Robotica (2017) volume 35, pp. 1341–1364. © Cambridge University Press 2016
doi:10.1017/S0263574716000096

Cooperative object transportation using parallel line formation with a circular shift

Gyuho Eoh*, Jae D. Jeon, Jung H. Oh and Beom H. Lee

Seoul National University (Department of Electrical and Computer Engineering)

(Accepted February 4, 2016. First published online: March 4, 2016)

SUMMARY

This paper presents a new cooperative object transportation technique using parallel line formation with a circular shift. Typical areas of research in relation to object transportation are 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. In this paper, the proposed technique does not need to consider the shape or the pose of objects, and equipped tools are not necessary for object transportation because objects are transported by pushing behavior only. Multiple robots create parallel line formation using a virtual electric dipole field and then push multiple objects into the formation. This parallel line is extended to the goal using cyclic motion by the robots and the objects are transported to the goal by pushing behavior. 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 technique.

KEYWORDS: Cooperative object transportation, Circular shift, Finite state machine, Electric dipole field.

1. Introduction

Object transportation is a typical problem in the field of cooperative robotics.[1,2] Many researchers have studied object transportation techniques using multiple robots[3–5] because a single robot cannot easily manipulate large or heavy objects. Additionally, a single robot should have greater capability than multiple robots when manipulating identically sized objects because a single robot requires many functions, such as sensing, transportation, and precise control.

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,[6,7] this technique has since gradually been applied to object transportation using mobile robots. Multiple robots grasp an object using robot manipulators and transport the object to the goal using the grasping technique.[8,9] This technique has an advantage in that 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. 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 multiple robots manipulate an object by pushing it without object-wrapping step.[10,11] In contrast to the grasping technique, a preliminary grasping action is unnecessary when using the pushing technique, enabling robots to transport a large object. In addition, a change in the position of the robots when modifying the transportation path is 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 easily be pushed out of control by the subtle pushing of the robots.11 In addition, it is necessary to consider the friction of the ground and the weight of the object for precise control, which

* Corresponding author. E-mail: geni0620@snu.ac.kr
E-mails: innocent88@snu.ac.kr, bulley85@snu.ac.kr, bhlee@snu.ac.kr

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

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 involved and the erratic motion of the object.[12,13]

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.[14,15] The object can be transported successfully without an escape using the caging technique. However, the condition of caging should be checked iteratively during the transportation process to guarantee that the object does not escape. The shape and the position of 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.

Finally, there are other transportation techniques using various tools in addition to those listed above. Some researchers have used a rope for cooperative object transportation.[16–18] In this technique, robots wrap and pull multiple objects using a rope to manipulate multiple objects to the goal. The robots, however, have restricted motion because they are tightly coupled with the rope. Other researchers investigated the object transportation problem using a stick.[19] 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. Robots can also manipulate an object in a narrow space using this technique. Multi-object transportation, however, is not considered to be feasible with this technique, like caging.

We present a new cooperative object transportation technique which uses a circular shift. Guider robots create parallel line formation by lining up in two rows around multiple objects, as shown in Fig. 1. The guider robots prevent the objects from escaping using their bodies and extend the parallel line using *circular shift motions*. The circular shift motion is cyclic 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 uses the advantages of the pushing and caging techniques for multi-object transportation. There is no need to grasp an object; the object is manipulated by pushing behavior only. Also, the proposed formation always maintains an enclosed shape to prevent the object from escaping. Unlike caging techniques, the geometric information of the object is unnecessary and multi-object transportation is possible with the proposed technique. This paper is an extended version of a previous paper which presents a decentralized object transportation framework using simple finite state machines.[20] In this paper, we describe detailed object transportation processes, consider kinematic and dynamic constraints for feasible robot motion, and conduct practical experiments in diverse environments.

2. Problem Description

2.1. Research terminology

The proposed multi-robot team consists of three roles: the *guider*, *leader*, and *pusher robot*. 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 robot* and a *tail robot*, respectively, as shown in Fig. 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. The names and the role descriptions of the proposed multi-robot team are summarized in Table I.

2.2. Assumptions

There are several assumptions when formulating the multi-object transportation problem. We propose a heterogeneous robot system and assume that the roles of each robot are predefined: guider, pusher,

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table I. The names and the 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 last 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

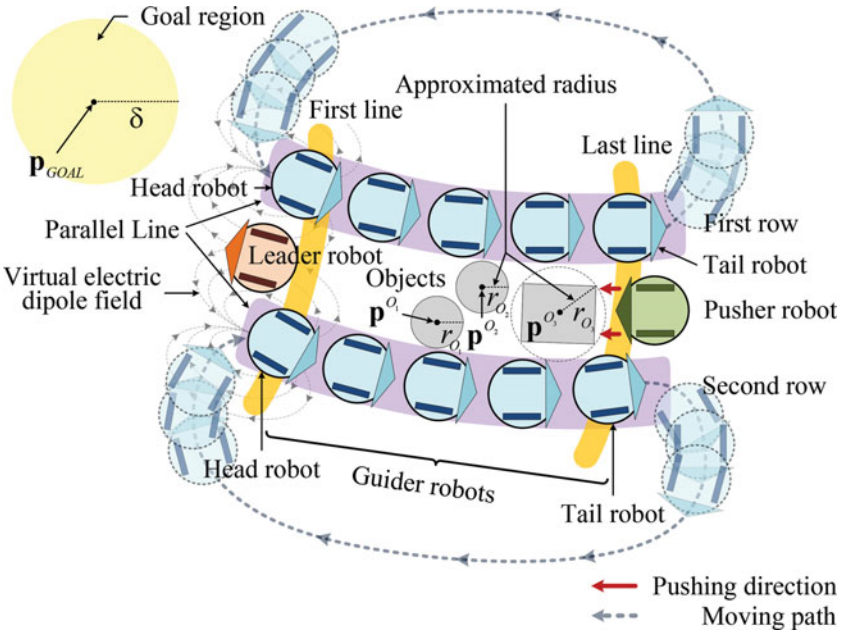


Fig. 1. The proposed multi-robot team for object transportation. A leader robot leads the robot team by following a global path, and a pusher robot pushes the object from behind. Multiple guider robots create parallel line formation by lining up with two rows to prevent the objects from escaping. The guider robots make an extended parallel line to the goal by iteratively moving from the last line to the 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.

