Ph.D. DISSERTATION

An Approach to Cooperative Robot Behavior for Object Transportation

물체 수송을 위한 협업 로봇의 행동 연구

BY

GYUHO EOH

FEBRUARY 2016

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

Abstracts

This dissertation presents two cooperative object transportation techniques according to the characteristics of objects: *passive* and *active*. The passive object is a typical object, which cannot communicate with and detect other robots. The active object, however, has abilities to communicate with robots and can measure the distance from other robots using proximity sensors. Typical areas of research in cooperative object transportation include grasping, pushing, and caging techniques, but these require precise grasping behaviors, iterative motion correction according to the object pose, and the real-time acquisition of the object shape, respectively. For solving these problems, we propose two new object transportation techniques by considering the properties of objects.

First, this dissertation presents a multi-agent behavior to cooperatively transport an active object using a sound signal and interactive communication. We first developed a sound localization method, which estimates the sound source from an active object by using three microphone sensors. Next, since the active object cannot be recalled by only a single robot, the robots organized a heterogeneous team by themselves with a pusher, a puller, and a supervisor. This self-organized team succeeded in moving the active object to

a goal using the cooperation of its neighboring robots and interactive communication between the object and robots.

Second, this dissertation presents a new cooperative passive object transportation technique using cyclic shift motion. The proposed technique does not need to consider the shape or the pose of objects, and equipped tools are also unnecessary for object transportation. Multiple robots create a parallel row formation using a virtual electric dipole field and then push multiple objects into the formation. This parallel row is extended to the goal using cyclic motion by the robots. The above processes are decentralized and activated based on the finite state machine of each robot. Simulations and practical experiments are presented to verify the proposed techniques.

Key words: Object transportation, Sound signal, Cyclic shift, Finite state machine, Virtual electric dipole field

Student Number: 2011-30243

Contents

List of Figures

List of Tables

Nomenclature

Notation	Description	Chapter
$\mathbf{p}_{GOAL}$	The goal point of object transportation	2
$\mathbf{p}^{O_i}$	The position of i^{th} object	2
r_i	The approximated radius of the i^{th} object O_i	2
θ_{err}	An included angle between the sound vector and the direction vector of puller	2
$s_{\max}$	The maximum sensing range	2, 4
$\mathcal{A}_S$, $\mathcal{A}_{PS}$, $\mathcal{A}_{PL}$	Supervisor, pusher and puller robot	3
$\mathcal{A}_O$	Active object	3
$\mathbf{p}_{PL}$, $\mathbf{p}_{PS}$, $\mathbf{p}_O$	The position of puller, pusher robot and active object	3
$\mathbf{f}_{PL}$, $\mathbf{q}_{PL}$	The direction and sound vector of puller robot	3
$s(t)$	The sound source signal	3
$h_k(t)$	The sound signal received by the microphone located in position k	3
d_k	Distance from k^{th} microphone to the sound source	3
$\zeta_k(t)$	The additive Gaussian noise at time k	3

E_k	The energy received by the k^{th} microphone	3
T	The time period of the sound source	3
c_Δ	The difference in sound level between two microphones on the left and right	3
$r_{SC}(\mathbf{x_1}, \mathbf{x_2})$	The radius of sound circle between $\mathbf{x_1}$ and $\mathbf{x_2}$	3
ς	The constant of sound circle	3
ϕ_i	Angle between robot i and the active object's center	3
θ_i	Heading angle of robot i	3
$\mathbf{R}, \mathbf{T}$	Rotational and transitional matrix	3
r	Radius of robot	3, B
${}^B\mathbf{u}_i$, ${}^A\mathbf{u}_i$	Initial and final posture coordinates	3
d_x, d_y	x and y coordinates of the direction vectors from the present point to the goal point	3
s_i	The distance measured at the i^{th} proximity sensor	3, 4, B
$\mathbf{S}_O^R$	Two proximity sensors which are located at ${}^A\mathbf{u}_i$ $(i = 0,\ 3)$	3
$\mathbf{S}_O^T$	Proximity sensor arrays	3
$\mathbf{S}_i^k$	The contact sensor set of robot i at state k	3
d_O^i	The distance between an active object and a robot	3
$\mathbf{D}(\mathbf{S}_O, \mathbf{S}_i)$	The output distances between proximity sensors belonging to $\mathbf{S}_O$ and $\mathbf{S}_i$	3
$\mathbf{S}_O^D$	The exclusive sensor's set at the measuring distance	3

$\mathbf{F}_e$	Virtual electric dipole function	4
$\tilde{\mathbf{F}}_e$	The modified virtual electric dipole function	4
$\mathbf{F}_e^{rot}$	The rotational electric dipole field function	4
p_c, γ_c	The current position and heading angle of the robot,	4
v_c, ω_c	Tangential and angular velocity of the robot,	4
a_c, α_c	Tangential and angular acceleration	4
$a_{\max}$, $\alpha_{\max}$	The maximum tangential and angular acceleration	4
p_t, γ_t	Target position and heading angle	4
$f_{x_c}^{rot}$, $f_{y_c}^{rot}$	Elements of function $\mathbf{F}_e^{rot}$	4
e_x, e_y, e_γ	The tangential, lateral and angular path error	4
ΔT	The sampling time interval	4
v_c^d, ω_c^d	The desired tangential and angular velocity	4
$\lambda(\psi)$	Index function of ψ-direction	4
N_{proxi}	The number of proximity sensors	4
$s_{desired}$	Desired distance	4
$\mathbf{p}_{head}^G$	The position of the head robot	4
$\mathbf{p}_{front\_head}^G$	The position of ahead of the head robot	4
ξ	The rotational angle of curved path	4
d_ε	The marginal distance between guider robots in the non-curved path	4

N_G	The number of guider robots	4
$N_{\min}$	The minimum necessary number of robots for caging	5
$dist_{\min}(obj)$	The minimum distance between the boundary points of an object	5
r_i^R	The radius of the i^{th} robot	5
r_{cage}	The virtual radius of caging formation	5
$F_k^r(x,y)$	The magnitude of the repulsive force with respect to the k^{th} robot	A
θ_k	The angle of the repulsive force with respect to the k^{th} robot	A
$f_k^r(x,y)$	The repulsive force with respect to the k^{th} robot	A
$f_{i,rep}^r(x,y)$	Total repulsive force of the i^{th} robot	A
$f_{i,rep}^O(x,y)$	The repulsive force of the i^{th} robot with respect to all objects	A
u_i^g, u_i^l, u_i^p	The controllers of the i^{th} guider, leader and pusher robots	A
$d_{\min}^o$, $d_{\max}^o$	The diameters of the minimum and maximum object	A
$A_{(i,j)}^G$	The state of j^{th} guider robot in i^{th} row	B

Chapter 1

Introduction

1.1 Background and Motivation

Object transportation is a typical problem in the field of robot applications [1, 2]. At first, many researchers have studied object transportation techniques using a single robot by grasping or pushing behavior [3]-[8]. However, there were several problems when a single robot was used for object transportation. First, a single robot cannot easily manipulate a large or heavy object. If an object is large, a robot cannot grasp the object by a manipulator or the object can be broken away from the desired path when pushing behavior is applied. Therefore, the complicated calculations considering geometric shapes need to keep the desired transportation path when a single robot is used [9]. Moreover, the robot cannot manipulate a heavy object by its own pushing power [10]. Second, a multi-object transportation is impossible by a single robot because

the robot cannot grasp or push multiple objects at the same time. Third, the object transportation will be failed when a robot is out of order because only single robot is used. In other words, the transportation system has no redundancy when a single robot is used for object transportation. Finally, a single robot should have greater capability than when multiple robots are used, because the single robot should be required many functions for object transportation, such as sensing ability, large power, and precise control. Multiple robots, however, can divide these diverse functions between the robots.

For solving the above problems, diverse cooperative object transportation techniques have been presented. They are divided into the four major categories of *grasping*, *pushing*, *caging*, and *tool-using*, as shown in Fig. 1.1. In the grasping technique, multiple robots grasp an object using their equipped manipulators and transport it to a goal [11]-[13]. In the pushing technique, multiple robots push an object using robot bodies. The object's pose is controlled using cooperative interaction between robots [14, 15]. The caging technique is based on object-wrapping; multiple robots approach, surround, and transport an object to the goal [16]-[19]. The tool-using technique is to use various tools for object transportation. For example, some researchers used a rope or a cable for cooperative object transportation in 2-D environment [20]-[23] and 3-D environment [24]. Another researcher used a stick [25, 26] or a pre-attached rope for object transportation [27].

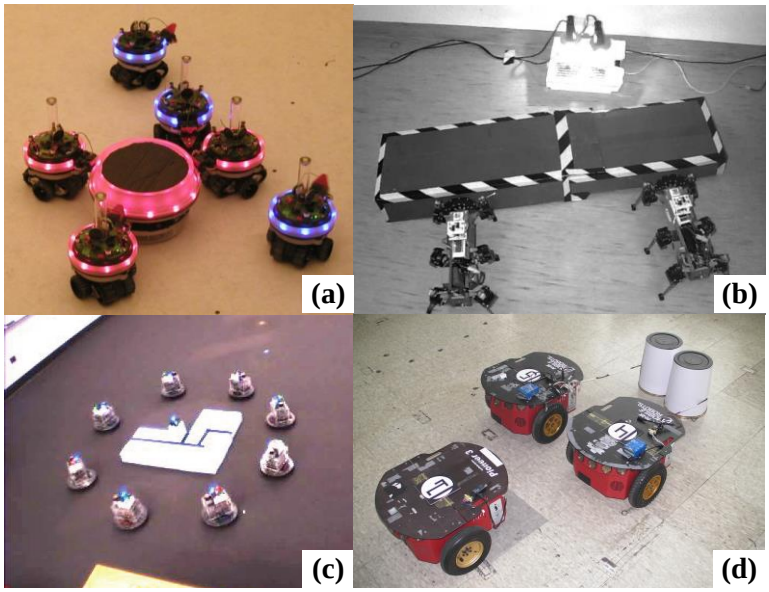

Figure 1.1 Diverse cooperative object transportation techniques. (a) Grasping [13] (b) Pushing [14] (c) Caging [16] (d) Tool-using [22]

Each of the above listed studies has both advantages and disadvantages. For example, the grasping technique guarantees stable transportation if robots succeed to grasp the object. However, this technique requires preliminary action such as gripping. On the contrary, there is no need to the preliminary action but stability is not guaranteed in the pushing technique. The caging technique enables objects to be robustly transported without grasping action but did not consider multi-object transportation. Multiple objects, however, can be transported at the same time if a rope or cable is used. Therefore, new object transportation techniques needs for manipulating multiple objects at the same time without any tools.

1.2 Related Work

1.2.1 The Categories of Object Transportation Techniques

As previously mentioned, cooperative object transportation techniques are divided into the four major categories of grasping, pushing, caging, and tool-using. While the grasping technique was originally developed as an efficient gripping method using the fingers of robot [28, 29], this technique has gradually been applied to object transportation using mobile robots. Multiple robots grasp an object using manipulators and transport it to the goal [11]-[13][30]. This technique has an advantage in that the success is guaranteed if the object is robustly grasped by the robots. However, the gripping condition, which refers to whether the robots can grasp the object using manipulators, should be examined in advance [31]. Additionally, the gripping of a large object is impossible when the prehensile size of the manipulator is smaller than the object.

The second object transportation technique is pushing, in which robots manipulate an object by pushing it without manipulators [14, 15, 32]. In contrast to the grasping technique, a preliminary grasping action is unnecessary and large-object transportation is possible in the pushing technique [33]. The change of robots' positions is also simpler than it is in the grasping technique because the robots can move around freely. The motion of the object, however, strongly depends on the points at which the force is applied; the object can be easily pushed out of control by the subtle pushing of

the robots [15]. In addition, it is necessary to consider the friction of the ground and the weight of the object for precise control, which complicates the object transportation task. Although some researchers have introduced a watcher or a leader who orders robots into action from a global viewpoint, it is still impossible to guarantee successful transportation even with this method due to the restricted viewpoint and the erratic motion of the object [34, 35].

Caging is the third object transportation technique based on object-wrapping. When using this technique, multiple robots approach, surround, and transport an object while maintaining object-wrapping without tools [16]-[19]. The object can be transported successfully without an escape using the minimum number of robots [36] and it is no need to consider the motion of objects. However, the caging condition should be checked iteratively during the transportation process to guarantee that the object does not escape. The shape and the vertices of an object should be known in real time to ensure a proper caging technique. In addition, multi-object transportation is not theoretically considered when using this technique. Some researchers studied various caging-like object transportation techniques [37]-[39] but they are also required the information of objects' shape like the caging.

Finally, there are other transportation techniques using various tools in addition to those listed above. Some researchers have used a rope or a cable for cooperative object transportation in 2-D environment [20]-[23] and 3-D environment [24]. In this technique, robots wrap and pull multiple objects

using a rope to manipulate multiple objects at the same time. The robots, however, have restricted motion because they are tightly coupled with the rope. Other researchers investigated the object transportation problem using a stick [25, 26]. With this technique, robots push or tumble over an object using a stick to facilitate transportation, which enables the robots to manipulate a large object to the goal. Multi-object transportation, however, is not considered to be feasible with this technique like the caging.

Above listed four techniques can be evaluated according to the following considerations, and the comparison results are summarized in Table 1.1.

- Successful object transportation
  - Are all objects successfully manipulated to the goal?
- Multi-object transportation
  - Is it possible to manipulate multiple objects at the same time?
- Rolling-object transportation
  - Is it possible to manipulate rolling objects, such as balls?
- The shapes of objects
  - Is it possible to manipulate irregular or unknown shaped objects?
- Whether tools are needed or not
  - Does it require additional tools for object transportation?
- The minimum number of robots
  - What is the minimum number of robots for object transportation?

TABLE 1.1

COMPARISON BETWEEN OBJECT TRANSPORTATION TECHNIQUES

	Grasping	Pushing	Caging	Tool-using
Related work	[11]-[13]	[14, 15] [34]	[16]-[19]	[20]-[26]
Multi-object transportation	Impossible	Impossible	Impossible	Possible
Rolling-object transportation	Impossible	Impossible	(partially) Possible	Impossible
The consideration of objects' shape	Yes	(partially) Yes	Yes	No
The requirement of additional tools	Yes	No	No	Yes
The minimum number of necessary robots	1	1	3	2

1.2.2 Sound Localization Techniques for Active Object Transportation

In the active object transportation technique, robots use sound signal for localization. There are two representative approaches to identify the position of a sound source using the physical differences in a sound signal: *interaural time difference* (ITD) and *interaural level difference* (ILD). The ITD is the time delay of the sound at different observation points. We can know a sound source's position from the difference between arrival times. This approach to position identification is well used in human-localization [40] because it is more accurate and robust in voice frequency, which is closely related to time delay. The ILD is caused by sound attenuation as the distance travelled

increases. According to the *inverse-square law* [41], the decrease in sound level, which is propagated by the point source, is inversely proportional to the total distance travelled in the free field. Due to the difference between the observation points, the sound levels of microphones differ from each other. This difference is used as a clue to identify the sound source [42, 43]. Although this cue has been known as the oldest theory of sound localization, the ILD localization has been rarely utilized in computer-based system. Recently, however, Birchfield and Gangishetty showed the possibility of using the ILD for sound localization [44]. They used two microphones for detecting sound signal and succeeded in localization accurately using only the sound level difference. Other studies showed joint evaluation between ITD and ILD [45, 46] for sound localization. The only one information is insufficient to localize a sound source, and thus, they combined the ITD with the ILD for accuracy.

1.3 Contributions

This dissertation presents two new object transportations. The first is *active object transportation* using sound-based localization and interactive communication. Unlike most of the previous studies [34], [47]-[49], robots can use partial abilities of the active object to accomplish the transportation. For example, the active object can emit a sound signal using its speaker to induce the robots to approach it without the help from a global positioning

system (GPS). Moreover, proximity sensing data from the active object are used to approach the assigned position around the active object. The robots can communicate and share the sensing information with the active object.

Second, we present a new *passive object transportation* technique which uses cyclic shift motion. In the proposed technique, some robots create parallel row formation by lining up in two rows around multiple objects, which are called guider robots. The guider robots prevent the objects from escaping using their bodies and extend the parallel row using cyclic shift motion in which the guider robots move from the last line to the first line one by one. A leader robot leads the robot team in front and a pusher robot pushes the objects toward the goal from behind until all objects arrive at the goal. The proposed technique has four advantages for object transportation as compared with the previous techniques. First, there is no need to grasp objects. A large object, which is large to grasp using a manipulator, can be transported. Second, additional tools, such as a rope or a stick, are not necessary for object transportation. The objects are manipulated by only pushing force which is applied by robot bodies. Third, multi-object transportation is possible because the proposed formation always maintains an enclosed shape to prevent the objects from escaping. The objects are wrapped by the robots while they are manipulated to the goal, which enables multiple objects to be transported at the same time. Rolling objects, such as balls, can be also transported by the same reason with the multi-object transportation. Finally, the geometric

information of the objects is unnecessary. In the grasping and caging techniques, robots should know the detailed geometric information of objects, such as the positions of vertices, the size, and the rotational angle of objects. However, the preliminary information of objects' shape is not necessary in the proposed technique. Only the approximated sizes of objects are required for object transportation, which is used to determine the necessary number of robots for object transportation.

1.4 Organization

This dissertation is organized as follows. In Chapter 2, the object transportation problem is described. The definitions of active and passive object are also presented in this section. Chapter 3 describes the active object transportation technique using a sound signal and interactive communication. The passive object transportation technique using cyclic motion is presented in Chapter 4. Chapter 5 and Chapter 6 show the simulation results and practical experiments of the proposed object transportation techniques, respectively. The discussion and conclusions of this dissertation are presented in Chapter 7 and Chapter 8, respectively.

Chapter 2

Object Transportation Problem

In this chapter, we address object transportation problem. First, the definitions of passive and active object are presented in Section 2.1. Then, the problem formation of object transportation is presented in Section 2.2. The assumptions of object transportation techniques are presented in Section 2.3.

2.1 Passive Object versus Active Object

We divide objects into two classes and define new terminologies according to their characteristics: *passive* and *active object*. The passive object denotes a common object which does not have any functions to interact with robots, such as a box and ball. It has no ability to communication, no sensing, no signal generation, and no mobility. Most of previous studies of object transportation assumed that an object was passive, and thus, robots

should know the shape and orientation of object by their own sensing abilities [15]-[19][34, 37, 48].

However, the active object has sensing and communication abilities unlike the passive object. It has abilities to communicate other robots, to detect other robots, and to produce specified signal. These abilities help the robots manipulate the object by interactive communication between the active object and robots. But, the active object has no mobility like the passive object. A few studies dealt with the active object because this concept was not presented before. Groß et al. introduced object transportation technique using LED signal [13]. Christensen et al. presented fault detection system using the synchronized flashing behavior between robots [50]. The active object can be considered in a special case, such as an immobile robot because of the fault.

Table 2.1 shows the comparison between the passive and active object according to their abilities. They have no mobility in common, but the active object has the abilities of communication, sensing, and signal generation.

TABLE 2.1

COMPARISON BETWEEN PASSIVE AND ACTIVE OBJECT

	Passive object	Active object
Communication with robots	No	Yes
Sensing ability	No	Yes
External signal generation	No	Yes
Mobility	No	No
Examples	Ball, Box	Faulty robot

2.2 Problem Formulation

The object transportation process is said to be a success if all objects are transported to the goal by a multi-robot team. The success of transportation is denoted as:

$$\| \mathbf{p}_{GOAL} - \mathbf{p}^{o_i} \| \leq \delta \quad \text{for} \quad \forall i \,, \tag{2.1}$$

where $\mathbf{p}_{GOAL} \in \mathbf{R}^2$ is the goal point, $\mathbf{p}^{o_i}$ is the position of i^{th} object, and δ is the radius of the goal region. This formulation is the same regardless of the passive or active object. The solutions of the object transportation problem will be presented as the forms of algorithms and robot controllers.

2.3 Assumptions

To solve the problem, the following assumptions are made.

First, all robots move in a two-dimensional plane, and the robots are modeled as circles. Various geometric primitives can be used for robot modeling instead of the circular model. However, we restricted the robot model so that it had a symmetric and simple shape.

Second, we assume that the roles of each robot are predefined as guider, pusher, puller, or leader before the transportation. The guider robots are assumed to be not moved by pushing force. The pusher robot has sufficient power to push multiple objects. The leader robot has the global path planner in advance, and thus, the leader robot can lead the multi-robot team to the goal.

Third, multiple objects are assumed to be gathered together before passive

object transportation such that

$$\| \mathrm{p}^{O_{i+1}} - \mathrm{p}^{O_i} \| < r_{i+1} + r_i + \varepsilon \quad \text{for} \quad \forall i, \tag{2.2}$$

where $\mathbf{p}^{O_i} \in \mathbf{R}^2$ and r_i are the position and the approximated radius of the i^{th} object O_i, respectively. The coefficient ε is a marginal constant between objects. The index i is sorted by the relative distance between objects. For example, object O_{i+1} is the closest object to object O_i among all objects. The approximated radius is the maximum length of a diagonal line of arbitrary objects.

Fourth, a symmetrical robot formation is assumed to be prearranged before passive object transportation. We do not consider an approaching method toward the objects so as to concentrate on the transportation process. However, the approaching method to the objects can be designed using other techniques, such as the potential field method [51] if the positions of objects are shared with all robots. The example of approaching method is presented in Appendix A.

Finally, we assumed that the sizes and number of objects are known in advance to determine the number of robots used in object transportation. Also, data association is assumed to be possible within a limited sensing range $s_{\max}$ by intercommunication.

Chapter 3

Active Object Transportation using a Sound Signal and Interactive Communication

In this chapter, we present an active object transportation technique. The active object has three following characteristics. First, the active object equipped with proximity sensors for detecting other robots. Second, the active object has a speaker to distress a sound signal, which helps robots to find the position of the active object. Finally, the active object cannot move by itself. For satisfying the above characteristics, we choose an immobile robot as an active object. This immobile robot has the normal abilities of robot, such as sensing, communication, and signal production, except mobility. This technique is the extended version of [35] and [52]. Additional analysis and explanations are presented as comparison with the previous versions.

3.1 Overview of Active Object Transportation

The active object transportation process is divided into three steps. The first step is position identification by generating a sound signal to identify the position of the active object. The second step is position assignment of the robots for pushing and pulling of the active object. The third step is the transportation of the active object through the cooperative behaviors between pusher and puller robots.

To manipulate the active object, a multi-robot team should be organized appropriately. The coordination model of the multi-robot team is inspired by how people carry large objects such as furniture. One person watches the workspace at a short distance from the object, and other people push or pull the object in accordance with that person's directions. This coordination is efficient to correct direction error and induce a goal. Therefore, we divided the robots into three roles to imitate human behavior: *supervisor*, *pusher*, and *puller*, as shown in Fig. 3.1. Each robot has different abilities according to its role. A supervisor robot ($\mathcal{A}_S$) emits a sound signal with a speaker to induce the active object to proceed to the goal. In addition, the supervisor watches all transportation processes and transmits screen data acquired from a wireless camera to the user. This is done so that the user can take appropriate action in the event of an unexpected situation. A pusher robot ($\mathcal{A}_{PS}$) pushes the active object from behind. We assumed that the pusher robot cannot move the active object alone due to a lack of power. Thus, we added a puller robot ($\mathcal{A}_{PL}$) and

which pulls the active object. The puller robot detects the sound signal in front of the active object, and it leads the robot team to the goal. Finally, the multi-robot team is made up of a supervisor, a pusher in the back, and a puller in the front utilizing the minimum number of robots necessary to perform a successful transportation.

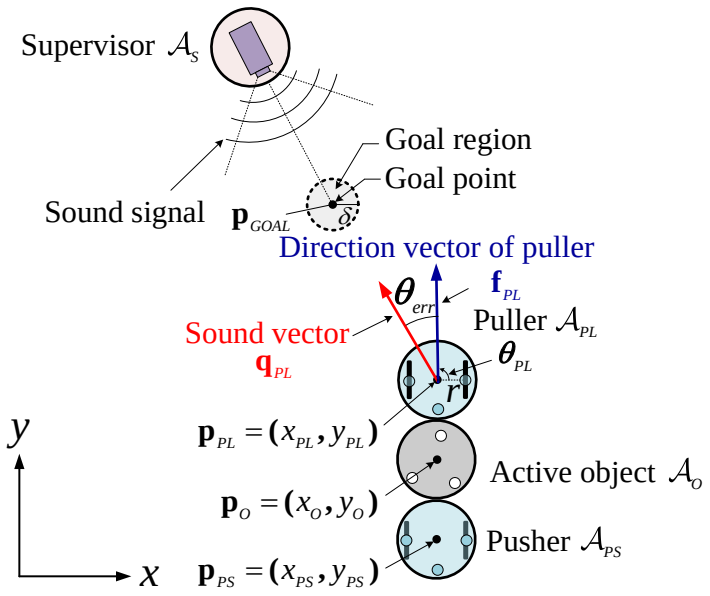

Figure 3.1 Graphical illustration of active object transportation

3.2 Sound Vector Generation using Triple Microphones

The positions of active object and goal are localized by a sound signal in the proposed technique. Thus, we describe the formulation of the *sound vector* (SV), which consists of a unit force and the direction of sound source,

by analyzing the sound signal. Determining the direction of the SV starts by acquiring candidate regions of a sound source using the ILD localization scheme proposed by Birchfield [44]. Two microphones on the robot's left and right are used to generate a *sound isocontour* (SI), and a microphone in the middle of the robot is used to generate a *sound circle* (SC) by the inverse-square law. Finally, we can generate the SV by combining the SI and SC. The magnitude of the SV is assumed as a unit vector for a convenient comparison with the formation vector. These processes for the generation of the SV are briefly presented in Table 3.1.

TABLE 3.1 THE METHODS OF THE SOUND VECTOR GENERATION

Purpose	Methods
Sound isocontour generation	Uses ILD proposed by Birchfield [44].
Sound circle generation	Uses inverse-square law [41].
Sound vector generation	Uses the intersection region of the SI and SC.

3.2.1 Sound Isocontour Generation using ILD

Suppose we have N microphones, then $s(t)$ is the sound signal; $h_k(t)$ is the sound signal received by the microphone located in position k. The measured sound signal can be modeled as

$$h_k(t) = \frac{s(t)}{d_k} + \zeta_k(t) \quad \text{for} \quad k = 1 \text{ to } N, \tag{3.1}$$

where d_k is the distance from the microphone located in position k to the

sound source, and $\zeta_k(t)$ is the additive Gaussian noise. An index k of each microphone is denominated according to its relative position. For example, 'L' and 'R' indicates left and right microphones in the robot, respectively.

Using (3.1), the energy received by the k^{th} microphone can be calculated as follows:

$$\mathrm{E}_k = \int_0^{\mathrm{T}} h_k^2(t)dt = \frac{1}{d_k^2}\int_0^{\mathrm{T}} s^2(t)dt + \int_0^{\mathrm{T}} \zeta_k^2(t)dt\,, \tag{3.2}$$

where T is the time period of the sound source.

