot to navigate through narrow and complex environments freely. The gate-type structure and two connectors equipped guarantee solid cart loading and unloading operations. This kind of design allows the light-weighted robot (about 200kg) to manipulate heavy carts up to around 1000 kg [83].

3.3.2 Hardware Configuration

The following hardware setup (Figure 3-9) ensures that Capper can perceive the environment, run algorithms, and communicate with other devices: (i) the main control computer has an Intel i7 processor and an SSD drive, which provide enough computing resources for applications, including machine vision, knowledge reasoning, and motion control; (ii) a wireless router that allows it to be connected with other on-site devices, including the manager PC and ELVs; and (iii) three industrial cameras and three/four laser-range finders that provide a detailed perception of the surroundings.

3.3.3 Mathematical Model

Kinematics

The parameters configuration as shown in Figure 3-10 is as follows:

$\Sigma(x, o, y)$: world coordinate;

$\Sigma(x', o', y')$: robot coordinate;

v: linear speed;

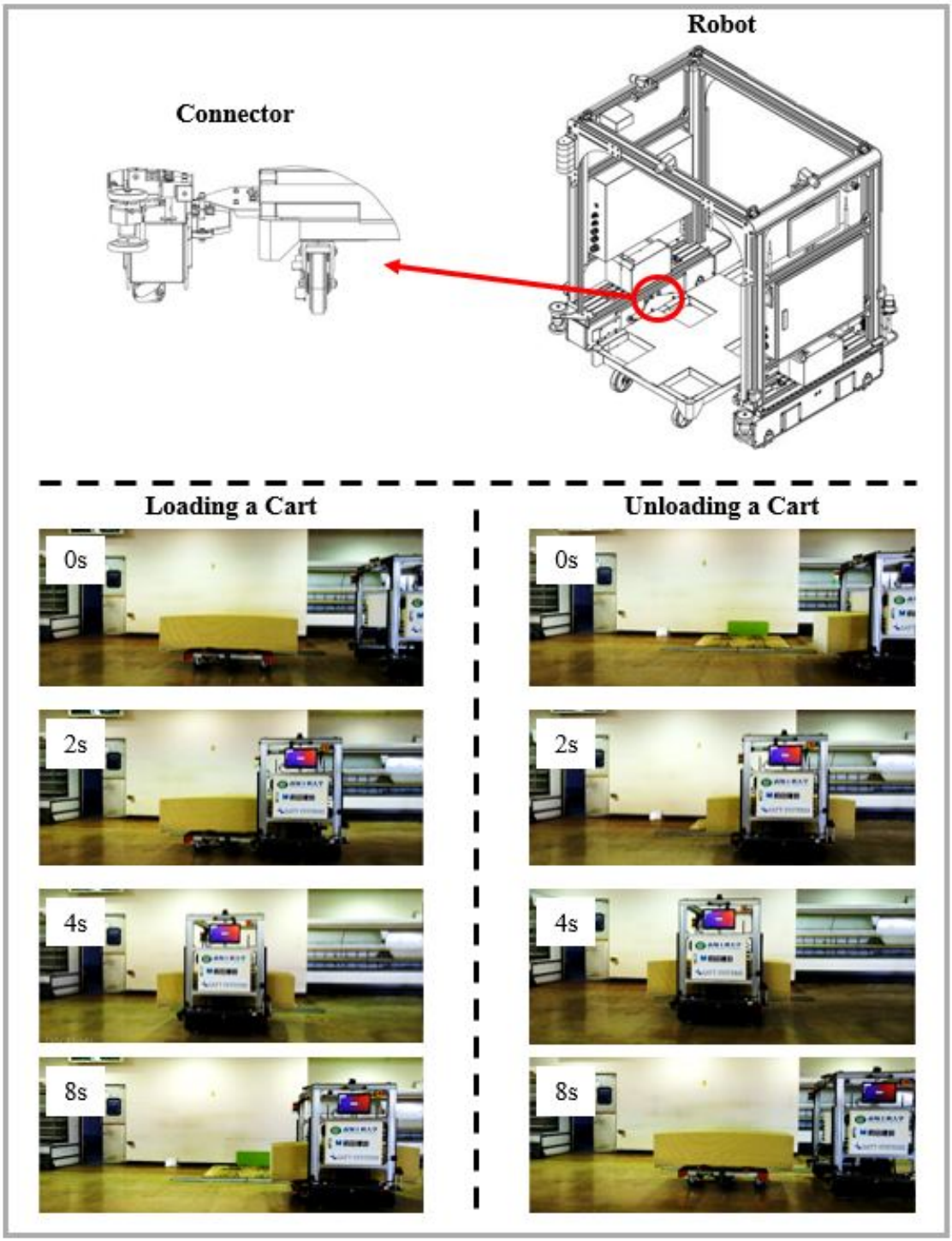


Figure 3-8: Mechanical design of Capper.

Figure 3-9: A gate-type material handling robot.

ω: rotational speed;

o': geometric center of the robot;

v_{im}: speed originated from wheel rotation;

v_{ip}: speed in the direction of the contact point;

v_i: real speed of wheel i;

$2L$: length of the robot;

$2W$: width of the robot;

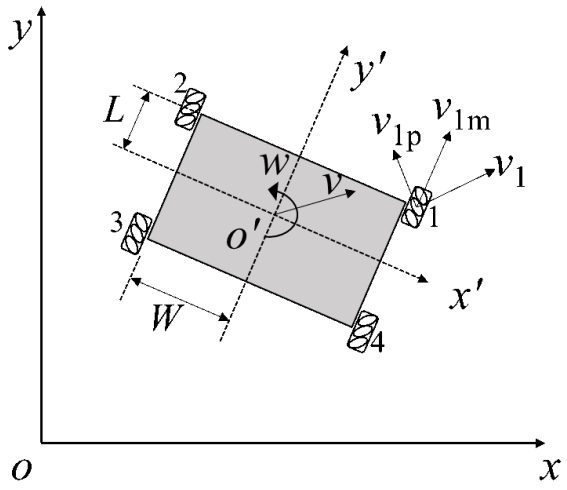


Figure 3-10: Kinematic model of the mobile platform.

Therefore, the inverse kinematics of the robot is given as:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -1 & 1 & (L+W) \\ 1 & 1 & -(L+W) \\ -1 & 1 & -(L+W) \\ 1 & 1 & (L+W) \end{bmatrix} \begin{bmatrix} v_{x'} \\ v_{y'} \\ \omega \end{bmatrix} \tag{3.1}$$

Dynamics

For simplification, the robot is treated as a rigid body. Parameters and coordinate systems are denoted as follows:

$\Sigma(x, o, y)$: world coordinate;

$\Sigma(x', o', y')$: robot coordinate;

v: robot speed;

o': geometric center of the robot;

$F_1 - F_4$: force of each wheel;

$2L$: length of the robot;

$2W$: width of the robot;

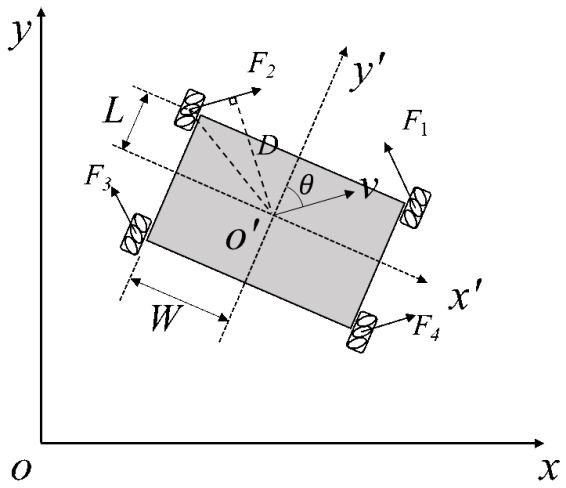


Figure 3-11: Dynamic model of the mobile platform.

M is the inertial matrix of the robot considering the geometric center,

$$M = \begin{bmatrix} M+m & 0 & 0 \\ 0 & M+m & 0 \\ 0 & 0 & I \end{bmatrix} \tag{3.2}$$

where M is the mass of the robot, m is the mass of the cart, and I is the inertia of the robot around the geometric center.

$\ddot{X}$ is the velocity matrix:

$$\ddot{X} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{\theta} \end{bmatrix}^T \tag{3.3}$$

where $\ddot{x}$, $\ddot{y}$ are the linear speed in the direction of x and y, respectively. Ans $\ddot{\theta}$ is the rotational speed.

F and f are the driving force matrix and friction force matrix, respectively:

$$F = \begin{bmatrix} F_1 & F_2 & F_3 & F_4 \end{bmatrix}^T \tag{3.4}$$

$$f = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix}^T \tag{3.5}$$

K^T is the nonlinear coupling relationship between the resultant force at the geometric center and the wheel driving forces:

$$K^T = \begin{bmatrix} -sin(\theta - \frac{\pi}{4}) & cos(\theta - \frac{\pi}{4}) & -sin(\theta - \frac{\pi}{4}) & cos(\theta - \frac{\pi}{4}) \\ cos(\theta - \frac{\pi}{4}) & sin(\theta - \frac{\pi}{4}) & cos(\theta - \frac{\pi}{4}) & sin(\theta - \frac{\pi}{4}) \\ D & -D & -D & D \end{bmatrix} \tag{3.6}$$

The dynamics model of the robot is derived as:

$$M\ddot{X} = K^T(F - f) \tag{3.7}$$

3.4 Material Transportation Task

In practice, a material handling task is conducted by repeating the following operations: (i) go to the loading zone; (ii) load the cart; (iii) go to the unloading zone; (iv) unload the cart. From the viewpoint of the automatic control system, four submodules are required including task planning, path planning, trajectory planning, and servo control. Task planning is currently conducted by on-site operators, by inputting material types, quantity, transportation targets with an UI system. Servo control and trajectory planning realize all the necessary sub-tasks such as loading/unloading of carts, obstacle avoidance, and ELV operations. The path planning in between is the most challenging part.

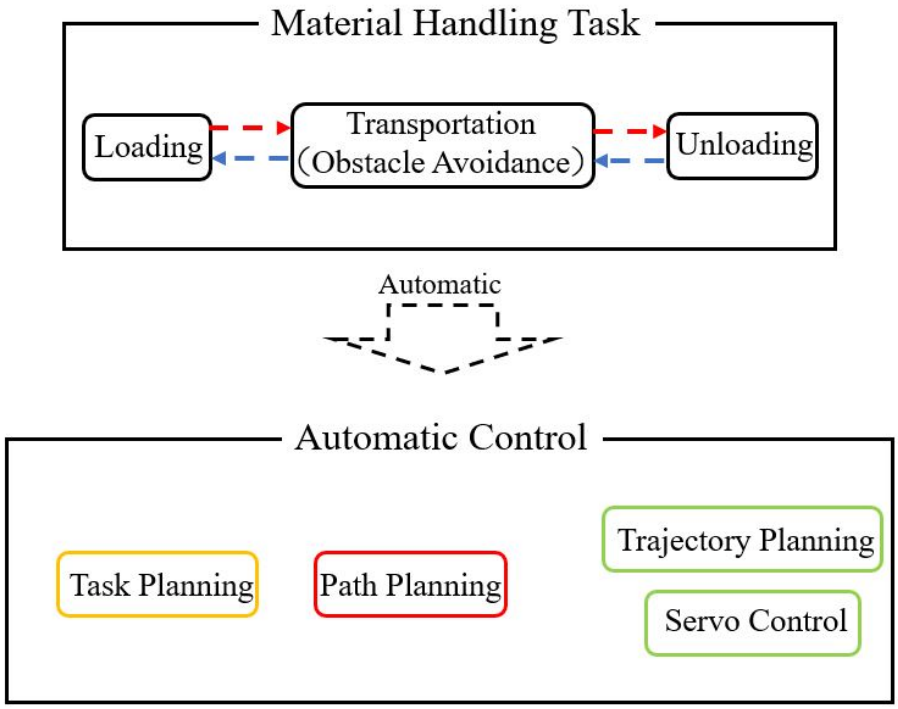


Figure 3-12: A material transportation task: from the viewpoint of robots.

3.5 Path Planning: A Guidance Based Approach

3.5.1 Knowledge Transfer from Employees to Robots

Material handling employees can work confronting dynamic and complex environments, as humans are quite talented in addressing vague and complex visual information. Limited by the level of robotic intelligence and machine vision technologies, attempting to address the problem purely by robotic systems is a long-term challenge. Instead, we argue that transferring the site cognition of employees to material transportation robots will be more realistic and efficient in construction sites.

It is worth noticing that a human employee in construction sites can navigate among loading/unloading zones easily, without the need for a global map or accurate self-localization. Instead, a high-level understanding of the environment and suitable inference accordingly is required. The inference process can be simulated by searching and scoring algorithms, but the current cutting-edge machine vision technologies are still far from what humans are capable of, especially considering complex and dynamic

environments.

From another perspective, anyone can find his room in a hotel even it is the first time that he checks in. The key is that given a room number as the target, we can easily decide where to proceed considering the current room number spotted. As Figure 3-13 shows, if room 140 is the target, given the spotted room 130 and 131, we can decide that going forward is the choice.

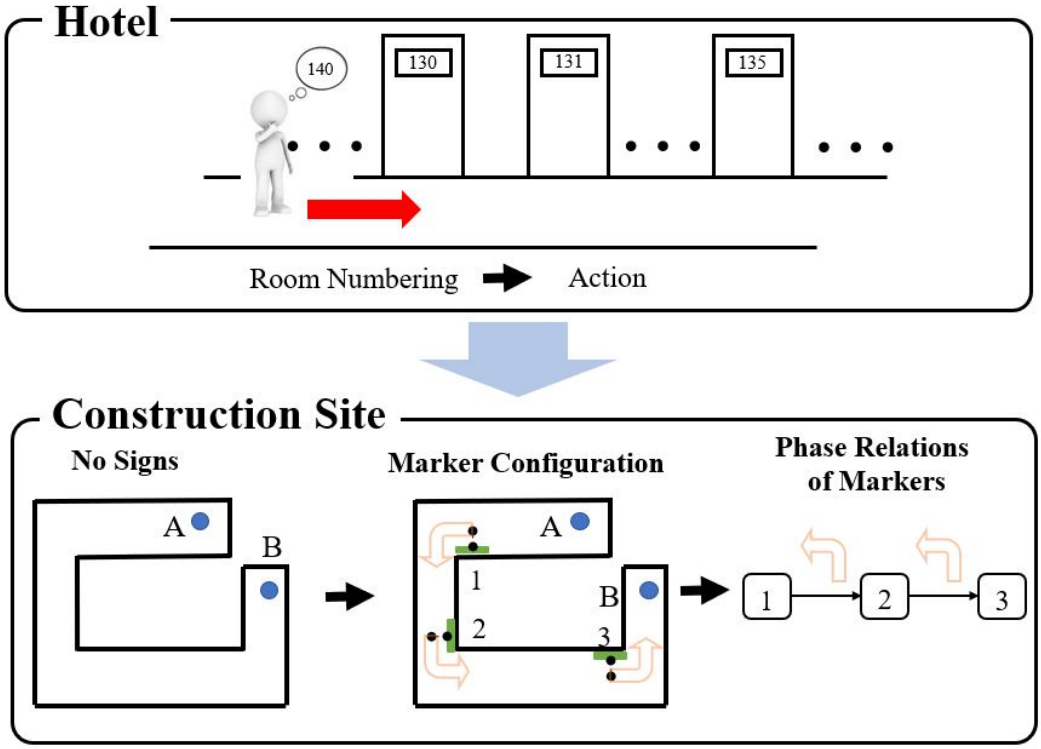


Figure 3-13: Directional guidance in hotels and construction sites.