and leader robot. Several different functions, such as sufficient pushing power to push multiple objects or global path planner, are necessary for successful multi-object transportation. The proposed heterogeneous system uses different robots for satisfying each function; each robot needs not have all functions. The guider robots are not moved by pushing behavior and recognize the row in which they line up. The pusher robot has sufficient power to push multiple objects. The leader robot has the goal path planner in advance and therefore the leader robot leads the robot team. Also, all objects are assumed to be gathered together before transportation such that $||\mathrm{p}^{o_{i+1}} - \mathrm{p}^{o_i}|| < r_{i+1} + r_i + \varepsilon, \forall i$, where $\mathrm{p}^{o_i} \in \mathrm{R}^2$ and r_i are the position and the approximated radius of ith object O_i, respectively. The coefficient ε is a marginal constant between objects. The index i of the object is given according to the relative distance. 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. The shapes of

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

objects are arbitrary; even rolling objects can be used for transportation. In addition, a symmetrical robot formation is assumed to be prearranged before object transportation. We do not consider an approaching method toward objects so as to concentrate on the transportation process. An approaching technique to the objects can be designed using other methods such as the potential field method.[21] The number of robots which are used in object transportation is shared with together in advance. Finally, data association is assumed to be possible within a limited sensing range $s_{\max}$ by intercommunication.

2.3. Multi-object transportation problem

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

$$||\mathbf{p}_{GOAL} - \mathbf{p}^{o_i}|| \leq 2\delta \textbf{ for } \forall i, \tag{1}$$

where $\mathbf{p}_{GOAL} \in \mathbf{R}^2$ is the goal point and δ is the radius of the goal region.

The solution to the multi-object transportation problem will be presented as algorithms and robot controllers. First, we present a basic formation generation (i.e., line formation) process to construct the proposed formation. Second, we present robot motion controllers which satisfy kinematic and dynamic constraints during the transportation process. Finally, the finite state machine of each robot will be presented.

3. Line Formation Generation using Multiple Robots

A key technique of the proposed method is that it generates parallel line formation using multiple robots. This formation consists of two symmetrical line formations, and the lines are extended to the goal using circular shift motion. The tail robot in the line formation moves from the last line to the first line; this is defined as a circular shift motion because it is analogous to the circular shift operator in combinatorial mathematics.[22] There are two problems associated with the circular shift motion. These 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, as shown in Fig. 1. 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 path deviation due to the dynamic constraint. To address these issues, we introduce the circular shift motion in Section 3.1. Then, we present the smooth path generation method for circular shift motion using a virtual electric dipole field in Section 3.2. Finally, the method which allows the following of the smooth path without path deviation is presented in Section 3.3.

3.1. Circular shift motion

A circular shift is a cyclic shift operation which takes place by element rearrangement in a tuple. The last or first element is transferred to the first or last element, respectively, and all other elements move one block forward or backward concurrently. The number of elements and the relative orders between the elements are reserved. The line formation of the guider robots is rearranged according to the circular shift operator, which is defined as circular shift motion. Figure 2 shows an example of circular 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 line 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 line formation can move backward using this type of circular shift motion.

The advantage of circular shift motion is easy formation maintenance because only a single robot moves for the forward or backward movements of the formation. The line 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 is difficult because each robot should be able to predict other robots' motions and the action of all robots should be synchronized. In the circular shift motion, however, the other robots are stopped while the last robot in line moves, which enables the robot formation to maintain a regular shape.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

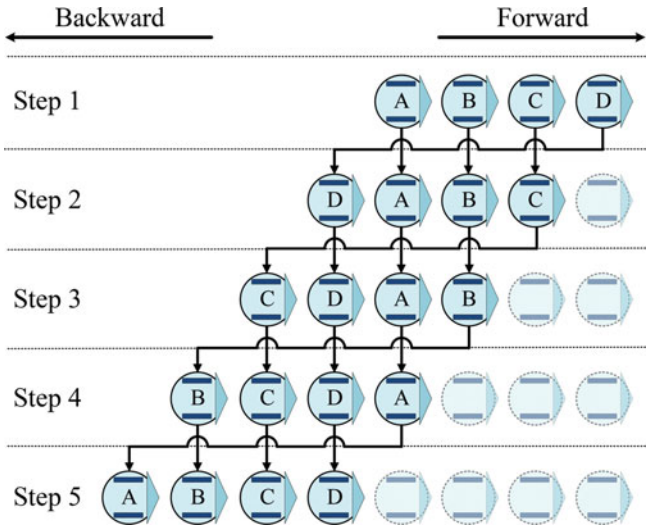


Fig. 2. Circular shift example. The line formation is extended in the backward direction using the circular shift motion. This formation maintains a regular distance between the robots when a single robot moves, which prevents objects from escaping.

3.2. Path generation using a virtual electric dipole field

In this section, we present the smooth path generation method for robots which use circular shift motion. For lining up two extended rows to the goal, the tail robots in each row should approach the head robots from a specific direction, respectively. We, therefore, adopt a smooth path generated by a virtual electric dipole field for approaching the robots. While the smooth path is longer than direct path due to circular shape, the nonholonomic guider robots can approach the other robots from a specific direction by following the smooth path.

In physics, an electric dipole field is generated using positive and negative electric charges or opposite currents which exist in close proximity.[23] 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.[24] The function $\mathbf{F}_e : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ is referred to as a *virtual electric dipole function*, and the derivation is given by

$$\mathbf{F}_e(x, y) = \begin{bmatrix} f_x & f_y \end{bmatrix}^{\mathrm{T}} = -\cos\theta \times \begin{bmatrix} -\cos\theta \\ -\sin\theta \end{bmatrix} + \sin\theta \times \begin{bmatrix} -\sin\theta \\ -\cos\theta \end{bmatrix} = \begin{bmatrix} \cos 2\theta \\ \sin 2\theta \end{bmatrix}, \tag{2}$$

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 virtual electric dipole field, and the successor robot denotes the robot approaching the predecessor robot, as shown in Fig. 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 (2), 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 virtual electric dipole field is modified by introducing an additional factor β, as follows:

$$\tilde{\mathbf{F}}_e(x, y) = [\tilde{f}_x \quad \tilde{f}_y]^{\mathrm{T}} = [\cos^2\theta - \beta\sin^2\theta \sin\theta\cos\theta + \beta \sin\theta \cos\theta]^{\mathrm{T}}. \tag{3}$$

The shape of the virtual electric dipole field moves closer to a circle as β becomes decrease, whereas it becomes more flattened as β increases, as expressed by (3). Finally, the successor robot is