To simplify the equation, let us assume that noise is ignored ($\zeta_k(t)=0$). If the coordinate of the sound source is (x,y), we can estimate the position of the sound source as follows:

$$\left(x-\frac{c_x}{c_\Delta}\right)^2 + \left(y-\frac{c_y}{c_\Delta}\right)^2 = \frac{16r^4\,\mathrm{E}_L\,\mathrm{E}_R}{c_\Delta^2}\,, \tag{3.3}$$

$$\begin{pmatrix} c_x = \mathrm{E}_L(x_i - r\sin\theta_i) - \mathrm{E}_R(x_i + r\sin\theta_i) \\ c_y = \mathrm{E}_L(y_i + r\cos\theta_i) - \mathrm{E}_R(y_i - r\cos\theta_i) \\ c_\Delta = \mathrm{E}_L - \mathrm{E}_R \end{pmatrix},$$

where the index i indicates the type of robots, r is the radius of robots and θ_i is the heading angle of robot i.

Equation (3.3) can be also written as (3.4) if the distance from the sound source to the left and right microphones is the same as $\mathrm{E}_L = \mathrm{E}_R$:

$$2c_x x + 2c_y y = c\,, \tag{3.4}$$

where $c = (\mathrm{E}_L - \mathrm{E}_R)(x_i^2 + y_i^2 + r_i^2 - 2rx_i \sin\theta_i + 2ry_i \cos\theta_i)$.

As shown in Fig. 3.2, we can generate a 3D-isocontour by (3.3) and (3.4) with the variation of energy level. The sound level difference is identical in the same height of the isocontour. From the SI, we can determine the possible location of sound source using the measured sound level difference from two microphones.

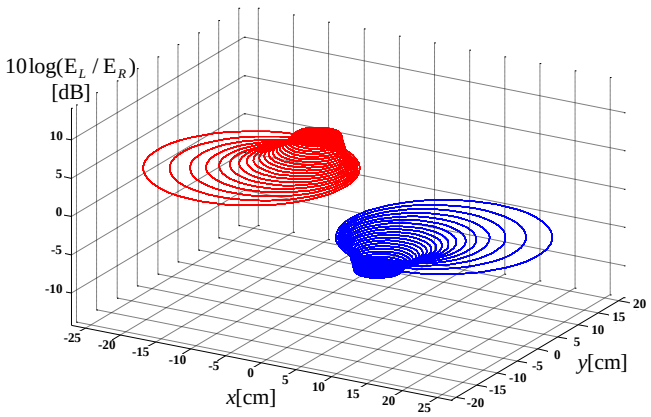

Figure 3.2 Three dimensional sound isocontour. The isocontour is drawn differently according to the ratio of the sound level.

In (3.3), c_Δ is the difference in sound level between two microphones on the left and right. If c_Δ is bigger, the radius of SI is smaller. It means that the estimated region of the sound source decreased. As a result, we can estimate the location of the sound source more accurately when sound level difference is large.

3.2.2 Sound Circle Generation using Inverse-square Law

We can roughly estimate the position of the sound source using the SI. The estimated position, however, is not reliable because the region of the SI is very wide. Thus, additional information is required to find the direction of the sound source precisely.

We use the characteristic of sound attenuation to acquire the additional information. As the sound propagates, it is attenuated by a factor inversely proportional to the distance travelled. This is called the inverse-square law of the attenuation. Applying this law, we can estimate the distance from the sound source to the measurement point. A new function $r_{SC}(\mathbf{x_1}, \mathbf{x_2})$ can be calculated by (3.5):

$$r_{SC}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\varsigma}{\|\mathbf{x_1}\text{-}\mathbf{x_2}\|} \quad (\mathbf{x_1} \neq \mathbf{x_2}), \tag{3.5}$$

where $\mathbf{x_1}$ and $\mathbf{x_2}$ are the measurement point and sound source point, respectively, and ς is an empirically determined proportionality constant. In (3.5), $\mathbf{x_1}$ and $\mathbf{x_2}$ are not the same because a robot does not collide with the active object which emits a sound signal. Using (3.5), we can define a set from the point to the same distance: sound circle (SC). The center point of SC is the measurement point and the radius of SC is the inverse distance between the sound source and measurement point. The SC is given by:

$$(x - x_i)^2 + (y - y_i)^2 = \{r_{SC}(\mathbf{x}_i, \mathbf{x}_O)\}^2, \tag{3.6}$$

where (x_i, y_i) is coordinate of pusher or puller robot.

3.2.3 Sound Vector Generation

So far, we derived the SI using ILD with two microphones and the SC using the inverse-square law with one microphone, as shown in Fig. 3.3(a) and (b), respectively. Combining with these two circles, we can get the two intersection points of the circles: (x_{SV}, y_{SV}). These points can possibly locate the sound source, as shown in Fig. 3.3(b) and be calculated by Pythagoras theorem applying to (3.3) and (3.6):

$$\begin{aligned} x_{SV} &= x_i + r_{SC}\cos\left(\tan^{-1}\frac{Y}{X} \pm \cos^{-1}\left(\frac{{r_{SC}}^2 - {r_{SI}}^2 + D^2}{2r_{SC}D}\right)\right) \\ y_{SV} &= y_i + r_{SC}\sin\left(\tan^{-1}\frac{Y}{X} \pm \cos^{-1}\left(\frac{{r_{SC}}^2 - {r_{SI}}^2 + D^2}{2r_{SC}D}\right)\right) \end{aligned}, \qquad (3.7)$$

where $X = \frac{c_x}{c_\Delta} - x_i$, $Y = \frac{c_y}{c_\Delta} - y_i$, $r_{SC} = r_{SC}(\mathbf{x}_i, \mathbf{x}_O)$, $r_{SI} = \frac{4r^2\sqrt{\mathrm{E}_L\,\mathrm{E}_R}}{c_\Delta}$, and $D = \sqrt{X^2 + Y^2}$.

However, the accuracy of the points is low because sound levels are constantly changing. One measured sound level is not reliable because noise and reverberation can be mixed with the direct sound during transmission. We, therefore, used sound data from five measurements to find the intersection precisely. The number of five is determined by considering the time delay of movement and the accurate estimation of sound source. Consequently, the intersection is not a point but regions, which are similar to a circular ring. The minimum calculated radius is the inner radius of the ring and the maximum

radius is the outer radius of ring, as shown in Fig. 3.3(c). These regions are denominated as the *Candidate Region* (CR).

If the ratio of the distance from the sound source to the two microphones is the same, we cannot know whether the CR is in front or not. This confusion is called the *front-back confusion* [53], which occurs because of an identical difference in the sound level. To eliminate this confusion, the sound level on the middle microphone (E_M) is compared with the average sound level between the left (E_L) and right (E_R) microphones. If the average of the sound levels at the left and right microphones is bigger than the sound level at the middle microphone, then the sound source is in front, as described in (3.8). If it is the other way, then sound source is in back. We can reduce to a half the region through (3.8) compared to the original CR. This reduction of the CR can be also described by θ_{err}; which is an included angle between the sound vector and the direction vector of puller. If the θ_{err} is in quadrant I, IV, then the sound source is located in front of measurement point. Otherwise, it means θ_{err} is in quadrant II, III, the sound source is located in back of measurement point. The direction of sound source is determined by given equation:

$$\textit{direction of sound source} = \begin{cases} \textit{forward} \quad \textit{if} \ \mathrm{E}_M \le \dfrac{\mathrm{E}_L + \mathrm{E}_R}{2} \ \textit{or} \ -90° \le \theta_{err} \le 90° \\ \textit{backward} \quad \textit{if} \ \mathrm{E}_M > \dfrac{\mathrm{E}_L + \mathrm{E}_R}{2} \ \textit{or} \ 90° < \theta_{err} < 270° \end{cases}, \qquad (3.8)$$

where $\theta_{err} = \tan^{-1}\dfrac{Y}{X} \pm \cos^{-1}\left(\dfrac{{r_{SC}}^2 - {r_{SI}}^2 + D^2}{2 r_{SC} D}\right)$.

Consequently, we can estimate the location of the sound source; it is defined as the sound vector (SV), as shown in Fig. 3.3(d).

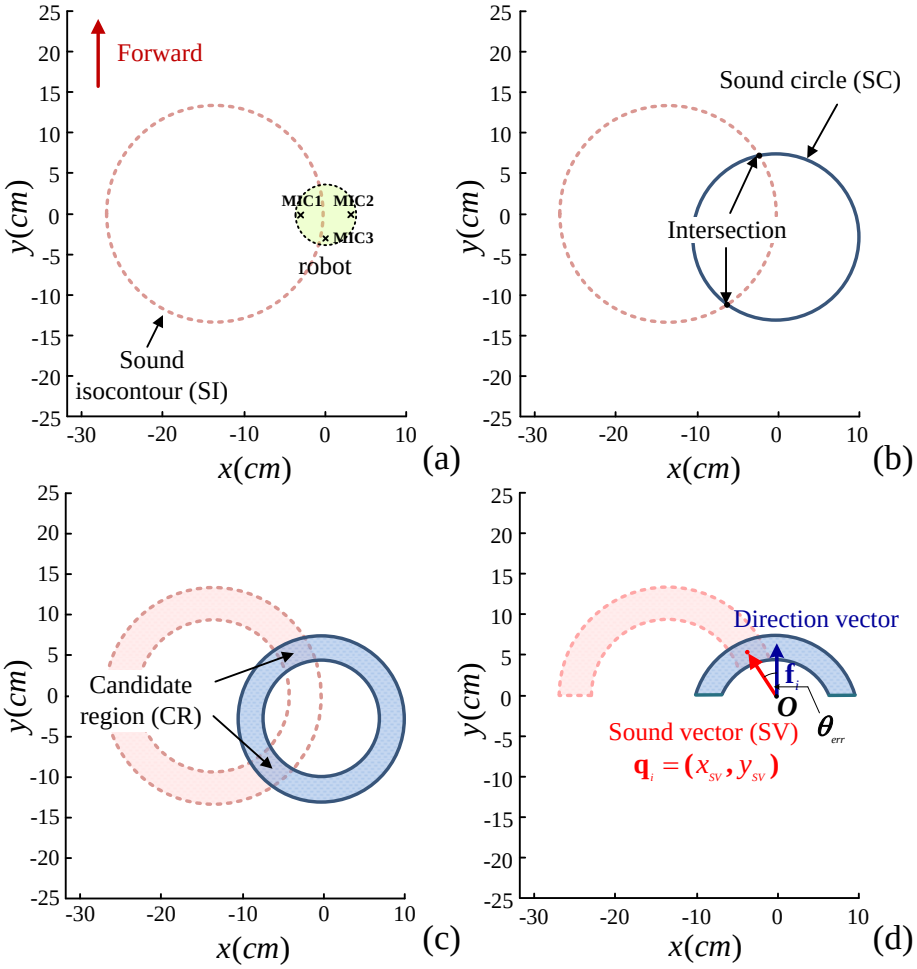

Figure 3.3 Process for the generation of the SV. The center of robot is the origin of Cartesian coordinate. (a) The generation of SI using ILD with two microphones. (b) The generation of SC using inverse-square law with one microphone. (c) The generation of CR by combining with the SI and SC. (d) The generation of SV. The front-back confusion is eliminated by comparison with triple microphones' relative sound level.

3.3 Cooperative Control Method using Interactive Communication

This section presents cooperative control method for active object transportation using the sound localization and interactive communication with robots. First, the role assignment of the multi-robot team is introduced in Section 3.3.1. Second, the positioning method of robots is presented in Section 3.3.2. Finally, the active object transportation process will be presented in Section 3.3.3.

3.3.1 Role Assignment of Multi-robot Team

The proposed multi-robot team is heterogeneous, and each robot thus has different capabilities according to their roles. For example, the supervisor robot equipped with the wireless camera to transmit the accident scene to the user. It has also a speaker to emit sound signals. The puller and the pusher robots have a manipulator with which to grasp the object, and the puller robot detects sound signals from the active object or from the supervisor robot using three microphones. Therefore, the roles of robots are predetermined by their capabilities before the transportation of an active object. The supervisor robot moves to the active object first and verifies that the scene is secured. Second, the puller moves to the active object and takes its position in front of the active object. Likewise, the pusher is positioned behind the active object.

3.3.2 Position Assignment of Multi-robot Team

Active object transportation is possible only when each robot is located in a position at which the robots can apply force to the active object. In addition, the heading directions of pusher and puller robots should nearly equal to maximize the applied force.

We defined *posture coordinates* to include two-dimensional coordinates and heading direction, as shown in Fig. 3.4. The posture coordinates is expressed as (3.9):

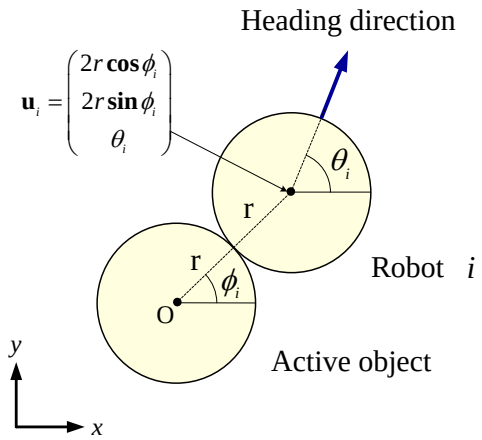

Figure 3.4 Posture coordinates of robot i

$$\mathbf{u}_i = \begin{bmatrix} \mathbf{p}_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} 2r\cos\phi \\ 2r\sin\phi_i \\ \theta_i \end{bmatrix}, \tag{3.9}$$

where ϕ_i is angle between robot i and the active object's center.

As already shown in Fig. 3.1, a puller and a pusher robot should be in front and behind of the active object, respectively, to complete the transportation. This position is defined as the *final posture coordinates* ${}^{A}\mathbf{u}_i$, which is expressed as the multiplication of transitional matrix and rotation matrix:

$$ {}^{A}\mathbf{u}_i = \mathbf{R}(\theta_i)\mathbf{T}(\mathbf{p}_i,\mathbf{p}_O)\,{}^{B}\mathbf{u}_i \quad \text{for} \quad (i = PL, PS), \tag{3.10} $$

$$ \mathbf{R}(\theta_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \dfrac{\pi}{2\theta_i} \end{bmatrix}, \mathbf{T}(\mathbf{p}_i,\mathbf{p}_O) = \begin{bmatrix} 1 & 0 & \dfrac{d_x}{\theta_i} \\ 0 & 1 & \dfrac{d_y}{\theta_i} \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.11} $$

where $\mathbf{R}$ is the rotational matrix, $\mathbf{T}$ is the transitional matrix, ${}^{B}\mathbf{u}_i$ is the initial posture coordinates, and $\mathbf{p}_O$ is the position of the active object. The notations d_x and d_y are x and y coordinates of the active object with respect to the global coordinate, respectively. The indices 'PL' and 'PS' indicates the puller and pusher robot, respectively.

All robots should have a direction vector that goes straight when the transportation starts; the heading direction should be $\pi/2$. The puller and pusher are located at $\mathbf{p}_{PL} = (0, 2r)$ and $\mathbf{p}_{PS} = (0, -2r)$, respectively if the origin is the center of the active object, as shown in Fig. 3.5.

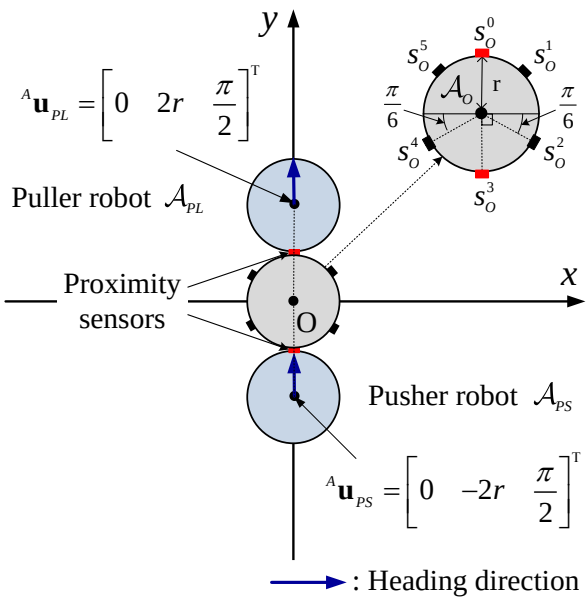

Figure 3.5 Position assignments of the robots. The puller robot should be in front of the active object, and the pusher robot should be behind the active object just before the multi-robot team escorts the active object.

For the positioning of puller and pusher robots, two proximity sensors located at ${}^{A}\mathbf{u}_i$ $(i = 0, 3)$ are defined as $\mathbf{S}_O^R = \{s_O^0, s_O^3\}$ in the sequence among the $\mathbf{S}_O^T = \{s_O^j \mid 0 \le j \le 5\}$ of $\mathcal{A}_O$ proximity sensor arrays, as shown in Fig. 3.5. The notation s_i^j indicates the proximity sensor of robot i at position j for $0 \le j \le 5$. The relative localization of robots is possible using the sensor information $\mathbf{S}_O^T$ in the close distance. These sensors have the role to make sure that all robots are in their assigned positions by measuring the relative distance to the active object.

3.3.3 Transportation Process of an Active Object

The object transportation is executed according to four commands, as shown in Table 3.2. First, if a sound signal is emitted by an active object, each robot approaches the object by the $Move-to-sound$ command, as shown in Fig. 3.6. As already described in Section 3.2, the robots generate a SV using the sound signal and follow this vector.

The distance between an active object and a robot is given by:

$$d_o^i = \min(\mathbf{D}(\bigcup_j s_o^j - \mathbf{S}_o^D, \bigcup_j s_i^j)) \quad \text{for} \quad (0 \le j \le 5), \tag{3.12}$$

where $\mathbf{D}(\mathbf{S}_o, \mathbf{S}_i)$ is the output distances between proximity sensors belonging to $\mathbf{S}_o$ and $\mathbf{S}_i$. The set $\mathbf{S}_o^D$ is the exclusive sensor's set at the measuring distance.

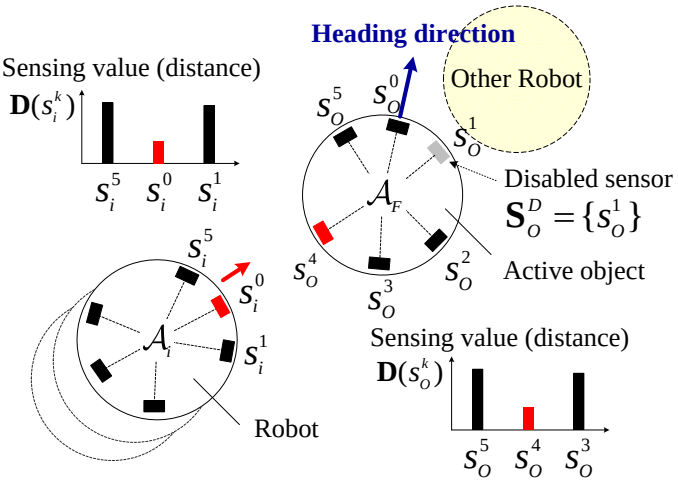

Figure 3.6 The approaching processes of the robots. This is just an example of approaching active object. The robots can approach the active object from any direction.

Second, if the robot succeeded in contacting the active object ($d_O^i \le \varepsilon$), it turns on the active object's border keeping a safe distance and stops at the assigned position satisfying (3.13). This process is called the $Follow-in-contact$. The active object cannot use the proximity sensor located at assigned position because the robot covers the sensor. Thus, this disable proximity sensor should be added to $\mathbf{S}_O^D$ after the robot's stop:

$$\arg\min(\mathbf{D}(\bigcup_j s_O^j - \mathbf{S}_O^D)) \in \mathbf{S}_O^R . \tag{3.13}$$

Third, the $Aligned-the-direction$ command carries out the robots spin to adjust the heading direction $\pi/2$ at the assigned position. This action needs to proceed to the goal by modifying the heading direction of robot. In the process, the robots check their proximity sensors whether sensing something or not. Each robot has a limited heading direction condition, as described in (3.14) and (3.15).

$$\mathcal{A}_{PL} : \min(\mathbf{D}(\bigcup_j s_{PL}^j)) > \varepsilon \ (j \ne 3) \text{ and } s_{PL}^3 \le \varepsilon , \tag{3.14}$$

$$\mathcal{A}_{PS} : \min(\mathbf{D}(\bigcup_j s_{PS}^j)) > \varepsilon \ (j \ne 0) \text{ and } s_{PS}^0 \le \varepsilon , \tag{3.15}$$

where ε is the constant that is used for 'sensing or not'. If the proximity sensor detects a distance which is smaller than ε , it means that a robot is present at the location.

Finally, if robots are ready to move, then $Move-to-goal$ command is carried out. The puller robot pulls the active object in front and adjusts its heading direction to correspond with sound signal direction. The pusher robot pushes the active object from behind for supplying force.

TABLE 3.2 COMMANDS OF THE MULTI-ROBOT TEAM

Command	Description
$Move-to-sound$	Robot moves to the active object emitting sound source
$Follow-in-contact$	Robot moves to the assigned position maintaining constant distance with the active object
$Aligned-the-direction$	Robot's heading direction is aligned to the goal
$Move-to-goal$	Robot team escorts active object to the goal

The entire process of the active object transportation is shown as Fig. 3.7. If the active object emits a sound signal, the supervisor robot approaches to the active object using the sound signal ($Move-to-sound$). If the supervisor is positioned at an appropriate location at which the user can cover the scene through the wireless camera, then the puller and pusher robots also move to the active object using the sound signal ($Move-to-sound$). The puller and pusher robots use proximity sensors to assign positions close to the active object ($Follow-in-contact$). The active object stops emitting sound signals after the robots are located in the desired positions, and the supervisor robot emits sound signals to induce the robot team. According to the sound signal of the supervisor robot and the sensor information of active object, the heading direction of robots aligns with the supervisor's position, which is the goal ($Aligned-the-direction$). When the alignment of the robots is completed, the robot team escorts the active object to the goal ($Move-to-goal$).

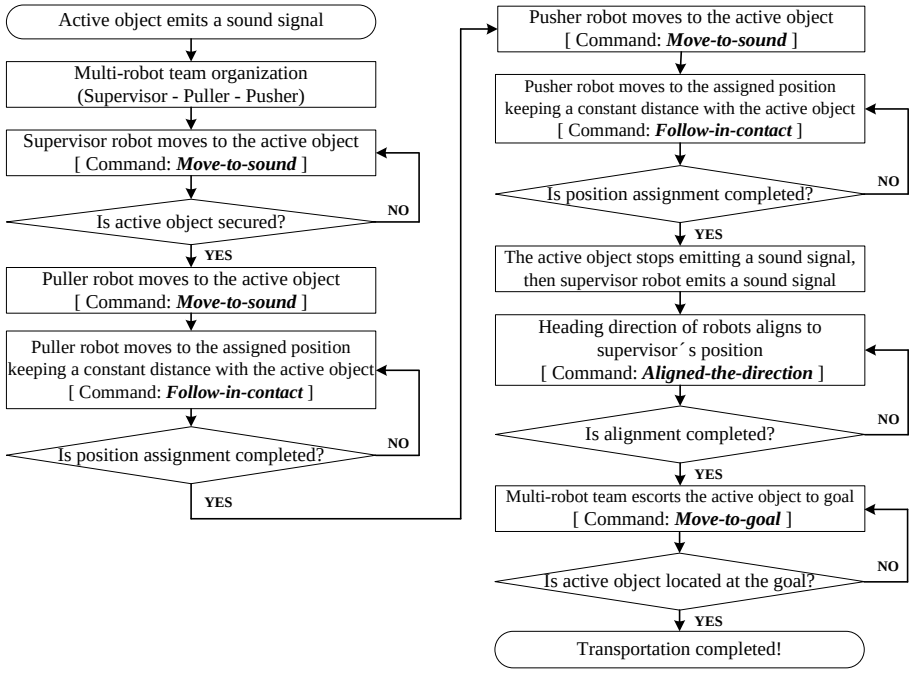

Figure 3.7 Object transportation process of an active object

Chapter 4

Passive Object Transportation using Cyclic Shift Motion

This chapter presents a passive object transportation technique using *cyclic shift motion*. Unlike an active object, a passive object cannot communicate with robots and recognize other robots. Therefore, we solve the passive object transportation problem by considering the cooperative behaviors of robots only. To address these issues, we present the overview of passive object transportation in Section 4.1. The roles of the proposed multi-robot team are presented in Section 4.2. In Section 4.3, we introduce the line formation which is a fundamental formation for the object transportation. Finally, we present a decentralized multi-object transportation technique using the line formation in Section 4.4.

4.1 Overview of Passive Object Transportation

A key idea of the passive object transportation is that it generates parallel row formation using multiple robots. This formation consists of two symmetrical row formations, and the rows are extended to the goal using cyclic shift motion, as shown in Fig. 4.1. The two rows on the right and left side of leader robot are defined as the first and the second row, respectively. The tail robots located in the last line in each row move from the last to first line; this is defined as a cyclic shift motion because it is analogous to the cyclic shift operator in combinatorial mathematics [54].

There are two problems associated with the cyclic shift motion, which are caused by kinematic and dynamic constraints. First, a tail robot should approach a head robot from a specific direction to line up in a row. To do this, the tail robot not only moves to the desired position, but it should also approach from a specific direction. Second, the tail robot should follow a smooth path without an abrupt turn or any path deviations. It is difficult to follow a desired path without the path deviation due to the dynamic constraint. These problems will be solved through following sections.

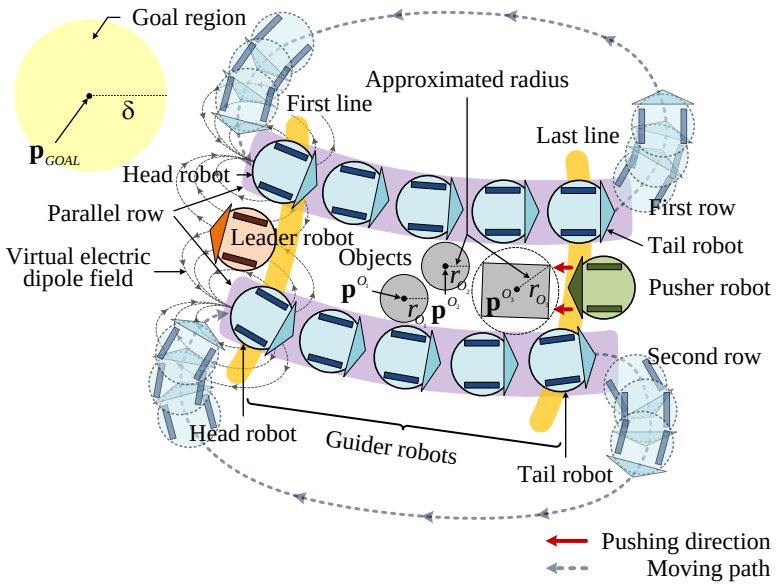

Figure 4.1 The proposed multi-robot team for passive object transportation. A leader robot leads the robot team by following a global path, and a pusher robot pushes the object from behind. Multiple guider robots create parallel row formation by lining up with two rows to prevent the objects from escaping. The guider robots make an extended parallel row to the goal by iteratively moving from the last to first line. More leader and pusher robots can be used for transportation in the case of multiple or large objects, but we illustrate only a leader and a pusher robot for simplicity in this figure.