In summary, the existence of ordered room numbers loses the requirement for high-level perception capabilities greatly. Reaching a desired hotel room can be achieved by: (i) look for a nearby room number; (ii) inference for the directional action concerning the target room number; and (iii) execute the action. For a person to find his room, neither map information nor self-localization is necessary, only two requirements are needed: (i) be able to recognize markers (room numbers); (ii) be able to inference considering the relations among markers.

In construction sites, what's missing between the humans' understanding of the environment and knowledge that a robot can use is a discrete abstraction. As Figure

3-13 indicates, by deploying markers into a construction site, an abstraction of the site can be obtained as a set of markers connected by phase relations.

As Figure 3-14 demonstrates, the system works as follows: (i) the operators on-site are responsible of observing the environment and defining available paths; (ii) also the operators are responsible of configuring the site by deploying necessary AR markers. (iii) an UI system (deployed on the managing PC) is the bridge between operators and robots, site configuration (including markers and the directional information that connects them) and task information are inputted by operators and then sent to the robots; (iv) eventually, the robot, will receive the date sent by UI, formalize them into a robotic knowledge base and a task list. As a result, the material transportation system will actually execute the tasks with respect to the knowledge base and AR markers perceived.

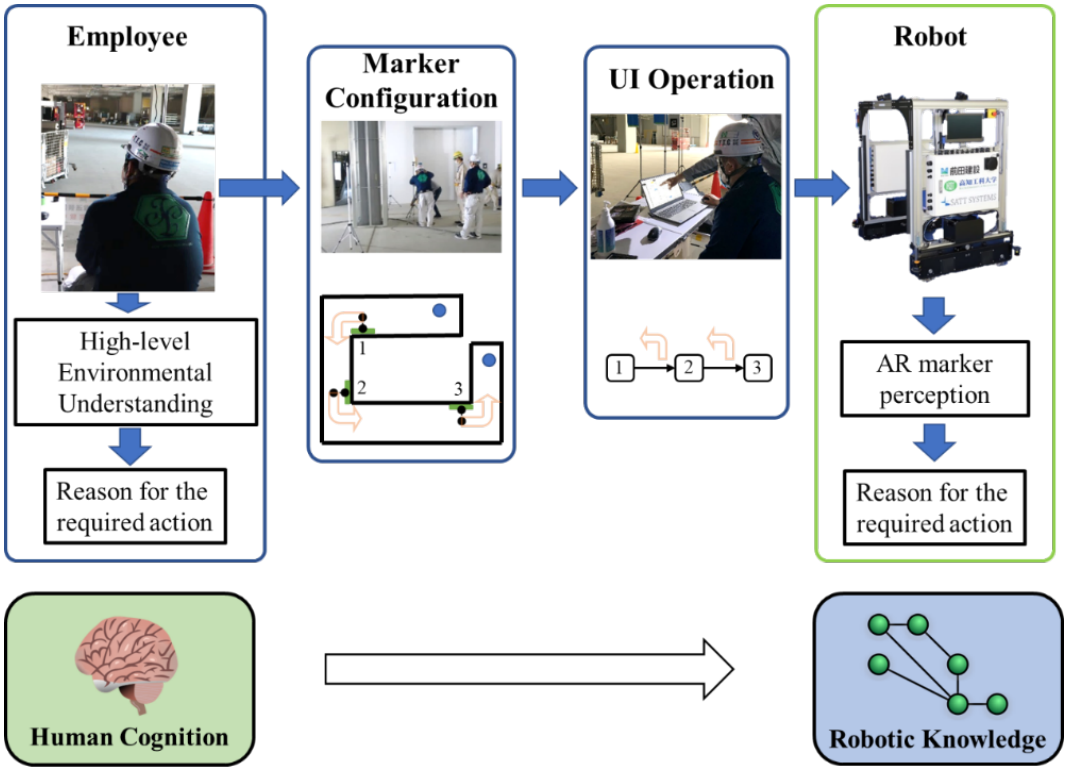


Figure 3-14: Overview of the knowledge transfer system.

In other words, by applying such a system, we will be able to map human cognition of the sites into a format of robotic knowledge, so that robots can conduct material

handling tasks similar to how employees will address the problem, but without the need for human working force.

3.5.2 Knowledge Transfer Method: HEIGA

We developed HEIGA to transfer knowledge from employees to robots, that allows an agent to reach its destination in an unstructured environment. It is inspired by the hallway exploration process searching for the desired hotel rooms.

Guidance Problem

In robotics, the navigational techniques involve locating the agent's position in a global frame. To clearly distinguish a guidance problem from a navigation problem, we now define a guidance problem.

Definition: the guidance problem in an unstructured environment configured with a series of info points is to enable an agent to move from the current location to the goal described with relative geometry relation of one of the info points.

Characteristics: a guidance problem (i) does not require mapping of the environment and (ii) does not require self-localization based on a global frame.

HEIGA

To address a guidance problem, a guidance approach typically requires the following: (i) a knowledge base comprising information about all the info points and their relationship; (ii) given a goal described with respect to an info point, directional guidance should be provided with every info point perceived, thus enabling an agent to reach its goal.

Figure 3-15 illustrates how a HEIGA system typically works. Firstly, the guidance knowledge base "Guidance KB" needs to be initialized and updated by human experts that are familiar with the field, who are also in charge of configuring the info points. Then, the "Guidance Core" will be able to provide a directional instruction considering each perceived info point and the guidance knowledge base. Meanwhile, agents can

either be robots, humans, or even cell phones capable of recognizing info points and querying for instructions.

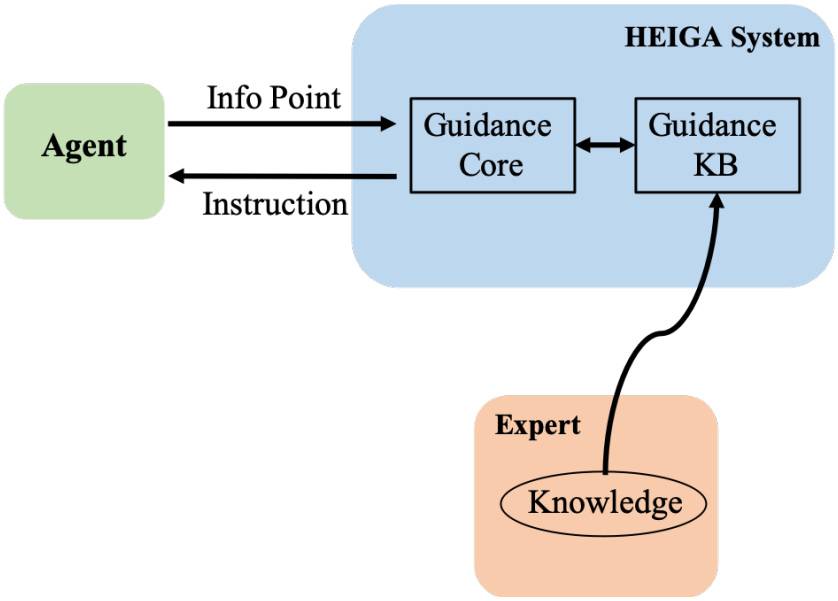


Figure 3-15: Schema of a HEIGA system

Relationship-Centered Guidance Knowledge Base

Guidance knowledge base is a kind of relationship-centered description. In other words, besides info points, how all the points are connected also matters. In the following example, we will demonstrate how a relational description is different from an absolute description.

Figure 3-16 shows two completely different fields A and B from the viewpoint of absolute geometry. However, if an agent needs to move from the start location to the goal location in A and B, both fields can be configured using six info points (from 1 to 6). Eventually, from the start location to the goal location, the sequence of perceived info points with directional instructions in between can be exactly the same.

In summary, info point-based relationship-centered description abstracts a field at a more abstracted level, which results in strong robustness confronting complex and dynamic environments. More specifically, a guidance knowledge base can quickly be established in a complex environment. When the environment changes, the knowledge

base could stay unchanged or be easily adjusted by modifying a few info points and their connections.

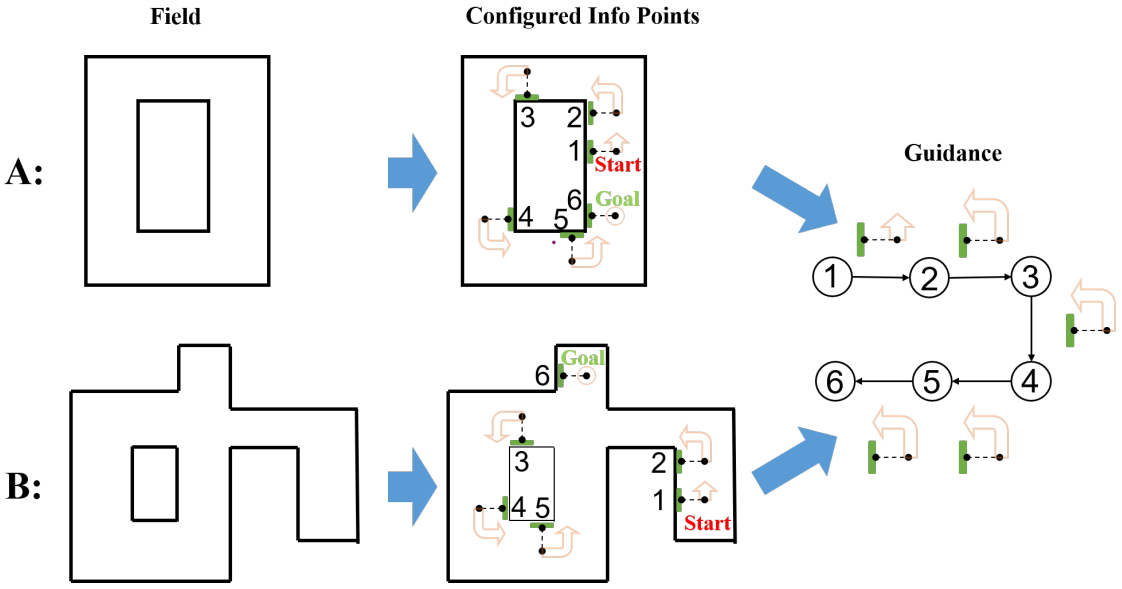


Figure 3-16: Relational description vs. absolute description

Definition of Guidance Graph The info point-based description is formulated as a kind of directed graph as:

$$G = (V, E) \tag{3.8}$$

where V is the set of vertexes, and E is the set of arcs. The vertices refer to the info points deployed in the environment while the arcs encode directional guidance information that allows an agent to be accurately guided by info points.

Besides, both the vertices and arcs were labeled so that various types of useful information can be included in the graph for further usage. Typically, one label set was integrated for vertices, which are denoted as $F = \{f_1, f_2, ...f_n\}$. With a label set F of size n, n types of information can be encoded. For instance, one commonly used property is the floor number indicating where the info point is located.

For arcs, two label sets were considered including the weight set $W = \{x \mid x \in \mathbb{R} \wedge 0 \leq x \leq 1\}$ (describing the cost of moving between two info points) and the directional guidance set $G = \{g_1, g_2, ...g_m\}$ (including m types of directional instructions). As a result, the following mapping is required to completely describe a guidance graph:

$$\alpha : V \to F, \beta_1 : E \to W, \beta_2 : E \to G \tag{3.9}$$

Graph Operations After defining the guidance graph, the following basic operations allow a guidance graph to be manipulated considering various scenarios:

1) Union

   Union allows new graphs to be added to the original graph, which can be used either when the knowledge base is initialized by combining multiple subgraphs or when updating is required considering new routes available.

   Info points on-site are typically deployed considering a number of paths among locations of interest. For instance, we define a set of locations as L, then $G_k^{L_i,L_j}$ $(L_i, L_j \in L)$ is the guidance graph describing one possible path from L_i to L_j, which is a independent line-type graph. Then the complete nest-type guidance graph consisting N paths can be obtained by the union of all the paths available: $\sum_{k=1}^{N} G_k^{L_i,L_j}$

2) Remove

   The removal of a vertex or an arc indicates an unused info point or a blocked path in real environments.

   However, if the situation is temporal, such that the path will only be blocked for an hour, then restoring the graph after removal will take extra effort since the information was previously deleted. Therefore, we suggest blocking the path by increasing the weight on the corresponded arc, then the path can be restored by simply recovering the weight.

Guidance Provision

With a guidance knowledge base established, the remaining procedure is how to provide guidance, given the query including the goal and the current info point in view.

A path can be described as a sequence of vertices and arcs:

$$p = v_0, e_1, v_1, e_2, ..., e_k, v_k \tag{3.10}$$

where v_0 is the start and v_k is the end of the path. Vertices v_{t-1} and v_t are end vertices of e_t $(t = 1, ..., k)$,and k is the length of the path. A zero-length path is a single vertex v_0. In a path, it is only allowed to visit a vertex or go through an arc once.

The shortest path p from the current info point to the goal info point can be effectively obtained with methods such as the A-star algorithm [84] or focussed D-star algorithm [85] (which can be chosen freely considering application requirements):

$$p = F(KB, I_s, I_g, w) \tag{3.11}$$

where KB is the knowledge base described with a guidance graph; I_s and I_g are the current and goal info points, respectively; and w refers to the property that should be considered while evaluating the path, which in default is the weights. The resulting path contains a series of info points and corresponding instructions leading to the goal.

In real applications though, the execution of the instructions does not always align with the expectations. Therefore, although a complete path is formed, only the part related to the current spotted info point will be trusted and provided as guidance, and instructions that rely on the rest of the path will be dropped. In this way, challenges such as failing to follow directional instructions and missing info points can be naturally addressed without the need for additional functions. Guidance is

defined as:

$$\zeta = \{g, P\} \tag{3.12}$$

where $g \in G$ is the direction guidance, P is a set containing labeled information of v_{i_0} and v_{i_1}. The properties of the current and next vertex provide information for possible further applications (e.g., ELV operations considering floor numbers).

The necessary guidance information from a path is extracted by an operator $f_e(.)$, which extracts the necessary data from path p and establishes an expected guidance $\zeta = f_e(p)$. Still there are special conditions, the complete guidance generation algorithm is shown in Table 3.1. Here we assume two special codes: 0 indicates stop and 255 indicates error.

In the algorithm, if $p = NULL$, then no path is obtained. This situation is generated when the source or goal info point is not in the graph or not connected. The code 255 is then returned to remind the agent to perform some actions such as recheck. When the length of the shortest path is 0, the goal info point is the current spotted one. Therefore, the task is finished and 0 is generated to request the agent to stop. Otherwise, executable guidance ζ will be returned.

Table 3.1: Guidance Generation Algorithm

Algorithm 1 Guidance Generation

Input: KB, I_s, I_g, w
Output: ζ
Calculate the path
$p = F(KB, I_s, I_g, w)$
# If no valid path is obtained
if $p = NULL$ **then**
  return $\{255, \}$
  # If the length of the path is 0
else if $len(p) = 0$ **then**
  return $\{0, \}$
  # If the length of the path is larger than 0
else if $len(p) > 0$ **then**
  return $f_e(p)$
end if

3.5.3 Material Transportation with HEIGA

How HEIGA can be used to solve the material transportation problem is described in Figure 3-17. A task list consisting of N task is denoted as $\{T_1, ..., T_N\}$. The marker of the loading zone and unloading zone are denoted as AR_l and AR_u, respectively. The target marker is denoted as AR_g, and S is the current working status. With such a workflow, a robot can change between "loading", "unloading", "reaching for loading/unloading zones" effectively, eventually leading to a list of material handling tasks being completed successfully.

3.6 Trajectory Planning: Actual Material Handling

In this section, we present how various functions required achieved by trajectory planning in the automatic transportation system.

3.6.1 Cart Manipulation

A material handling robot needs to locate and join a cart into one unit in order to perform efficient transportation tasks, as well as autonomously self-detach so that the carts can be relocated to desired locations. In this work, we present a novel two-stage latching system that enables robots to catch carts in complex and uncontrolled construction sites.