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

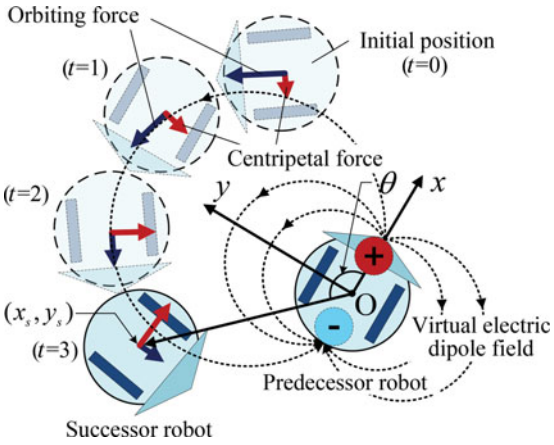


Fig. 3. A virtual electric dipole field is generated by a combination of centripetal and orbiting forces. The virtual electric dipole field is created by assuming that a positive charge exists at the head of the predecessor robot and a negative charge exists at the tail of the predecessor robot. A successor robot is able to approach the predecessor robot from the behind by following the virtual electric dipole field.

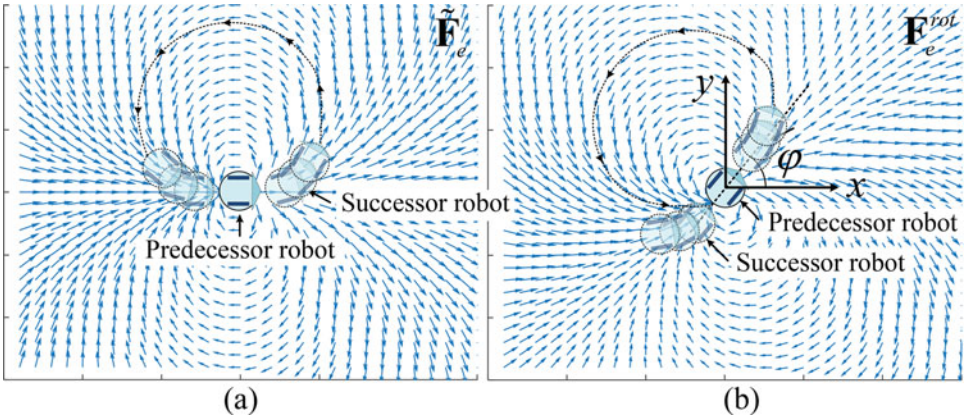


Fig. 4. The dashed line indicates the projected path by the successor robot. (a) The successor robot follows the original virtual electric dipole field $\tilde{\mathbf{F}}_e(x, y)$, which is generated by (3), arriving at the tail of the predecessor robot. (b) If the virtual electric dipole field is rotated by φ by (4), 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.

able to approach the predecessor robot from a specific direction using a combination of centripetal and orbiting forces.

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, multiple guider robots should stand in a line on the leftward side. We, therefore, modify the approaching angle by rotating the virtual electric dipole function $\tilde{\mathbf{F}}_e(x, y)$ which is generated by (3). The rotational electric dipole function $\mathbf{F}_e^{rot} : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ with angle φ is described below:

$$\mathbf{F}_e^{rot}(x, y, \varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \tilde{\mathbf{F}}_e(x, y). \tag{4}$$

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(b).

3.3. Following a smooth path using a bang-bang controller

A smooth path is generated by the virtual electric dipole field, as described in Section 3.2. To line up in a row, guider robots should follow the smooth path without path deviation considering both kinematic and dynamic constraints. We, therefore, apply a *bang-bang controller* to the path tracking

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

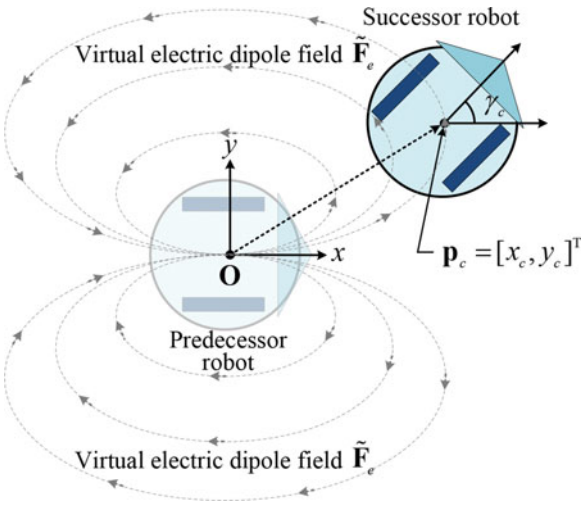


Fig. 5. Two-wheeled differential-drive robot configuration with respect to the predecessor robot.

method to satisfy the kinematic and dynamic constraints.[25] A mobile robot can smoothly change its driving velocity or direction using the bang-bang control scheme.

Here, we assume that two-wheeled differential mobile robots are used in the proposed technique, as shown in Fig. 5. The kinematic equation of the mobile robot is described as

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\gamma}_c \end{bmatrix} = \begin{bmatrix} v_c \cos(\gamma_c) \\ v_c \sin(\gamma_c) \\ \omega_c \end{bmatrix}, \tag{5}$$

where $\mathbf{p}_c = [x_c, y_c]^{\tau}$ is the current 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, as 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,

$$|a_c| \leq a_{\max}, |\alpha_c| \leq \alpha_{\max}, \tag{6}$$

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

If the robots follow the virtual electric dipole field $\mathbf{F}_e^{rot}$, the target position $\mathbf{p}_t = [x_t, y_t]^{\tau}$ is given by

$$\mathbf{p}_t = \mathbf{F}_e^{rot}(x_c, y_c) + \mathbf{p}_c, \tag{7}$$

and target heading angle γ_t is given by

$$\gamma_t = \tan^{-1}\left(\frac{f_{y_c}^{rot}}{f_{x_c}^{rot}}\right), \tag{8}$$

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

$$\begin{bmatrix} e_x \\ e_y \\ e_\gamma \end{bmatrix} = \begin{bmatrix} \cos(\gamma_t) & \sin(\gamma_t) & 0 \\ -\sin(\gamma_t) & \cos(\gamma_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t - x_c \\ y_t - y_c \\ \gamma_t - \gamma_c \end{bmatrix}, \tag{9}$$