4.2 Multi-robot Team Organization

The proposed multi-robot team consists of three roles: *guider*, *leader*, and *pusher*. The guider robots prevent objects from breaking away from the robot team by lining up in two rows around the objects. Some guider robots have other names according to their relative order; the guider robots in the first and last line are called a head and tail robot, respectively, as shown in Fig. 4.1. A

leader robot is located in front of the robot team and leads the robot team to the goal using the global path planner. In addition, the leader robot prevents the objects from escaping by blocking the head entrance of the proposed formation, which is the space between the head robots. It is possible for two or more leader robots to be used for object transportation when multiple or large objects are transported. A pusher robot which is located between the tail robots manipulates the objects by pushing them from behind. The pusher robot does not allow the escape of objects by blocking the back entrance in the same way used by the leader robot. Also, multiple pusher robots can be used according to the need. The names and the role descriptions of the proposed multi-robot team are summarized in Table 4.1.

TABLE 4.1

THE NAMES AND ROLE DESCRIPTIONS USED IN THE MULTI-ROBOT TEAM

Name	Role description	
Leader robot	1. located in front of the robot team between the head robots 2. has the global path planner for in advance 3. prevents the objects from escaping 4. transmits an approaching angle to the guider robot which is approaching to the head robot	
Pusher robot	1. located behind the robot team between the tail robots 2. pushes the objects to the goal using the robot body 3. prevents the objects from escaping	
Guider robot	Head robot	1. located in the first line of each row
	The remaining robots	1. line up with two rows around the objects 2. prevent the objects from escaping
	Tail robot	1. located in the last line of each row 2. follows the virtual electric dipole field to approach the head robot

4.3 Row Formation Generation using Multiple Robots

In the proposed technique, the generation of parallel row formation is basic behavior for object transportation. We, therefore, present the generation method of parallel row formation in this section. First, the cyclic shift motion is defined and described in Section 4.3.1. Then, we present the smooth path generation method to show the cyclic shift motion in Section 4.3.2. Finally, the robot controllers which allow the following of the smooth path without path deviation is presented in Section 4.3.3.

4.3.1 Cyclic Shift Motion

A cyclic shift operation takes place by element rearrangement in a tuple [54]. The last or first element is transferred to the first or last element, respectively, and all other elements move forward or backward concurrently. The number and relative orders of the elements are reserved. The row formation of the guider robots is rearranged according to the cyclic shift operator, which is defined as cyclic shift motion. Figure 4.2 shows an example of cyclic shift motion using robots. Initially, there are four robots which are lined up a row in the order of A, B, C and D in step 1. In step 2, robot D moves to the rear of robot A. Thus, the order becomes D, A, B and C. The row formation is made one step backward when robot D moves. These actions are executed iteratively. For example, robot C moves to the rear of robot D in the next step; the formation order then becomes C, D, A and B.

Finally, the row formation can move backward using the cyclic shift motion.

The advantage of cyclic shift motion in formation control is easy formation maintenance because only a single robot moves forward or backward. The row formation should maintain a regular distance between the robots to prevent the object from escaping. If all robots move together to maintain the formation, keeping a certain distance between robots is difficult. This is because each robot should predict other robots' motions, and the action of all robots should be synchronized. However, in the cyclic shift motion, the other robots are stopped while one robot in line moves, which enables the robot formation to maintain a regular shape.

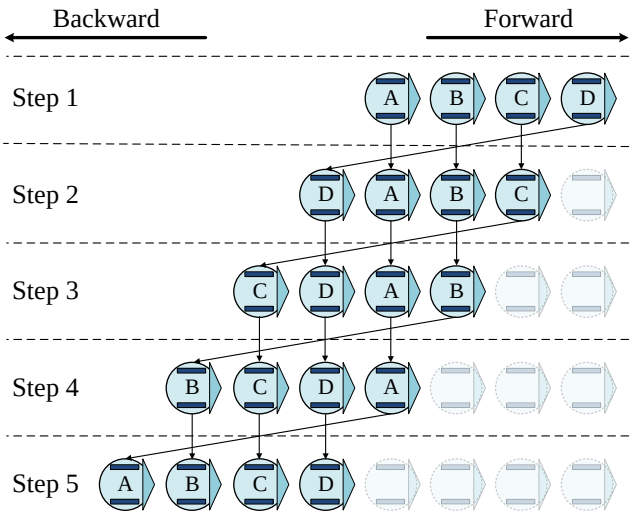

Figure 4.2 The example of cyclic shift motion. The row formation is extended in the backward direction using the cyclic shift motion.

4.3.2 Path Generation using Virtual Electric Dipole Field

In this section, we present the smooth path generation method for the cyclic shift motion. We use a *virtual electric dipole field* (VEDF) for the smooth path generation because this field has the characteristic which allows a tail robot to approach a head robot from a specific direction.

In physics, an electric dipole field is generated using positive and negative electric charges or opposite currents which exist in close proximity [55]. A precise derivation of the electric dipole field is complicated; thus, we use a simple electric dipole field equation which is modified by a single parameter [56]. The function $\mathbf{F}_e : \mathbf{R}^2 \to \mathbf{R}^2$ is referred to as a *virtual electric dipole function*, and the derivation is given by:

$$\mathbf{F}_e(x,y) = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = -\cos\theta \times \begin{bmatrix} -\cos\theta \\ -\sin\theta \end{bmatrix} + \sin\theta \times \begin{bmatrix} -\sin\theta \\ -\cos\theta \end{bmatrix} = \begin{bmatrix} \cos 2\theta \\ \sin 2\theta \end{bmatrix}, \quad (4.1)$$

where θ is the angle between a *predecessor* and a *successor robot* such that $\theta = \tan^{-1}(y/x)$. The predecessor robot represents a reference robot, the center of which is the origin of a VEDF, and the successor robot denotes the robot approaching the predecessor robot, as shown in Fig. 4.3. These terms are relative according to the robot position. If a successor robot arrives at a predecessor robot, the successor robot becomes a new predecessor robot and another moving robot becomes a new successor robot. Generally, the predecessor robot is a head robot and the successor robot is a tail robot in the proposed technique.

In (4.1), the first term $[-\cos\theta \quad -\sin\theta]^{\mathrm{T}}$ is the centripetal force, which is used for the approach toward the predecessor robot, and the second term $[-\sin\theta \quad -\cos\theta]^{\mathrm{T}}$ is the orbiting force, which is used for orbiting around the predecessor robot. Two parameters $(-\cos\theta,\ \sin\theta)$ are weighting factors of the centripetal and orbiting forces, respectively. Additionally, the shape of the VEDF is modified by introducing an additional factor β, as follows:

$$\tilde{\mathbf{F}}_e(x,y) = [\tilde{f}_x \quad \tilde{f}_y]^{\mathrm{T}} = [\cos^2\theta - \beta\sin^2\theta \quad \sin\theta\cos\theta + \beta\sin\theta\cos\theta \ ]^{\mathrm{T}}. \quad (4.2)$$

The shape of the modified VEDF moves closer to a circle as β becomes decrease, whereas it becomes more flattened as β increases. Finally, the successor robot is able to approach the predecessor robot from a specific direction using a combination of centripetal and orbiting forces.

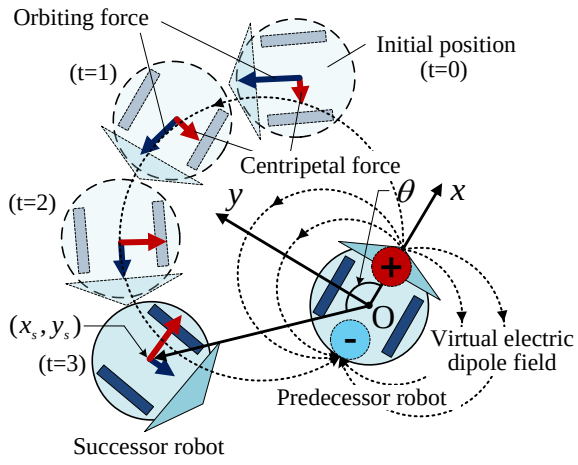

Figure 4.3 A VEDF is generated by a combination of centripetal and orbiting forces. The VEDF is created by assuming that a positive charge exists at the head of the predecessor robot and a negative charge exists at the tail of the predecessor robot. A successor robot is able to approach the predecessor robot from the behind by following the VEDF.

Objects can be manipulated to a goal by changing the direction of the transportation path. For example, if we want to transport an object to the left direction, multiple guider robots should stand in a line on the left side. We, therefore, modify the approaching angle by rotating the virtual electric dipole function $\tilde{\mathbf{F}}_e(x, y)$ which is generated by (4.2). The rotational electric dipole function $\mathbf{F}_e^{rot} : \mathbf{R}^3 \to \mathbf{R}^2$ with angle φ is described below.

$$\mathbf{F}_e^{rot}(x, y, \varphi) = \begin{bmatrix} f_{x_c}^{rot} \\ f_{y_c}^{rot} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \tilde{\mathbf{F}}_e(x, y) \tag{4.3}$$

Here, the value of φ is determined by the direction of the global path and the successor robots move along the rotated vector field, as shown in Fig. 4.4(b).

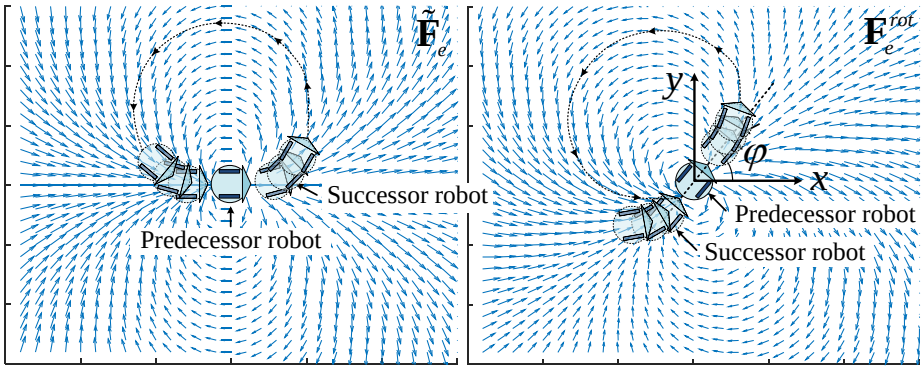

Figure 4.4 The dashed line indicates the projected path by the successor robot. (a) The successor robot follows the original VEDF, which is generated by (4.2), arriving at the tail of the predecessor robot. (b) If the VEDF is rotated by φ by (4.3), the successor robot is able to line up in the φ-direction. The robot team can modify the direction of the transportation path by changing the value of φ.

4.3.3 Path Following using Bang-bang Controller

A smooth path is generated by the VEDF, as described in Section 4.3.2. To line up in a row, guider robots should follow the smooth path without the deviation considering both kinematic and dynamic constraints. We, therefore, apply a *bang-bang controller* to the path tracking method to satisfy the kinematic and dynamic constraints [57]. A mobile robot can smoothly change its driving velocity using the bang-bang control scheme.

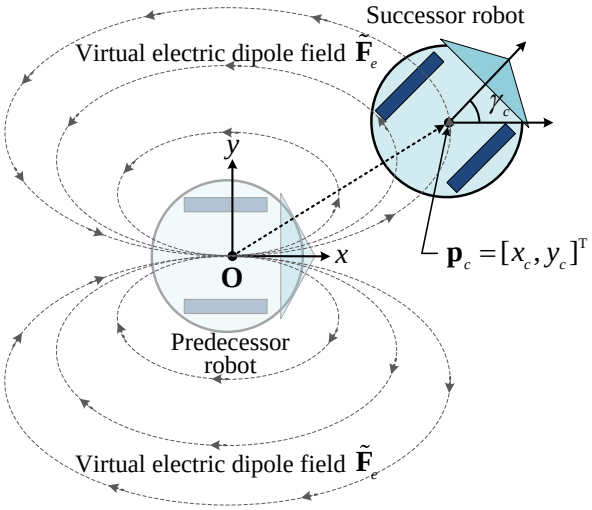

Figure 4.5 The configuration of two-wheeled differential-drive robot with respect to the predecessor robot

Here, we assume that two-wheeled differential mobile robots are used in the proposed technique, as shown in Fig. 4.5. The kinematic equation of the mobile robot is described as

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\gamma}_c \end{bmatrix} = \begin{bmatrix} v_c \cos(\gamma_c) \\ v_c \sin(\gamma_c) \\ \omega_c \end{bmatrix}, \tag{4.4}$$

where $[x_c, y_c]^{\mathrm{T}}$ is the position of the robot, γ_c is the current heading angle of the robot, and v_c and ω_c denote the tangential and angular velocity of the robot, respectively. All measured data are relative with respect to the predecessor robot because guider robots do not have a global positioning system and use limited sensing information for path following. The robots have limited tangential and angular accelerations a_c and α_c, respectively,

$$|a_c| \le a_{\max}, \ |\alpha_c| \le \alpha_{\max}, \tag{4.5}$$

where $a_{\max}$ and $\alpha_{\max}$ are the maximum tangential and angular accelerations, respectively.

If the robots follow the VEDF $\mathbf{F}_e^{rot}$, the target position $\mathbf{p}_t = [x_t, y_t]^{\mathrm{T}}$ is given by $\mathbf{p}_t = \mathbf{F}_e^{rot}(x_c, y_c) + \mathbf{p}_c$ and target heading angle γ_t is given by:

$$\gamma_t = \tan^{-1}\left(\frac{f_{y_c}^{rot}}{f_{x_c}^{rot}} \right), \tag{4.6}$$

where $f_{x_c}^{rot}$ and $f_{y_c}^{rot}$ are elements of function $\mathbf{F}_e^{rot}(x_c, y_c)$, respectively. Therefore, the path error with respect to the target position is described as

$$\begin{bmatrix} e_x \\ e_y \\ e_\gamma \end{bmatrix} = \begin{bmatrix} \cos(\gamma_t) & \sin(\gamma_t) & 0 \\ -\sin(\gamma_t) & \cos(\gamma_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t - x_c \\ y_t - y_c \\ \gamma_t - \gamma_c \end{bmatrix}, \tag{4.7}$$

where e_x, e_y and e_γ are the tangential, lateral and angular path error, respectively.

The desired angular velocity $\omega_c^d(k+1)$ is calculated as follows:

$$\omega_s(k) = \omega_t(k) + [2\alpha_{\max} | \gamma_t(k) - \gamma_c(k) |]^{1/2} \operatorname{sgn}(\omega_t(k) - \omega_c(k)), \tag{4.8}$$

$$\omega_t(k) = \frac{\gamma_t(k) - \gamma_c(k)}{\Delta T}, \quad \alpha_c(k) = \mu\left(\frac{\omega_s(k)}{\Delta T}, \ \alpha_{\max} \right), \tag{4.9}$$

$$\omega_c^d(k+1) = \omega_c(k) + \alpha_c(k)\Delta T, \tag{4.10}$$

where $\operatorname{sgn}(\cdot)$ is the sign operator, ΔT is the sampling time interval and the index k denotes the time. The clamping function $\mu(a, b)$ in (4.9) is defined as follows:

$$\mu(a, b) = \begin{cases} a & \text{for } |a| < |b| \\ |b| \operatorname{sgn}(a) & \text{for } |a| \geq |b| \end{cases}. \tag{4.11}$$

The desired tangential velocity $v_c^d(k+1)$ is calculated in a similar manner to that of the angular velocity as follows:

$$\dot{e}_x(k) = v_t(k) - v_c(k)\cos(e_\gamma(k)) + \omega_t(k)e_y(k), \tag{4.12}$$

$$v_s(k) = \dot{e}_x(k) + [2a_{\max} | e_x(k) |]^{1/2} \operatorname{sgn}(e_x(k)), \quad a_c(k) = \mu\left(\frac{v_s(k)}{\Delta T}, \ a_{\max} \right) \tag{4.13}$$

$$v_c^d(k+1) = v_c(k) + a_c(k)\Delta T. \tag{4.14}$$

In (4.12), the derivative of tangential path error $\dot{e}_x$ is calculated using ω_t, which means that $v_c^d(k+1)$ is closely related to the target angular velocity. For example, if a robot should sharply rotate according to (4.10), the robot decreases its tangential velocity according to (4.14). Therefore, the robots are able to follow the VEDF without path deviations using (4.10) and (4.14) as controllers, which enables guider robots to line up in a row.

4.4 Multi-object Transportation by a Decentralized Multi-robot Team

This section addresses the object transportation process by a decentralized multi-robot team. The transportation process is executed by the *finite state machine* (FSM), which consists of multiple states and events. The state transition is caused by three factors: (a) the acquired distances from proximity sensors, (b) the lining-up order of the guider robots, and (c) a trigger command. This information is acquired through internal sensors and by short-range communication. In Section 4.4.1, we present the method by which the sensor information is acquired, the numbering method of the guider robots, and the communication method of the trigger command. Each state in the finite state machine has different actions and algorithms, which is explained in Section 4.4.2. The state transition processes is presented in Section 4.4.3.

4.4.1 Information Acquisition Method for Finite State Machine

We present the information acquisition method for triggering events in the FSM. First, we assume that the proximity sensors on the robots are uniformly distributed around the boundary, as shown in Fig. 4.6. The x-axis direction of local coordinate is the heading of the robots, and the distance measured at the i^{th} proximity sensor is denoted as s_i. The index function $\lambda(\psi)$ at the distance of $s_{\lambda(\psi)}$ is the closest sensor index of the ψ-direction:

$$\lambda(\psi) = \underset{k}{\operatorname{argmin}} \left(\left| \psi - \frac{2\pi}{N_{prox}} k \right| \right), \tag{4.15}$$

where ψ is the angle of the sensor and N_{prox} is the number of proximity sensors equipped onto a robot. This allows the determination of the measured distance in the ψ-direction from $s_{\lambda(\psi)}$.

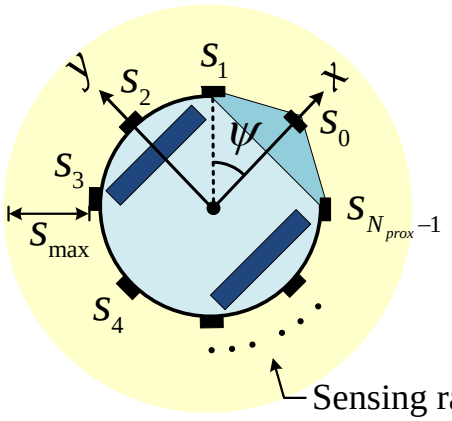

Figure 4.6 The arrangement of proximity sensors

Second, the lining-up order of robots is a key index for a state transition. The order index can be calculated by a *gradient formation algorithm*, which perceives long-range distances by means of local communication [58]. We adapt and modify this algorithm appropriately for lining up, as described in Table 4.2. The front and rear direction denote the heading direction of robot and the opposite direction of that, respectively. The notation $s_{\max}$ is the maximum measured distance from a proximity sensor; if $s_{\lambda(\psi)} = s_{\max}$, there is

no robot from the $\lambda(\psi)$-direction. A moving guider robot has the $(-1)^{\text{th}}$ order index (lines 3-4) and a tail robot has the 0^{th} order index (lines 5-6). Initially, all guider robots in the row formation have the maximum order index (line 8) and they compare the order index with that of the front robot by local communication. If the front robot has a smaller order index than me, the order index is changed to the index of the front robot (lines 9-10). Then, the robot adds one index to its self-index (line 11) and transmits the self-index to other neighboring robots (line 12). All guider robots are able to recognize their orders with respect to a global view through the above iterative processes.

TABLE 4.2 ALGORITHM FOR THE ASSIGNMENT OF RELATIVE ORDER IN ROW FORMATION

Algorithm	Gradient formation (*for guider robot*)
Input	s_i : the distance acquiring from i^{th} proximity sensor *order*(front) : the order index of the front robot *order*(self) : the order index of the self-robot
Output	*order*(self) : the order index of the self-robot
1:	**loop**
2:	*check order*(front)
3:	if $s_{\lambda(0)} = s_{\max}$ and $s_{\lambda(\pi)} = s_{\max}$
4:	*order*(self) $\leftarrow -1$
5:	else if $s_{\lambda(0)} = s_{\max}$ and $s_{\lambda(\pi)} < s_{\max}$
6:	*order*(self) $\leftarrow 0$
7:	else
8:	*order*(self) $\leftarrow$ *order*(max)
9:	if *order*(front) $<$ *order*(self)
10:	*order*(self) $\leftarrow$ *order*(front)
11:	*order*(self) $\leftarrow$ *order*(self) $+1$
12:	*transmit order*(self)

Third, there are two sharing commands between robots in the proposed technique: *ready* and *action*. To begin the transportation, the pusher robot transmits an action command to neighboring guider robots. In contrast, the leader robot transmits a ready command to neighboring guider robots to end a cycle of transportation. A guider robot always waits for a trigger command from neighboring robots and transmits its trigger command once to other neighboring robots when there is a change in the trigger command. For example, if a guider robot receives a ready or an action command from an anonymous robot, the guider robot then transmits the ready or the action command to other neighboring robot once, respectively.

4.4.2 Finite State Machines (FSMs)

The guider, the leader, and the pusher robots in the proposed multi-robot team have different FSMs, as shown in Fig. 4.7. The guider robot has three states and three events, the pusher has two states and two events, and the leader robot also has two states and two events, respectively. Each robot carries out their actions according to their states, and state transitions are led by diverse events. All events in the FSMs are summarized in Table 4.3. The proposed FSMs enable the decentralized multi-robot team to transport the objects by systematic cooperation.

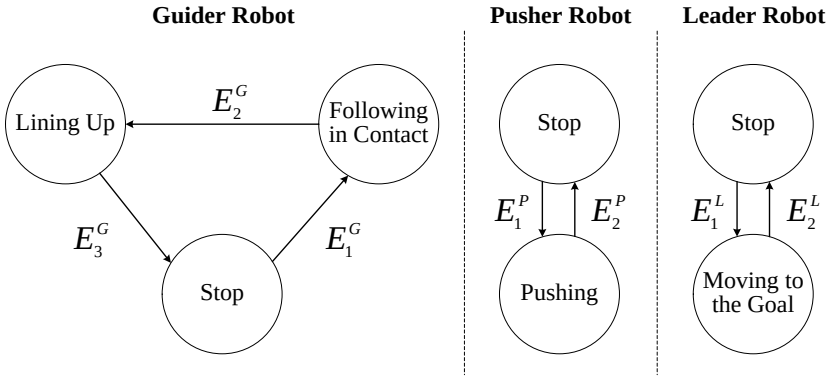

Figure 4.7 The FSMs of the guider, pusher, and leader robots

TABLE 4.3 THE EVENT DESCRIPTIONS IN THE FSMS

Robot	Event	Descriptions
Guider	E_1^G	*receive 'action'* and $order(\text{self}) = 0$
	E_2^G	$\| \mathbf{p}_{self}^G - \mathbf{p}_{head}^G \| < 2r + s_{\max}$
	E_3^G	$\| \mathbf{p}_{self}^G - \mathbf{p}_{head}^G \| < s_\varepsilon$
Pusher	E_1^P	*receive 'ready'* and $order(\text{left}) = 0$ and $order(\text{right}) = 0$
	E_2^P	$order(\text{left}) = 1$ and $order(\text{right}) = 1$
Leader	E_1^L	$order(\text{left}) = \frac{N_G}{2} - 2$ and $order(\text{right}) = \frac{N_G}{2} - 2$
	E_2^L	$s_{\lambda(\pi/2)}^L \geq s_{\max}$ and $s_{\lambda(-\pi/2)}^L \geq s_{\max}$

4.4.2.1 The FSM of Guider Robots

The FSM of the guider robots consists of the *following-in-contact*, the *lining-up* and the *stop* states. Table 4.4 shows the following-in-contact algorithm of guider robots. In the following-in-contact state, a guider robot

follows the periphery of the line formation while maintaining a desired distance $s_{desired}$. If the guider robot cannot detect any robot at the maximum measured distance $s_{\max}$, the guider robot follows the line formation by clockwise or counterclockwise motion (lines 2-3). The turning direction depends on the row in which the guider robot is located. For example, the guider robots in the first and second row move counterclockwise and clockwise to follow the line formation, respectively, as shown in Fig. 4.8(b). In addition, if the guider robot is closer to the line formation than the desired distance $s_{desired}$, the guider robot turns further away from the line formation (lines 4-5). In contrast, if the guider robot is further away from the line formation, the guider robot turns closer to the line formation (lines 6-7). If the guider robot maintains the desired distance $s_{desired}$, the guider robot moves forward (line 8-9). The target tangential velocity v_t and the target angular velocity ω_t are predefined by the system designer, which satisfy the maximum acceleration constraints. In this case, the target position $\mathbf{p}_t$ is $\mathbf{p}_c + v_t \Delta T$ and the target heading angle γ_t is $\gamma_c + \omega_c \Delta T$. The notation *counterclockwise*^{1st} indicates that a robot turns counterclockwise if the robot is located in the first row. Similarity, the notation *clockwise*^{2nd} indicates that a robot turns clockwise if the robot is located in the second row. The following-in-contact state is changed to the lining-up state if a head robot is detected in the sensing range.

TABLE 4.4

ALGORITHM FOR THE FOLLOWING-IN-CONTACT STATE

Algorithm	Following-in-contact (*for guider robot*)
Input	$s^G_{\lambda(\cdot)}$: The sensor information of guider robot
Output	action of the guider robot
1:	**loop**
2:	if $\forall s^G_{\lambda(\cdot)} \geq s_{\max}$
3:	*move forward and* {*counterclockwise*^{1st}, *clockwise*^{2nd}}
4:	else if $s^G_{\lambda(\pi/2)} < s_{desired}$ or $s^G_{\lambda(-\pi/2)} < s_{desired}$
5:	*move forward and* {*clockwise*^{1st}, *counterclockwise*^{2nd}}
6:	else if $s_{desired} < s^G_{\lambda(\pi/2)} \leq s_{\max}$ or $s_{desired} < s^G_{\lambda(-\pi/2)} \leq s_{\max}$
7:	*move forward and* {*counterclockwise*^{1st}, *clockwise*^{2nd}}
8:	else
9:	*move straightforward*

Table 4.5 shows the algorithm for the lining-up state. In the lining-up state, a guider robot approaches the head robot located in the first line of the row. The VEDF $\mathbf{F}_e^{rot}$ is generated with respect to the local coordinate of the head robot (line 2). The origin of $\mathbf{F}_e^{rot}$ is located at $\mathbf{p}^G_{head}$, and its x-axis is parallel to the $\mathbf{p}^G_{front\_head} - \mathbf{p}^G_{head}$ vector, where $\mathbf{p}^G_{head}$ is the position of the head robot and $\mathbf{p}^G_{front\_head}$ is the position of ahead of the head robot. The rotational angle φ of the VEDF is determined by the global path planner when the guider robot approaches the leader robot within communication range. The guider robot calculates its tangential velocity v^d and rotational velocity ω^d using (4.10) and (4.14), respectively (line 3). The guider robot can then move along the VEDF using v^d and ω^d without path deviation (line 4).