A cart is labeled by a set of AR markers. The number of the markers can be adjusted considering different applications. In the first stage (Figure 3-18), marker tracking will be conducted that drives the robot into a position ready for further operation. In the second stage (Figure 3-19), lase-range finders will be used to locate the cart and conduct accurate latching action. The latching task will be considered successful when the cart is connected by two lockers.

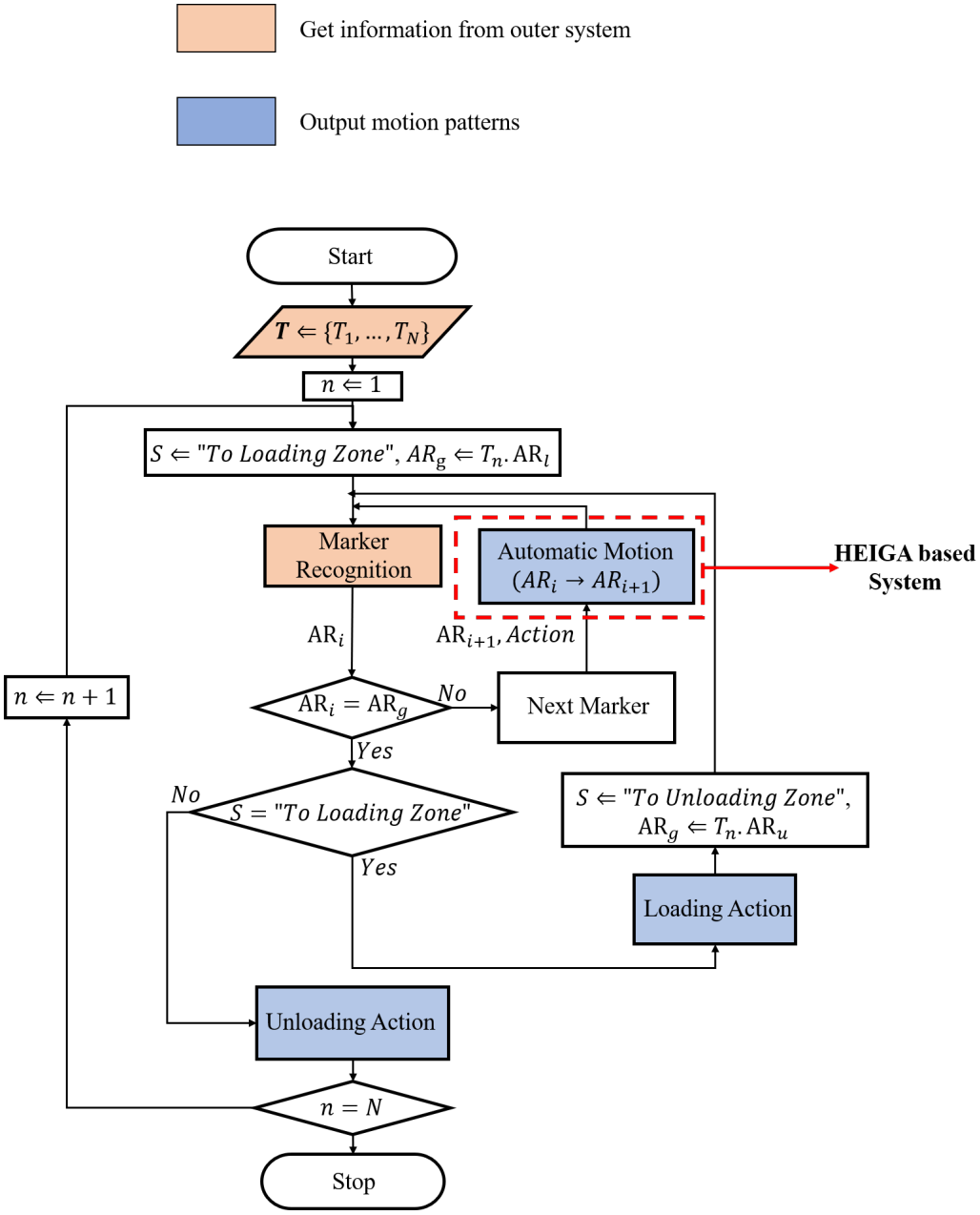


Figure 3-17: Material transportation with HEIGA.

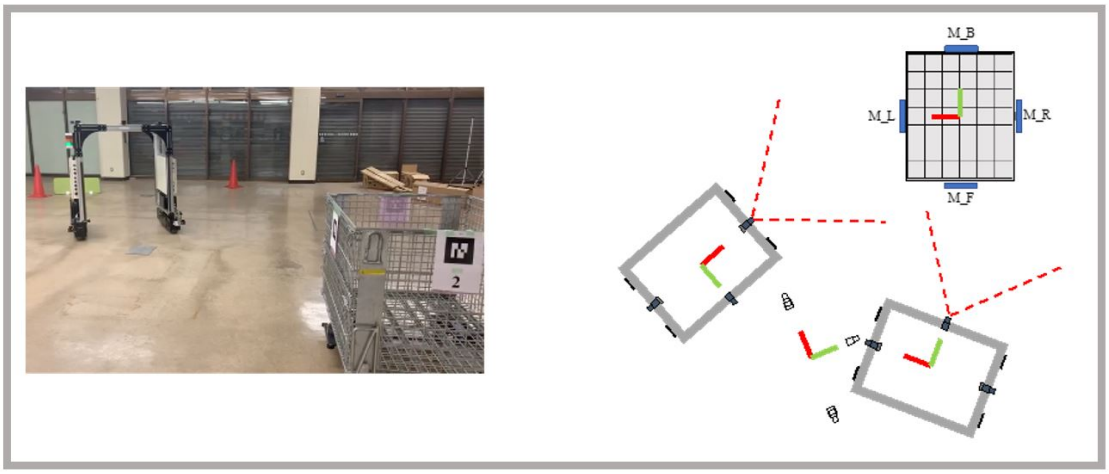


Figure 3-18: Loading action: first stage.

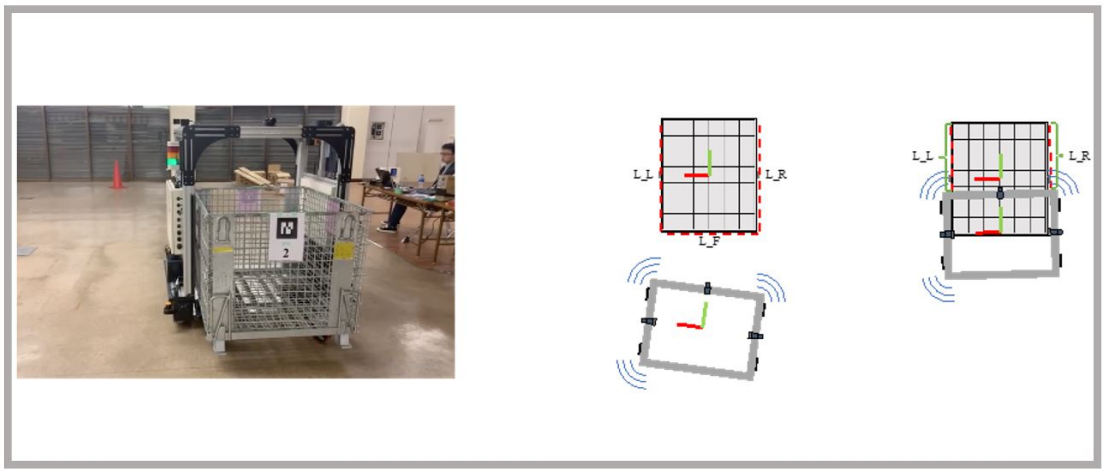


Figure 3-19: Loading action: second stage.

3.6.2 Directional Instructions: Perception and Execution

A HEIGA-based system can provide directional instructions regarding each info point spotted. However, two problems need to be addressed to take advantage of this approach: (i) suitable choice of info points so that directional instructions can be provided accordingly and (ii) ability to execute directional instructions.

For humans, signs or text information can be used as info points, considering our robust vision processing and understanding capabilities. However, info points that can be recognized by a robot are limited. Available methods include deep learning-based or marker-based object detection. Besides, people can easily follow directional

instructions based solely on vision. However, it is difficult for a robot to perform similar actions in a complex and unstructured environment. For a robot to follow a directional instruction based on an info point, a local reference source is required for distance and angle information.

To achieve a system with high stability and low complexity, we chose AR markers to serve as info points in the HEIGA system and as a reference source in the motion execution system.

Figure 3-20 is an example of the info points used on-site. The machine reading area occupies most of the board area so that Capper can recognize it with cameras. Moreover, there are customized areas that provide information for humans, intelligent hardware scans, etc. Additionally, info points can be deployed in the environment based on multiple configuration patterns.

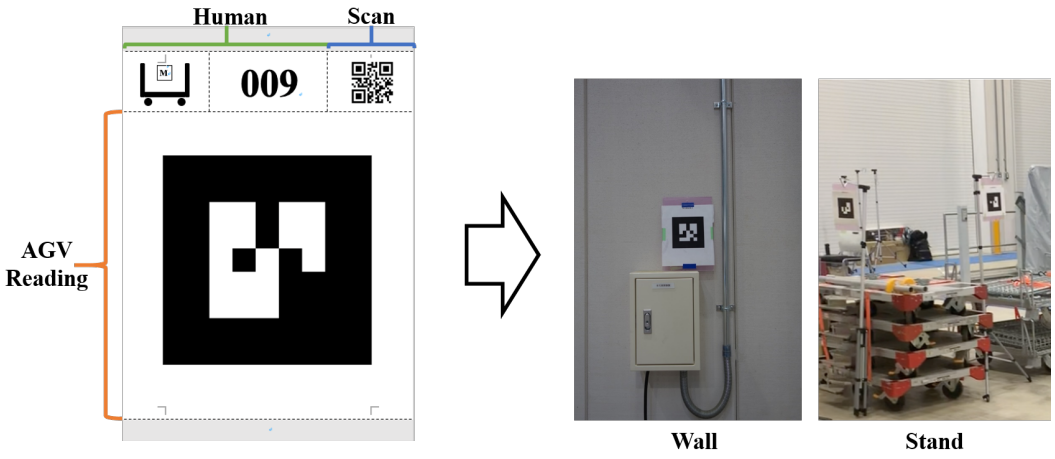


Figure 3-20: AR marker-based info point boards and use cases

Figure 3-21 shows the two roles of AR markers. In Figure 3-21-1, six types of directional instructions numbered 1-6 are presented. Here markers are info points, and only a directional instruction will be provided when the marker is perceived. In Figure 3-21-2, a two-marker set is used to ensure a solid corner turn action when the No.4 directional instruction is provided considering the primary marker. By designing different marker configuration patterns, various motion capabilities can be achieved, including following directional guidance, moving in/out of an ELV, accurate picking/releasing, and autonomous charging.

In more detail, in Figure 3-22, we use practical examples to illustrate how the guidance patterns are achieved using AR markers. Capper is capable of a wide-range perception (three industrial cameras are used, including the left camera, front camera, and right camera). Regardless of the starting position, as long as an AR marker is detected, the triggered navigation task will guide the robot to the specified direction. While being guided, the robot tracks its position relative to the maker through visual recognition (markers are within view range) and speed accumulation (markers are out of view range). The positioning is neither global nor highly precise, but sufficient in completing the given guidance task. Three examples are given for guidance pattern No.1 and No.2. It is worth noting that to improve the stability of guidance, pattern No. 2 uses a set of two AR markers. The rest patterns (No.3, No.4, No.5, No.6) can be configured similarly.

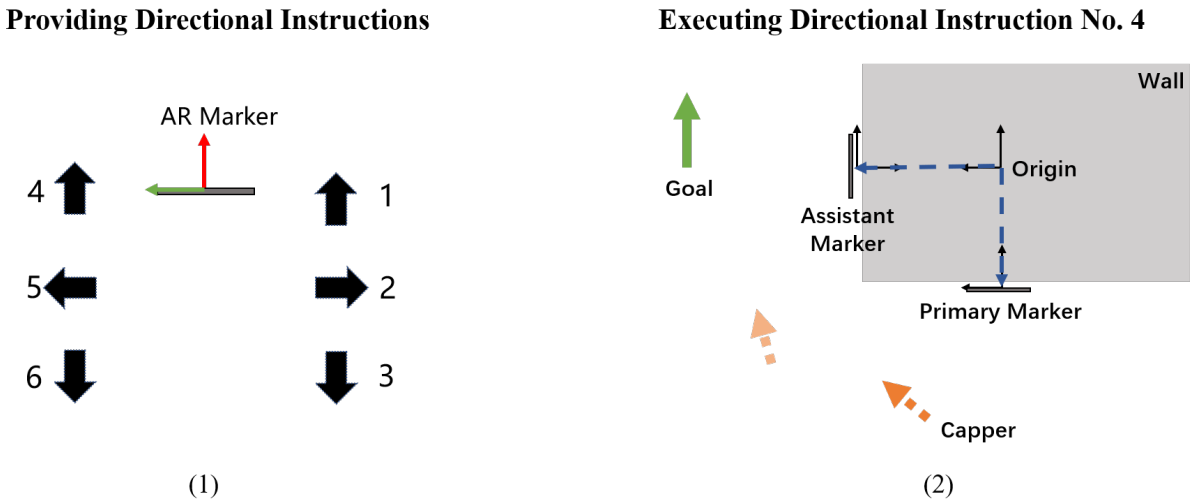


Figure 3-21: Two roles of AR markers

Typically, the number of info points is smaller than the number of AR markers deployed. In other words, some of the AR markers have the functions of info points, while some of them only function as reference resources. For instance, as Figure 3-23 shows, markers 200, 210, 220, 101, and 102 perform roles of both info points and reference resources, while markers 201, 202, 211, 212, and 221 only function as reference resources inside picking/releasing zones 200, 210, and 220, respectively.