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

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

The desired angular velocity $\omega_c^d(k+1)$ is calculated using the bang-bang control, as follows:

$$\omega_s(k) = \omega_t(k) + [2\alpha_{\max}|\gamma_t(k) - \gamma_c(k)|]^{1/2}\text{sgn}(\omega_t(k) - \omega_c(k)), \tag{10}$$

$$\omega_t(k) = \frac{\gamma_t(k) - \gamma_c(k)}{\Delta T}, \tag{11}$$

$$\alpha_c(k) = \mu\left(\frac{\omega_s(k)}{\Delta T}, \alpha_{\max}\right), \tag{12}$$

$$\omega_c^d(k+1) = \omega_c(k) + \alpha_c(k)\Delta T, \tag{13}$$

where $\text{sgn}(\cdot)$ is the sign operator, ΔT is the sampling time interval, and the index k denotes the time. The clamping function $\mu(a, b)$ in (12) is defined as follows:

$$\mu(a, b) = \begin{cases} a & \text{for } |a| < |b| \\ |b|\text{sgn}(a) & \text{for } |a| \geq |b| \end{cases}. \tag{14}$$

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

$$\dot{e}_x(k) = v_t(k) - v_c(k)\cos(e_\gamma(k)) + \omega_t(k)e_y(k), \tag{15}$$

$$v_s(k) = \dot{e}_x(k) + [2a_{\max}|e_x(k)|]^{1/2}\text{sgn}(e_x(k)), \tag{16}$$

$$a_c(k) = \mu\left(\frac{v_s(k)}{\Delta T}, a_{\max}\right), \tag{17}$$

$$v_c^d(k+1) = v_c(k) + a_c(k)\Delta T. \tag{18}$$

In (15), the derivative of tangential path error $\dot{e}_x$ is calculated using the target angular velocity ω_t, which means that the desired tangential velocity $v_c^d(k+1)$ is closely related to the target angular velocity. For example, if a robot should sharply rotate according to (13), the robot decreases its tangential velocity according to (18). Therefore, the robots are able to follow the virtual electric dipole field without path deviations using (13) and (18) as controllers, which enables guider robots to line up in a row.

4. Object Transportation by a Decentralized Multi-robot Team

This section addresses the object transportation process by a decentralized multi-robot team using circular shift motion, as described in Section 3. 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 line-up order of the guider robots, and (c) a trigger command. This information can be acquired through internal sensors and by short-range communication, enabling the proposed multi-robot team to become a decentralized system. In Section 4.1, we present the methods by which the sensor information is acquired, the numbering order 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 will be explained in Section 4.2. The state transition processes for object transportation will be presented in Section 4.3.

4.1. Necessary information acquisition for the finite state machine design

In this section, we present the information acquisition method for triggering events before the FSM is described in detail. First, we assume that the proximity sensors on the robots are uniformly distributed around the boundary, as shown in Fig. 6. The x-axis direction of local coordinate is the heading of the robots, and the distance measured at the ith proximity sensor is denoted as s_i. The index function

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

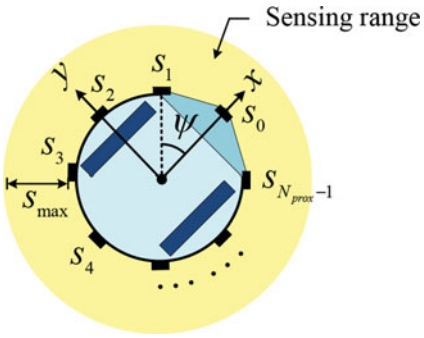


Fig. 6. The arrangement of proximity sensors.

$\lambda(\psi)$ at the distance of $s_{\lambda(\psi)}$ is the closest sensor index of the ψ-direction. It is described as

$$\lambda(\psi) = \underset{k}{\operatorname{argmin}} \left(\left| \psi - \frac{2\pi}{N_{prox}} k \right| \right), \tag{19}$$

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 $s_{\lambda(\psi)}$.

Second, the line-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.[26] We adapt and modify this algorithm appropriately for lining up, as described in Algorithm 1. 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)th order index (lines 3–4) and a tail robot has the 0th order index (lines 5–6). Initially, all guider robots in the line 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.

Algorithm 1 Gradient formation (*for guider robot*)