TABLE 4.5 ALGORITHM FOR THE LINING-UP STATE

Algorithm	Lining-up (*for guider robot*)
Input	$\mathbf{p}_{head}^{G}$: the position of the head robot $\mathbf{p}_{front\_head}^{G}$: the position of ahead of the head robot φ : the rotational angle of the VEDF
Output	v^d : the desired tangential velocity of the guider robot ω^d : the desired angular velocity of the guider robot
1:	**loop**
2:	*generate virtual electric dipole field* $\mathbf{F}_e^{rot}$ with respect to the local coordinate of $\mathbf{p}_{head}^{G}$ and $\mathbf{p}_{front\_head}^{G}$
3:	*calculate* v^d *and* ω^d using (4.10) and (4.14), respectively
4:	*approach the head robot* using v^d and ω^d

If the guider robot is too close to the head robot ($\|\mathbf{p}_{self}^{G} - \mathbf{p}_{head}^{G}\| < s_{\varepsilon}$), the lining-up state is changed to the stop state, where s_{ε} is marginal sensing range. The measurement method of distance between the guider robot and head robot is to check the sensor value of the front direction ($s_{\lambda(0)}^{G}$). This is because the guider robot should be positioned behind the head robot if the robot follows the VEDF without path deviation.

When the guider robot is in the stop state, the robot stops its motion, waits for a trigger command, and continuously checks whether its order index is 0^{th} or not.

4.4.2.2 The FSM of a Pusher Robot

The FSM of a pusher robot consists of the *pushing* and *stop* states. In the pushing state, the pusher robot transmits an action command to neighboring

guider robots (line 1) and moves forward while maintaining the desired gap s_{gap} between the tail robots (line 8), as described in Table 4.6. If the pusher robot is too close to or too far away from the guider robots, the pusher robot adjusts its heading direction toward the center of the parallel row formation (lines 3-6). When two or more pusher robots are used, they share their sensing information for position arrangement. For example, if more than two pusher robots are used in the transportation process, the sensing distances $s^{P}_{\lambda(\pi/2)}$ and $s^{P}_{\lambda(-\pi/2)}$ are acquired from the leftmost and rightmost pusher robot by intercommunication, respectively.

In the stop state, the pusher robot stops moving, waits for a ready command from the neighboring guider robots, and checks whether the left and the right robots have the 0^{th} order index.

TABLE 4.6 ALGORITHM FOR THE PUSHING STATE

Algorithm	Pushing (*for pusher robot*)
Input	$s^{P}_{\lambda(\cdot)}$: The sensor information of pusher robot s_{gap} : The desired distance from a guider robot
Output	action of the pusher robot
1:	*transmit 'action' command*
2:	**loop**
3:	**if** $s^{P}_{\lambda(\pi/2)} > s_{gap}$
4:	*move forward and counterclockwise*
5:	**else if** $s^{P}_{\lambda(-\pi/2)} > s_{gap}$
6:	*move forward and clockwise*
7:	**else**
8:	*move straightforward*

4.4.2.3 The FSM of a Leader Robot

The FSM of a leader robot consists of the *moving-to-the-goal* and *stop* states. Table 4.7 shows the algorithm for the moving-to-the-goal state. In the moving-to-the-goal-state, the leader robot transmits a ready command to neighboring robots and moves to the goal by the global path planner $\mathbf{P}(k)$ until there are no guider robots on the left or right sides. Also, the leader robot transmits the rotational angle φ of the VEDF to the $(-1)^{\text{th}}$ guider robot approaching to the leader robot when the guider robot is located within communication range.

In the stop state, the leader robot stops moving and checks the order index of both sides of the guider robots. If the order indices of both sides are $N_G / 2 - 2$, which means that a guider robot in both rows (totally two robots) is approaching to the leader robot, then the state is changed to the moving-to-the-goal. Similar to the pusher robot, two or more leader robots can be used for object transportation, and they share their sensing information.

TABLE 4.7 ALGORITHM FOR THE MOVING-TO-THE-GOAL STATE

Algorithm	Moving-to-the-goal (*for leader robot*)
Input	*order*(left) : the order index of the left robot *order*(right) : the order index of the right robot
Output	action of the leader robot
1:	*transmit 'ready' command*
2:	**loop**
3:	*move to the goal according to the global path planner* $\mathbf{P}(k)$
4:	*transmit rotational angle* φ *to* $(-1)^{\text{th}}$ *guider robot*

4.4.3 Object Transportation Process

In this section, we describe the object transportation processes using the FSMs. The transportation process is not wholly separated, but we divide the transportation process into four steps to simplify the explanation.

Initially, all robots are in the stop state and share a ready command, as shown in Fig. 4.8(a). In this case, event E_1^P is satisfied and the state of the pusher robot changes from the stop to pushing state; the pusher robot begins to push the objects and transmits an action command to the neighboring guider robots. In the next step, the pusher robot stops moving because event E_2^P is satisfied; the order indices of the left and right guider robots are both one, as shown in Fig. 4.8(b). The tail robots in two rows begin to move along each row, maintaining the desired distance. In the third step, the order indices of the guider robots are changed and the tail robot is replaced by the next guider robot because the previous tail robot leaves the row, as shown in Fig. 4.8(c). The order index of a moving guider robot is -1; the order index of the head robot becomes $N_G/2-2$, where N_G is the total number of guider robots. This situation satisfies event E_1^L and the leader robot begins to approach the goal. In the final step, a moving guider robot is close to the head robot and the state is changed to the lining-up state by event E_2^G. The moving guider robot generates a VEDF with respect to the head robot and follows it using the (4.10) and (4.14) controllers. The rotational angle of the VEDF is received from the leader robot. In addition, the leader robot stops moving and

transmits a ready command to other neighboring robots because there are no robots on the left and right sides. This satisfies event E_2^L.

Above four steps are executed iteratively until all objects are located in the goal region. The boundaries between steps are vague, which means that all robots show continuous motion during the transportation. The state and event descriptions according to the processes are summarized in Table 4.8.

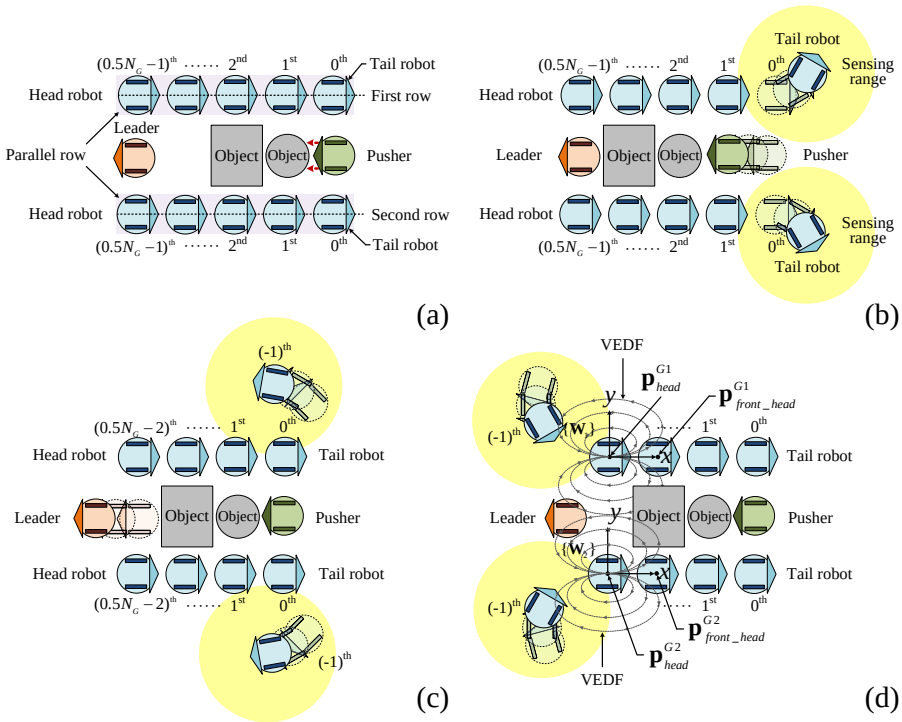

Figure 4.8 Object transportation steps. (a) All robots are initially stationary. A pusher robot begins to push objects according to event E_1^P. (b) Two tail robots begin to follow each row according to event E_1^G and a pusher robot stops between the first indexed robots. (c) The leader robot begins to approach the goal according to the global path. (d) The moving guider robots detect the head robot in each row and line up by following the VEDF according to event E_2^G. The leader robot stops when there are no guider robots on either side.

TABLE 4.8 STATE AND EVENT DESCRIPTIONS ACCORDING TO THE TRANSPORTATION PROCESS

Figure	Command	Robot	Event	State
Figure 4.8(a)	Ready	Guider	-	Stop
		Pusher	E_1^P	Stop → Pushing
		Leader	-	Stop
Figure 4.8(b)	Action	Guider	E_1^G	Stop → Following-in-contact
		Pusher	E_2^P	Pushing → Stop
		Leader	-	Stop
Figure 4.8(c)	Action	Guider	-	Following-in-contact
		Pusher	-	Stop
		Leader	E_1^L	Stop → Moving-to-the-goal
Figure 4.8(d)	Ready	Guider	E_2^G	Following-in-contact → Lining-up
		Pusher	-	Stop
		Leader	E_2^L	Moving-to-the-goal → Stop

4.4.4 Formation Constraints for Curved Transportation Path

In the previous sections, we dealt with the straight transportation path only. However, the curved transportation path is necessary for manipulating the objects to the goal. The guider robots should maintain rectangular formation in the curved path because the extremely inequality of row length precludes the objects to be manipulated. Thus, the distances between guider robots in each row should be adjusted to maintain the rectangular formation. In this

section, we present three constraints of rotational angle for following the curved transportation path.

First, the curved path has a feasible curvature: $90° < \xi < 270°$ where ξ is the rotational angle of path. If this constraint is not satisfied, the distance between two rows is less than zero. It means that the arrangement of the robot formation is infeasible.

Second, the minimum-sized object should not escape the robot formation in the curved path. This constraint subdivides into two categories according to the quadrant of the rotational angle: $90° < \xi \le 180°$ (quadrant II) and $180° < \xi < 270°$ (quadrant III). If the rotational angle belongs to the quadrant II, the constraint to prevent the smallest object escaping from the robot formation is given by:

$$d_{\varepsilon} + \frac{2b\sin\xi}{N_G} < \min_i(2r^{O_i}) \quad \text{for} \quad 90° < \xi \le 180°, \tag{4.16}$$

where d_{ε} is the marginal distance between guider robots in the non-curved path, b is the distance between the rows, N_G is the number of guider robots, and r^{O_i} is the radius of i^{th} object. Thus, $d_{\varepsilon} + 2b\sin\xi / N_G$ is the gap between robots in the longer row. Figure 4.9 shows the robot formation in the curved path if the rotational angle belongs to $90° < \xi \le 180°$.

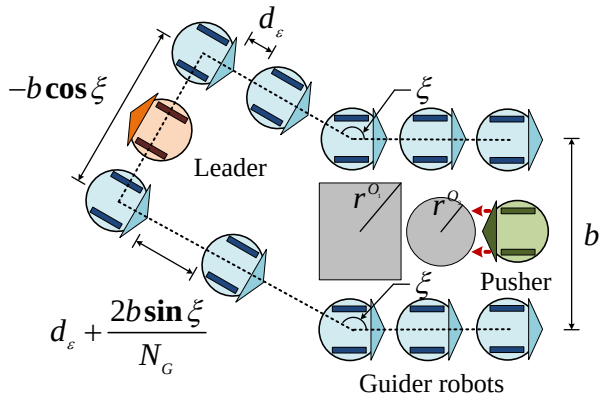

Figure 4.9 Curved path description ($90° < \xi \leq 180°$)

In a similar way, the constraint of rotational angle in the quadrant III is given by:

$$d_\varepsilon - \frac{2b\sin\xi}{N_G} < \min_i(2r^{O_i}) \quad \text{for} \quad 180° < \xi < 270°. \tag{4.17}$$

Figure 4.10 shows the curved path if the rotational angle belongs to $180° < \xi < 270°$.

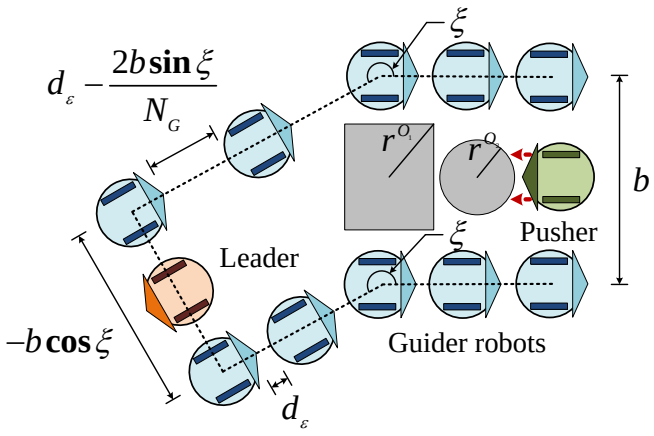

Figure 4.10 Curved path description ($180° < \xi < 270°$)

Third, the maximum-sized object should pass the robot formation in the curved path. The width of the curved path should be wide to pass the maximum object. This constraint can be described as:

$$-b\cos\xi > \max_i(2r^{O_i}), \tag{4.18}$$

where $-b\cos\theta$ is the width of the curved path.

As a result, the constraints of rotational angle are given as (4.16), (4.17) and (4.18). Therefore, the leader robot can design the global path by considering the above constraints.

Chapter 5

Simulation Results

5.1 Simulation Environment

We tested the passive object transportation technique by simulations. The active object transportation was verified by a practical experiment because the errors of sound signal and proximity sensors can be considered only in the practical experiment. Our simulation was conducted using the MATLAB simulator. Figure 5.1 shows the simulation environment; two differently sized circular objects are transported. Sixteen robots, including a leader robot and a pusher robot, are used in the proposed technique. The radii of the two objects are 18 cm and 14 cm, respectively, and the radius of all of the robots is 10 cm. The maximum tangential and angular accelerations of the guider robots are 1 cm/s^2 and 5 $°/s^2$, respectively. The initial positions of the robots and objects are given in Table 5.1, and the parameter information is given in Table 5.2.

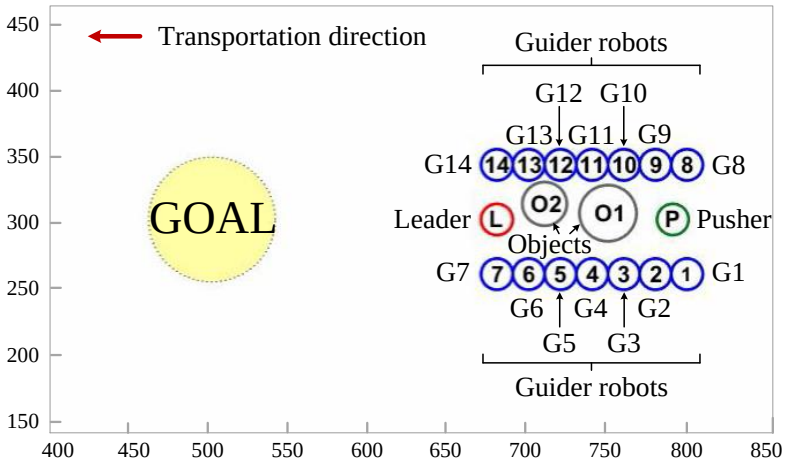

Figure 5.1 Simulation environment of the proposed technique

TABLE 5.1 THE NOTATIONS AND INITIAL POSITIONS OF THE ROBOTS AND OBJECTS IN THE SIMULATIONS

Name	Notation	Initial Position	Name	Notation	Initial Position
Leader	L	$\mathbf{p}^{L}(t_0)=[680,310]^{\mathrm{T}}$	Guider 6	G6	$\mathbf{p}_6^{G}(t_0)=[700,275]^{\mathrm{T}}$
Pusher	PS	$\mathbf{p}^{PS}(t_0)=[800,310]^{\mathrm{T}}$	Guider 7	G7	$\mathbf{p}_7^{G}(t_0)=[680,275]^{\mathrm{T}}$
Goal	GOAL	$\mathbf{p}_{GOAL}=[500,310]^{\mathrm{T}}$	Guider 8	G8	$\mathbf{p}_8^{G}(t_0)=[800,345]^{\mathrm{T}}$
Object 1	O1	$\mathbf{p}^{O1}(t_0)=[750,325]^{\mathrm{T}}$	Guider 9	G9	$\mathbf{p}_9^{G}(t_0)=[780,345]^{\mathrm{T}}$
Object 2	O2	$\mathbf{p}^{O2}(t_0)=[710,325]^{\mathrm{T}}$	Guider 10	G10	$\mathbf{p}_{10}^{G}(t_0)=[760,345]^{\mathrm{T}}$
Guider 1	G1	$\mathbf{p}_1^{G}(t_0)=[800,275]^{\mathrm{T}}$	Guider 11	G11	$\mathbf{p}_{11}^{G}(t_0)=[740,345]^{\mathrm{T}}$
Guider 2	G2	$\mathbf{p}_2^{G}(t_0)=[780,275]^{\mathrm{T}}$	Guider 12	G12	$\mathbf{p}_{12}^{G}(t_0)=[720,345]^{\mathrm{T}}$
Guider 3	G3	$\mathbf{p}_3^{G}(t_0)=[760,275]^{\mathrm{T}}$	Guider 13	G13	$\mathbf{p}_{13}^{G}(t_0)=[700,345]^{\mathrm{T}}$
Guider 4	G4	$\mathbf{p}_4^{G}(t_0)=[740,275]^{\mathrm{T}}$	Guider 14	G14	$\mathbf{p}_{14}^{G}(t_0)=[680,345]^{\mathrm{T}}$
Guider 5	G5	$\mathbf{p}_5^{G}(t_0)=[720,275]^{\mathrm{T}}$			

(unit: cm)

TABLE 5.2 PARAMETER INFORMATION IN THE SIMULATIONS

Name	Notation	Value	Name	Notation	Value
All robots	r	10 cm	Pusher robot	v_{PS}	20 cm/s
	$s_{\max}$	55 cm		s_{gap}	5 cm
Guider robot	$v_{\max}$	65 cm/s	Leader robot	v_L	20 cm/s
	$a_{\max}$	26 cm/s^2	Object	r_{O_1}	18 cm
	$\omega_{\max}$	30°/s		r_{O_2}	14 cm
	$\alpha_{\max}$	5°/s^2	Goal	δ	40 cm
	$s_{desired}$	35 cm	The parameter of VEDF	β	2
	s_g	3 cm			

5.2 Simulation Result of Passive Object Transportation

We tested the proposed object transportation technique, as shown in Fig. 5.2. Initially, all robots were stopped and shared a ready command between them, which enables event E_1^P to occur. The pusher robot (P) began to push the objects and transmitted an action command to the neighboring robots (G1 and G8) at 0 seconds. At the same time, two guider robots in each row (G1 and G8) began to follow the row in contact according to event E_1^G. After 1 second, the pusher robot stopped because event E_2^P was satisfied; the order indices of the left and right guider robots (G2 and G9) with respect to the pusher robot were both 1. Two guider robots in each row (G1 and G8) left their respective rows at 1 second, and thus, the order indices of the head robots (G7 and G14) in the rows were changed to $N_G / 2 - 2$. The leader robot (L) began to approach the goal according to event E_1^L and transmitted a ready command to the neighboring guider robots (G7 and G14). The moving guider robots (G1 and G8) detected the head robots in each respective row

which triggered the state transition from the following-in-contact to lining-up state according to event E_2^G at 7 seconds. At the same time, the leader robot stopped according to event E_2^L. The state transition of the guider robot from the lining-up to stop state occurs by according to event E_3^G when the guider robots (G1 and G8) arrive in the neighborhood of the head robot at 8 seconds. These processes were executed iteratively until the two objects (O1 and O2) were transported to the goal. The travel time was 93 seconds and the average speed of object transportation was 2.7 cm/s.

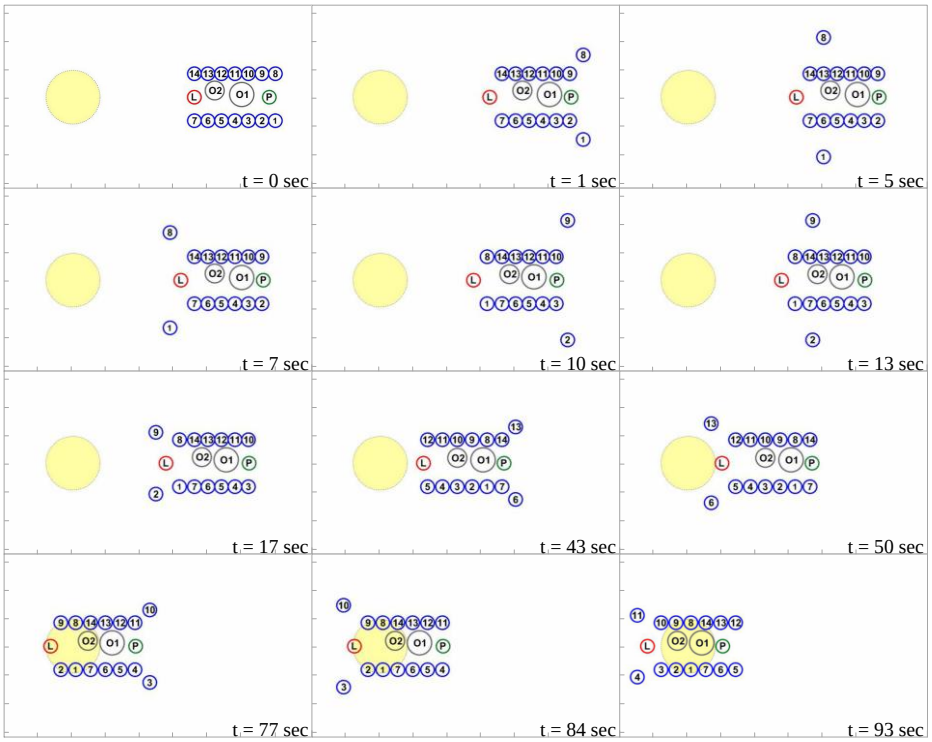

Figure 5.2 Simulation result of the proposed technique

Figure 5.3 shows the state transition results of the simulation. For a simple graphical description, we use a new index, termed the *state index*, as described in Table 5.3. The state index denotes the state of robot, which has different meanings according to robot. For example, state index 1 of the guider robot indicates the following-in-contact state, and state index 1 of the pusher robot indicates the pushing state. The pusher robot initiates the transportation by changing from the stop state to the pushing state, as shown in Fig. 5.3. At nearly the same time (at 1 second), the states of the leader and guider robot 1 (G1) change to the moving-to-the-goal and following-in-contact state, respectively. The states of the leader and pusher robots return to the stop state quickly because the travelling distance to approach the goal and the pushing distance are both short. The state of the guider robot changes from the following-in-contact to lining-up state after the head robot is detected at 7.3 seconds. After the lining-up state is completed, the guider robot is stopped by the stop state at 7.5 seconds. The identical process is executed iteratively with other guider robots until all objects are transported to the goal. For example, guider robot 2 (G2) started to move, and its state changed from 7.5 to 16 seconds.

TABLE 5.3 THE STATES ACCORDING TO THE STATE INDEX IN FIG. 5.3

State index	Guider robot	Pusher robot	Leader robot
0	Stop	Stop	Stop
1	Following-in-contact	Pushing	Moving-to-the-goal
2	Lining-up	-	-

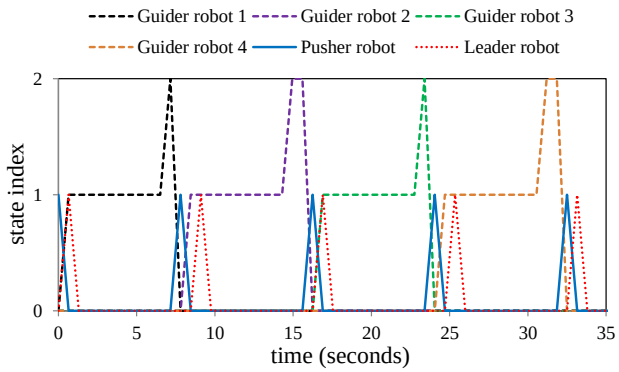

Figure 5.3 The state transition results of the simulation. Only some of the guider robots are presented in the graph for a simple description; there were 14 guider robots in the simulation.

A key motion of the proposed transportation technique is the lining up of the guider robots. Therefore, we examine at this stage the motion of a single guider robot in detail. All guider robots have identical states and algorithms; thus, it is sufficient to analyze a single guider robot to verify the proposed technique. Figure 5.4 shows the tangential and angular velocity of a guider robot (G1) for a total of 9 seconds. Initially, the angular velocity and the tangential velocity of the guider robot were 0 °/s and 0 cm/s, respectively. In the following-in-contact state, the heading direction of the guider robot changed frequently, as shown in Fig. 5.6, as the guider robot modified its heading direction to maintain a regular distance from the row. On the other hand, the tangential velocity was relatively low and constant because the guider robot needs not change the tangential velocity frequently during the following-in-contact state. The state of the guider robot changed to the lining-

up state at 7.3 seconds. At this point, the tangential velocity increased dramatically at first and decreased at the end of the lining-up state to track the VEDF. This tangential velocity profile is the result of the bang-bang control scheme via (4.14). The angular velocity was also calculated using (4.10) in the lining-up state. Figure 5.5 shows the tangential and the angular accelerations of the guider robot (G1) according to the velocity profile. The guider robot did not exceed the maximum acceleration in any case, which shows that the dynamic constraint is satisfied.

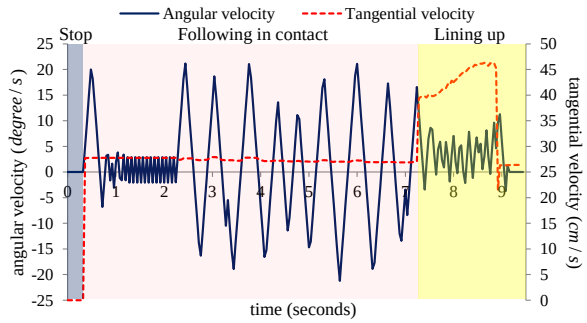

Figure 5.4 The tangential and angular velocity of a guider robot (G1) for 9 seconds.

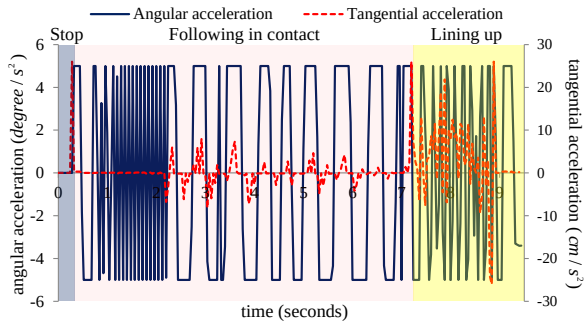

Figure 5.5 The tangential and rotational acceleration of a guider robot (G1) for 9 seconds.

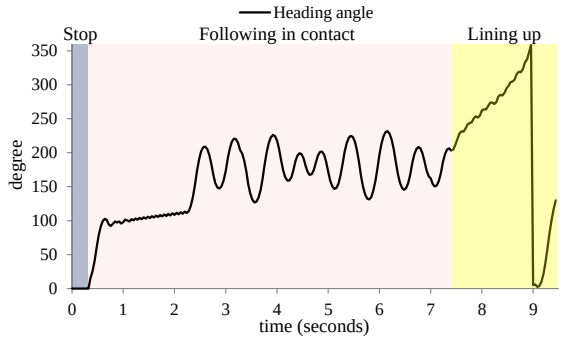

Figure 5.6 The heading direction of a guider robot (G1) for 9 seconds. The guider robot frequently adjusted its heading direction in the following-in-contact state for maintaining the desired distance from the row. The dramatic decrease in the lining-up state is caused by the angular translation between $0°$ and $360°$

The above simulation was conducted in the ideal environment assuming that there are no localization errors. However, there is noise in real environment when robots detect objects or other robots, which cause localization errors. We, therefore, verified the proposed technique by considering zero-mean Gaussian noise with standard deviation (σ) changing from 0 to 3 at an interval of 0.25, as shown in Fig. 5.7. A hundred simulations were conducted at each standard deviation, and total number of simulations was 1300. The success rate drops under 70% after σ is 2 because the relative localization error is large with respect to the size of robot ($r = 10\,\text{cm}$).