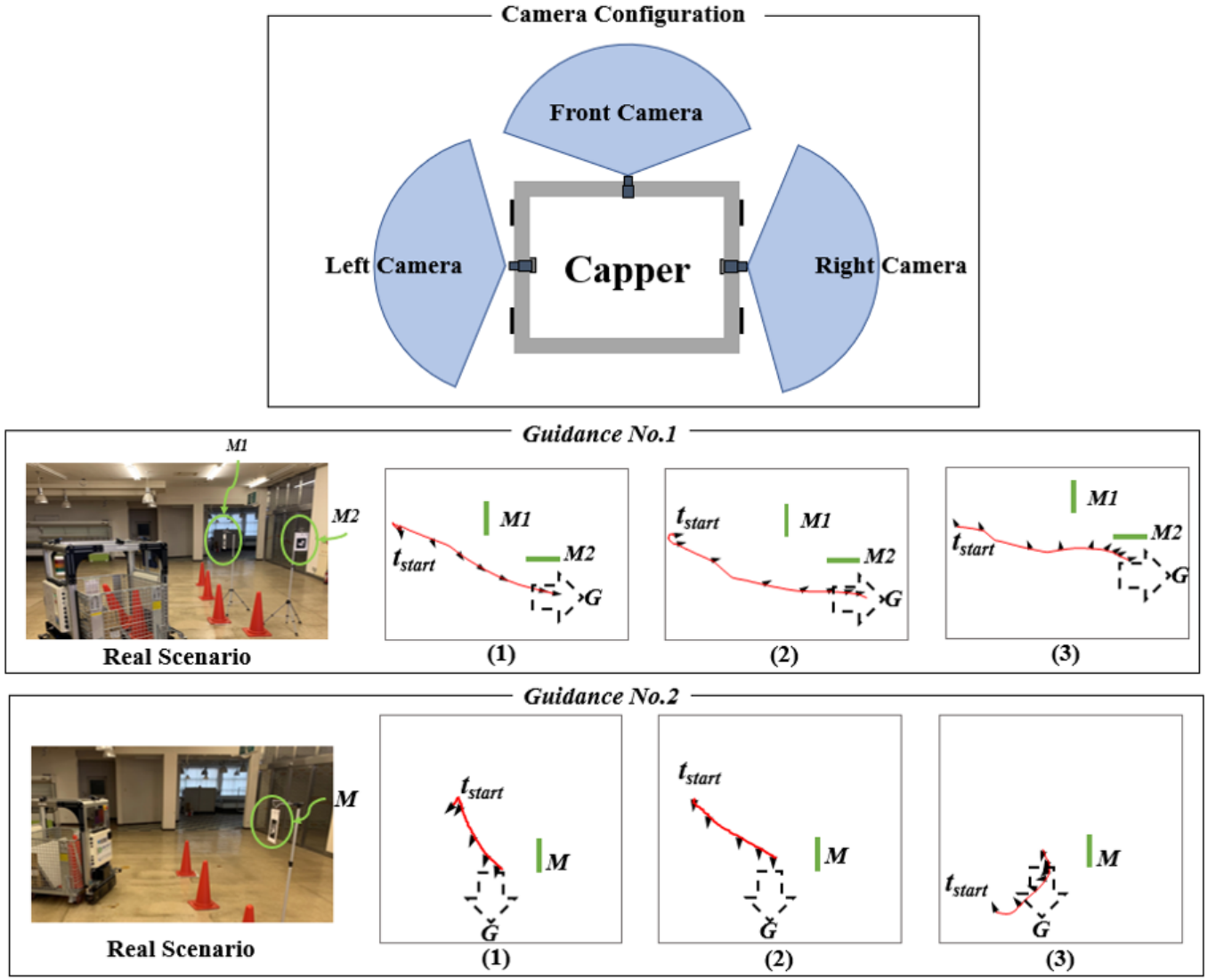


Figure 3-22: Local marker tracking examples

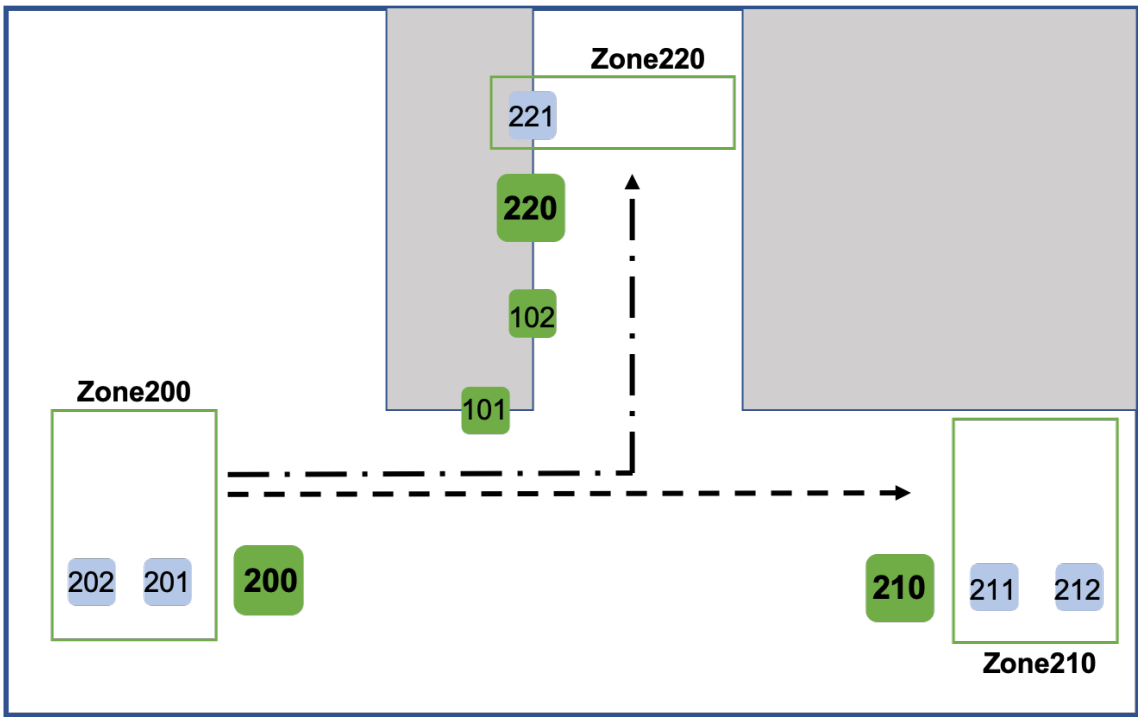


Figure 3-23: Usage of AR markers

3.6.3 Construction Site Configuration with Info Points

In the previous section, we proposed a graph-based description of a HEIGA system. In this section, the approach is further configured to address the issues peculiar to construction sites. The construction knowledge graph is denoted as $G_{cons} = (V_{cons}, E_{cons})$. For each info point as well as the floor information that is useful in multi-floor construction sites, special functions need to be marked for further usage. Therefore, $F = \{f_1, f_2\}$. If an info point is configured in 2F and also shows where the charge station is, then $F = \{\text{“}2F\text{”}, \text{“}ChargeStation\text{”}\}$.

For arcs, two label sets are considered including the weight set $W = \{x \mid x \in \mathbb{R} \wedge 0 \leq x \leq 1\}$and the directional guidance set $G = \{1, 2, 3, 4, 5, 6\}$ as introduced in Figure 10-1.

Hence, the following mappings are required to label each vertex and arc with the necessary information:

$$\alpha : V_{cons} \rightarrow F, \beta_1 : E_{cons} \rightarrow W, \beta_2 : E_{cons} \rightarrow G \tag{3.13}$$

With AR markers deployed in the construction environment and the HEIGA system adequately configured, Capper can reach any location in a construction site as long as there is an info point set.

In summary, the automatic control system is presented in Figure 3-24. The knowledge base is a directed graph, where each node represents a marker and each vertex describes the guidance information between markers. Given a valid marker detection, the path planning will inference considering the task and knowledge base. It's worth noticing that, the output of the path planning module is not an actual path including a series of points with respect to a global frame. Instead, a target that guides the robot to the right/left/front/backward will be assigned to the trajectory planning module. Eventually, robotic motion is conducted by the trajectory planning module.

3.7 Experiments

The automatic material transportation robots are expected to be deployed into real construction sites. However, deploying robots under continues development into construction sites is both dangerous and surely will have impact on the original construction schedules. Therefore, newly designed robots and algorithms need to be well evaluated before being deployed into real construction sites.

In this projects, we have conducted a three-stage evaluation including: (i) development and evaluation of robotic systems including hardware and algorithms in an abandoned supermarket named ACOOP; (ii) evaluation of long-distance transportation and vertical transportation (with an elevator) tasks in a construction lab named ICI; (iii) integrated assessment of material handling in multiple real construction sites.

3.7.1 Abandoned Supermarket: ACOOP

An abandoned supermarket named "ACOOP" near college campus was used for initial evaluation of all the hardware and algorithms. Figure 3-25 shows the basic information of ACOOP. The empty supermarket provided sufficient space for plane move-

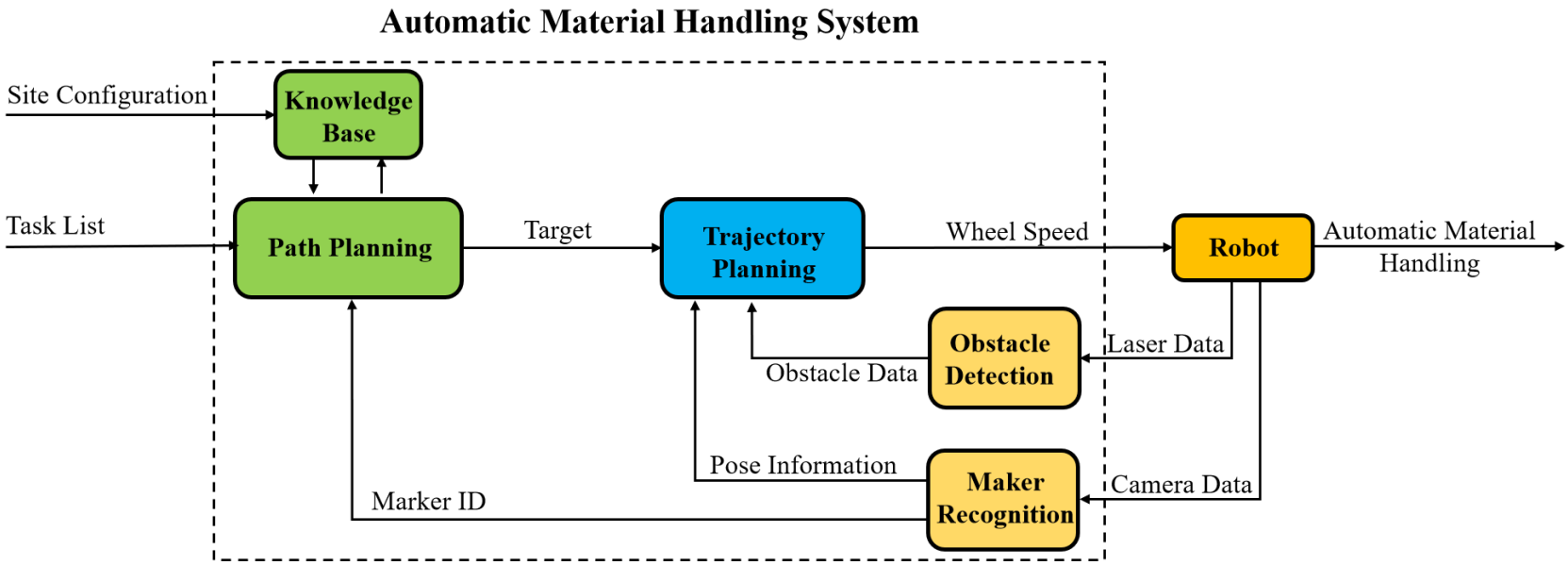


Figure 3-24: Automatic material handling flowchart.

ment. Additionally, carts, traffic cones, and gypsum panels were used to simulate different site configurations. In this fully controlled environment, unlimited types of simulated scenarios can be configured to test the system from different angles, without disrupting any other activities.

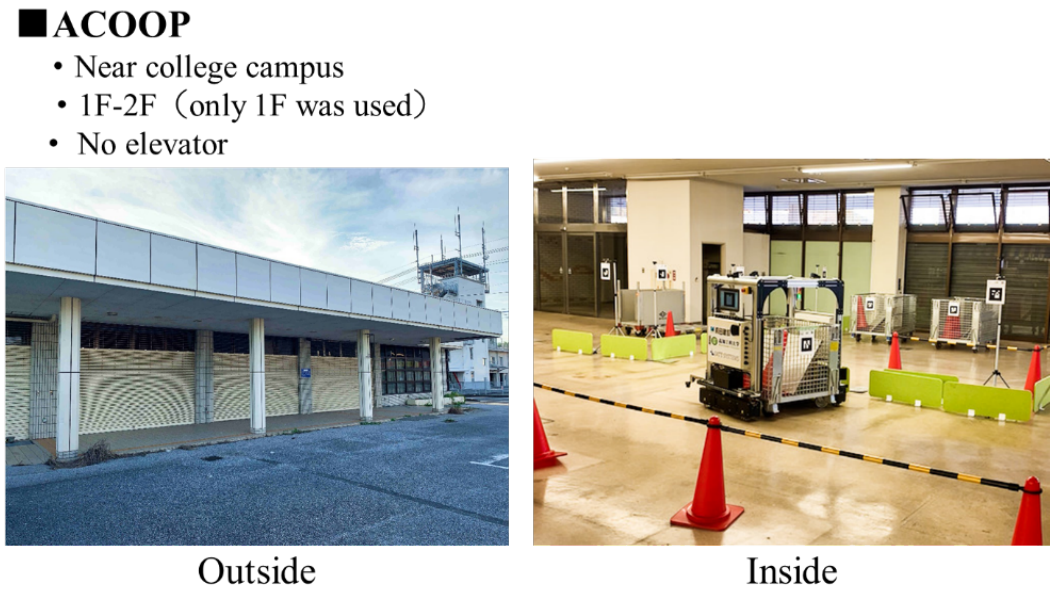
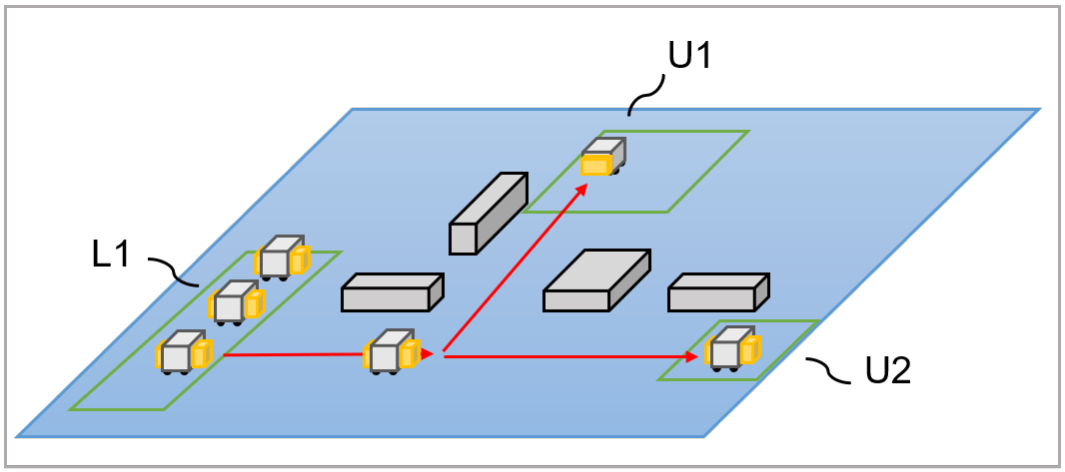


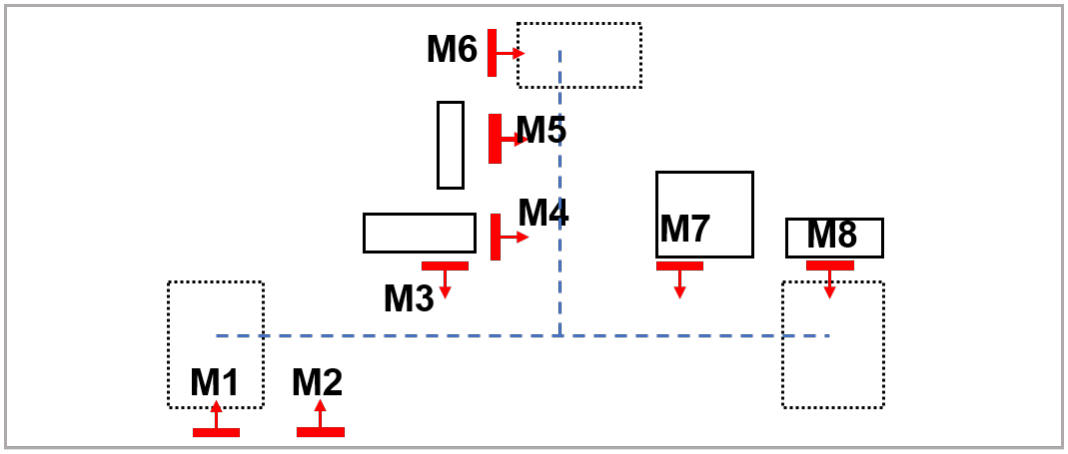
Figure 3-25: Basic information of ACOOP.