s_i: the distance acquiring from i^{th} proximity sensor
Input $order$(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:      checkorder(front)
3:    if s_λ(0) = s_max and s_λ(π) = s_max
4:      order(self) ← −1
5:    else if s_λ(0) = s_max and s_λ(π) < s_max
6:      order(self) ← 0
7:    else
8:      order(self) ← order(max)
9:      if order(front) < order(self)
10:       order(self) ← order(front)
11:     order(self) ← order(self) + 1
12:   transmitorder(self)
```

Third, there are two sharing commands between robots in the proposed transportation technique: *ready* and *action*. To begin the transportation phase, the pusher robot transmits an action command to neighboring guider robots. In contrast, the leader robot transmits a ready command to neighboring

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table II. The event descriptions in the FSM.

Robot	Event	Descriptions
Guider robot	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 robot	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 robot	E_1^L	$order(\text{left}) = \frac{N_G}{2} - 2$ and $order(\text{right}) = \frac{N_G}{2} - 2$
	E_2^L	$s_{\lambda(\pi/2)} > s_{\max}$ and $s_{\lambda(-\pi/2)} > s_{\max}$

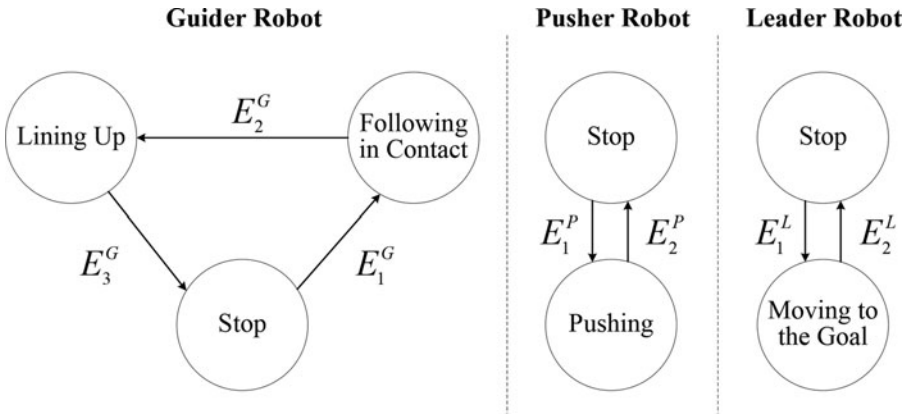


Fig. 7. The finite state machine of the guider, pusher, and leader robots.

guider robots to end a transportation phase. 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.

4.2. Finite state machine for object transportation

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

4.2.1. The states of guider robots. The FSM of the guider robots consists of the *following-in-contact*, the *lining-up* and the *stop* states. The guider robots recognize only the other guider robots which are in the same row. For example, a guider robot in the first row only recognizes the guider robots in the first row.

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. 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

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

acceleration constraints. In this case, the target position p_t is $\mathrm{p}_c + v_t \Delta T$ and the target heading angle γ_t is $\gamma_c + \omega_c \Delta T$. The following-in-contact algorithm is described in Algorithm 2. 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 located in the sensing range.

Algorithm 2 Following-in-contact (for guider robot)

Input s_i: the distance acquiring from the i^{th} proximity sensor
Output action of the guider robot

1: **loop**
2: **if** $\forall s_i > s_{\max}$
3: *move forward and* $counterclockwise^{1st}$, $clockwise^{2nd}$
4: **else if** $s_{\lambda(\pi/2)} < s_{desired}$ or $s_{\lambda(-\pi/2)} < s_{desired}$
5: *move forward and* $clockwise^{1st}$, $counterclockwise^{2nd}$
6: **else if** $s_{desired} < s_{\lambda(\pi/2)} \leq s_{\max}$ or $s_{desired} < s_{\lambda(-\pi/2)} \leq s_{\max}$
7: *move forward and* $counterclockwise^{1st}$, $clockwise^{2nd}$
8: **else**
9: *move straightforward*

In the lining-up state, a guider robot approaches the head robot in the row. A virtual electric dipole field $\mathbf{F}_e^{rot}$ is generated with respect to the local coordinate of the head robot (line 2). The origin of $\mathbf{F}_e^{rot}$ is located at $\mathbf{p}_{head}^G$, and its x-axis is parallel to the $\mathbf{p}_{front_head}^G - \mathbf{p}_{head}^G$ vector, where $\mathbf{p}_{head}^G$ is the position of the head robot and $\mathbf{p}_{front_head}^G$ is the position of ahead of the head robot. The rotational angle φ of the virtual electric dipole field is transmitted by the leader robot 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 (13) and (18), respectively (line 3). The guider robot can then move along the virtual electric dipole field using v^d and ω^d without path deviation (line 4). This process is described in Algorithm 3.

Algorithm 3 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 virtual electric dipole field
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 (13) and (18), 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. 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 0th or not.