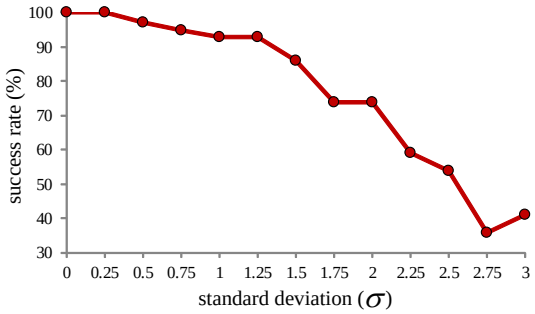

Figure 5.7 The success rate graph by changing the standard deviation in Gaussian noise with zero-mean. The success rate drops under 70% if the standard deviation is more than 2.

5.3 Comparison Results with Other Passive Object Transportation Techniques

We conducted on two comparative object transportation simulations for showing the advantages of the proposed technique. The first is leader-follower based object transportation, and the second is caging based object transportation. The leader-follower technique was developed for the formation control of multi-robot system, but this technique can be applied to object transportation area [59]. The caging technique was originally developed for guaranteeing object-wrapping using a robot manipulator, but this is extended to the object transportation using mobile robots [16]. These techniques used different methods with the proposed technique for object transportation, but they have identical principles which are to wrap an object using multiple robots. Thus, we chose these methods for comparison with the proposed technique.

5.3.1 Simulation Result of Leader-follower Technique

We simulated the object transportation method based on the leader-follower formation control. Desai et al. presented a formation control technique by maintaining distance l_d and orientation ψ_d from a leader, which is called as $l-\psi$ *controller* [60]. Figure 5.8 shows the initial positions of robots for object transportation. Robots (1~5, 8~12) followed the front robots, respectively, using the $l-\psi$ controller by maintaining relative distance and orientation. For example, the robot 12 followed the leader robot by maintaining relative distance and orientation, and the robot 11 also followed the robot 12. Using the $l-\psi$ controller, all robots can maintain the regular formation by the linkage connection between robots.

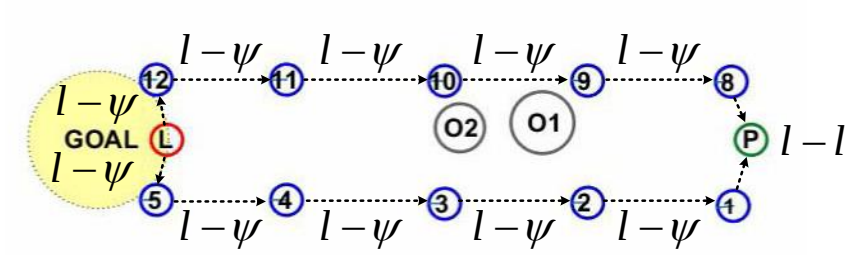

Figure 5.8 The position assignment of leader-follower based formation control for object transportation

Figure 5.9 shows the simulation result of object transportation using the leader-follower control. The desired distance and orientation between robots are 20 cm and 180°, respectively. The initial distance between robots was 27

cm. The robots followed the front robots to narrow the distance. The small object (O2) escaped at 6 seconds because of the localization error by Gaussian noise. If the $l-\psi$ controller is used for object transportation, the localization error of a robot affects the motion of other robots. This error is accumulated while the robot team proceeds to the goal. Thus, the $l-\psi$ controller cannot guarantee the maintenance of distance between robots. Finally, only the large object (O1) arrived in the goal, but the small object (O2) could not arrive in the goal.

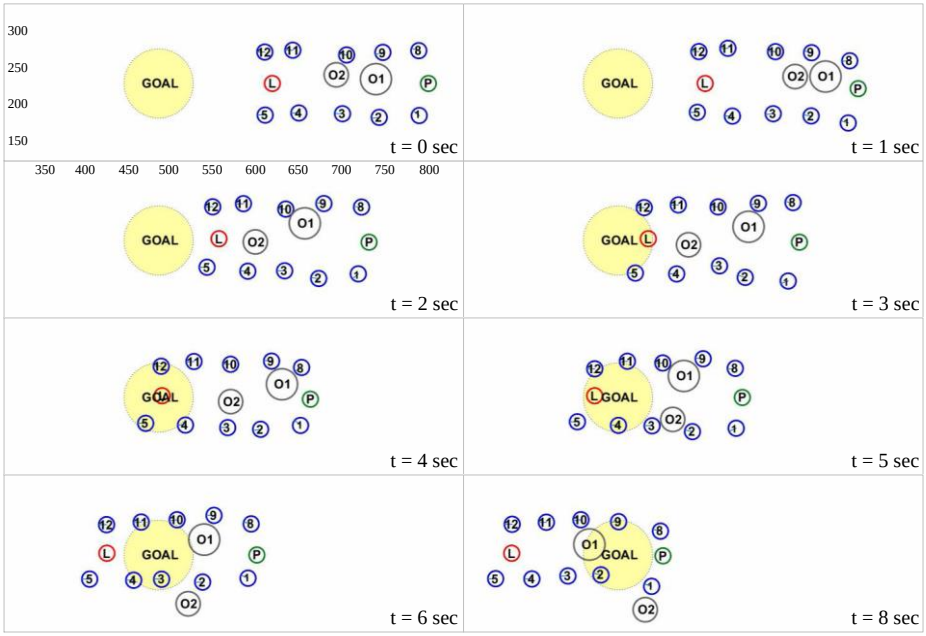

Figure 5.9 Simulation result of object transportation using leader-follower control with Gaussian noise (mean: 0, standard deviation: 0.5)

Figure 5.10 shows the success rate of the object transportation using the leader-follower formation control. The success rate became fewer than 80% if the standard deviation is more than 0.5. In the leader-follower technique, all robots move together. If a robot has some motion or localization error, other robots are affected this error. Thus, the small value of standard deviation decreases the success rate of object transportation by comparison with the proposed technique.

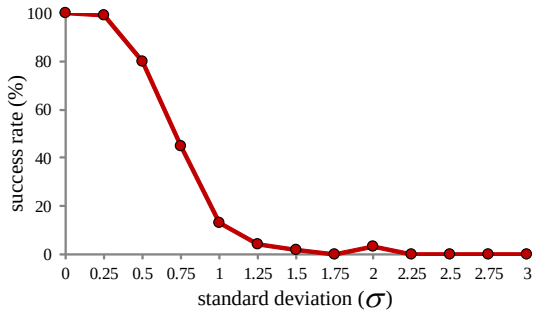

Figure 5.10 The success rate of object transportation using the leader-follower formation control with Gaussian noise. The success rate dropped sharply under 80% if the standard deviation is more than 0.5.

5.3.2 Simulation Result of Caging Technique

The caging technique cannot be applied for multi-object transportation because this technique considers a single object only. However, we extended the caging technique to multiple objects by applying the wrapping condition with respect to the large object. We presented two caging simulations for showing diverse cases; the first is the minimum number of robots, and the

second is the enough number of robots.

The minimum necessary number of robots for caging based object transportation is given by $N_{\min}$ [16, 61]:

$$N_{\min} = \frac{\pi r_{cage}}{\frac{1}{2} dist_{\min}(obj) + \max_i r_i^R}, \tag{5.1}$$

where $dist_{\min}(obj)$ is the minimum distance between the boundary points of an object, r_i^R is the radius of the i^{th} robot and r_{cage} is the virtual radius of caging formation:

$$r_{cage} = \frac{1}{2} dist_{\max}(obj) + \max_i r_i^R + \varepsilon . \tag{5.2}$$

In this simulation, $dist_{\min}(obj)$ is 18 cm, $dist_{\max}(obj)$ is $18\sqrt{2}$ cm, and r_i^R is 10 cm, respectively, according to Table 5.2. If we set ε as 3 cm, the minimum necessary number of robots $N_{\min}$ is 4.25. The five, therefore, is the minimum number of robots in the caging technique. Figure 5.11 shows the caging-based object transportation simulation when the minimum number of robots was used. The virtual rectangular object in Fig. 5.11 denotes the virtual rectangle which wraps the largest circular object. This approximation is needed for applying the caging technique to multi-object transportation. After 4 second, two objects were overlapped graphically, which means that object transportation is infeasible. This is because there is no enough space to include a small object in the caging formation.

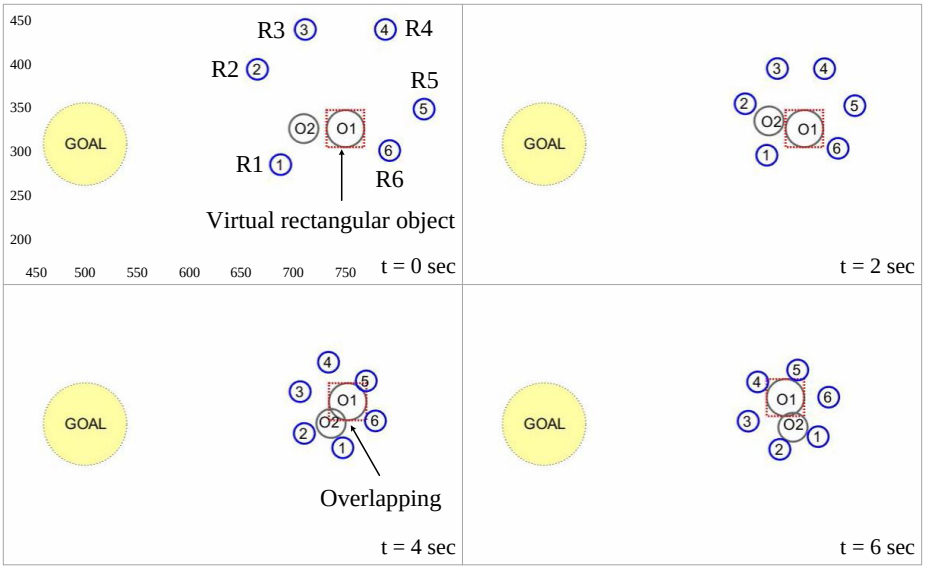

Figure 5.11 Caging based object transportation when the minimum number of robots was used. The overlapping between objects occurred at 4 seconds because the small object (O2) was not considered.

In the second case, we used enough number of robots to secure the free space. The maximum number of robots for caging based object transportation is given by [16]:

$$N_{\max} = \frac{\pi r_{cage}}{\max_i 2r_i^R}. \tag{5.3}$$

Thus, we determined the number of robots as eight because of $N_{\max} = 8.1$. Figure 5.12 shows the caging-based object transportation simulation when the enough number of robots was used. We added zero-mean Gaussian noise with standard deviation ($\sigma = 0.5$) for verifying the robustness to error. In contrast to the previous caging technique case, two objects could be manipulated in the

beginning because the caging formation is sufficiently large to wrap the small and large object together. However, the small object (O2) escaped from the caging formation at 10 seconds because the gap between robots could not be maintained due to the Gaussian noise; the gap is enough to wide for the small object.

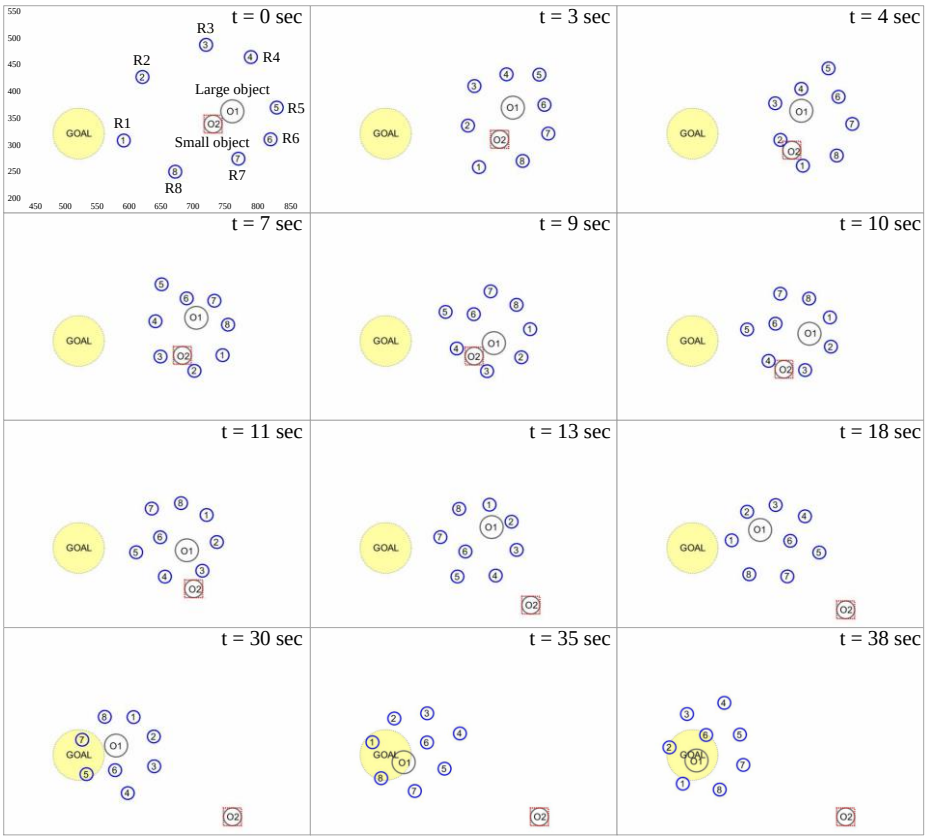

Figure 5.12 Caging based object transportation when the sufficient number of robots to wrap the two objects was used. The eight is the sufficient number of robots for the large and small object (O1 and O2) transportation. However, the small object escaped from the caging formation after 10 seconds because the distance between neighboring robots is too wide to wrap the small object.

Figure 5.13 shows the success rate of the caging-based object transportation. The success rate became fewer than 80% if the standard deviation is more than 0.25. In the caging technique, the distances between the neighboring robots should be smaller than the size of small object for successful transportation. However, it is difficult to maintain constant distances between robots if the localization or motion errors exist. This problem is caused by the simultaneous movements of robots in common with the formation control case.

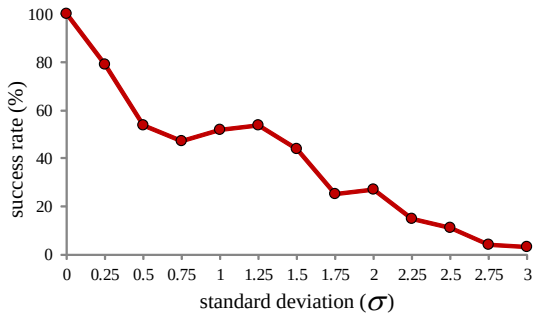

Figure 5.13 The success rate of the caging-based object transportation with Gaussian noise. The success rate dropped sharply under 80% if the standard deviation is more than 0.25.

Chapter 6

Practical Experiments

6.1 Experimental Environments

We conducted practical experiments to verify the proposed techniques. The size of the experimental space is 140cm×140cm. We use two types of differential-drive robots in the experiments: the E-puck and the Elisa-3 types. The maximum velocities of the E-puck and Elisa-3 robots are 2 cm/s and 6 cm/s, respectively, and the diameters of these robots are 7 cm and 5 cm, respectively. These robots are appropriate for the experiments of multi-robot team because the costs of robots are relatively low; the E-puck is $1020 and Elisa-3 is $390 [62].

In the active object transportation, the E-puck robots are used as supervisor, pusher, and puller robot. The pusher and puller robot can localize the active object using a speaker and three microphones. A sound wave at 1

kHz is emitted from the E-puck robot and the maximum acquisition speed of microphones is 33 kHz. A wireless camera (320×240 pixels) is attached to the supervisor robot, as shown in Fig. 6.1(a). A Velcro replaces the robot manipulator for simple grasping, as shown in Fig. 6.1(b). In such a situation, the accuracy of the proposed sound localization method can be highly dependent on the hardware, i.e., the performance of the microphones, the geometrical array of the microphones, and the type of sound signal. These factors are difficult to change due to the restriction of the robot platform. In addition, the distance between the microphones is relatively short, at 6.2 cm, which leads to low localization accuracy because the sound localization method using the ILD has a wide candidate region when a small difference in the sound levels exist according to (3.3). To solve these problems related to the limited hardware specifications, we installed an absorption wall (4.4cm× 6cm×8cm) onto the puller and pusher robots, as shown in 6.1(b). The absorption wall makes the ILD high artificially, which raises the accuracy of the sound localization process. This additional equipment is considered as a component of our system.

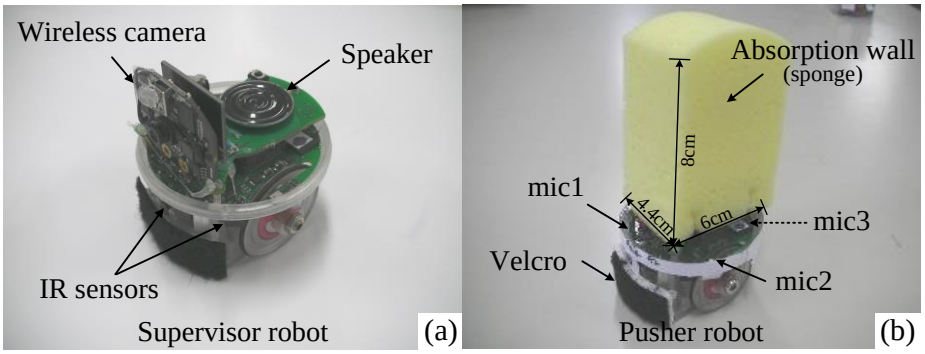

Figure 6.1 The E-puck robots were used in the experiment of active object transportation. A speaker and three microphones were used for localization.

In the passive object transportation, the E-puck robots are used as leader and pusher robots, and the Elisa-3 robots are used as guider robots. A shovel is attached onto the front of the pusher robot to collect objects easily, as shown in Fig. 6.2(a). However, the pusher robot does not grasp or attach onto the objects using the shovel. A paper cup, two ping-pong balls, and an L-shaped object are used as the manipulated objects. The ping-pong balls are used to verify multiple and movable object transportation and the L-shaped object is used to verify large-object transportation. We attached rectangular ID tags (6cm×6cm) onto each robot for position tracking. The tracking system captures all ID tags in real time using an overhead camera and acquires the positions of the robots. The localization errors of our tracking system are described in Table 6.1. The E-puck and Elisa-3 robots have eight infrared radio sensors in various places on their bodies, but the sensing range is less than 6 cm. This range is insufficient for detecting the other robots in our

experimental environment. We, therefore, had help from the position information acquired by the overhead camera, but it was assumed that all robots have a limited sensing range and the ability to communicate in the practical experiment. Thus, we used only partial information from among all localization data, which means that the experimental environment satisfied the condition of decentralized system. All of the robots and objects used in this experiment are shown in Fig. 6.2.

TABLE 6.1 THE LOCALIZATION ERRORS OF POSITION TRACKING SYSTEM

Coordinate	Mean (cm)	Standard deviation (cm)
x	0.77	0.68
y	0.95	0.94

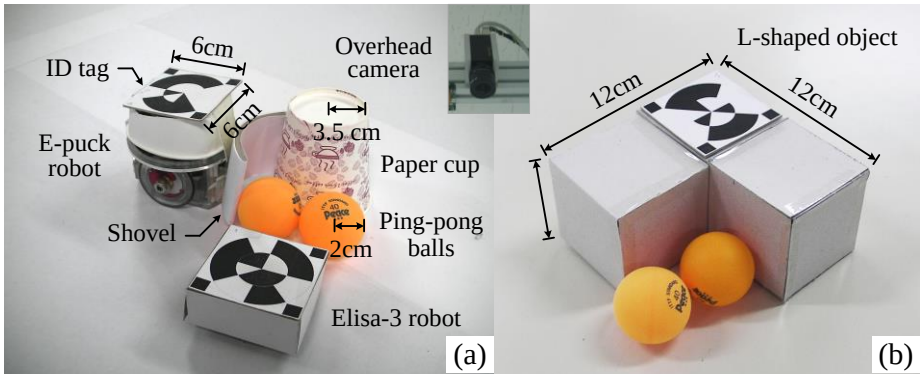

Figure 6.2 Two types of differential-drive robots were used in the experiment of passive object transportation: the E-puck and Elisa-3 robots. A paper cup and two ping-pong balls were used as objects. The ID tags were attached to each robot for position tracking.

6.2 Experimental Results of Active Object Transportation

The experiments of active object transportation are divided into two sections depending on their purpose. First, we investigated the estimation of the SV by varying the distance and direction. Second, we described and analyzed the object transportation processes: approaching to the active object, assigning the positions of the robots, and manipulating the active object.

6.2.1 Experimental Result of the SV Estimation

We estimated the direction angle of the SV by varying the distance and angle from sound source. For example, we measured all sound levels from the microphone by varying from 10cm to 50cm but maintaining an angle of 180°, as shown in Table 6.2 at the first row. As the travelled distance increased, the influence of the noise increased. Thus, the CR was getting wider because the radius of SC increased rapidly thorough small level differences at a long distance; the error of the direction angle of SV increased. More seriously, the accuracy of the direction angle decreased when the ‘left-right’ distinction of the location of the sound source was wrong. The SI can be added not only on one side but also on the other side. The other side can be used to calculate an inaccurate CR with the wrong data. In real experiments, this confusion appeared at a measuring distance of more than 30cm; the accuracy of the direction vector decreased rapidly, as shown in Table 6.2 at the fourth and fifth columns.

TABLE 6.2 ESTIMATION RESULTS OF SOUND VECTOR'S DIRECTION

Angle \ Distance	10cm	20cm	30cm	40cm	50cm
180°	176.9 (3.1)	162.1 (17.9)	159.8 (20.2)	151.1 (28.9)	135.8 (44.2)
210°	206.1 (3.9)	199.8 (10.2)	192.3 (17.7)	176.5 (33.5)	179.5 (30.5)
240°	233.1 (6.9)	229.7 (10.3)	223.0 (17.0)	217.9 (22.1)	169.9 (70.1)
270°	220.4 (49.6)	249.0 (21.0)	226.0 (44.0)	230.3 (39.7)	213.7 (56.3)

(The value inside the parentheses shows the estimation error of the real SV's direction angle)

6.2.2 Experimental Result of Active Object Transportation

The active object transportation begins from the sound signal detection. An active object emitted a sound signal, and a puller robot detected this sound signal; then, it approached the active object, as shown in Fig. 6.3(a) (*Move – to – sound*). As we already mentioned in Section 3.2.3, the heading direction of puller robot continuously changed at a relatively long detecting distance because of the SV estimation errors. Although the direction of the SV was not correct, the distinction of left and right was reliable. Therefore, the wrong path by trial and error could be corrected. In addition, as the sound signal came closer, the accuracy of the estimation became higher according to the result of the previous section. Thus, the puller robot succeeded in approaching the active object. A supervisor robot watched these processes, and transmitted to user from wireless camera, as shown in Fig. 6.3(b).

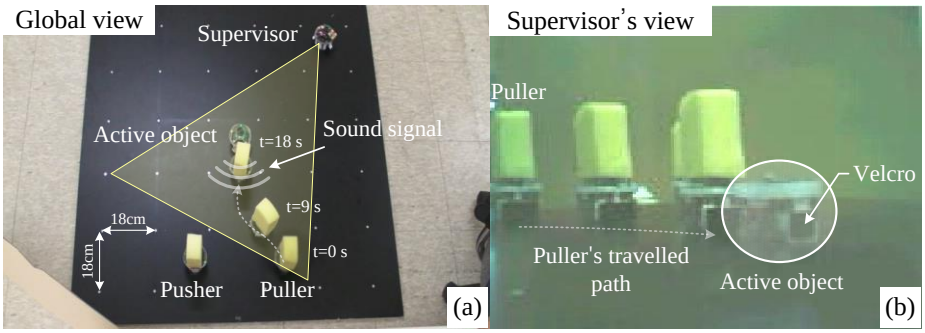

Figure 6.3 The approaching step of puller robot to an active object (a) A puller robot approached the active object emitting a sound signal. (b) This process was transmitted to the user from the supervisor robot.

When the puller robot arrived at the active object, the next step was to move in front of the active object, as shown in Fig. 6.4(a). For travelling around the active object, the puller robot adjusted its heading direction to the right. Then, the puller robot moved in front of the active object maintaining a regular distance ($Follow-in-contact$). If the puller robot was located in front of the active object, then the puller stopped and contacted the active object using the Velcro ($Aligned-the-direction$). In Fig. 6.4(b), the right graph illustrates the IR sensor data when the puller robot moved around the active object. When the puller robot started to move to s_O^3 at 18 second, the puller robot moved to the right around the active object and the measuring distance decreased in the order of s_O^2 and s_O^1. Finally, when the puller robot arrived at the front of the active object, the s_O^0 decreased and then the puller robot stopped. The pusher robot followed the same procedure as the puller robot except moving around the active object. The whole processes took 27 seconds.

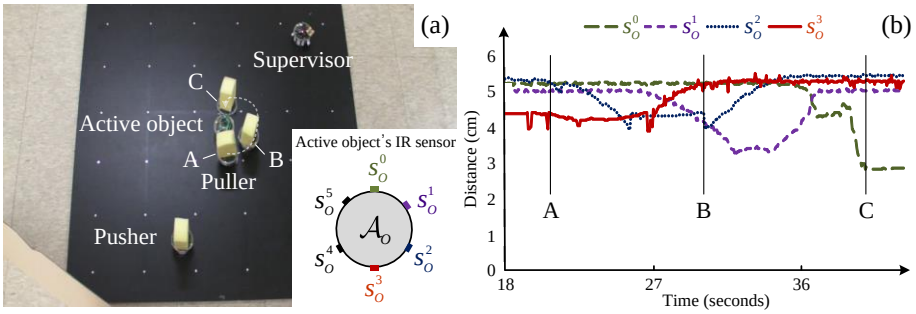

Figure 6.4 The puller robot moved in front of the active object. (a) The whole processes. (b) Variation of the IR sensor data according to the puller's movement. The puller robot turned right at the A, the puller robot was located to the right side of active object at the B, and the C is complete time that the puller robot contacted the front of the active object. The maximum sensing range is 5.5 cm.

If the pusher and puller robot were ready to escort the active object, then they applied force to the active object, as shown in Fig. 6.5. The puller robot adjusted the heading direction and pulled the active object by detecting the sound signal ($Move-to-goal$). The pusher robot applied force to help move the active object. The two robots headed for the direction which was generated by the supervisor's sound signal. The transportation processes updated continuously until the active object was in the goal region. In the experiment, we decided the radius of goal region as 9 cm in consideration of the radius of robot and marginal distance.