Figure 3-26 shows the route of one of the conducted trails. Capper needed to move two carts from zone L1 to U1 and one cart from L1 to U2. Eight markers numbered from M1 to M8 were used. The info points, along with their relationships, were entered into the system to establish a guidance knowledge base. We evaluated the effectiveness of HEIGA in providing directional instructions and the effectiveness of Capper at executing them. Figure 3-27 shows four photos of the trial.

In ACOOP, all the required functions including cart manipulation, obstacle avoidance, marker-based knowledge base have been evaluated. However, due to the filed limitation, long-distance transcription and vertical transportation (ELV) yet need further evaluation.



(1) Route Configuration



(2) Info Point Configuration

Figure 3-26: ACOOP: Trial 1 (marker configuration).

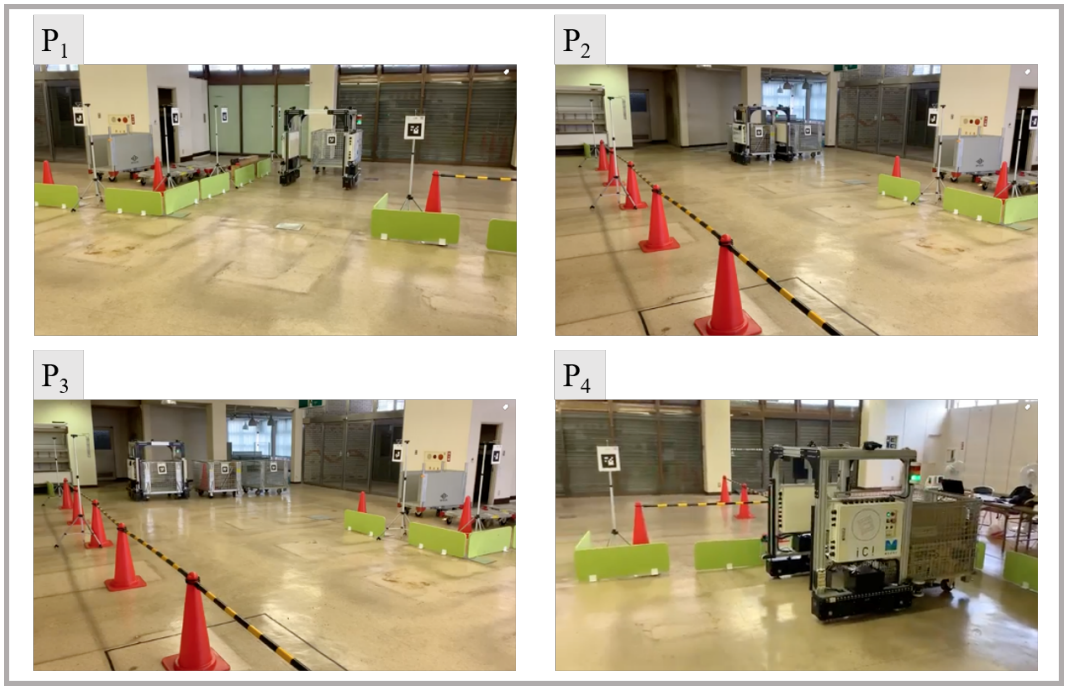


Figure 3-27: ACOOP: Trial 1 (photos).

3.7.2 Construction Laboratory: ICI

We performed the next phase of experiments in the construction laboratory of our cooperated construction company. The laboratory is named ICI and has a massive indoor space, as it was designed to conduct construction-related experiments.

Compared with ACOOP, ICI can be configured with scenarios that are closer to real construction sites, including more construction materials, and equipment. More importantly, the lab allows the evaluation for long-distance horizon transportation and vertical transportation with an elevator (Figure 3-28).

Here, we demonstrate two experiments conducted in the ICI lab.

1) **Long-Distance and Vertical Task**. Although basic automatic control algorithms have been fully investigated back in the experiments in ACOOP. The actual long-distance transportation with an elevator is what will frequently be required in real construction sites. Therefore, a similar configuration was conducted in ICI to evaluate the system considering both long-term horizon transportation, and vertical transportation. Despite that only six markers were used,

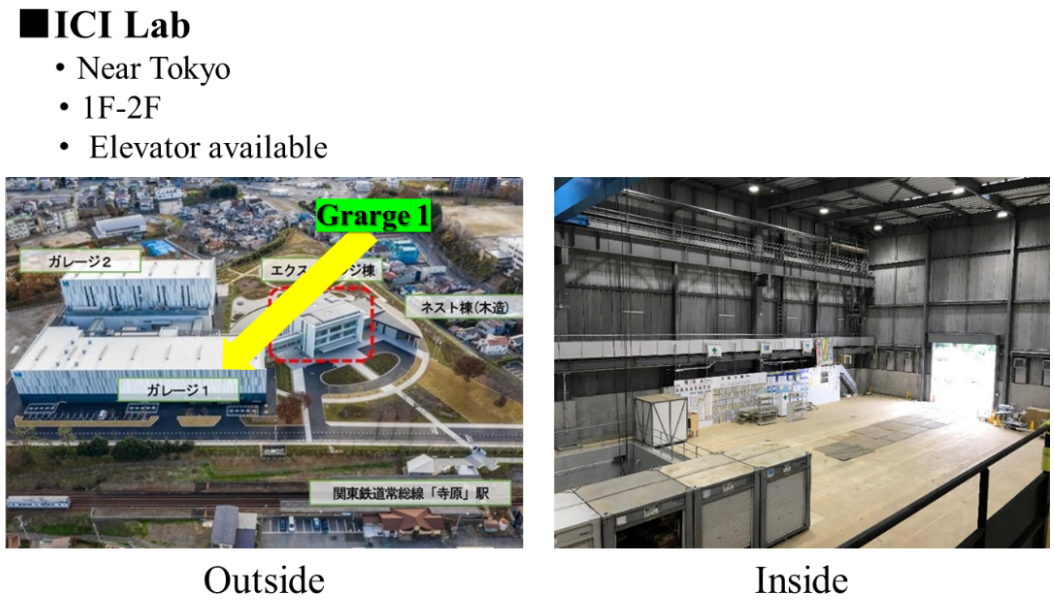


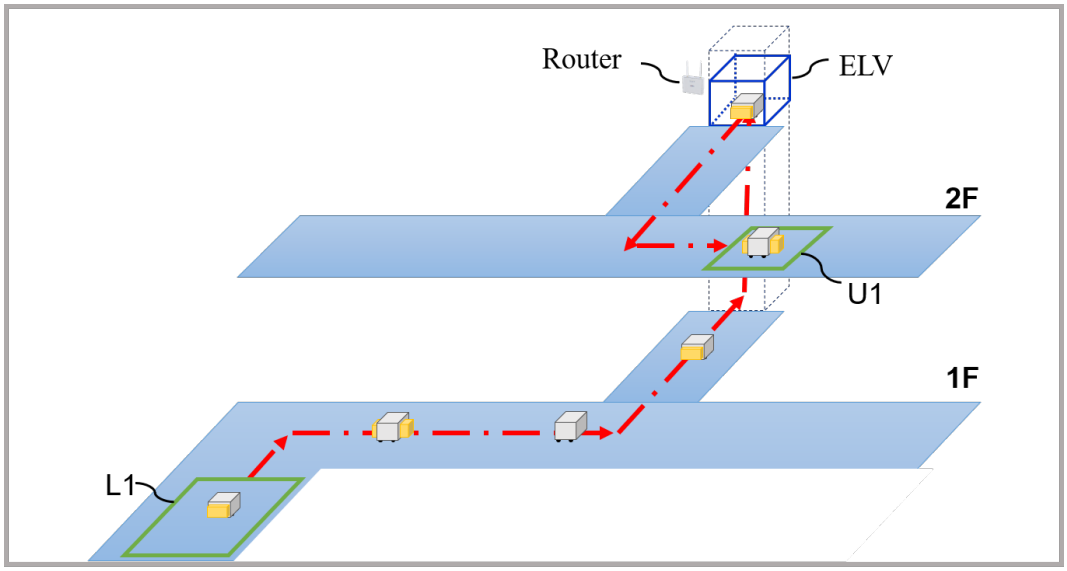
Figure 3-28: Basic information of ICI lab.

the transportation from 1F to 2F for around 100m was succeeded.

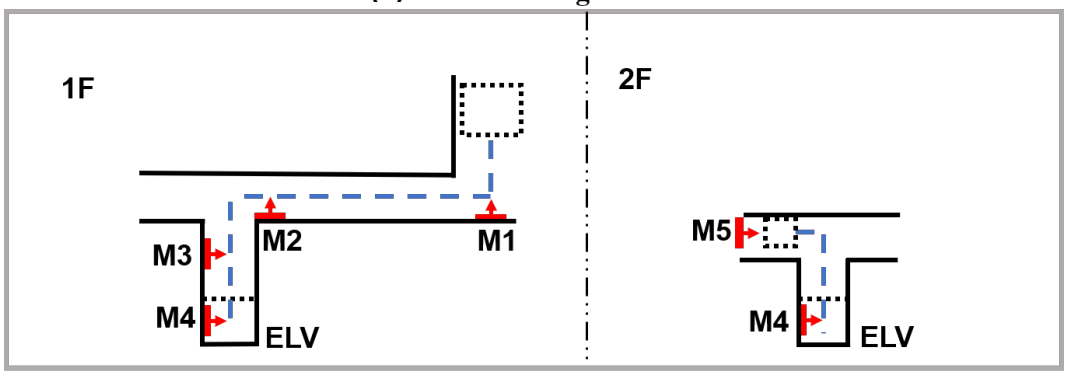
As Figure 3-29 shows, Capper picked the cart in L1, went to 2F through the ELV, dropped the cart in U1, returned to 1F through the ELV, and finally returned to L1. Figure 3-30 shows the photos captured during the trial. As demonstrated, the ELV was just 10 cm wider than the length of Capper carrying a cart. In this scenario, although only five AR markers were deployed (markers on the carts are not included), the field was successfully configured.

2) **Transportation in an Environment Closer to Real Sites**. Another important difference from ACOOP to a real construction site is the environment that a robot may perceives. Therefore, in ICI, we simulated an environment with carts loaded with materials and various building machines. Nine markers marked form M_1 to M_9 were used to configure the environment.

Figure 3-31 illustrates the route: two carts were initially placed in L1; Capper was then instructed to move them to U1 and U2, respectively. Meanwhile, a wide-range camera was configured from a bird viewpoint (Figure 3-32).



(1) Route Configuration



(2) Info Point Configuration

Figure 3-29: ICI: Trial 1 (marker configuration).

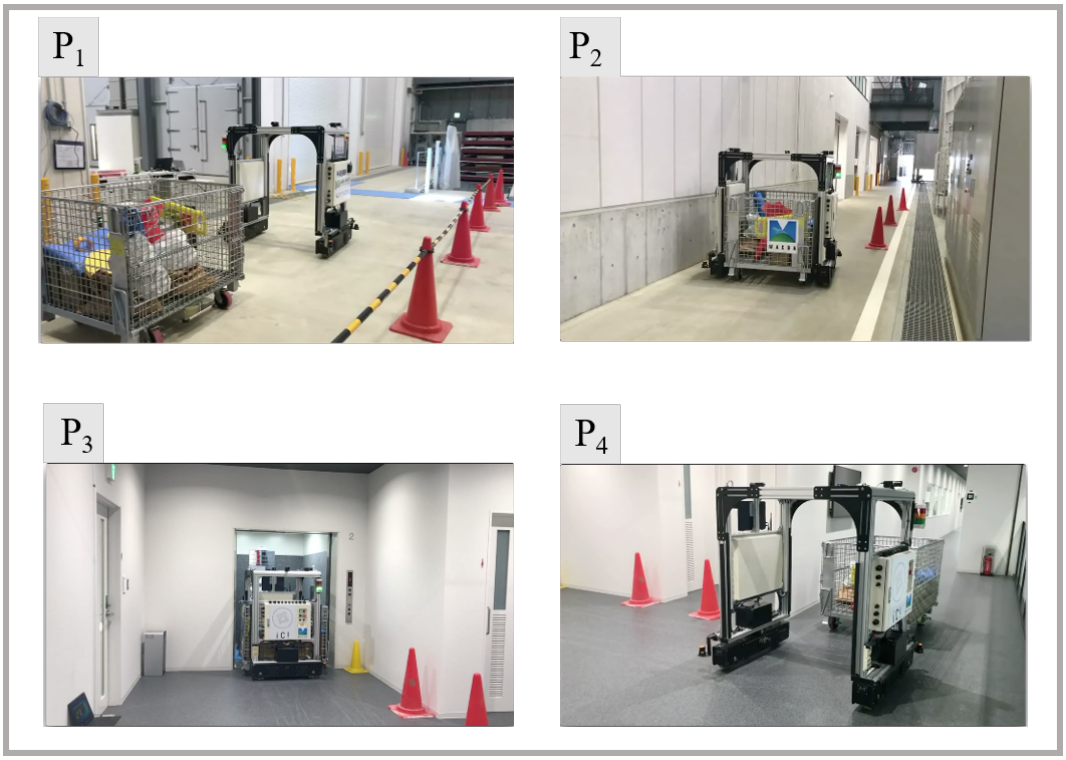


Figure 3-30: ICI: Trial 1 (photos).

3.7.3 Construction Site in Shinagawa, Tokyo

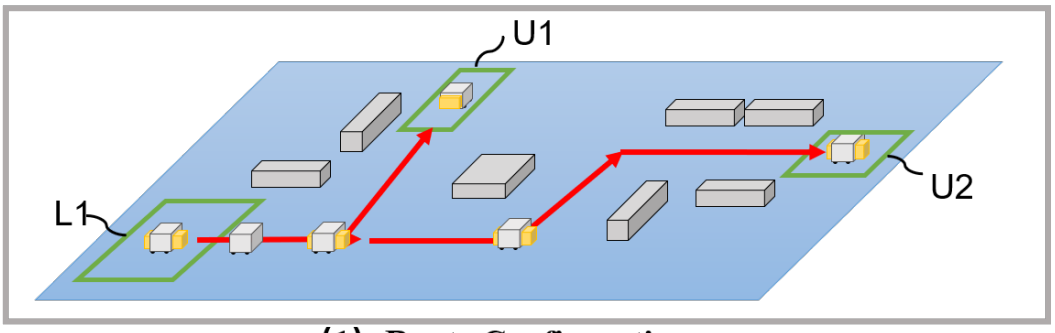
The Yamato construction sit as shown in Figure 3-33 was designed as a logistics center and office for the Yamato cooperation, which is located in Shinagawa, Tokyo. The building has ten floors and an elevator available for vertical transportation.

For first time, our newly developed material handing robots were deployed into a real construction site. A cart transportation and recycle task was presented.

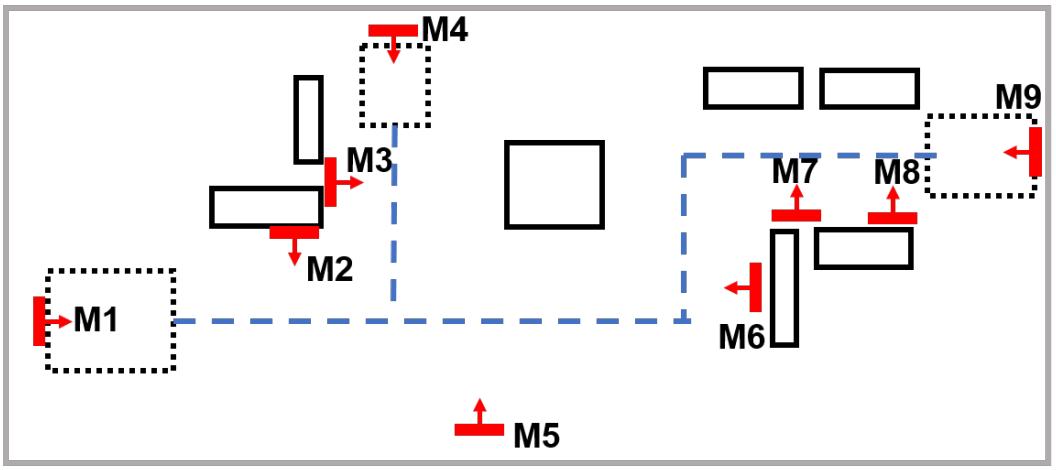
As Figure 3-34 shows, Capper loaded a cart in 1F, went to 5F through the ELV, dropped the cart in U1, picked another cart in U2, and eventually returned to 1F. Figure 3-35 shows the photos that were taken during the trial.