4.2.2. The states of pusher robot. The FSM of the pusher robot consists of the *stop* and *pushing* states. 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 0th order index. 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 Algorithm 4. 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 line 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 distance

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

$s_{\lambda(\pi/2)}$ is acquired from the leftmost pusher robot and the sensing distance $s_{\lambda(-\pi/2)}$ is acquired from the rightmost pusher robot by intercommunication.

Algorithm 4 Pushing (for pusher robot)

Input $order$(left): the order index of the left robot
$order$(right): the order index of the right robot
Output action of the pusher robot
transmit 'action' command
Loop
  if $s_{\lambda(\pi/2)} > s_{gap}$
    move forward and counterclockwise
  else if $s_{\lambda(-\pi/2)} > s_{gap}$
    move forward and clockwise
  else
    move straightforward

4.2.3. The states of leader robot. The FSM of the leader robot consists of the *moving-to-the-goal* and *stop* states. In the moving-to-the-goal-state, the leader robot transmits a ready command to neighboring robots and moves to the goal using the global path planner $\mathbf{P}(k)$ until there are no guider robots on the left or right sides. In addition, the leader robot transmits the rotational angle φ of the virtual electric dipole field to the (-1)th guider 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. Similar to the pusher robot, two or more leader robots can be used for object transportation, and they share their sensing information.

Algorithm 5 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
transmit 'ready' command
Loop
  move to the goal according to the global path planner $\mathbf{P}(k)$
  transmit rotational angle φ *to* $(-1)^{\text{th}}$ *guider robot*

4.3. Object transportation process

In this section, we describe the object transportation processes using the FSM. The transportation process is not wholly separated, but we divide the transportation process into four steps to simplify the explanation, as shown in Fig. 8.

Initially, all robots are in a stop state and share a ready command, as shown in Fig. 8(a). In this case, event E_1^P is satisfied and the state of the pusher robot changes from the stop state to the 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 1, as shown in Fig. 8(b). The tail robots in two rows also begin to move along each row, maintaining the desired distance. During the third step, the order indices of the guider robots change and the tail robot is replaced by the next guider robot because the previous tail robot leaves the row, as shown in Fig. 8(c). The order index of a moving guider robot is -1; the order index of the head robot becomes $0.5N_G - 2$, 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. During 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 virtual electric dipole field with respect to the head robot and follows it using the (13) and (18) controllers. The rotational angle of the virtual electric dipole field 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 additional robots on the left or the 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 an

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

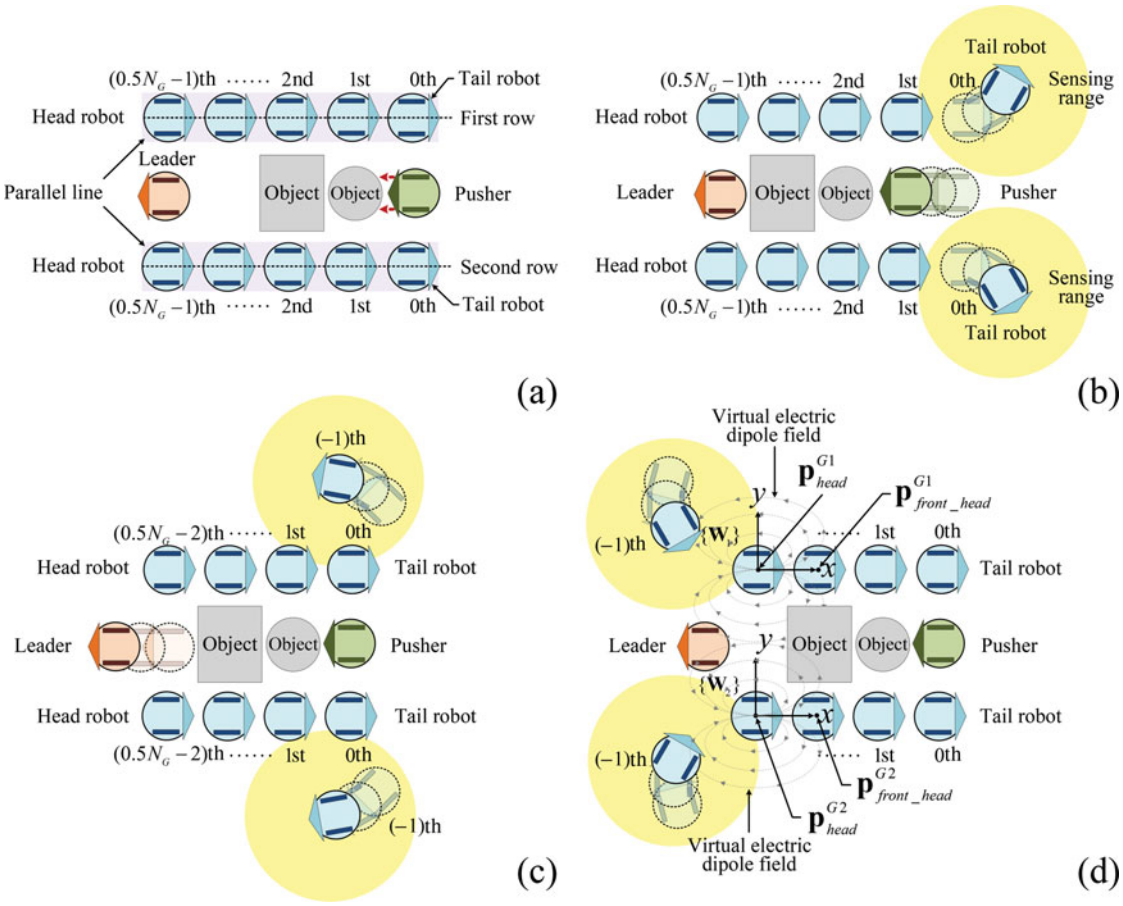


Fig. 8. Object transportation steps. (a) All robots are initially stationary. A pusher robot begins to push objects according to event E_1^P. (b) Two tail robots begin to follow each row according to event E_1^G and a pusher robot stops between the first indexed robots. (c) The leader robot begins to 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 virtual electric dipole field according to event E_2^G. The leader robot stops when there are no guider robots on either side.

actual application. The state and event descriptions according to the processes are summarized in Table III.

5. Simulations