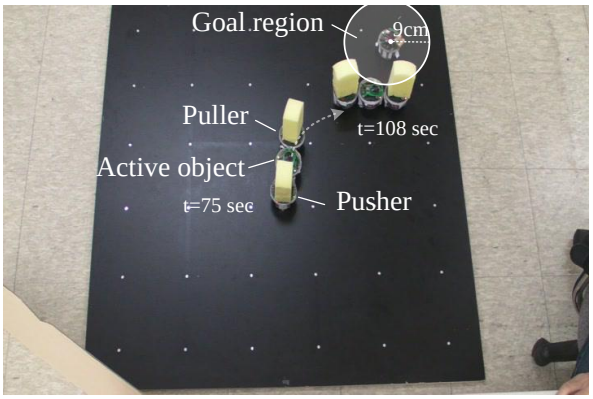

Figure 6.5 The multi-robot team manipulated the active object to goal region. When the active object was in the goal region, the object transportation was completed.

Success rate according to step is illustrated in Table 6.3. We attempted the same experiment 30 times, and measured the travelled time of a pusher robot. The success rate of the $Move-to-sound$ and $Aligned-the-direction$ commands were high because turning around at present place was not difficult. However, the success rate of $Move-to-goal$ was relatively low, because the puller robot could not change heading direction easily during pulling the active object. Consequently, even though each step has relatively high success rate, total successful rate was 60%.

TABLE 6.3 SUCCESS RATE ACCORDING TO STEP

Step	Travelled time (seconds)	Success rate
$Move-to-sound$	18	28 out of 30 (93%)
$Follow-in-contact$	27	25 out of 30 (83%)
$Aligned-the-direction$	3	28 out of 30 (93%)
$Move-to-goal$	33	21 out of 30 (70%)
Total success rate		18 out of 30 (60%)

(Travelled time is relevant to the pusher robot only)

6.3 Experimental Results of Passive Object Transportation

For verifying the passive object transportation technique, we conducted on three kinds of practical experiments according to following conditions: straight path, curved path, and large-object transportation. We can verify the path generation technique using the VEDF and the realization of the FSM through the straight path. The curved path is used to verify the path modification of the rotational VEDF. Finally, more than two leader and pusher robots can be used for the large-object transportation. The large-object transportation experiment is conducted to verify the communication method and position assignments between robots.

6.3.1 Small-object Transportation with Straight Path

The first experiment involved small-object transportation with a straight path, as shown in Fig. 6.6. We used two E-puck robots as the leader (L) and as the pusher robot (P). Also, eight Elisa-3 robots were used for the guider robots (G1~G8). The E-puck robot is larger and stronger than the Elisa-3 robot; thus, the E-puck robot is more appropriate as a pusher robot which should have the power to manipulate multiple objects. The initial and goal position of the objects were (110, 95) cm and (25, 95) cm, respectively. Initially, a pusher robot (P) pushed multiple objects in the pushing state from behind until the pusher robots were located between the guider robots (G2 and G6) for 7

seconds, as shown in Fig. 6.6. At the same time, two tail robots (G1 and G5) followed each row in the following-in-contact state for 18 seconds. The moving guider robots (G1 and G5) detected the head robot (G4 and G8) within the maximum sensing range ($s_{\max} = 18\,\text{cm}$) at 18 seconds, which triggered a change to the lining-up state. The guider robots generate a VEDF ($\varphi = 0°$) with respect to the two head robots in each row and followed it. If the moving guider robots (G1 and G5) arrived at the head robots (G4 and G8), the guider robots became the new head robots. In addition, the leader robot (L) approached the goal in the moving-to-the-goal state and had their positions reassigned according to midpoint between the head robots, as shown in Fig. 6.6, at 34 seconds. All robots repeated these processes until all of the objects were successfully manipulated to the goal. The bending angle of the first and the second rows were -2.1° and 1.1°, respectively, which means that the guider robots lined up with a nearly straight path. The total travel time was 275 seconds and the average travel distance of the eight guider robots was 278 cm. The trajectories of the robots are illustrated in Fig. 6.7.

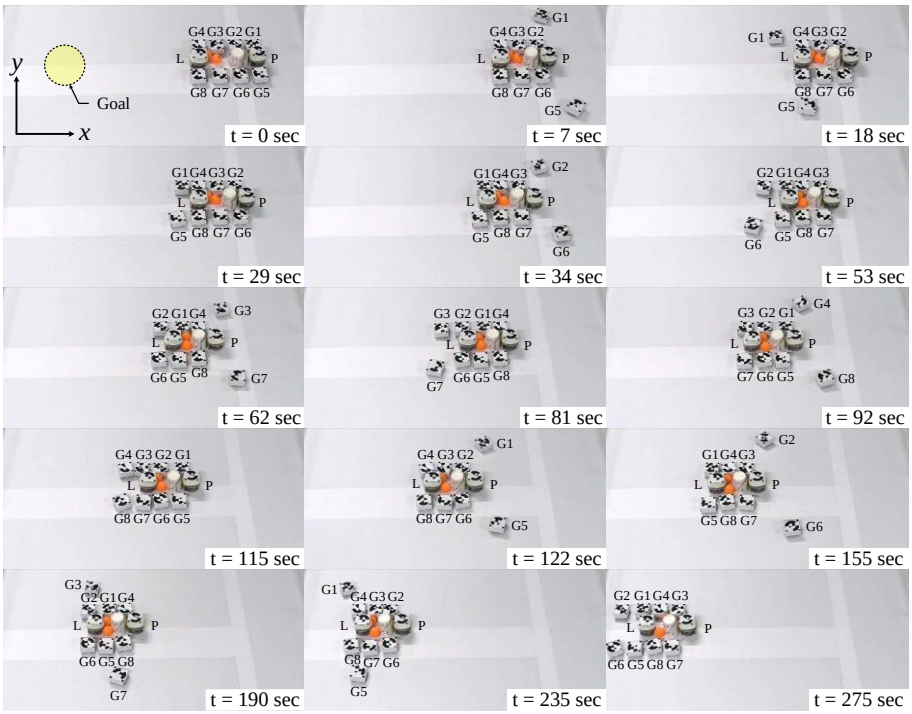

Figure 6.6 Multi-object transportation with a straight path. Three objects, including rolling objects, were transported successfully in 275 seconds.

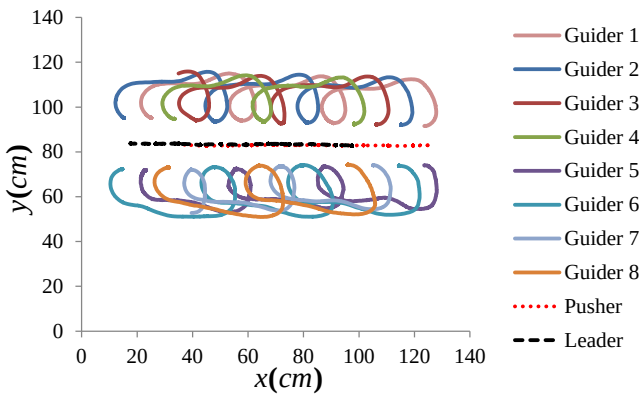

Figure 6.7 The trajectories of the robots during the transportation of small-object with a straight path.

The above experiment was conducted in our position tracking system which has the localization errors as described in Table 6.1. However, the localization techniques can be changed or can acquire inaccurate information according to experimental environments. Therefore, we added artificial Gaussian noise to our tracking system for verifying the robustness of our system. Three cases of experiments were conducted by varying localization errors assumed to have zero-mean Gaussian noise with standard deviation from 1 to 2 with 0.5 intervals. Each experiment has 10 trials, and the total trials are 30 times. Table 6.4 and Fig. 6.8 show the summary of the experimental results and the trajectories of robots according to localization errors, respectively. The success rate decreased as the standard deviation of Gaussian noise increases. The mean of completion time were not directly related with the standard deviation of Gaussian noise, but the variation of completion time increased when the standard deviation of Gaussian noise is large. This is because the robots showed erratic motion due to large localization error, as shown in Fig. 6.8. The guider robots showed unnecessary motion in the following-in-contact state and they could not stand in a line straight in the lining-up state, as shown in Fig. 6.8(c).

TABLE 6.4 THE RESULTS ACCORDING TO LOCALIZATION ERRORS

Standard deviation (cm) (Gaussian noise with zero-mean)	Successful trials	Completion time (seconds)	
		Mean	Standard deviation
1	8 out of 10	326.3	16.4
1.5	7 out of 10	332.0	30.8
2	6 out of 10	315.3	27.0

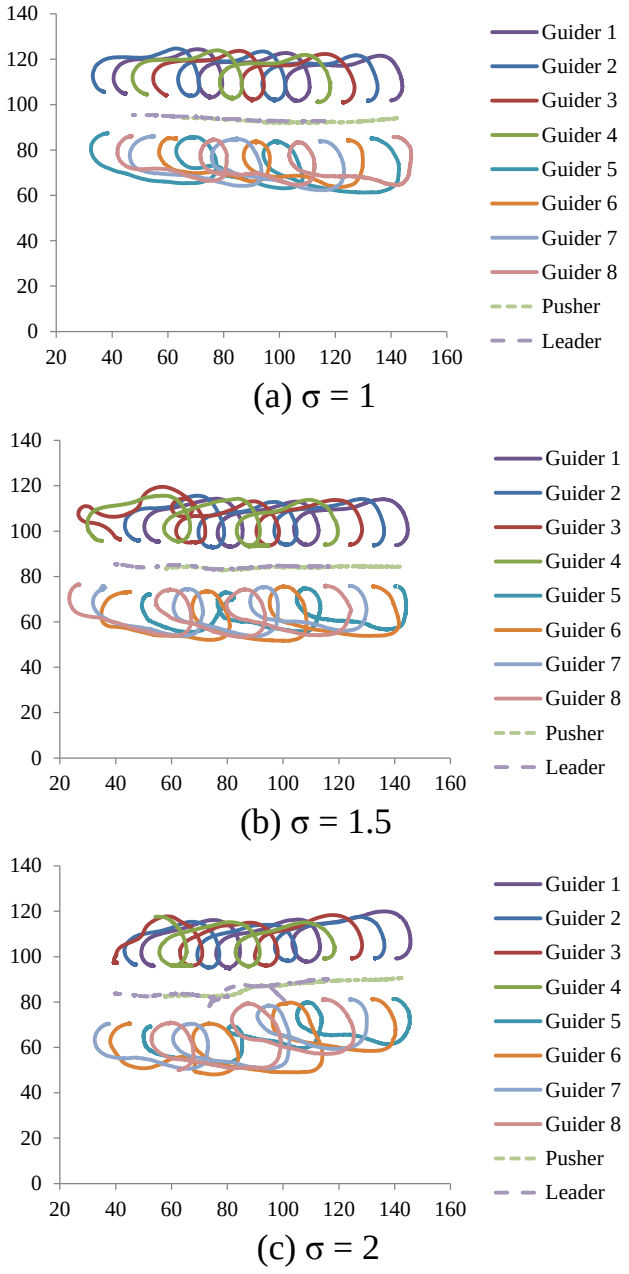

Figure 6.8 The trajectories of robots according to localization errors with zero-mean Gaussian noise. The standard deviation (σ) has an effect on the trajectories of robots and the success rate of transportation. (a) σ=1 (b) σ=1.5 (c) σ=2.

6.3.2 Small-object Transportation with Curved Path

The second experiment involved small-object transportation with a curved path. In this experiment, the initial and goal position of the objects were (120, 80) cm and (30, 50) cm, respectively. The basic transportation process was identical to the previous object transportation experiment with a straight path, except for the rotational angle of lining up. In this experiment, we determined the rotational angle as 30°. Thus, moving guider robots (G1 and G5) approached the head robots (G4 and G8, respectively) from a 30° direction when they executed the lining-up state, as shown in Fig. 6.9 at 47 seconds. The VEDF was generated using (4.3), and the rotational angle φ was 30°. The pusher robot pushed objects in the bending direction because the parallel row formation of the guider robots was bent to 30° direction. The bending angles of the first and the second rows were 22.2° and 31.2°, respectively. This means that the guider robots in the first row did not line up 30° direction than the guider robots in the second row because of the movement error of robots. The total travel time was 400 seconds and the average travel distance of the eight guider robots was 479.4 cm. The trajectories of the robots are illustrated in Fig. 6.10.

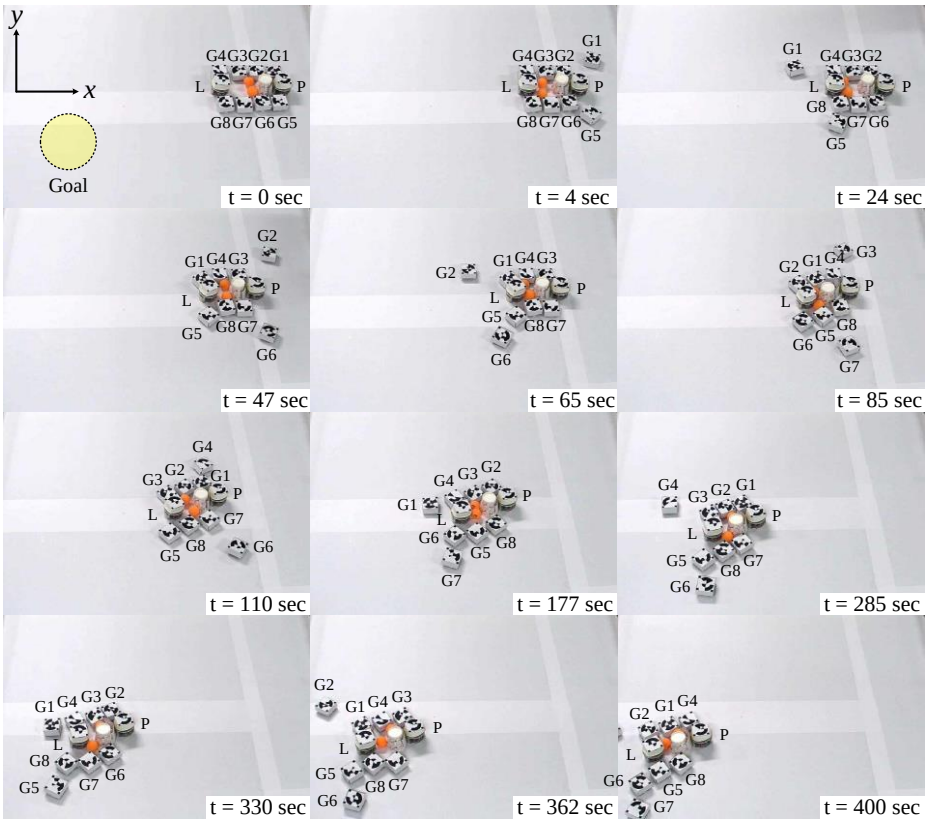

Figure 6.9 Multi-object transportation with a curved path. Three objects, including rolling objects, were transported successfully in 400 seconds.

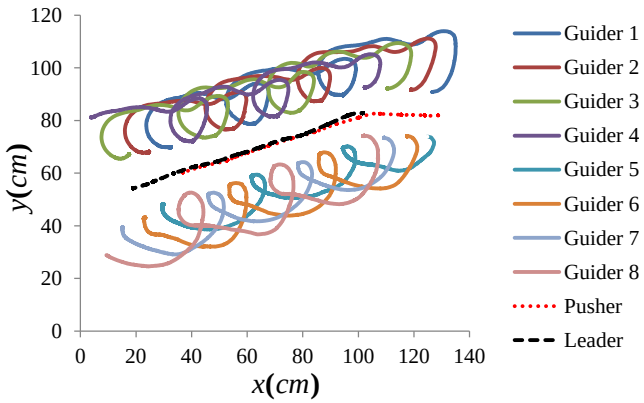

Figure 6.10 The trajectories of the robots in the transportation experiment with a curved path ($\varphi = 30°$)

6.3.3 Large-object Transportation

The last experiment assessed large-object transportation. Unlike the previous experiments, multiple leader and pusher robots were used for the large-object transportation. We used four E-puck robots for the two leader robots (L1 and L2) and for the two pusher robots (P1 and P2), respectively. Also, ten Elisa-3 robots were used for the guider robots (G1-G10). The initial and goal position of the L-shaped object were (30, 110) cm and (100, 70) cm, respectively. In this experiment, identical FSMs with small-object transportation were applied, except for the communication method between the pusher robots and the leader robots, respectively. For example, two pusher robots share the orders of the left and right guider robots through the local communication. In Fig. 6.11 at 0 seconds, the P1 robot not only recognizes the order index of G1 but can also detect the order index of G6 through communication with the P2 robot. Likewise, the P2 robot can detect the order index of G1 through the P1 robot. The two leader robots can also share their information through communication. The large L-shaped object was transported by multiple pusher robots and leader robots using this method. The total travelling time for large-object transportation was 180 seconds. The trajectories of the robots are illustrated in Fig. 6.12.

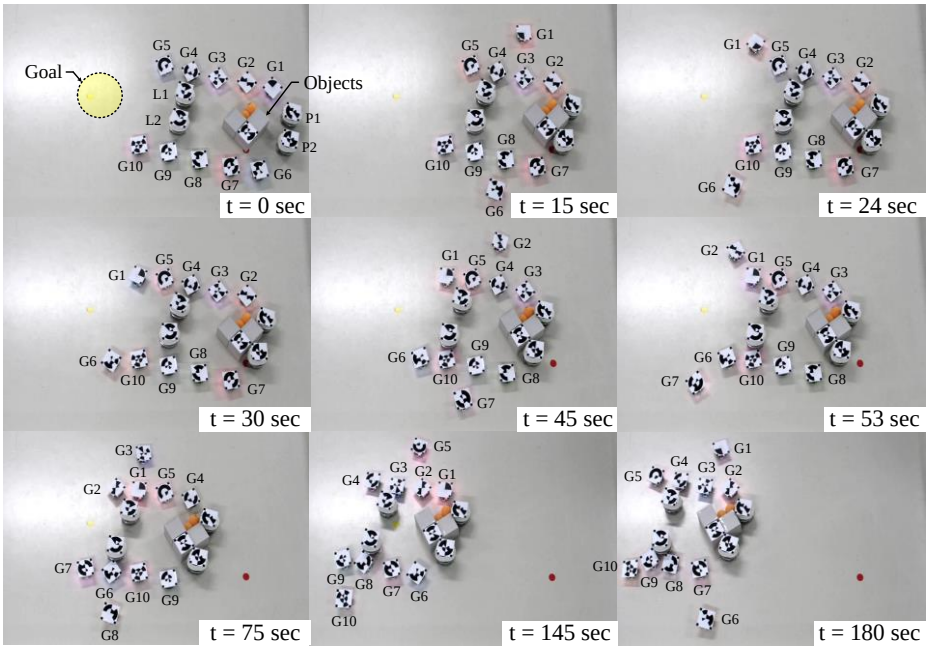

Figure 6.11 Large-object transportation using multiple pusher robots and leader robots. Three objects, including large or rolling objects, were transported successfully in 180 seconds.

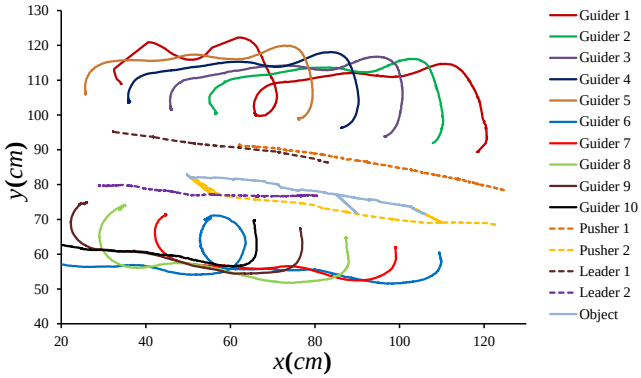

Figure 6.12 The trajectories of the robots during large-object transportation

6.4 Comparison Result with Caging Technique

We have conducted on comparative experiment for showing the strength of proposed technique, as shown in Fig. 6.13. The comparative method is the caging technique [16, 17]. Five E-puck robots were used for caging because the five is suitable for caging condition by (5.1). The initial position of L-shaped object was (101, 71) cm and the position of goal was (31, 110) cm. The L-shaped object was successfully transported by caging technique. Two ping-pong balls, however, escaped from the robot team at 24 and 56 seconds, respectively, because multiple robots took only the L-shaped object into account for guaranteeing caging condition. As a result, the caging technique was inappropriate for multi-object transportation because this technique was not originally developed for the multi-object transportation.

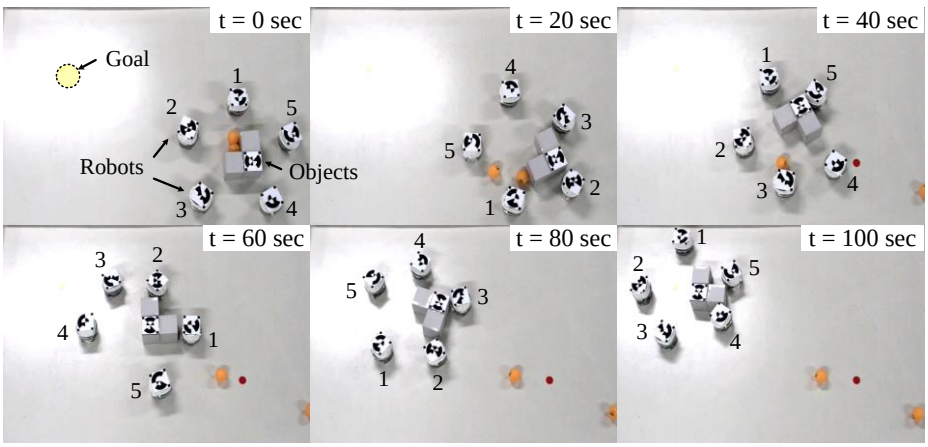

Figure 6.13 Five E-puck robots twirl and transport the multiple objects in the caging technique. An L-shaped object was successfully transported by robots. Two ping-pong balls, however, escaped from robot team at 24 and 56 seconds because multiple objects were not considered in the caging technique.

Chapter 7

Discussion

Various object transportation techniques by multi-robot team have been presented many researchers because of the advantages of cooperation. However, most of previous studies needed the frequent repositioning of robots or additional tools for object transportation [11]-[15]. Also, rolling or multiple objects could not be transported. We, therefore, proposed two object transportation techniques for solving the above problems: active and passive object transportations.

The active object transportation technique is a new method using interactive communication between robots and an object. Previous studies have considered objects as only something to transport; they did not consider that acquired information from objects' sensors is used for transportation. We, however, used the sensing information and sound signal from an object for

localization and the repositioning of robots, as described in Section 3. This technique can be realized using relatively low cost robot ($1020) and low performance of microphones (33kHz) [62]; it is feasible to apply a multi-robot application. The active object transportation technique has great significances that the robot positioning and localization problems of object transportation are solved by viewing objects from a different angle.

In the passive object transportation technique, robots can transport multiple objects without any tools. Multiple objects are manipulated through two lining-up row made by robot formation. This technique is useful to transport multiple objects at the same time by limiting the motion of objects to one side. In addition, this technique is based on decentralized system and needs not the information of objects; it is suitable to apply to a real environment. The proposed technique was verified diverse simulations (Section 5.2) and practical experiments (Section 6.3) in this dissertation; multiple and rolling objects could be transported with straight and curved paths. This technique can be applicable to specific fields such that there is insufficient information about objects or multiple objects should be transported at the same time. For example, an unmanned garbage collector and freight transportation in indoor environment are appropriate for applying the proposed technique.

Nonetheless, there are some issues for future works. First, the accuracy of localization should be improved by considering reverberation and noise in the

active object transportation technique. Our experiments were conducted in the environment for which it can be minimized the effect of reverberation. However, there is much noise in a real environment; there is a need to improve sound localization method by considering errors. Second, an approaching method to objects should be presented in the passive object transportation. For applying our technique, it is necessary to form a regular formation in advance of the transportation; this process was omitted in our technique. Appendix A shows an example about this process but it needs an additional assumption that all robots share their positions. Third, the proposed technique should be applied in a static environment where multiple obstacles exist. The consideration of obstacles is essential for applying the passive object transportation technique to a real situation. A brief method for solving this problem is presented in Appendix B. Finally, mathematical analysis for diverse robot formation is required for stable and efficient object transportation.

Chapter 8

Conclusions

This dissertation presented two object transportation techniques using cooperative robot behaviors. We classified the objects according to their characteristics: active and passive objects. The active object was manipulated using sound-based localization technique and interactive communication. The passive objects were transported using a virtual electric dipole field and finite state machines.

The results of active object transportation indicated that an active object can be retrieved by robots using a cooperative control scheme. The robots each generate a sound vector to identify the position of the active object with a sound signal. A combination of a sound isocontour and a sound circle was used for the generation of the sound vector, and the sensing data from the active object were used to assign the positions of the robots. The proposed

technique used interactive real-time communication between the active object and the robots for data sharing.

In the passive object transportation, multiple objects were manipulated by a decentralized robot team using parallel row formation with cyclic shift motion. The proposed multi-robot team consists of a leader, a pusher, and guider robots; the leader robot has a global path planner and leads the team, the pusher robot pushes objects from behind, and the guider robots create a parallel row formation using the virtual electric dipole field to guide the objects. The main contribution of this technique is that multiple robots create an extended parallel row toward the goal using cyclic motion and transport objects using the finite state machine of each robot. Multiple objects could be transported successfully without additional tools, and the shape information of the objects is unnecessary for transportation. The kinematic and dynamic constraints for a two-wheeled differential-drive robot were also satisfied. The proposed technique can be applied to various real object transportation applications, such as a garbage collector, foraging, a tennis ball collector and transport vehicles.

Appendix A: The Approaching Phase of Passive Object Transportation

The symmetrical formation was assumed to be prearranged before passive object transportation in the proposed technique. However, robots should approach multiple objects for transportation. We, therefore, present the approaching method to the objects, which is defined as *approaching phase*. For approaching to the objects, we modify the assumption which was described in Section 2.3; all robots know the positions of objects in everywhere.

A.1 Approaching Phase

All robots should be gathered around multiple objects and make two rows to manipulate them. To do this, the robots approach the objects by following the VEDF as generated in Section 4.3.2. The origins of the vector fields are the center of the head robots in each row.

There, however, are no head robots initially. Thus, we adopt two *virtual robots* for initial references of the VEDF. We assume that virtual robots exist near the objects and follow the VEDF which is generated with respect to the virtual robots. The virtual robots are generated by the following process. First,

we consider multiple objects as a single box by approximation, as shown Fig. A.1(a). The single box is termed *object box*, whose height is x_h and width is x_w. Second, we define two virtual robots whose positions are χ-away on the y-axis from two right vertices of the object box. Finally, guider robots follow the VEDF which are generated with respect to the virtual robots.