3.7.4 Construction Site in Nagoya

As Figure 3-36 shows, the Fuji Machinery construction site was designed as a new factory for the Fuji Machinery Cooperation. The factory has three floors, and elevators can be used for material transportation.



(1) Route Configuration



(3) Info Point Configuration

Figure 3-31: ICI: Trial 2 (marker configuration).

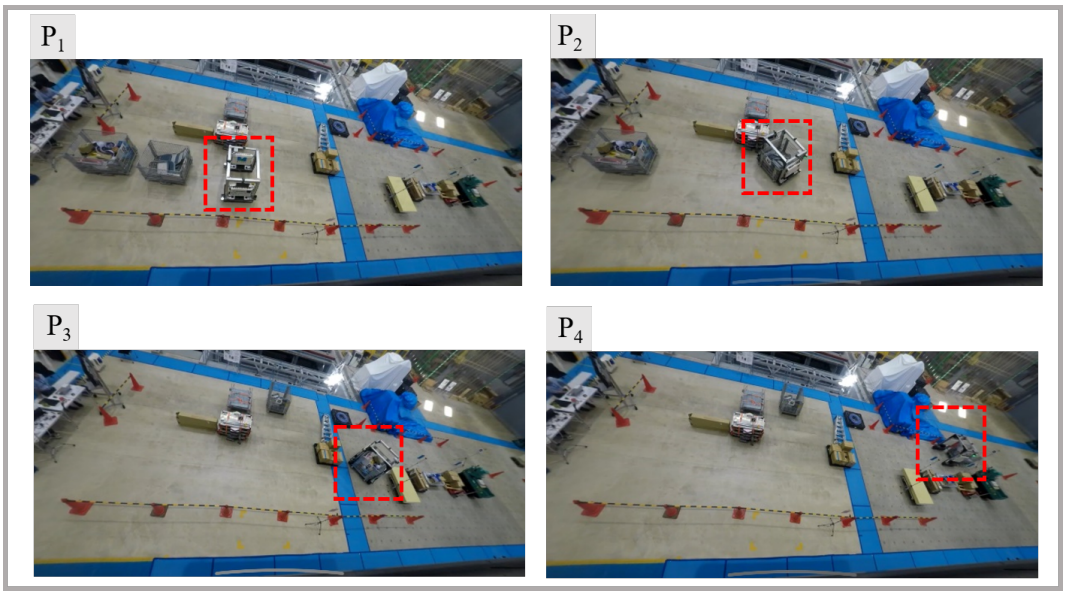


Figure 3-32: ICI: Trial 2 (photos).

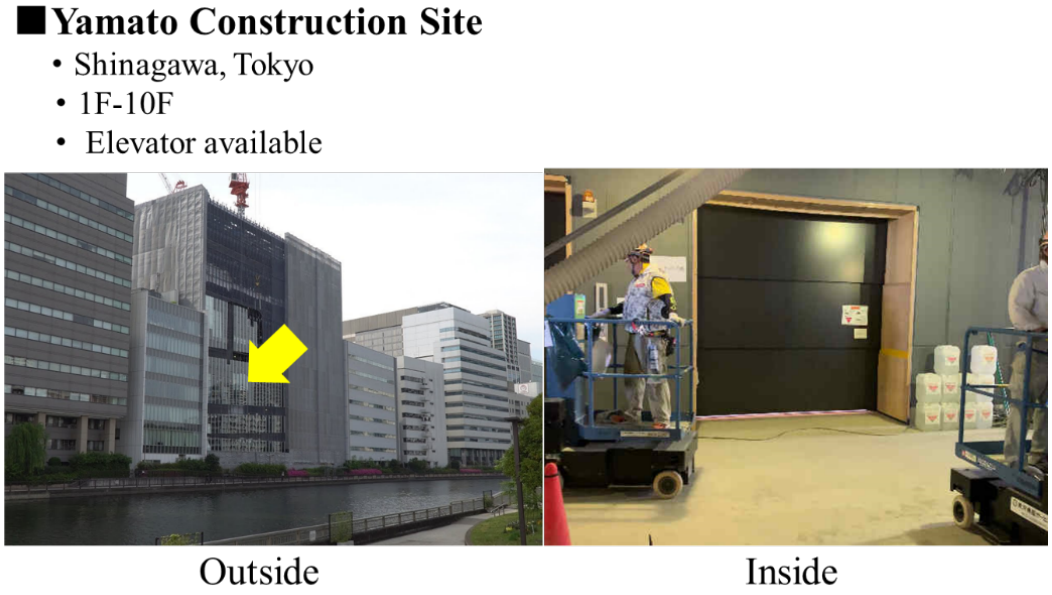
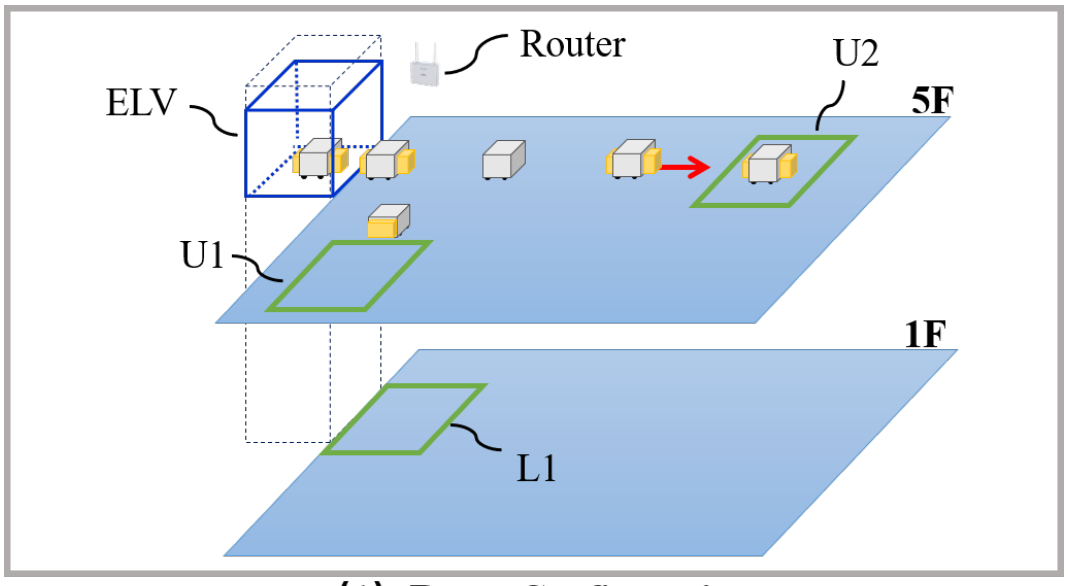


Figure 3-33: Basic information of the Yamato construction site.

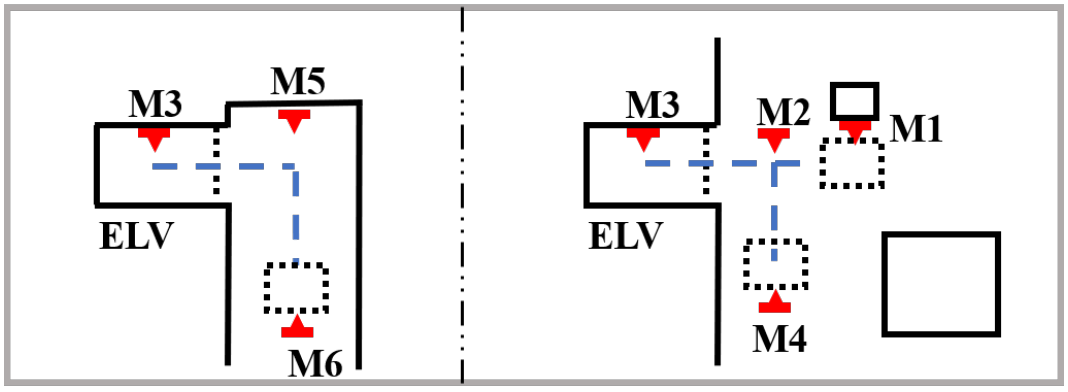
1) **Coordinate Material Transportation**. In a multi-floor wide construction sites, coordination is required for a more effective material handling system. More detailly, one robot would be responsible for relocating all the carts from 1F to their target floors respectively. Meanwhile, multiple robots will be deployed in each floor, while the master robot leaves the cart near the elevator and return, the slave robots in each floor will continue to load the carts and unload them to the desired unloading zones.

   We spent a few days in the site especially for the evaluation of coordinate material handling. Figure 3-37 shows one of the trials. A total of 12 markers were used in both 1F and 3F. The master robot was commanded to relocate a cart from L1 to U1, and the slave robot was deployed in 3F, waiting for the cart being relocated into U1, then conduct the transportation task from U1 to U2. Photos during the trial are presented in Figure 3-38.

2) **Transportation of Building Materials**. As the last experiment introduced in the work, we conducted material transportation which was on the original schedule of the construction site. In other words, instead of experimental ma-



(1) Route Configuration



(2) Info Point Configuration

Figure 3-34: Shinagawa: Trial 1 (marker configuration).

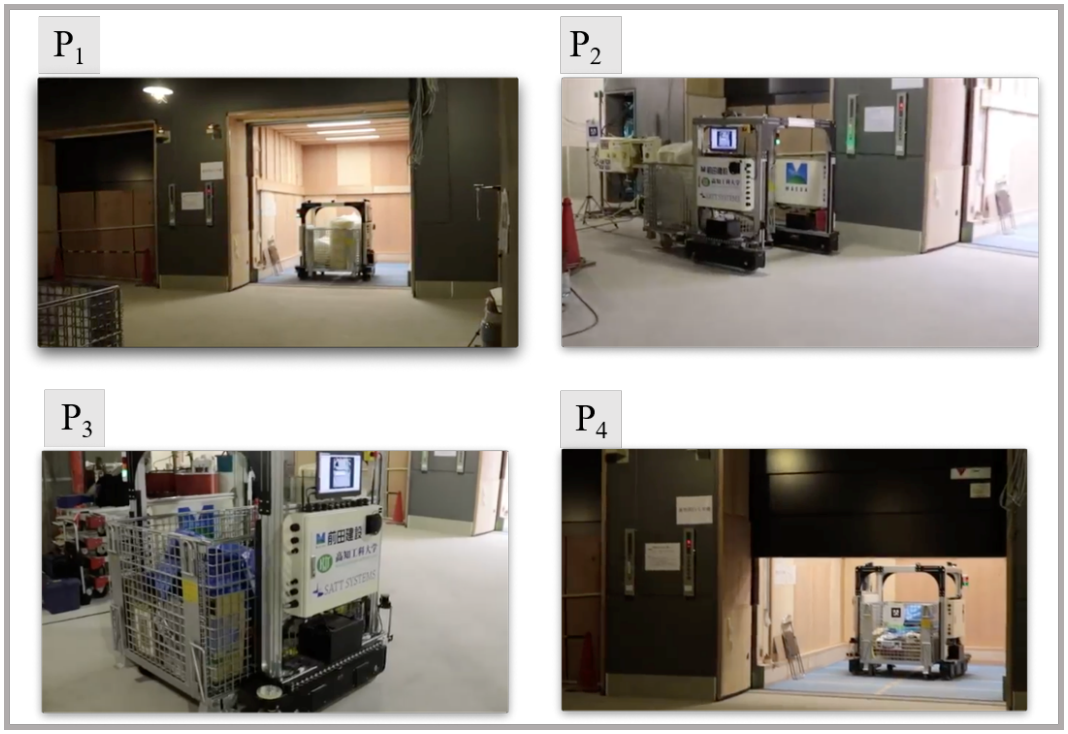


Figure 3-35: Shinagawa: Trial 1 (photos).

Figure 3-36: Basic information of Fuji Machinery Construction Site.

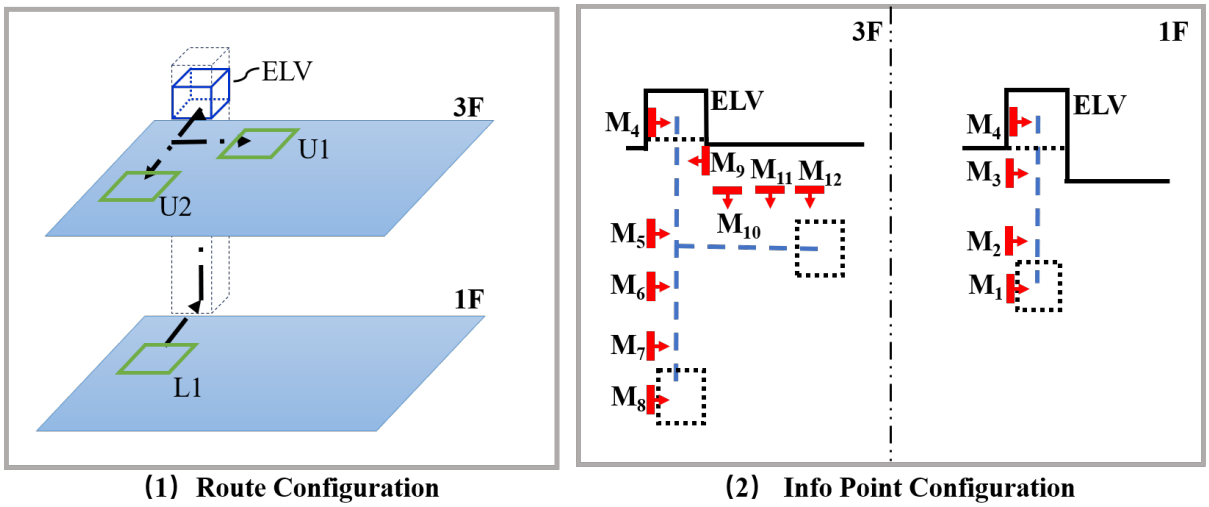


Figure 3-37: Nagoya: Trial 1 (marker configuration).