5.1. Simulation environment

We tested the proposed object transportation technique by means of simulations. Our simulation was conducted using the MATLAB simulator. Figure 9 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 and the radius of all of the robots is 10 cm. Also, the guider robots have dynamic constraints. The maximum tangential and angular accelerations are 1 cm/s^2 and 5 °/s^2, respectively. The initial positions of the robots and objects are given in Table IV, and the parameter information is given in Table V.

5.2. Simulation results

We tested the proposed object transportation technique, as shown in Fig. 10. 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 s. 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 s, the pusher robot stopped because event E_2^P was satisfied;

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table III. State and event descriptions according to the transportation process.

Figure	Command	Robot	Event	State
Figure 8(a)	Ready	Guider	–	Stop
		Pusher	E_1^P	Stop → Pushing
		Leader	–	Stop
Figure 8(b)	Action	Guider	E_1^G	Stop → Following in contact
		Pusher	E_2^P	Pushing → Stop
		Leader	–	Stop
Figure 8(c)	Action	Guider	–	Following in contact
		Pusher	–	Stop
		Leader	E_1^L	Stop → Moving to the goal
Figure 8(d)	Ready	Guider	E_2^G	Following in contact → Lining up
		Pusher	–	Stop
		Leader	E_2^L	Moving to the goal → Stop

Table IV. The notations and initial positions of the robots and objects in the simulation.

Name	Notation	Initial position (cm)	Name	Notation	Initial position (cm)
Leader robot	L	$\mathbf{p}^L(t_0) = [680, 310]^{\mathrm{T}}$	Guider robot 6	G6	$\mathbf{p}_6^G(t_0) = [700, 275]^{\mathrm{T}}$
Pusher robot	PS	$\mathbf{p}^{PS}(t_0) = [800, 310]^{\mathrm{T}}$	Guider robot 7	G7	$\mathbf{p}_7^G(t_0) = [680, 275]^{\mathrm{T}}$
Goal	GOAL	$\mathbf{p}_{GOAL} = [500, 310]^{\mathrm{T}}$	Guider robot 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 robot 9	G9	$\mathbf{p}_9^G(t_0) = [780, 345]^{\mathrm{T}}$
Object 2	O2	$\mathbf{p}^{O_2}(t_0) = [710, 325]^{\mathrm{T}}$	Guider robot 10	G10	$\mathbf{p}_{10}^G(t_0) = [760, 345]^{\mathrm{T}}$
Guider robot 1	G1	$\mathbf{p}_1^G(t_0) = [800, 275]^{\mathrm{T}}$	Guider robot 11	G11	$\mathbf{p}_{11}^G(t_0) = [740, 345]^{\mathrm{T}}$
Guider robot 2	G2	$\mathbf{p}_2^G(t_0) = [780, 275]^{\mathrm{T}}$	Guider robot 12	G12	$\mathbf{p}_{12}^G(t_0) = [720, 345]^{\mathrm{T}}$
Guider robot 3	G3	$\mathbf{p}_3^G(t_0) = [760, 275]^{\mathrm{T}}$	Guider robot 13	G13	$\mathbf{p}_{13}^G(t_0) = [700, 345]^{\mathrm{T}}$
Guider robot 4	G4	$\mathbf{p}_4^G(t_0) = [740, 275]^{\mathrm{T}}$	Guider robot 14	G14	$\mathbf{p}_{14}^G(t_0) = [680, 345]^{\mathrm{T}}$
Guider robot 5	G5	$\mathbf{p}_5^G(t_0) = [720, 275]^{\mathrm{T}}$			

(unit: cm).

Table V. Parameter information in the simulation.

	Notation	Value		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	Electric dipole field parameter	β	2
	s_ε	3 cm			

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 s, and thus, the order indices of the head robots (G7 and G14) in the rows were changed to $0.5N_G - 2$. 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 the lining-up state according to event E_2^G at 7 s. 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 the 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 s. These processes were executed iteratively until the two objects (O1 and O2) were transported to the goal. The travel time was 93 s and the average of object transportation speed was 2.7 cm/s.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

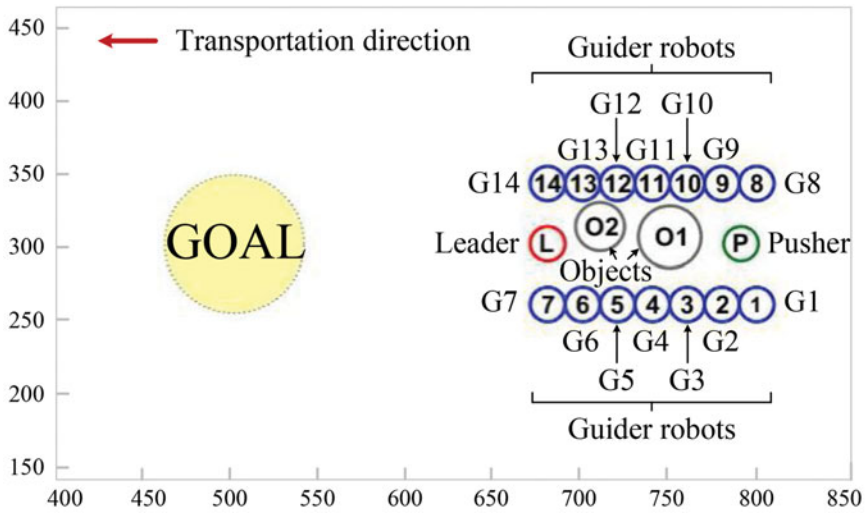


Fig. 9. Simulation environment of the proposed method.

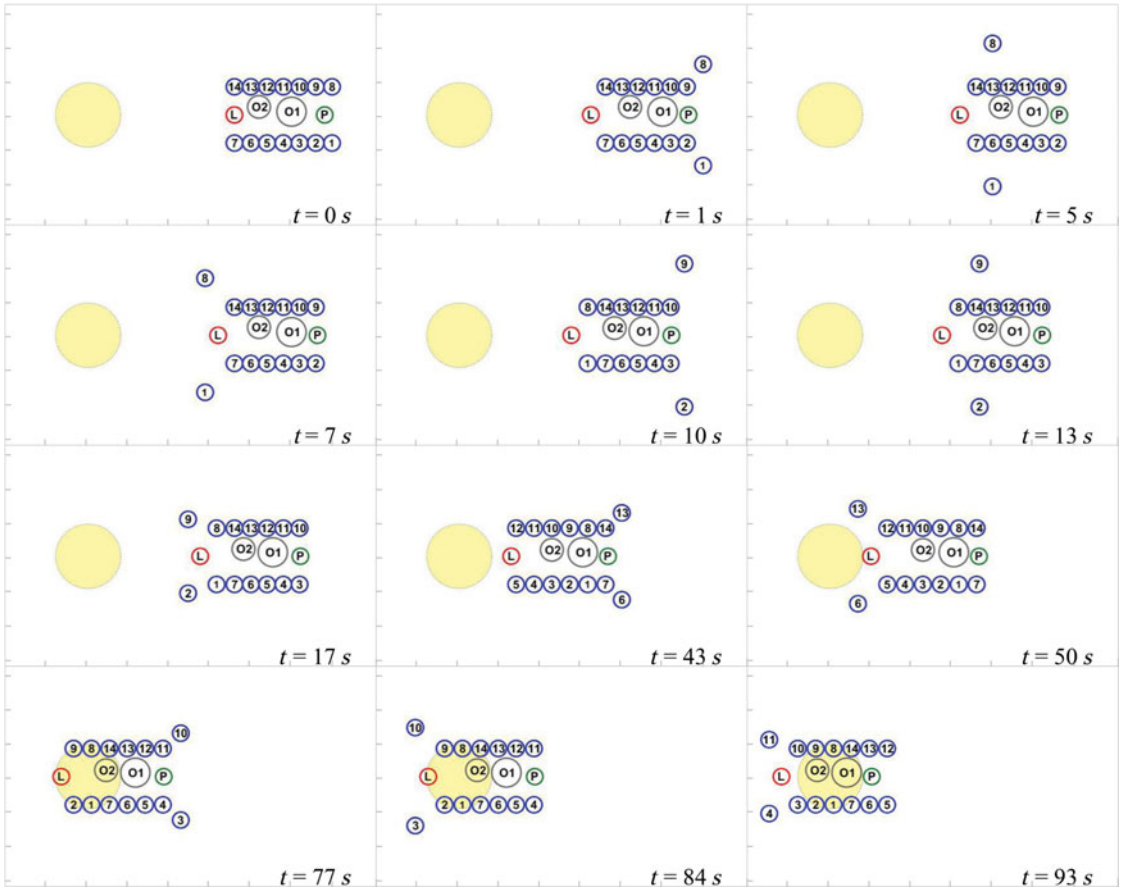


Fig. 10. Simulation results of the proposed technique.

Figure 11 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 VI. 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. 11. At nearly the same time (at 1 s), the states of the leader and guider robot 1 (G1) change to the moving-to-the-goal state and the following-in-contact state, respectively.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table VI. The states according to the state index in Fig. 11.