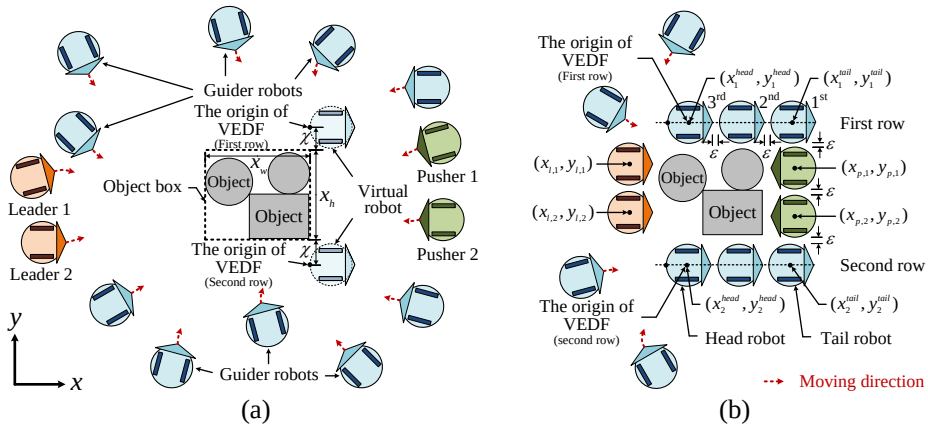

Figure A.1 The robot motions of the approaching phase. (a) Initially, guider robots follow arbitrary VEDFs generated with respect to two virtual robots. (b) Guider robots approach the objects using the controller (A.6) and leader and pusher robots move using the controller (A.15). The representative leader robot counts the number of guider robots in each row and distributes the guider robots uniformly by changing the following VEDF of the guider robots. The pusher and the leader robots move to the middle points between the tail robots and the middle points between the head robots using internal division formula, respectively.

The guider robots should avoid collisions with other robots and objects during the approaching phase. The potential field method is used to avoid collisions [51]. The magnitude of the repulsive force with respect to the k^{th} robot is given as

$$F_k^r(x,y) = \zeta\left(\frac{1}{(x-x_k)^2 + (y-y_k)^2}\right), \tag{A.1}$$

where (x_k, y_k) is the position of the k^{th} robot and ζ is a positive constant value to adjust the force ratio. The angle of the repulsive force with respect to the k^{th} robot is given as

$$\theta_k = \tan^{-1}((y_k - y)/(x_k - x)). \tag{A.2}$$

The repulsive force with respect to the k^{th} robot is described as

$$f_k^r(x,y) = \begin{bmatrix} F_k^r \cos\theta_k \\ F_k^r \sin\theta_k \end{bmatrix}, \tag{A.3}$$

and total repulsive force of the i^{th} robot is given as

$$f_{i,rep}^r(x,y) = \sum_{k\neq i}^{n} f_k^r(x,y), \tag{A.4}$$

where n is the number of other robots. The repulsive force by the objects is analogous to (A.4). If m is the number of objects, the repulsive force of the i^{th} robot with respect to all objects is given as

$$f_{i,rep}^o(x,y) = \sum_{k}^{m} f_k^o(x,y), \tag{A.5}$$

where $f_k^o(x,y)$ is the repulsive force with respect to the k^{th} object.

The controller of the i^{th} guider robot $u_{i,j}^g$ is derived from the combination of (4.3), (A.4) and (A.5). This formula is given:

$$u_{i,j}^g(x,y) = \tilde{\mathbf{F}}_e(x,y) + f_{i,rep}^r(x,y) + f_{i,rep}^o(x,y), \tag{A.6}$$

where the origin of the VEDF is the tail of the j^{th} robot. Each guider robot lines up one by one according to (A.6) until all guider robots completed two rows, as shown in Fig. A.1(b).

The leader and pusher robots move to the region between the head robots and the region between the tail robots, respectively, by means of attractive force, as shown in Fig A.1(b). The magnitudes of the attractive forces are proportional to the distance between the robots as follows:

$$F_k^l(x,y) = -\xi\{(x - x_k^l)^2 + (y - y_k^l)^2\}, \tag{A.7}$$

$$F_k^p(x,y) = -\xi\{(x - x_k^p)^2 + (y - y_k^p)^2\}. \tag{A.8}$$

Here, k is the index of the leader and pusher robot, ξ is the proportional positive constant, and the negative sign ahead of ξ denotes the attractive force. The coordinates (x_k^l, y_k^l) and (x_k^p, y_k^p) are the desired positions of the leader and pusher robots, respectively. The desired positions of the leader and pusher robots are related to the number of leader and pusher robots for manipulation. Thus, we apply an internal division formula,

$$\begin{bmatrix} x_k^l \\ y_k^l \end{bmatrix} = \frac{1}{n_l + 1}\begin{bmatrix} (n_l - k + 1)x_2^{head} + kx_1^{head} \\ (n_l - k + 1)y_2^{head} + ky_1^{head} \end{bmatrix}, \tag{A.9}$$

$$\begin{bmatrix} x_k^p \\ y_k^p \end{bmatrix} = \frac{1}{n_p + 1}\begin{bmatrix} (n_p - k + 1)x_2^{tail} + kx_1^{tail} \\ (n_p - k + 1)y_2^{tail} + ky_1^{tail} \end{bmatrix}. \tag{A.10}$$

where (x_1^{head}, y_1^{head}), (x_2^{head}, y_2^{head}), (x_1^{tail}, y_1^{tail}) and (x_2^{tail}, y_2^{tail}) are the positions of the head and tail robots of the first and second rows, respectively. The values of n_l and n_p correspondingly indicate the number of leader and pusher robots. Also, the directions of the attractive forces are given as

$$\theta_{l,k} = \tan^{-1}((y_k^l - y)/(x_k^l - x)), \tag{A.11}$$

$$\theta_{p,k} = \tan^{-1}((y_k^p - y)/(x_k^p - x)). \tag{A. 12}$$

The attractive forces of the leader and pusher robots are acquired by (A. 7-8) and (A. 11-12) as follows:

$$f_{att}^{l,k}(x,y)=\begin{bmatrix} F_k^l \cos\theta_k^l \\ F_k^l \sin\theta_k^l \end{bmatrix}, \tag{A.13}$$

$$f_{att}^{p,k}(x,y)=\begin{bmatrix} F_k^p \cos\theta_k^p \\ F_k^p \sin\theta_k^p \end{bmatrix}. \tag{A.14}$$

The controllers of the leader and pusher robots, therefore, are given by combining (A.4-5) and (A.13-14):

$$\begin{bmatrix} u^{l,k} \\ u^{p,k} \end{bmatrix}=\begin{bmatrix} f_{att}^{l,k}(x,y)+f_{l,rep}^{r}(x,y)+f_{l,rep}^{o}(x,y) \\ f_{att}^{p,k}(x,y)+f_{p,rep}^{r}(x,y)+f_{p,rep}^{o}(x,y) \end{bmatrix}. \tag{A.15}$$

The numbers of leader and pusher robots should be sufficient to block the front and behind to wrap multiple objects. This condition is expressed as

$$\varepsilon < d_{\min}^{o}, \quad n_l > \frac{d_{\max}^{o}-\varepsilon}{2r+\varepsilon}, \quad n_p > \frac{d_{\max}^{o}-\varepsilon}{2r+\varepsilon}, \tag{A.16}$$

where $d_{\min}^{o}$ and $d_{\max}^{o}$ are the diameters of the minimum and maximum object, respectively.

Equation (A.16) is only valid when the number of guider robots in two rows is identical because more leader or pusher robots will be necessary when different numbers of guider robots are used in each row. In an extreme case, object transportation is impossible if most of the guider robots crowd into a specific line. Therefore, an equal distribution of guider robots with respect to each row is necessary. The process of creating an equal distribution of guider robots is described as follows:

1) Initially, guider robots choose and move along arbitrary VEDF which are generated with respect to two virtual robots.
2) Guider robots send a message to a leader robot regarding the selection of a VEDF.
3) The leader robot counts the number of guider robots (n_1 and n_2) with respect to each row when messages are received from the guider robots. If guider robots crowd the specific row, the leader robot orders further guider robots to change the following VEDF. The changing condition is $|n_1 - n_2| > \kappa$, where κ is the boundary criteria of congestion. The value of κ is normally 0 or 1 for an equal distribution.
4) The index of the guider robots is gradually increased from the last to first line because a new indexing number is generated when the tail robot arrives at the head robot.

One of leader or pusher robots becomes a representative when multiple leader or pusher robots are used. The representative leader robot collects data and determines how to arrange the positions of the guider robots. The guider robots make two rows by repeating processes 1) to 4) above. If all robots complete the gathering and arrangement process, the approaching phase is completed.

A.2 Experimental Result of Approaching Phase

In the approaching phase, two virtual robots whose positions were (112, 86) cm and (99, 62) cm generated two VEDF, and two guider robots (G1 and G6) moved to the area behind the virtual robots according to (A.6), as shown in Fig. A.2 at 15 seconds. If the two guider robots arrived at the virtual robots within 4 cm, the guide robots stopped and became new head robots; the value of 4 cm is the desired marginal space ε between the robots. Two other guider robots (G2 and G7) approached the head robots (G1 and G6) using the VEDF, as shown in Fig. A.2 at 30 seconds. If the two guider robots (G2 and G7) arrived at the head robots (G1 and G6), these robots became new head robots because they were located in the first line of the rows. Head robots, therefore, are changed continuously because the definition of a head robot is the first-line robot in each row. All guider robots lined up in two rows by the iteration of the same process. If all guider robots (G1~G10) are completed in terms of their positioning, the pusher and leader robots moved to the middle region between the tail robots and the head robots via (A.15). The positions of the pusher and leader robots were determined by the internal division points between the tail robots and the head robots. We applied internal division ratios of 1:2 and 2:1, respectively, because two pusher and two leader robots were used. The approaching phase was completed when all robots did not have to move any longer according to (A.6) and (A.15). The completion time was 78 seconds and the trajectories of robots are described in Fig. A.3.

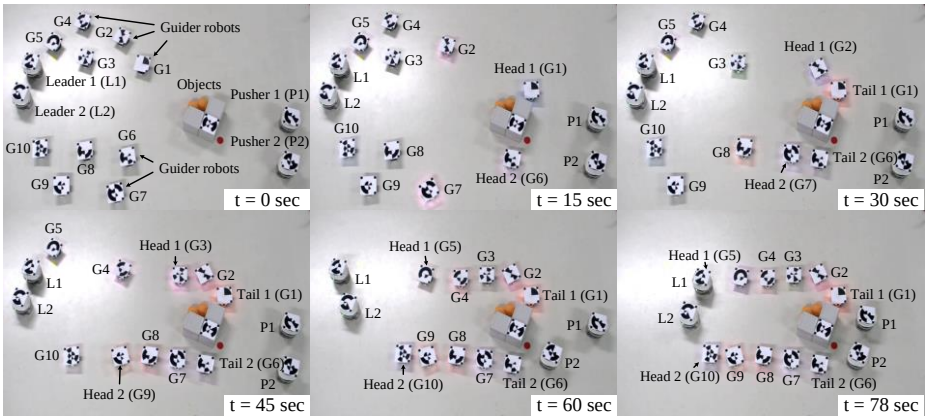

Figure A.2 The experimental result of approaching phase

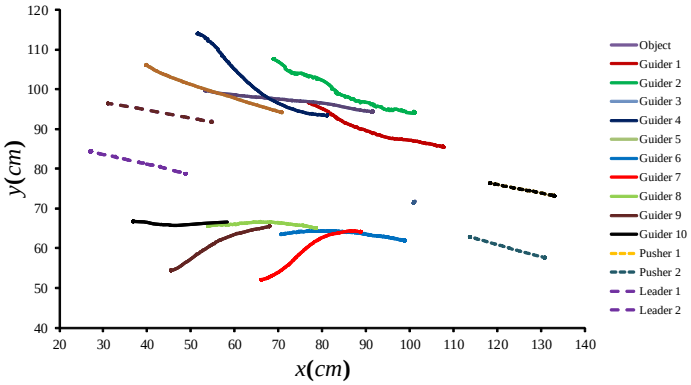

Figure A.3 The trajectories of the robots in approaching phase

Appendix B: Object Transportation in a Static Environment

B.1 Overview

In the previously proposed passive object transportation technique, we assumed that obstacles do not exist for simplifying the problem. However, there are multiple obstacles in real environment, and thus, we should consider these obstacles for successful object transportation. If our previously proposed technique is applied to transport objects without modification in a static environment in which obstacles exist, the guider robot collides with the obstacles, as shown in Fig. B.1(a).

Most previous studies on this problem considered the obstacles as something to avoid only [63]-[65]. They mainly focused on the minimum travelling time and stable motion while the robots avoid the obstacles. Our previously proposed technique can be also applied in a static environment without any modification, as shown in Fig. B.1(b). However, there are two problems as follows. First, a large region needs for object transportation because the leader robot should design the global path by considering the size of obstacles also. Second, there is the limitation of guider robots' motion because of obstacles. The guider robots cannot line up in a row precisely in this case.

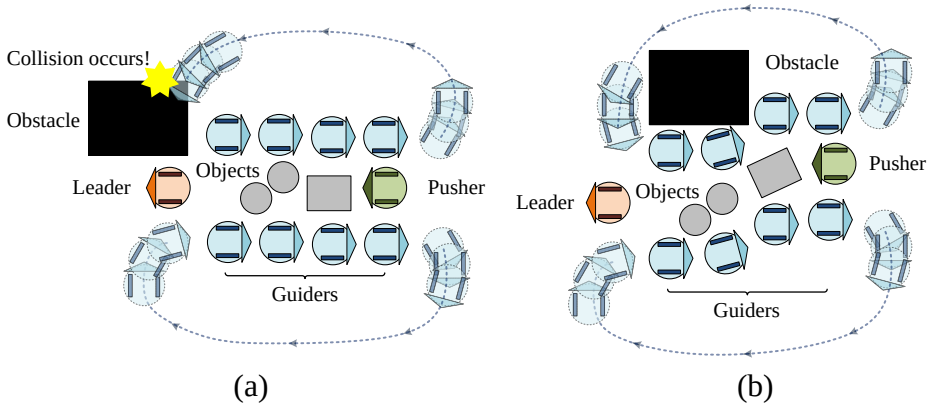

Figure B.1 (a) Collision can occur in a static environment when guider robots approach to the head robot because the obstacles prevent the movement of guider robot. (b) It needs excessive space to transport objects and guider robots are difficult to be located in the desired position.

We, therefore, present a new object transportation technique that obstacles replace the part of guider robots. If the guider robot is located in the next to obstacle, the obstacle is used as a wall by preventing object escaping, as shown in Fig. B.2(a). This method has three advantages by comparison with the previous technique. First, the object transportation is possible using fewer robots because the obstacles replace the role of guider robots which is preventing the objects from escaping. Second, object transportation is possible within small region. The objects can be transported through narrow way using the new method, as shown in Fig. B.2(b). Finally, this new method is robust to sensor inaccuracy and localization errors. It needs not to consider the movement of robot in the row where obstacles are located because the objects cannot escape through obstacles.

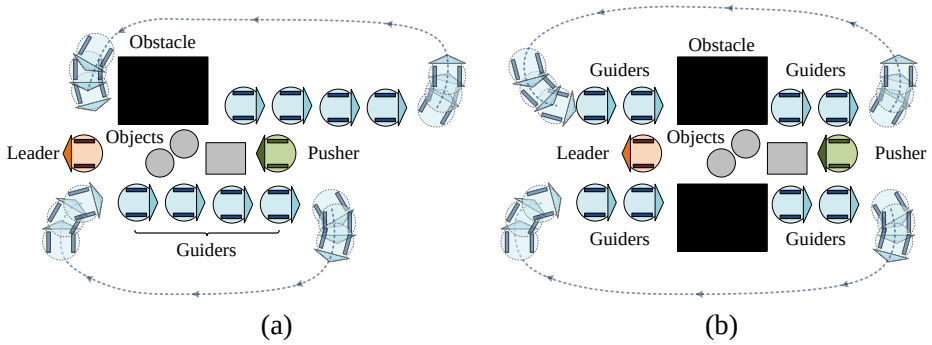

Figure B.2 (a) An obstacle can help object transportation by preventing the escape of objects. (b) Objects can be transported through narrow way using the new method.

B.2 Object Transportation Problem in a Static Environment

The problem and assumptions of object transportation in a static environment are identical with the previous study, as described in (2.1). However, there are three different points by comparison with the previous version. First, all obstacles are convex polygons in which no line segment between two points on the boundary ever goes outside the polygon [66]. In other words, all interior angles of convex polygon are less than 180°. The local minima can occur if obstacles have other shapes, such as concave hull or parabolic shape. Second, a leader robot can communicate with all guider robots in a real time. The leader robot knows the states and lining-up order of guider robots by communicating them, which means that the new proposed technique is based on the hybrid system. The leader robot orders the guider robots to move according to their state. Then, the guider robots move to the

head robot by avoiding other robots and obstacles using their proximity sensor information. Finally, the leader robot has the global path planner considering all obstacles before the object transportation. Various path planner can be used for the global path planning, such as visibility graph [67], A* [68] and Dijkstra algorithms [69]. However, the path planning of multi-robot team is out of scope in this dissertation; we do not describe the path planning methods in detail.

B.3 Multi-object Transportation using Hybrid System

The previously proposed technique should be modified for transporting objects in a static environment where multiple obstacles exist. For the modification, there are three considerations as follows. First, the origin selection method of VEDF should be changed because the back of head robot can be obstacles. In the existing technique, the origin of VEDF was the next of head robot. If a moving guider robot follows this existing VEDF, the guider robot will collide with obstacles, as shown in Fig. B.1(a). Second, the state transition method of guider robots should be changed. In the existing proposed technique, the state transition totally depends on the lining-up order of guider robots by the gradient algorithm. However, the lining-up order cannot be calculated due to obstacles in a static environment. Therefore, the new command architecture needs for the state transitions of guider robots. Finally, the robot formation can be asymmetric by obstacles, as shown in Fig.

B.2(a). Thus, a new formation maintenance method needs because the existing proposed technique considers only symmetric case.

We can solve the above problems by giving more authority to the leader robot as follows. First, the origin of VEDF is assigned with respect to the position of leader robot, not the head robot. The guider robots can generate the VEDF using this method whether the obstacles exist or not. Second, the leader robot orders guider robots to change their states. Finally, a symmetrical formation is regenerated by the command of leader robot after all guider robots pass through the region where obstacles exist.

B.4 New Finite State Machines

In this section, new finite state machines for object transportation are proposed. The state diagrams of the new FSMs are identical to the previous version, as already shown in Fig. 4.7. However, there is a little change in states and events, as shown in Table B.1. The leader robot orders guider robots to change their states. For example, if there is no guider robot in a leader robot's left or right side, the leader robot orders the guider robot which is lined up left or right side to execute following-in-contact state ($\bar{E}_1^G$). In addition, if a guider robot approaches the leader robot within the maximum sensing range $s_{\max}$, the leader robot orders the guider robots to change its state to the lining-up state ($\bar{E}_2^G$). A pusher and a leader robot decide whether their states are changed or not according to their sensor information.

TABLE B.1 THE EVENTS OF THE NEW FSMS IN A STATIC ENVIRONMENT

Robot	Event	Descriptions		Commander
		1st row	2nd row	
Guider	$\bar{E}_1^G$	$s^L_{\lambda(\pi/3)} \geq s_{\max}$	$s^L_{\lambda(-\pi/3)} \geq s_{\max}$	Leader
	$\bar{E}_2^G$	$\| \mathbf{p}^G_{self} - \mathbf{p}^L \| < 2r + s_{\max}$		Leader
	$\bar{E}_3^G$	$\| \mathbf{p}^G_{self} - \mathbf{p}^L_1 \| < s_\varepsilon$	$\| \mathbf{p}^G_{self} - \mathbf{p}^L_2 \| < s_\varepsilon$	Guider
Pusher	$\bar{E}_1^P$	$s^P_{\lambda(\pi/2)} \geq s_{\max}$ or $s^P_{\lambda(-\pi/2)} \geq s_{\max}$ or $\| \mathbf{p}^G_i - \mathbf{p}^P \| > s_{\max}$ for $\forall i$		Pusher
	$\bar{E}_2^P$	$s^P_{\lambda(\pi/2)} < s_{\max}$ and $s^P_{\lambda(-\pi/2)} < s_{\max}$		Pusher
Leader	$\bar{E}_1^L$	$s^L_{\lambda(\pi/2)} < s_{\max}$ and $s^L_{\lambda(-\pi/2)} < s_{\max}$		Leader
	$\bar{E}_2^L$	$s^L_{\lambda(\pi/2)} \geq s_{\max}$ or $s^L_{\lambda(-\pi/2)} \geq s_{\max}$		Leader

B.4.1 The States of Guider Robots

For guider robot, the following-in-contact and stop states are identical to the previously proposed technique, as already described in Section 4.4.2. Therefore, we describe the lining-up state only. The major difference is that guider robots should choose the origin of VEDF according to the position of leader robot. The guider robot which belongs to the first and second row generates two VEDFs with respect to the right ($\mathbf{p}_1^L$) and left reference points ($\mathbf{p}_2^L$) of leader robot, respectively. These reference points are as follows:

$$\mathbf{p}_1^L = \mathbf{p}^L + \begin{bmatrix} (2r+\varepsilon)\cos\left(\gamma^L - \frac{\pi}{2}\right) \\ (2r+\varepsilon)\sin\left(\gamma^L - \frac{\pi}{2}\right) \\ 0 \end{bmatrix}, \quad \mathbf{p}_2^L = \mathbf{p}^L + \begin{bmatrix} (2r+\varepsilon)\cos\left(\gamma^L + \frac{\pi}{2}\right) \\ (2r+\varepsilon)\sin\left(\gamma^L + \frac{\pi}{2}\right) \\ 0 \end{bmatrix}, \quad \text{(B.1)}$$

where $\mathbf{p}^L$ is the position of leader robot. In (B.1), the third element of matrix indicates the rotational angle of VEDF. The rest procedure of lining-up state is the identical to the previous version, and the modified lining-up state algorithm is described in Table B.2.

TABLE B.2 ALGORITHM FOR THE LINING-UP STATE IN A STATIC ENVIRONMENT

Algorithm	Lining-up (*for guider robot*)
Input	$\mathbf{p}^L$: the position of leader robot $\mathbf{p}_1^L$, $\mathbf{p}_2^L$: the desired origins of VEDF with respect to leader robot (1st and 2nd rows)
Output	v^d : the desired tangential velocity of the guider robot ω^d : the desired angular velocity of the guider robot
1:	**loop**
2:	*generate virtual electric dipole field* $\mathbf{F}_e^{rot}$ according to the local coordinate of $\mathbf{p}_1^L$ or $\mathbf{p}_2^L$ with respect to the relative row.
3:	*approach the desired position* using v^d and ω^d by (4.10) and (4.14), respectively

B.4.2 The States of a Pusher Robot

For pusher robot, the stop state is the same as the previously proposed technique, as already described in Section 4.4.2. In the pushing state, the pusher robot does not transmit command to other robots unlike the previous technique. The pusher robot executes simply pushing action according to the relative distance from guider robots. The algorithm for the pushing state is described in Table B.3.

TABLE B.3 ALGORITHM FOR THE PUSHING STATE IN A STATIC ENVIRONMENT

Algorithm	Pushing (*for pusher robot*)
Input	$s^{P}_{\lambda(\cdot)}$: The sensor information of pusher robot
Output	action of the pusher robot
1:	**loop**
2:	**if** $s^{P}_{\lambda(\pi/2)} \geq s_{\max}$
3:	*move forward and counterclockwise*
4:	**else if** $s^{P}_{\lambda(-\pi/2)} \geq s_{\max}$
5:	*move forward and clockwise*
6:	**else**
7:	*move straightforward*

B.4.3 The States of a Leader Robot

For leader robot, the moving-to-the-goal state is identical to the previously proposed technique except that the leader robot does not order anyone, as shown in Table B.4. The global path planner $\bar{\mathbf{P}}(k)$ has a feasible path planning trajectory which considers multiple obstacles. In the stop state, the leader robot orders guider robots to change their states according to the sensor information of leader and the state of guider robot. Table B.5 shows the algorithm for the stop state in a static environment. If all guider robots have stop state, the leader robot orders the tail robot to change its state to the following-in-contact (line 3-7). If the distance between the guider robot and the desired origin of VEDF is less than sensing range, the leader robot orders the $(-1)^{\text{th}}$ moving guider robot to change to the lining-up state (line 8-11).

TABLE B.4 ALGORITHM FOR THE MOVING-TO-THE-GOAL STATE IN A STATIC ENVIRONMENT

Algorithm	Moving to the goal (*for leader robot*)
Output	action of the leader robot
1:	**loop**
2:	*move to the goal according to the global path planner* $\bar{\mathbf{P}}(k)$

TABLE B.5 ALGORITHM FOR THE STOP STATE IN A STATIC ENVIRONMENT

Algorithm	Stop (*for leader robot*)
Input	$A^G_{(i,j)}$: the state of j^{th} guider robot in i^{th} row $\mathbf{p}^G_{-1}$: the position of guider robot which belongs to following-in-contact state $s^L_{\lambda(\cdot)}$: the sensor information of leader robot
Output	action of the leader robot
1:	**loop**
2:	$v^L \leftarrow 0$
3:	**if** $\forall A^G_{(i,j)} = stop$
4:	**if** $s^L_{\lambda(\pi/3)} \geq s_{\max}$
5:	$A^G_{(1,tail)} \leftarrow following-in-contact$
6:	**if** $s^L_{\lambda(-\pi/3)} \geq s_{\max}$
7:	$A^G_{(2,tail)} \leftarrow following-in-contact$
8:	**if** $\| \mathbf{p}^G_{-1} - \mathbf{p}^L_1 \| < 2r + s_{\max}$
9:	$A^G_{(1,-1)} \leftarrow lining-up$
10:	**if** $\| \mathbf{p}^G_{-1} - \mathbf{p}^L_2 \| < 2r + s_{\max}$
11:	$A^G_{(2,-1)} \leftarrow lining-up$

B.5 Simulation Results

Figure B.3 and B.4 show the simulation of new object transportation techniques in a static environment. The simulation environments are identical to the previous simulation, as already described in Section 5.1 except obstacles. An obstacle is a rectangle of which size is 100×65 cm.

B.5.1 Simulation Result: An Obstacle

At first, we simulated the proposed technique in an environment where a rectangular-shaped obstacle exists, as shown in Fig. B.3. The leader robot could not detect anything on its left side because an obstacle was located on its right side, as shown in Fig. B.3 at 0 second. Therefore, the leader robot ordered the tail robot (G1) located in the second row to change its state from the stop to following-in-contact. Then, the G1 robot began to follow the boundary of the second row, as shown in Fig. B.3 at 1 second. If the G1 robot approached to the origin of VEDF, its state is changed by leader robot's order from the following-in-contact to lining-up state. Likewise, the G2, G3, and G4 robot showed the same actions. At 26 second, two tail robots (G5 and G8) in the first and second row began to move together because the leader robot detected that there are no robots both side. This means that there is no obstacle on the both side of leader robot. The guider robots avoided the obstacle using the following-in-contact state while they move. The guider robots showed the same action by the lining-up state after the robots pass the

obstacles. The rest of simulation result is analogous to the previous technique which is already shown in Section 5. Total travelled time was 138 seconds.