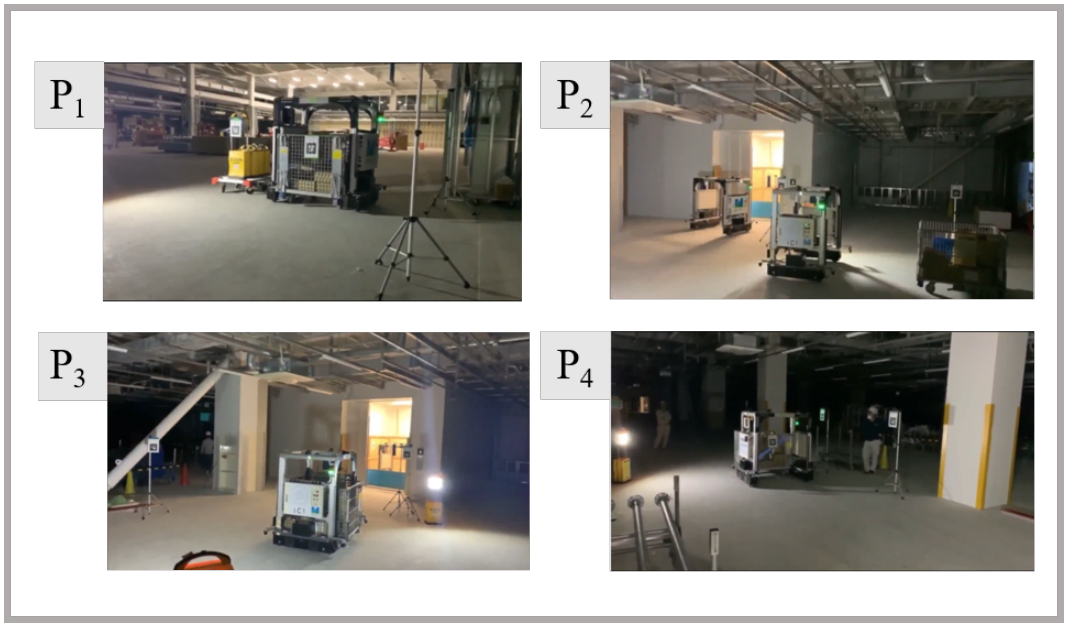
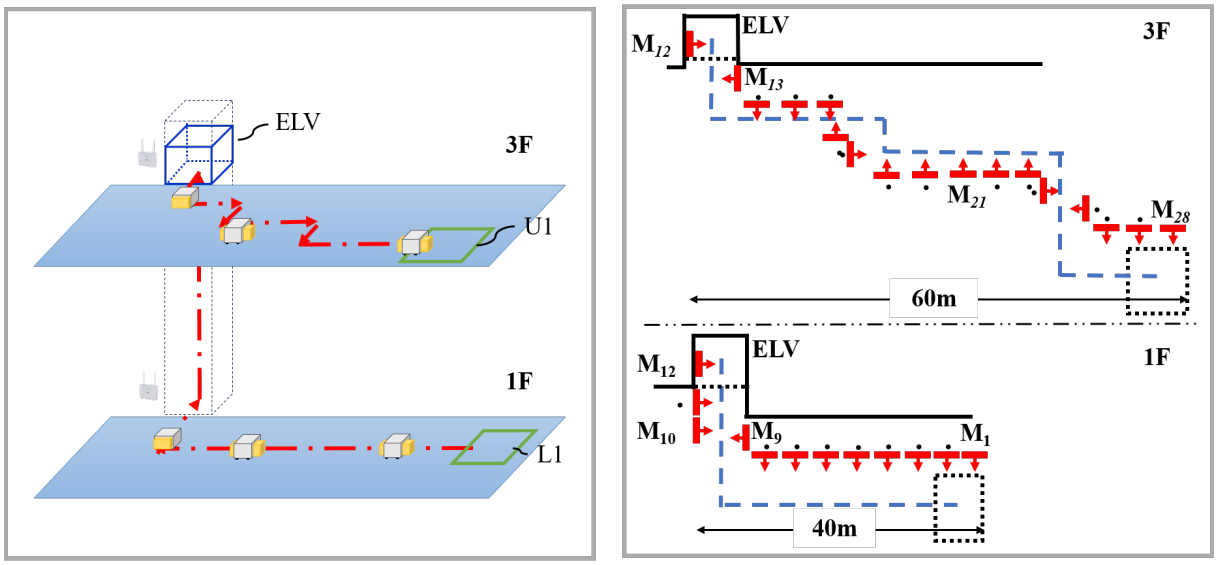


Figure 3-38: Nagoya: Trial 1 (photos).

terial handling tests, the trials conducted actually contributed to the site. A single cart was loaded with 40 tanks of painting oil, which added up to 600kg. During the two-day trial in the nights between PM 9:00 to AM 2:00, a total of 20 carts (800 tanks) were transported from 1F to 3F.

As Figure 3-39 shows, 28 markers were deployed. During the night, the construction site can be completely dark. Three LED lights were configured on the robot, so that the robot can perceive the environment without requiring for on-site lighting. Figure 3-40 shows the photos taken during the trial.



(1) Route Configuration (2) Info Point Configuration

Figure 3-39: Nagoya: Trial 2 (marker configuration).

3.8 Discussion

By deploying the HIEGA based material transportation system into three different types of sites, HEIGA has been proved to be effective in enabling autonomous material transportation in unstructured and dynamics indoor environments. Since robots are expected to work outside human working hours, rather than competing with employees considering transportation speed, the construction working efficiency will guaranteed to be increased.

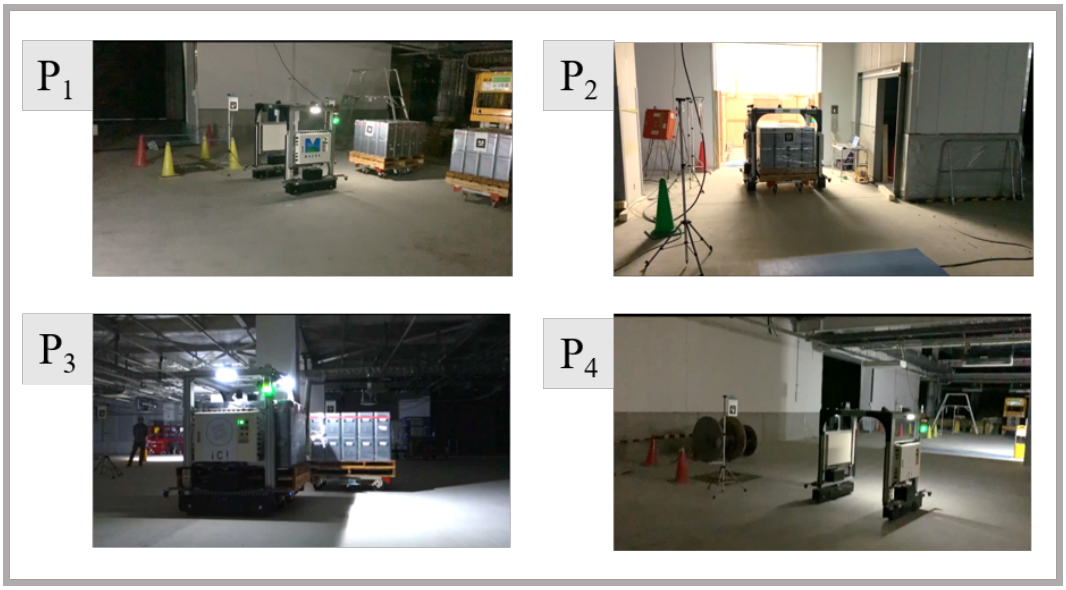


Figure 3-40: Nagoya: Trial 2 (photos).

For the widely investigated navigation methods (either using tape, beacons or mark-less solutions such as SLAM), the core idea, is self-localization in a global frame. In other words, once accurate self-positioning can be achieved, navigating to any coordinate is obtainable. The biggest challenge is that obtaining an accurate self-localization becomes harder and harder as the environment gets more complex and dynamic.

The most important characteristic of HEIGA is that the performance of the system will not be influenced dramatically by the complexity or the dynamics of the environment. From ACOOP, ICI, to real construction sites, the performance of the system has not been influenced. In other words, we eventually decouple the performance of the automated handling system from environmental complexity and dynamics.

3.9 Conclusions and Future Work

In this paper, we have proposed a guidance approach called HEIGA. By deploying info points at critical locations and building a guidance knowledge base accordingly, an agent can reach its destination without the need for self-localization in a global frame. The approach has been evaluated in three types of fields, as a result, HEIGA

based automated material transportation:

1) keeps humans out of heavy manipulation and repetitive tasks (material transportation), thus reducing the risk of harms and repetitive motion injuries;

2) improve construction working efficiency dramatically since material handling can be conducted automatically in longer working hours.

Most importantly, as far as we are concerned, for the first time, we have achieved fully autonomous material transportation in uncontrolled multi-floored construction sites (the transported items are real construction materials that are scheduled to be used).

Our future work would focus on two directions aiming at higher transportation working efficiency: (i) coordinated transportation that allows more kinds of carts to be manipulated with multiple robots working together; and (ii) relay transportation on a larger scale that allows a team of robots to conduct multi-floored transportation in a much higher efficiency.

Chapter 4

Conclusion

Our main contributions in this work is summarized as follows. With the goal of contributing to a declining birthrate and an aging population: (i) on one hand, we tried to achieve high-level cognitive intelligence on personal care robot KUT-PCR so that caring tasks can be conducted as required where no care givers are available. As a result, the quality of life of the care recipients can be improved while the caring cost will be decreased; (ii) on the other hand we focus on motion intelligent, that enables autonomous material transportation in construction sites with Capper, so that the effect caused by the lack of young labor in construction sites can be effectively decreased.

As all the presented approaches have been evaluated in a real household environment and construction sites respectively, we still face great challenges in deploying the robots into homes and hospitals, or all kinds of construction sites needing automatic handling. In the future work, we have planned continues development and evaluation schedule, so that the algorithms now developed in the laboratory can be evaluated and deployed into the society and make certain contribution.

Bibliography

[1] Koichiro Shiba, Naoki Kondo, and Katsunori Kondo. Informal and formal social support and caregiver burden: the ages caregiver survey. *Journal of Epidemiology*, 26(12):622–628, 2016.

[2] T Barnett, Pathmavathy Namasivayam, and DAA Narudin. A critical review of the nursing shortage in malaysia. *International Nursing Review*, 57(1):32–39, 2010.

[3] Ahmad Aboshaiqah. Strategies to address the nursing shortage in saudi arabia. *International Nursing Review*, 63(3):499–506, 2016.

[4] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation*, pages 1386–1383, 2017.

[5] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Yuichi Taguchi, Tim K Marks, and Rama Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, 31(8):951–973, 2012.

[6] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[11] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the 2012 Asian Conference on Computer Vision*, pages 548–562. Springer, 2012.

[13] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings of the 2001 International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE, 2001.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the 2014 European Conference on Computer vision*, pages 740–755. Springer, 2014.

[15] Jay M. Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David MS Johnson, et al. Segicp: Integrated deep semantic segmentation and pose estimation. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5784–5789, 2017.

[16] Lasitha Piyathilaka and Sarath Kodagoda. Affordance-map: A map for context-aware path planning. In *Proceedings of the 2014 Australasian Conference on Robotics and Automation*, 2014.

[17] Stephan Sehestedt, Sarath Kodagoda, and Gamini Dissanayake. Robot path planning in a social context. In *Proceedings of the 2010 IEEE Conference on Robotics, Automation and Mechatronics*, pages 206–211, 2010.

[18] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. In *Proceedings of the 2016 IEEE international conference on robotics and automation*, pages 5729–5736, 2016.

[19] Ming Ding, Ryojun Ikeura, Toshiharu Mukai, Hiromichi Nagashima, Shinya Hirano, Kazuya Matsuo, Minghui Sun, Shigeyuki Hosoe, et al. Comfort estimation during lift-up using nursing-care robot—riba. In *Proceedings of the 2012 International Conference on Innovative Engineering Systems*, pages 225–230, 2012.

[20] Matthias Sczesny-Kaiser, Oliver Höffken, Mirko Aach, Oliver Cruciger, Dennis Grasmücke, Renate Meindl, Thomas A Schildhauer, Peter Schwenkreis, and Martin Tegenthoff. Hal® exoskeleton training improves walking parameters and

normalizes cortical excitability in primary somatosensory cortex in spinal cord injury patients. *Journal of NeuroEngineering and Rehabilitation*, 12(68):1–11, 2015.

[21] Takashi Yamamoto, Koji Terada, Akiyoshi Ochiai, Fuminori Saito, Yoshiaki Asahara, and Kazuto Murase. Development of human support robot as the research platform of a domestic mobile manipulator. *ROBOMECH Journal*, 6(1):4, 2019.

[22] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[23] Patric Beinschob and Christoph Reinke. Graph slam based mapping for agv localization in large-scale warehouses. In *Proceedings of the 2015 IEEE International Conference on Intelligent Computer Communication and Processing*, pages 245–248, 2015.

[24] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Proceedings of the 2014 European conference on computer vision*, pages 834–849, 2014.

[25] Chenguang Yang, Kunxia Huang, Hong Cheng, Yanan Li, and Chun-Yi Su. Haptic identification by elm-controlled uncertain manipulator. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(8):2398–2409, 2017.

[26] Giuseppe Di Leo, Consolatina Liguori, Antonio Pietrosanto, and Paolo Sommella. A vision system for the online quality monitoring of industrial manufacturing. *Optics and Lasers in Engineering*, 89:162–168, 2017.

[27] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems*, 45(1):51–61, 2014.

[28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521:436–444, 2015.

[29] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization-based full body control for the darpa robotics challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.

[30] Nikolaus Correll, Kostas E Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M Romano, and Peter R Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 15(1):172–188, 2016.

[31] Marc Hanheide, Moritz Göbelbecker, Graham S Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, et al. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 247:119–150, 2017.

[32] Marco Hutter, Christian Gehring, Andreas Lauber, Fabian Gunther, Carmine Dario Bellicoso, Vassilios Tsounis, Péter Fankhauser, Remo Diethelm, Samuel Bachmann, Michael Blösch, et al. Anymal-toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918–931, 2017.

[33] Patrick M. Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics*, 33(3):509–522, 2017.

[34] Jiwoong Kim and Woojin Chung. Localization of a mobile robot using a laser range finder in a glass-walled environment. *IEEE Transactions on Industrial Electronics*, 63(6):3616–3627, 2016.

[35] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. Travi-navi: Self-deployable indoor navigation system. *IEEE/ACM Transactions on Networking*, 25(5):2655–2669, 2017.

[36] G Ajay Kumar, Ashok Kumar Patil, Rekha Patil, Seong Sill Park, and Young Ho Chai. A lidar and imu integrated indoor navigation system for uavs and its application in real-time pipeline classification. *Sensors*, 17(6):1–24, 2017.

[37] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence*, 247:45–69, 2017.

[38] Toshiaki Tsuji, Jun Ohkuma, and Sho Sakaino. Dynamic object manipulation considering contact condition of robot with tool. *IEEE Transactions on Industrial Electronics*, 63(3):1972–1980, 2015.

[39] Kai Chen, Fangkai Yang, and Xiaoping Chen. Planning with task-oriented knowledge acquisition for a service robot. In *Proceedings of the 2016 International Joint Conference on Artificial Intelligence*, pages 812–818, 2016.

[40] Shiqi Zhang, Mohan Sridharan, and Jeremy L. Wyatt. Mixed logical inference and probabilistic planning for robots in unreliable worlds. *IEEE Transactions on Robotics*, 31(3):699–713, 2015.

[41] Håkan LS Younes and Michael L Littman. Ppddl1. 0: An extension to pddl for expressing planning domains with probabilistic effects. *Techn. Rep. CMU-CS-04-167*, pages 1–23, 2004.

[42] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26(1):191–246, 2006.

[43] Gi Hyun Lim, Il Hong Suh, and Hyowon Suh. Ontology-based unified robot knowledge for service robots in indoor environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3):492–509, 2010.

[44] Moritz Tenorth and Michael Beetz. Representations for robot knowledge in the knowrob framework. *Artificial Intelligence*, 247:151–169, 2017.

[45] Hongyan Guo, Hong Chen, Fang Xu, Fei Wang, and Geyu Lu. Implementation of ekf for vehicle velocities estimation on fpga. *IEEE Transactions on Industrial Electronics*, 60(9):3823–3835, 2012.

[46] Lingguang Song and Neil N. Eldin. Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling. *Automation in Construction*, 27:32–39, 2012.

[47] Souravik Dutta, Yiyu Cai, Lihui Huang, and Jianmin Zheng. Automatic replanning of lifting paths for robotized tower cranes in dynamic bim environments. *Automation in Construction*, 110:1–19, 2020.

[48] Farzaneh Golkhoo and Osama Moselhi. Optimized material management in construction using multi-layer prceptron. *Canadian Journal of Civil Engineering*, 46(10):909–923, 2019.

[49] A. Ahmadian F.F., A. Akbarnezhad, T. H. Rashidi, and S.T. Waller. Importance of planning for the transport stage in procurement of construction materials. In *Proceedings of The 31st International Symposium on Automation and Robotics in Construction and Mining*, pages 466–473, 2014.

[50] Naoko Muramatsu and Hiroko Akiyama. Japan: Super-aging society preparing for the future. *The Gerontologist*, 51(4):425–432, 2011.

[51] Franco Fraccaroli and Jürgen Deller. Work, aging, and retirement in europe: Introduction to the special issue. *Work, Aging and Retirement*, 1(3):237–242, 2015.

[52] Ning Jackie Zhang, Man Guo, and Xiaoying Zheng. China: Awakening giant developing solutions to population aging. *The Gerontologist*, 52(5):589–596, 2012.

[53] Mi Pan, Thomas Linner, Wei Pan, Huimin Cheng, and Thomas Bock. Structuring the context for construction robot development through integrated scenario approach. *Automation in Construction*, 114:1–16, 2020.

[54] Carlos Balaguer and Mohamed Abderrahim. *Robotics and Automation in Construction*. InTech, 2008.

[55] Daniel Claes, Frans Oliehoek, Hendrik Baier, and Karl Tuyls. Decentralised online planning for multi-robot warehouse commissioning. In *Proceedings of the 16th International Conference on Antonomous Agents and Multiagent Sytems*, pages 492–500, 2017.

[56] Golnaz Habibi, Zachary Kingston, William Xie, Mathew Jellins, and James McLurkin. Distributed centroid estimation and motion controllers for collective transport by multi-robot systems. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, pages 1282–1288, 2015.

[57] Roberto Pinillos, Samuel Marcos, Raul Feliz, Eduardo Zalama, and Jaime Gómez-García-Bermejo. Long-term assessment of a service robot in a hotel environment. *Robotics and Autonomous Systems*, 79:40–57, 2016.

[58] Thomas Bock. Construction robotics. *Autonomous Robots*, 22:201–209, 2007.

[59] L. Armesto and J. Tornero. Automation of industrial vehicles: A vision-based line tracking application. In *Proceedings of the 2009 IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–7, 2009.

[60] Mark Taylor, Sam Wamuziri, and Ian Smith. Automated construction in japan. *Civil Engineering*, 156(1):34–41, 2003.

[61] Shaoping Lu, Chen Xu, and Ray Y. Zhong. An active rfid tag-enabled locating approach with multipath effect elimination in agv. *IEEE Transactions on Automation Science and Engineering*, 13(3):1333–1342, 2016.

[62] John J. Enright and Peter R. Wurman. Optimization and coordinated autonomy in mobile fulfillment systems. In *Proceedings of the 9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots*, pages 33–38, 2011.

[63] Tomoya Fujimoto, Jun Ota, Tamio Arai, Tsuyoshi Ueyama, and Tsuyoshi Nishiyama. Semi-guided navigation of agv through iterative learning. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 968–973, 2001.

[64] Zhi Song, Xinyu Wu, Tiantian Xu, Jianquan Sun, Qinshi Gao, and Yong He. A new method of agv navigation based on kalman filter and a magnetic nail localization. In *Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics*, pages 952–957, 2016.

[65] José M. Bengochea-Guevara, Jesus Conesa-Muñoz, Dionisio Andújar, and Angela Ribeiro. Merge fuzzy visual servoing and gps-based planning to obtain a proper navigation behavior for a small crop-inspection robot. *Sensors*, 16(3):1–23, 2016.

[66] F. Serhan Daniş and Ali Taylan Cemgil. Model-based localization and tracking using bluetooth low-energy beacons. *Sensors*, 17(11):1–23, 2017.

[67] Weiquan Ye, Zhijun Li, Chenguang Yang, Junjie Sun, Chun-Yi Su, and Renquan Lu. Vision-based human tracking control of a wheeled inverted pendulum robot. *IEEE Transactions on Cybernetics*, 46(11):2423–2434, 2016.

[68] Tokyu construction and thk begins construction robot demonstration test, [Online]. Available https://www.thk.com/?q=jp/node/20107. (2019), Accessed date: 30 May 2020.

[69] Robots under autonomous control at robot lab, [Online]. Available https://www.shimz.co.jp/en/company/about/news-release/2018/2018006.html. (2018), Accessed date: 30 May 2020.

[70] Automatic transfer system to improve productivity at the construction site of a new building, [Online]. Available https://www.obayashi.co.jp/news/detail/news20181221_1.html. (2018), Accessed date: 30 May 2020.

[71] A logistics system that autonomously transports materials at construction sites, [Online]. Available https://www.obayashi.co.jp/news/detail/news20200219_1.html. (2020), Accessed date: 30 May 2020.

[72] T. Tsumura. Survey of automated guided vehicle in a japanese factory. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pages 1329–1334, 1986.

[73] Chang Boon Low and Danwei Wang. Gps-based path following control for a car-like wheeled mobile robot with skidding and slipping. *IEEE Transactions on Control Systems Technology*, 16(2):340–347, 2008.

[74] Aswin N Raghavan, Harini Ananthapadmanaban, Manimaran S Sivamurugan, and Balaraman Ravindran. Accurate mobile robot localization in indoor environments using bluetooth. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 4391–4396, 2010.

[75] Davide Ronzoni, Roberto Olmi, Cristian Secchi, and Cesare Fantuzzi. Agv global localization using indistinguishable artificial landmarks. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pages 287–292, 2011.

[76] Raffaello D'Andrea. Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4):638–639, 2012.

[77] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pages 1271–1278, 2016.

[78] Henry Carrillo, Ian Reid, and José A. Castellanos. On the comparison of uncertainty criteria for active slam. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pages 2080–2087, 2012.

[79] Carlos Estrada, José Neira, and Juan D. Tardós. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[80] Georg Gerstweiler, Emanuel Vonach, and Hannes Kaufmann. Hymotrack: A mobile ar navigation system for complex indoor environments. *Sensors*, 16(1):1–19, 2016.

[81] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6622–6631, 2019.

[82] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. Handheld augmented reality indoor navigation with activity-based instructions. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 211–220, 2011.

[83] Kazuo Okuhata, Shingo Ino, Yoshinobu Mizobuchi, Hajime Okamura, Shuoyu Wang, and Guang Yang. Proposal of structure of automatic interior material transfer robot adapting to construction site. In *Proceedings of the 2020 Annual Conference of the Robotics Society of Japan*, 2020.

[84] Junfeng Yao, Chao Lin, Xiaobiao Xie, Andy JuAn Wang, and Chih-Cheng Hung. Path planning for virtual human motion using improved a* algorithm. In *Proceedings of the Seventh International Conference on Information Technology*, pages 1154–1158, 2010.

[85] Anthony Stentz. The focussed d* algorithm for real-time replanning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1652–1659, 1995.

List of Publications

Disclaimer: some content of the thesis has been modified from a part of the following publications, which were originally written by the author of this thesis.

Journals

[1] Guang Yang, Shuoyu Wang, and Junyou Yang, Desire-Driven Reasoning for Personal Care Robots, IEEE Access, Vol. 7, pp. 75203-75212, 2019

[2] G. Yang, S. Wang, H. Okamura, B. Shen, Y. Ueda, T. Yasui, T. Yamada, Y. Miyazawa, S. Yoshida, Y. Inada, S. Ino, K. Okuhata, Y. Mizobuchi, "A Hallway Exploration-Inspired Guidance Approach and Its Application sin Autonomous Material Transportation in Construction Sites", Automation in Construction. (Provisional Acceptance)

[3] Guang Yang, Shuoyu Wang, Junyou Yang, Bo Shen, and Peng Shi, Pose Estimation of Daily Containers for a Life-support Robot, International Journal of Innovative Computing, Information and Control, Vol. 14, No.4, pp. 1545-1552, 2018.

[4] Guang Yang, Shuoyu Wang, Context-aware Local Planning for Personal Care Robots, International Journal of Innovative Computing, Information and Control , Vol. 15, No.6, pp. 2385–2392, 2019.

[5] Guang Yang, Shuoyu WANG, Bo SHEN, Hayato ENOKI, Kenji ISHIDA, and Kazuo OKUHATA, Yoshinobu MIZOBUCHI and Shingo INO, Personal Care

Robot: Mechanism, Motion Control, and Pose Estimation, International Journal of Biomedical Soft Computing and Human Sciences, Vol. 23, No. 2, pp. 83-89, 2018

[6] Guang Yang, Shuoyu Wang, Junyou Yang and Peng Shi, "Desire-driven Knowledge-based Life-assist Robot with Required Preference", IEEE Transactions on Systems Man Cybernetics-Systems. (Under Review)

[7] 楊 光, 王 碩玉, 岡村 甫, 猪野 真吾, 奥畑 一男, 溝渕 宣誠, 資材自動搬送ロボットの自律化制御,日本ロボット学会誌. (Under Review)

[8] 奥畑 一男,猪野 真吾,溝渕 宣誠, 岡村 甫, 王 碩玉, 楊 光,建築現場に適応する内装資材自動搬送ロボットの構造提案 ,日本ロボット学会誌. (Under Review)

International Conferences

[1] Guang Yang, Shuoyu Wang, Junyou Yang, and Bo Shen, Active Pose Estimation of Daily Objects, The 2018 IEEE International Conference on Mechatronics and Automation (ICMA 2018), pp. 837-842, Changchun, August, 2018.

[2] Guang Yang, Shuoyu Wang , Junyou Yang, and Bo Shen, Viewpoint Selection Strategy for a Life Support Robot , 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR 2018), pp. 82-87, Shenyang, August, 2018.

[3] Guang Yang, Shuoyu Wang, Junyou Yang, and Bo Shen, Estimating the Poses of Daily Containers for a Life-support Robot, International Conference on Innovative Computing, Information and Control (ICICIC 2018), Lianyungang, August, 2018.

[4] Guang Yang, Shuoyu Wang, Modeling and Control of a Personal Care Robot Considering Posture Adjustment, The 2018 IEEE International Conference on

Mechatronics and Automation (ICMA 2019), pp. 1414-1418, Tianjin, August, 2019.

[5] Guang Yang, Shuoyu Wang, Context-aware Local Planning for Personal Care Robots, International Conference on Innovative Computing, Information and Control (ICICIC 2019), Seoul, August, 2019.

[6] Guang Yang, Shuoyu Wang, and Peng Shi: Desire-Driven Reasoning considering Status-Based Knowledge Description for Personal Care Robots, The 2020 IEEE International Conference on Mechatronics and Automation (ICMA 2020), Beijing, October, 2020.

[7] Guang Yang, Shuoyu Wang , Junyou Yang, and Peng Shi, Risk-Aware Motion Control for Care Robots, 2020 IEEE International Conference on Industrial Artificial Intelligence (IAI 2020), Shenyang, October 2020.

Domestic Conferences

[1] Guang Yang, Shuoyu Wang, and Bo Shen, Pose Estimation of Daily Containers using RGB-D Camera, 信学技報, vol. 117, no. 442, PRMU2017-160, pp. 89-91, 和歌山, 2018年2月.

[2] 楊 光, 王 碩玉，潘 博, 榎 勇人，石田 健司，和食 一男，溝渕 宣誠，猪野 真吾, 生活支援ロボット 基 本構造、運動制御、対象物認識, 第31回バイオメディカル・ファジィ・システム学会年次大会 講演論文集(BMFSA2018)，BMFSA2018-30，金沢，2018年11月．

[3] 楊 光, 王 碩玉, 楊 俊友, 生活支援ロホットの運動制御～局所的環境情報に基つく経路計画法,第 37回日本ロボット学会学術講演会(RSJ2019) , 東京, 2019年9月.

[4] 楊 光, 王 碩玉, 岡村 甫, 猪野 真吾, 奥畑 一男, 溝渕 宣誠, 資材自動搬送ロボットの自律化制御, 第38回日本ロボット学会学術講演会(RSJ2020) , オンライン開催, 2020年10月.

[5] 森 優，王 碩玉，瀋 博，楊 光，上田 康浩，安井 利彰, 測域センサによる建築資材運搬ロボットの障害物回避法、第31回バイオメディカル・ファジィ・システム学会年次大会 講演論文集(BMFSA2018)，金沢，2018年11月．

[6] 佐藤 春陽, 王 碩玉，瀋 博，楊 光， 生活支援ロボットの実用化に向けた色認識による反応動作システムの実現, 日本知能情報ファジィ学会 ソフトロボティックス研究会 第16回ポトラック＆ワークショップ,発表番号1-1, 高知, 2018年11月

[7] 佐藤 春陽，王 碩玉，瀋 博, 楊 光，寝たきり高齢者の生活支援を目的とした色認識での意思伝達方法, 日本機械学会中国四国学生会第49回学生員卒業研究発表講演会講演論文集，S1410，山口，2019年3月.

[8] 佐藤 春陽, 王 碩玉, 瀋 博, 楊 光, 画像情報を用いた生活支援ロボットへの生活欲求の意思伝達方法,第32回バイオメディカル・ファジィ・システム学会年次大会 (BMFSA2019), 山梨, 2019年11月．

[9] 森 優, 王 碩玉，瀋 博，楊 光，上田 康浩，安井 利彰, 建築現場における自動運搬ロボットの障害物回避行動について,第20回計測自動制御学会, 香川, 2019年12月.

[10] 中村 俊一, 王 碩玉, 瀋 博，楊 光, 生活支援ロボットによる冷蔵庫内の物体位置検知, 日本知能情報ファジィ学会 ソフトロボティックス研究会 第17回ポトラック＆ワークショップ, 高知, 2019年12月.

[11] 奥畑 一男,猪野 真吾,溝渕 宣誠, 岡村 甫, 王 碩玉, 楊 光, 建設現場における自律型資材搬送ロボットの走行安定性および車体軽量化に関する機体構造について, 第38回日本ロボット学会学術講演会(RSJ2020) , オンライン開催, 2020年10月.

Patents

[1] 特許願（出願中）特願2019-013801　発明の名称：自動搬送システム　発明

者： 岡村　甫、王　碩玉、楊　光、瀋　博、安井　利彰、山田　哲也、宮澤　友基、稲田　雄大

[2] 特許願（出願中）特願2020-013126 発明の名称：自動搬送システム 発明者： 岡村　甫、王　碩玉、楊　光、瀋　博、安井　利彰、山田　哲也、宮澤　友基、稲田　雄大、吉田　彩斗美

Acknowledgement

During the past three years of SSP program, I received valuable help from many people. Their opinions and suggestions help me greatly.

First, my heartiest thanks to my supervisor, Prof. Shuoyu Wang, for his helpful guidance, valuable suggestions and constant encouragement both in my study and in my life. His profound insight and accurateness about my study taught me so much.

Also, I would like to express my sincere gratitude to my co-supervisors Prof. Junyou Yang and Prof. Kyoko Shibata for the continuous support of my study and related research, for their patience, motivation, and immense knowledge.

Besides, I would like to thank the rest of my thesis committee: Prof. Koichi Oka, and Prof. Yukinobu Hoshino, for their insightful comments, but also for the questions which drive me to widen my research from various perspectives.

Also, my sincere thanks go to the team from ICI lab (Maeda Cooperation), who has provided tremendous assistance that allows the evaluation of robots in real construction sites to be possible. And thanks to the team from SATT Systems, who have conducted a wonderful job including the mechanical and electrical design. Without all their help, there is no chance that the presented results in this thesis can be obtained. I will always remember the days and nights we fought together on the construction sites, which is hard yet memorable.

Finally, I would like to express my deep gratefulness to my family and friends, especially my parents and my wife. Only with their selfless support, concern, and love, can I overcome those difficulties and pursue my dream till now. They are the source of my strength.

During this three years at Kochi University of Technology, I was able to devote

myself to scientific research, which was pleasant and fulfilling. I have learned from the best, and worked with excellent teams. From here, I am full of confidence and expectation for the future!