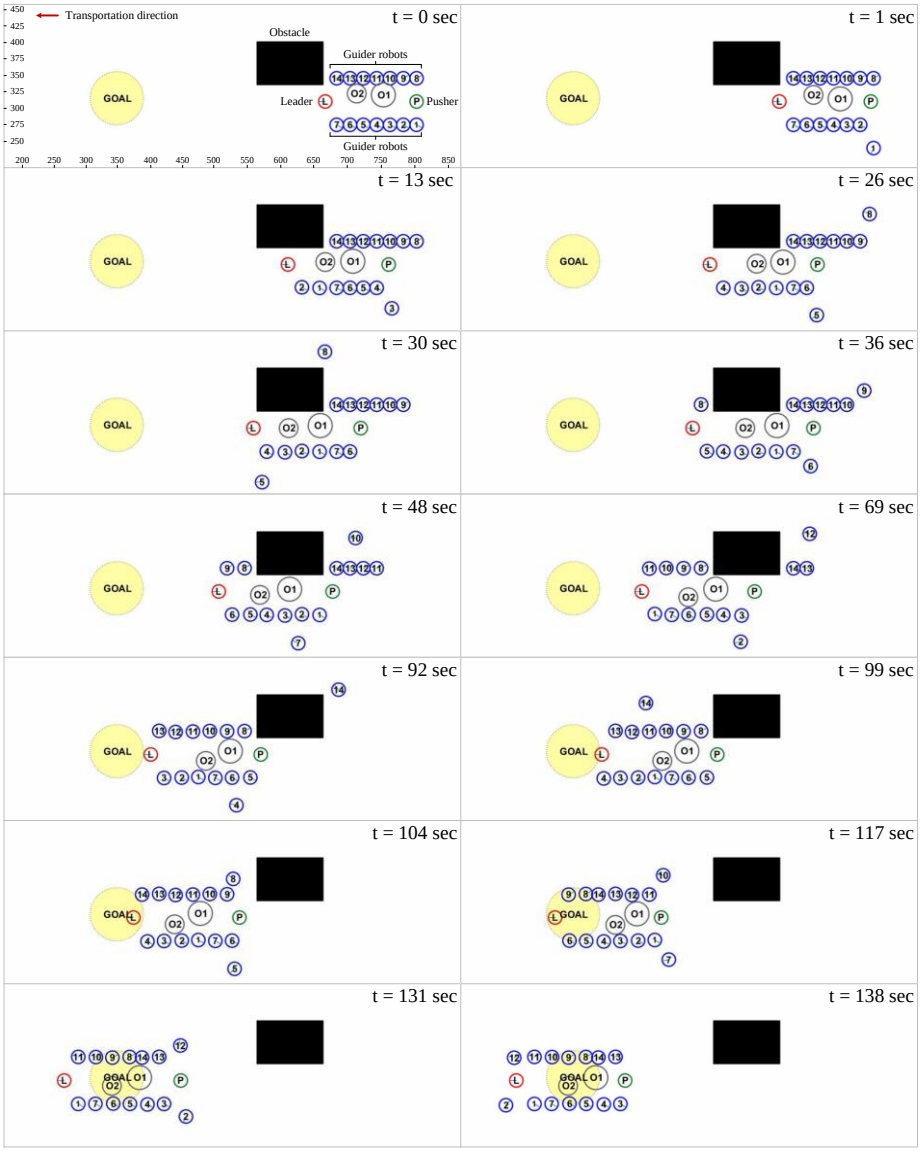

Figure B.3 Simulation result in a static environment where a rectangular obstacle exists.

B.5.2 Simulation Result: Two Obstacles

We also simulated the proposed technique in the environment where two obstacles exist, as shown in Fig. B.4. In this case, two rectangular obstacles broke away the ways of guider robots. The objects were transported by passing through the gap between the obstacles, and thus, the guider robots should go a long way round to avoid obstacles. If the pusher robot is located between obstacles, it pushes objects along the obstacles, as shown in Fig. B.4 at 73 seconds. Two obstacles replace the role of guider robots in this case. The obstacles were successfully transported in 113 seconds. This travelled time was shorter than one-obstacle case because two obstacles shortened the travelled distance that the guider robots should move.

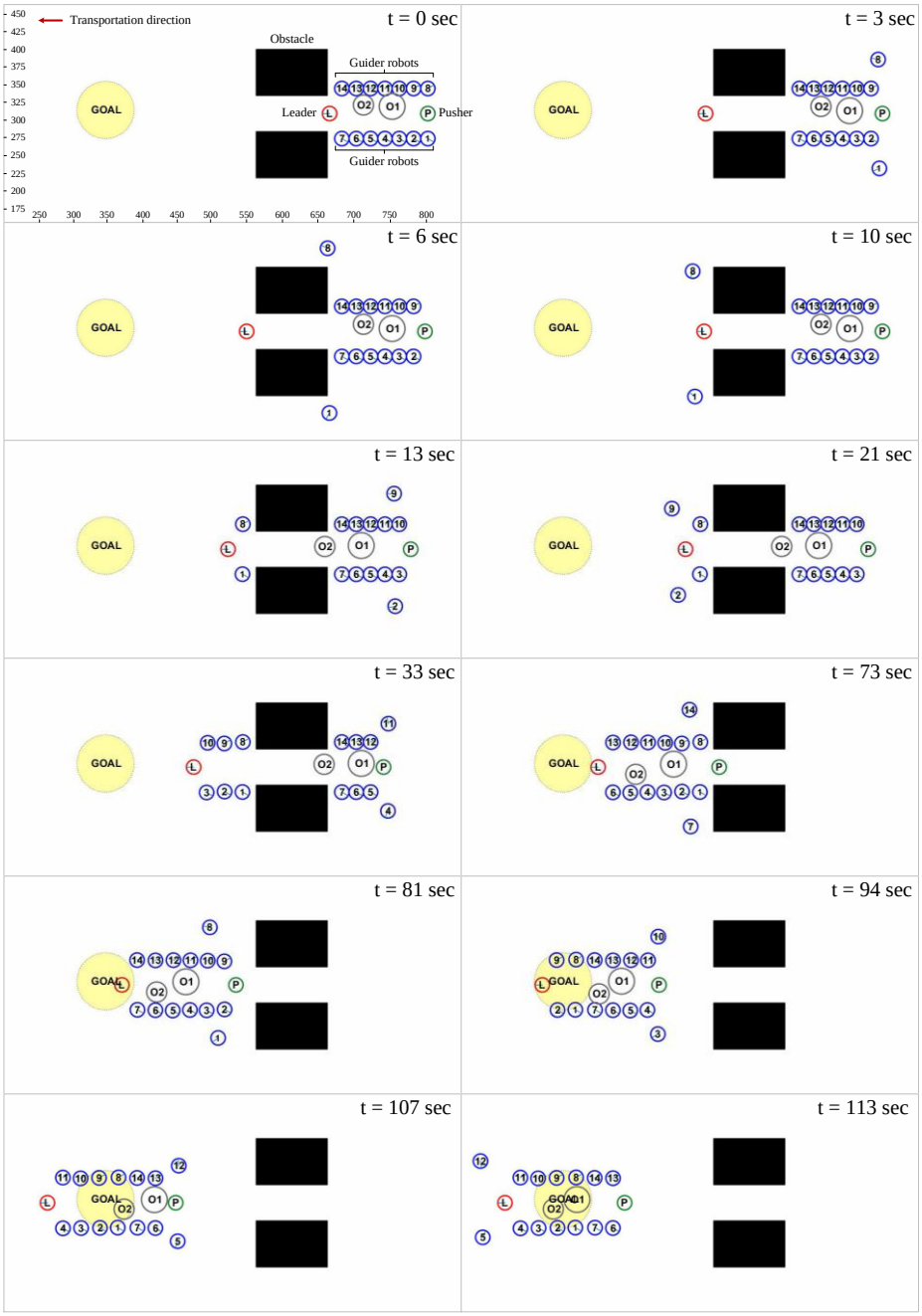

Figure B.4 Simulation results in a static environment where two obstacles exist.

B.6 Practical Experiment

We also conducted on a practical experiment in a static environment. The experimental environment is identical to the previous practical experiment as already described in Section 6.1 except an obstacle. The shape of the obstacle is rectangle, as shown in Fig. B.5 at 0 second. At first, a leader robot began to move-to-the-goal by event $\bar{E}_1^L$ because there are two guider robots on both sides. If the leader robot arrived at the position where there are no guider robots on both sides, it stopped and ordered tail robots to move according to the event $\bar{E}_1^G$, as shown in Fig. B.5 at 8 seconds. The tail robots approached to the head robots by the following-in-contact and lining-up states. For approaching the head robots, the guider robots followed the row of guider robots for 30 seconds. If the guider robots were closed to near the leader robot, they generated and followed the VEDF with respect to the relative position of leader robot. At the same time, a pusher robot pushed the objects, as shown in Fig. B.5 at 30 seconds. The leader robot ordered the guider robot located in the second row because there is an obstacle in the first row only, after 48 seconds. Therefore, the guider robots in the second row (G4, G5, and G6) moved alone while the leader robot passed the obstacle. Two guider robots on both row moved together after the guider robots passed the region located in the obstacle, as shown in Fig. B.5 at 163 seconds. The rest processes were analogous to the previous steps. Total travelled time was 363 seconds, and the trajectories of robots were illustrated in Fig. B.6.

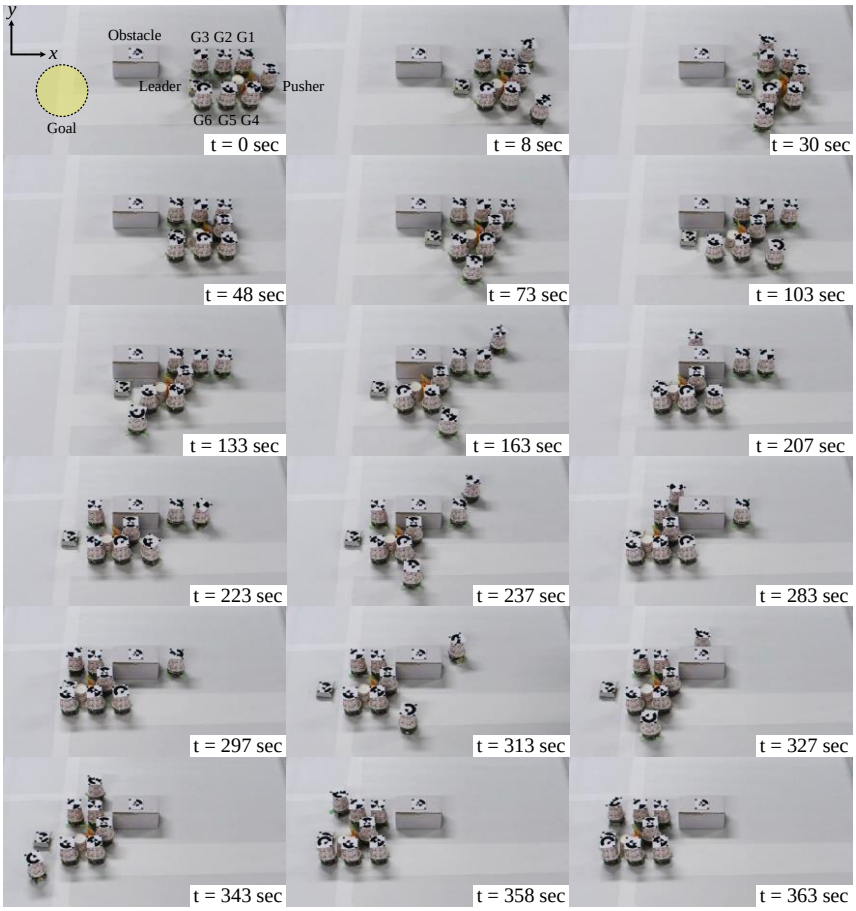

Figure B.5 Object transportation in a static environment. A rectangular-shaped box prevents the objects from escaping.

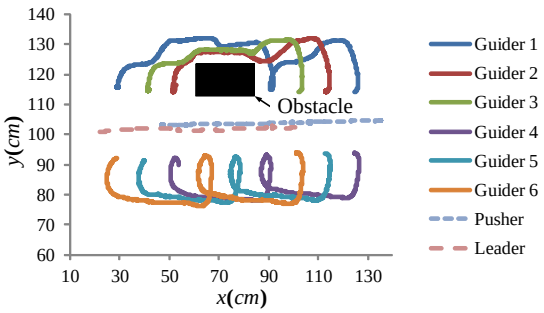

Figure B.6 The trajectories of the robots during the transportation

Bibliography

[1] T. Arai, E. Pagello, and L. E. Parker, “Guest editorial: Advances in multirobot systems,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655-661, 2002.

[2] Y. U. Cao, A. S. Fukunaga, A. B. Kahng, and F. Meng, “Cooperative mobile robotics: Antecedents and directions,” *Autonomous Robots*, vol. 4, no. 1, pp. 7-27, 1997.

[3] A. Bicchi and V. Kumar, “Robotic Grasping and Contact: A Review,” in *Proc. of IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 24-28, 2000, pp. 348-353.

[4] O. B. Kroemer, R. Detry, J. Piater, and J. Peters, “Combining active learning and reactive control for robot grasping,” *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1105-1116, 2010.

[5] K. M. Lynch, “Nonprehensile Robotic Manipulation: Controllability and Planning,” *Ph. D. Dissertation*, The Robotics Institute, Carnegie Mellon University, March 1996.

[6] D. Nieuwenhuisen, A. F. Stappen, and M. H. Overmars, “Path Planning for Pushing a Disk using Compliance,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2-6, 2005, pp. 714-720.

[7] K. Harada, J. Nishiyama, Y. Murakami, and M. Kaneko, “Pushing Multiple Objects using Equivalent Friction Center,” in *Proc. of IEEE International Conference on Robotics and Automation*, Washington, USA, May 11-15, 2002, pp. 2485-2491.

[8] T. Mericli, M. Veloso, and H. Levent Akin, “Push-manipulation of complex passive mobile objects using experimentally acquired motion models,” *Autonomous Robots*, vol. 38, no. 3, pp.317-329, 2015.

[9] M. de Berg and D. H. P. Gerrits, “Computing Push Plans for Disk-Shaped Robots,” in *Proc. of IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 3-7, 2010, pp. 4487-4492.

[10] G. Eoh, J. D. Jeon, J. S. Choi, and B. H. Lee, “Multi-robot Cooperative Formation for Overweight Object Transportation,” in *Proc. of IEEE/SICE International Symposium on System Integration*, Kyoto, Japan, December 20-22, 2011, pp. 726-731.

[11] R. Groß, F. Mondada, and M. Dorigo, “Transport of an object by six pre-attached robots interacting via physical links,” in *Proc. of the IEEE*

International Conference on Robotics and Automation, Orlando, USA, May 15-19, 2006, pp. 1317-1323.

[12] Z. Liu, H. Kamogawa, and J. Ota, "Fast grasping of unknown objects through automatic determination of the required number of mobile robots," *Advanced Robotics*, vol. 27, no. 6, pp. 445-458, 2013.

[13] R. Groß, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Proc. of IEEE International Conference on Robotics and Automation*, Orlando, USA, May 15-19, 2006, pp. 2558-2564.

[14] M. J. Mataric, M. Nilsson, and K. T. Simsarian, "Cooperative multi-robot box-pushing," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, PA, August 5-9, 1995, pp. 556-561.

[15] Q. Li and S. Payandeh, "Manipulation of convex objects via two-agent point-contact push," *The International Journal of Robotics Research*, vol. 26, no. 4, pp. 377-403, 2007.

[16] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-Robot Manipulation via Caging in Environments with Obstacles," in *Proc. of IEEE International Conference on Robotics and Automation*, Pasadena, USA, May 19-23, 2008, pp. 1471-1476.

[17] G. A. S. Pereira, M. F. M. Campos, and V. Kumar, "Decentralized

Algorithms for Multi-Robot Manipulation via Caging," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 783-795, 2004.

[18] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Grasping by Caging: A Promising Tool to Deal with Uncertainty," in *Proc. of IEEE International Conference on Robotics and Automation*, MN, USA, May 14-18, 2012, pp. 5142-5149.

[19] Z. Wang, Y. Hirata, and K. Kosuge, "Dynamic Object Closure by Multiple Mobile Robots and Random Caging Formation Testing," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 9-15, 2006, pp. 3675-3681.

[20] B. Donald, L. Gariepy, and D. Rus, "Distributed Manipulation of Multiple Objects using Ropes," in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 24-28, 2000, pp. 450-457.

[21] P. Cheng, J. Fink, V. Kumar, and J. Pang, "Cooperative Towing with Multiple Robots," *Journal of Mechanisms and Robotics*, vol. 1, no. 1, pp. 1-8, August, 2008.

[22] G. Eoh, K. W. Lee, J. H. Oh, S. H. Lee, and B. H. Lee, "Cooperative Multiple-Object Towing with Linked Robots," *Journal of Communication and Computer*, vol. 10, no. 3, pp. 385-393, 2013.

[23] S. Bhattacharya, S. Kim, H. Heidarsson, G. S. Sukhatme, and V. Kumar, "A topological approach to using cables to separate and manipulate sets of objects," *The International Journal of Robotics Research*, vol. 34, no.6, pp. 799-815, 2015.

[24] Q. Jiang and V. Kumar, "The Inverse Kinematics of Cooperative Transport with Multiple Aerial Robots," *IEEE Transaction on Robotics*, vol. 29, no. 1, pp.136-145, 2013.

[25] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion Planning of Multiple Mobile Robots for Cooperative Manipulation and Transportation," *IEEE Transaction on Robotics and Automation*, vol. 19, no. 2, pp. 223-237, 2003.

[26] A. Yamashita, J. Sasaki, J. Ota, and T. Arai, "Cooperative manipulation of objects by multiple mobile robots with tools," in *Proc. of the 4th Japan-France/2nd Asia-Europe Congress on Mechatronics*, 1998, pp. 310-315.

[27] M. N. Ahmadabadi and E. Nakano, "A "Constrain and Move" Approach to Distributed Object Manipulation," *IEEE Transaction on Robotics and Automation*, vol. 17, no. 2, pp. 157-172, 2001.

[28] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319-334, 1999.

[29] Y. H. Liu, “Qualitative test and force optimization of 3d frictional form-closure grasps using linear programming,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 163-173, 1999.

[30] M. Dogar, R. A. Knepper, A. Spielberg, C. Choi, H. I. Christensen, and D. Rus, “Multi-scale assembly with robot teams,” *The International Journal of Robotics Research*, vol. 34, no. 13, pp.1645-1659, 2015.

[31] S. Erhart and S. Hirche, “Internal Force Analysis and Load Distribution for Cooperative Multi-Robot Manipulation,” *IEEE Transaction on Robotics*, vol. 31, no. 5, pp.1238-1243, 2015.

[32] T. Meriçli, M. Veloso, H. L. Akın, “Push-manipulation of complex passive mobile objects using experimentally acquired motion models,” *Autonomous robots*, vol. 38, no.3, pp.317-329, 2015.

[33] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, “Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots,” *IEEE Transaction on Robotics*, vol. 31, no.2, pp. 307-321, 2015.

[34] B. P. Gerkey and M. J. Matari´c, “Sold!: Auction methods for multi-robot coordination,” *IEEE Transactions on Robotics and Autonomous Systems*, vol. 18, no. 5, pp. 758-768, 2002.

[35] G. Eoh, J. S. Choi, and B. H. Lee, “Faulty Robot Rescue by Multi-Robot Cooperation,” *ROBOTICA*, vol. 31, no. 8, pp. 1239-1249, 2013.

[36] W. Wan, R. Fukui, M. Shimosaka, T. Sato, and Y. Kuniyoshi, "Cooperative Manipulation with Least Number of Robots via Robust Caging," in *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, July 11-14, 2012, pp. 896-903.

[37] A. Sudsang, F. Rothganger, and J. Ponce, "Motion Planning for Disc-Shaped Robots Pushing a Polygonal Object in the Plane," *IEEE Transaction on Robotics and Automation*, vol. 18, no. 4, pp. 550-562, 2002.

[38] V. Shinde, A. Dutta, and A. Saxena, "Experiments on multi-agent capture of a stochastically moving object using modified projective path planning," *ROBOTICA*, vol. 31, no. 2, pp. 267-284, 2013.

[39] C. C. Loha and A. Traechtlera, "Cooperative transportation of a load using nonholonomic mobile robots," *International Symposium on Robotics and Intelligent Sensors*, vol. 41, 2012, pp. 860-866.

[40] P. Aarabi, "Robust multi-source sound localization using temporal power fusion," in *Proc. of Sensor Fusion: Architectures, Algorithms, and Applications V (AeroSense'01)*, Orlando, Florida, March 22, 2001, pp. 255-264.

[41] D. Davis and E. Patronis, *Sound System Engineering*, CRC Press, 2014.

[42] J. Backman and M. Karjalainen, “Modelling of human directional and spatial hearing using neural networks,” in *Proc. of IEEE International Conference of Acoustics, Speech, and Signal Processing*, MN, USA, April 27-30, 1993, pp.125-128.

[43] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, The MIT Press, 2001.

[44] S. T. Birchfield and R. Gangishetty, “Acoustic localization by interaural level difference,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 18-23, 2005, pp. 1109-1112.

[45] W. Cui, Z. Cao, and J. Wei, “Dual-microphone source location method in 2-D space,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, May 14-19, 2006, pp. 845-848.

[46] M. Raspaud, H. Viste, and G. Evangelista, “Binaural source localization by joint estimation of ILD and ITD,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp.68-77, 2010.

[47] L. E. Parker, *Heterogeneous Multi-Robot Cooperation*, PhD Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute Technology, February 1994.

[48] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, “Cooperative Localization and Control for Multi-Robot

Manipulation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, HI, October 29-November 03, 2001, pp. 631-636.

[49] G. Eoh, Y. Cho, and B. H. Lee, "Reconfigurable Formation Control for Multi-object Transportation," in *Proc. of the International Conference on Distributed Autonomous Robotic Systems*, Daejeon, South Korea, November 2-5, 2014, pp. 304-305.

[50] A. L. Christensen, R. O'Grady, and M. Dorigo, "From Fireflies to Fault-Tolerant Swarms of Robots," *IEEE Transaction on Evolutionary Computation*, vol. 13, no. 4, pp. 754-766, 2009.

[51] J. Ren, K. A. McIsaac, and R. V. Patel, "A potential model using generalized sigmoid functions," *IEEE Transaction on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 37, no. 2, pp. 477-484, 2007.

[52] G. Eoh, "Cooperative Fault Robot Rescue using Sound Signal Detection," *Master thesis*, Seoul National University, 2011.

[53] H. Li, S. Ishikawa, Q. Zhao, M. Ebana, H. Yamamoto, and J. Huang, "Robot navigation and sound based position identification," in *Proc. of IEEE International Conference on Systems, Mans and Cybernetics*, Montreal, Canada, October 7-10, 2007, pp. 2449-2454.

[54] Z. Li, G. Liu, Y. Yang, and J. You, "Scale- and Rotation-Invariant Local

Binary Pattern Using Scale-Adaptive Texton and Subuniform-Based Circular Shift," *IEEE Transaction on Image Processing*, vol. 21, no. 4, pp. 2130-2140, 2012.

[55] D. K. Cheng, *Field and Wave Electromagnetics*, Addison-Wesley, 1989, second edition.

[56] T. Igarashi, Y. Kamiyama, and M. Inami, "A dipole Field for Object Delivery by Pushing on a Flat Surface," in *Proc. of the IEEE International Conference on Robotics and Automation*, Anchorage, USA, May 3-7, 2010, pp. 5149-5119.

[57] K. C. Koh and H. S. Cho, "A Smooth Path Tracking Algorithm for Wheeled Mobile Robots with Dynamic Constraints," *Journal of Intelligent and Robotic Systems*, vol. 24, no. 4, pp. 367-385, 1999.

[58] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795-799, 2014.

[59] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A Vision-Based Formation Control Framework," *IEEE Transaction on Robotics and Automation*, vol. 18, no. 5, pp. 813-825, 2002.

[60] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of

multiple mobile robots," in *Proc. of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 16-20, 1998, pp. 2864-2869.

[61] G. Eoh, S. Lee, T. Lee, and B. H. Lee, "Distributed Object Transportation using Virtual Object," *Journal of Industrial and Intelligent Information*, vol. 2, no. 1, pp. 20-25, 2014.

[62] Available online 2015/11/02: http://www.roadnarrows-store.com/products/robots/micro-robots/e-puck-elisa.html.

[63] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760-772, 1998.

[64] D. Fox and W. Burgard, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp.22-33, 1997.

[65] B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Transactions on Systems, Main, and Cybernetics*, vol. 17, no. 1, pp. 21-31, 1987.

[66] M. Dimakos and A. S. Fokas, "The Poisson and the Biharmonic Equations in the Interior of a Convex Polygon," *Studies in Applied Mathematics*, vol. 134, no. 4, pp.456-498, 2015.

[67] H. Huan and S. Chung, “Dynamic Visibility Graph for Path Planning,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 28-October 2, 2004, pp. 2813-2818.

[68] S. M. Persson and I. Sharf, “Sampling-based A* algorithm for robot path-planning,” *The International Journal of Robotics Research*, vol. 33, no. 13, pp.1683-1708, 2014.

[69] T. Guan-zheng, H. Huan, and A. Sloman, “Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm,” *Journal of Central South University of Technology*, vol. 13, no. 1, pp.80-86, 2006.

초 록

논문은 물체의 특징(능동 및 수동)에 따른 두 가지 물체 수송 방법을 제안한다. 수동 물체는 보통의 물체로써, 다른 로봇과 통신이 불가능하고 스스로 센싱(sensing)할 수 있는 기능이 없다. 하지만 능동 물체는 스스로 로봇과 통신이 가능하고, 물체에 장착되어 있는 근접 센서를 이용하여 다른 로봇을 감지할 수 있다. 로봇 협업을 이용한 대표적인 물체 수송 방법은 움켜쥐기(grasping), 밀기(pushing), 감싸기(caging) 방법이 있는데, 이 방법들은 각각 세밀한 움켜쥐기, 물체의 위치에 따른 반복적인 로봇의 움직임 보정, 실시간으로 물체의 위치를 획득하는 과정이 요구된다. 이러한 문제들을 해결하기 위하여 본 논문에서는 물체의 특징을 고려한 다음의 두 가지 물체 수송 방법을 제안하였다.

첫째, 본 논문은 능동 물체와 로봇과의 상호 통신 및 소리 신호를 이용한 협업 물체 수송 방법을 제안한다. 이를 위해 먼저 3개의 마이크를 이용하여 음원의 위치를 추정하는 방법을 개발하였다. 또한, 단일 로봇만으로는 물체를 수송할 수 없기 때문에 '밀기-당기기-감독' 로봇으로 구성된 로봇 팀을 조직하였다. 이렇게 제안된

로봇 팀은 능동 물체와 로봇 간의 상호 통신 및 협업을 통해 물체를 목적지까지 성공적으로 수송할 수 있었다.

둘째, 본 논문은 순환 시프트 움직임을 고려한 새로운 수동 물체 수송 방법을 제안한다. 제안한 방법은 물체의 위치 정보나 모양을 알 필요가 없으며, 물체 수송을 위한 추가적인 도구도 필요하지 않다. 물체 수송을 위해 먼저 여러 대의 로봇들이 가상 전기쌍극자 필드를 이용하여 두 줄로 늘어서고, 그 사이의 공간으로 밀기 로봇이 물체들을 밀어 넣는다. 이러한 두 줄 대형은 순환 시프트 움직임을 통해 목적지까지 이어지게 된다. 이러한 방식으로 다수의 물체들을 목적지까지 수송할 수 있었다. 이 방법은 비중앙식 방법이고, 유한 상태 기계를 기반으로 동작한다. 제안된 방법은 시뮬레이션과 실제 실험을 통해 검증되었다.

주요어: 물체 수송, 소리 신호, 순환 시프트, 유한 상태 기계, 가상 전기쌍극자 필드

학 번: 2011-30243