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

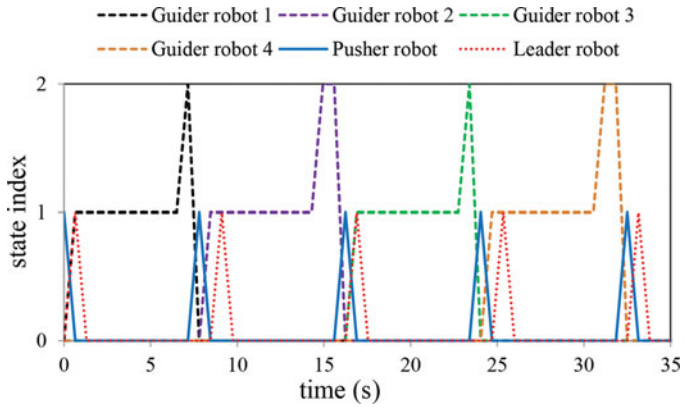


Fig. 11. The state transition results of the simulation. Only some of the guider robots are presented in the graph for a simple description; there were 14 guider robots in the simulation.

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 state to the lining-up state after the head robot is detected at 7.3 s. After the lining-up state is completed, the guider robot is stopped by the stop state at 7.5 s. 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 s.

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 12 shows the tangential and angular velocity of a guider robot (G1) for a total of 9 s. 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. 13, 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 s. At this point, the tangential velocity increased dramatically at first and decreased at the end of the lining-up state to track the virtual electric dipole field. This tangential velocity profile is the result of the bang-bang control scheme via (18). The angular velocity was also calculated using (13) in the lining-up state. Figure 14 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.

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. 15. 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 ($r = 10$cm).

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

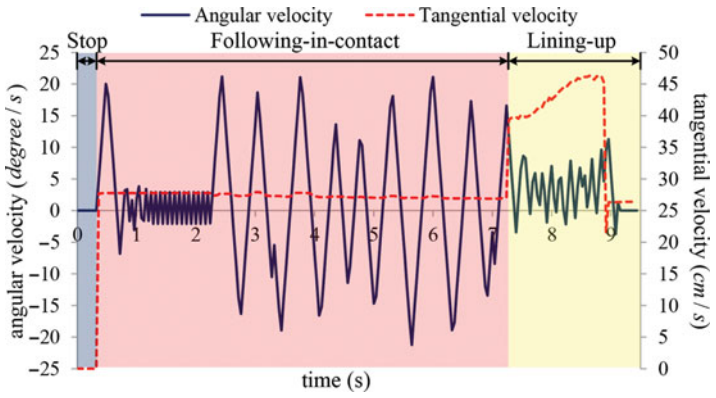


Fig. 12. The tangential and angular velocity of a guider robot (G1) for 9 s.

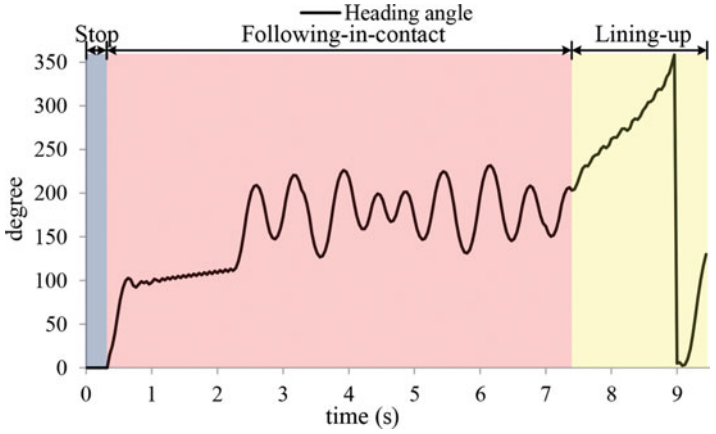


Fig. 13. The heading direction of a guider robot (G1) for 9 s. The dramatic decrease in the lining-up state is caused by the angular translation between 0° and 360°.

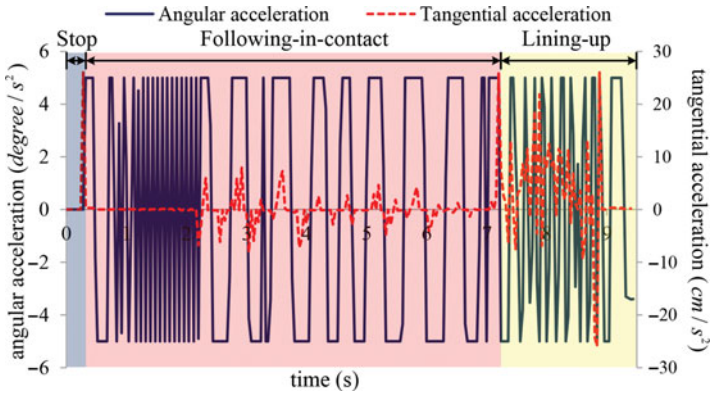


Fig. 14. The tangential and rotational acceleration of a guider robot (G1) for 9 s.

6. Practical Experiments

6.1. Experimental environments

We also conducted practical experiments to verify the proposed technique. The size of the experimental space was 140 cm × 140 cm. We used two types of differential-drive robots in the experiments: the E-puck and the Elisa-3 types. The E-puck robots are used as the 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. However, the pusher robot does not grasp or attach onto the objects using

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table VII. The localization errors of position tracking system.

Coordinate	Mean (cm)	Standard deviation (cm)
X	0.77	0.68
Y	0.95	0.94

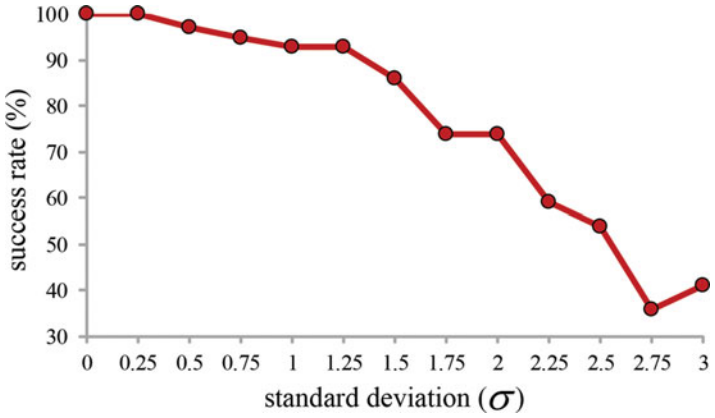


Fig. 15. The success rate graph by changing the standard deviation in Gaussian noise with zero-mean. The success rate drops under 70% after standard deviation is 2.

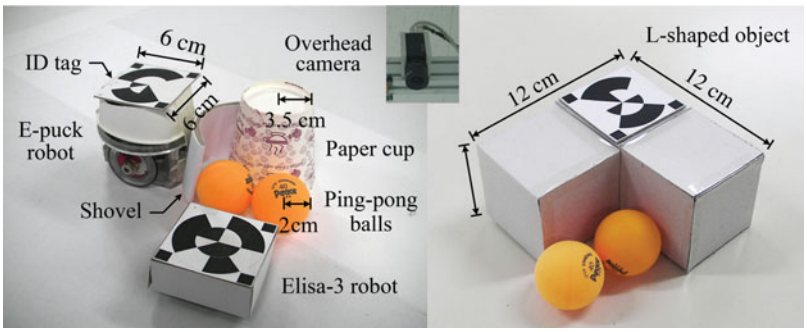


Fig. 16. Two types of differential-drive robots were used in the experiment: the E-puck and the Elisa-3 robots. A paper cup and two ping-pong balls were used as objects. ID tags were attached to each robot for position tracking.

the shovel. The maximum velocities of the E-puck and Elisa-3 robots were 2 cm/s and 6 cm/s, respectively, and the diameters of these robots are 7 cm and 5 cm, respectively. 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 movable and multi-object transportation and the L-shaped object is used to verify large-object transportation. We attached rectangular ID tags (6 cm $\times$ 6 cm) onto each robot for position tracking. The tracking system captured all ID tags in real time using an overhead camera and acquired the positions of the robots. The localization errors of our tracking system are described in Table VII.[27] 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, which was acquired in this case 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 practical experiment satisfied the condition of decentralized system. All of the robots and objects used in this experiment are shown in Fig. 16.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

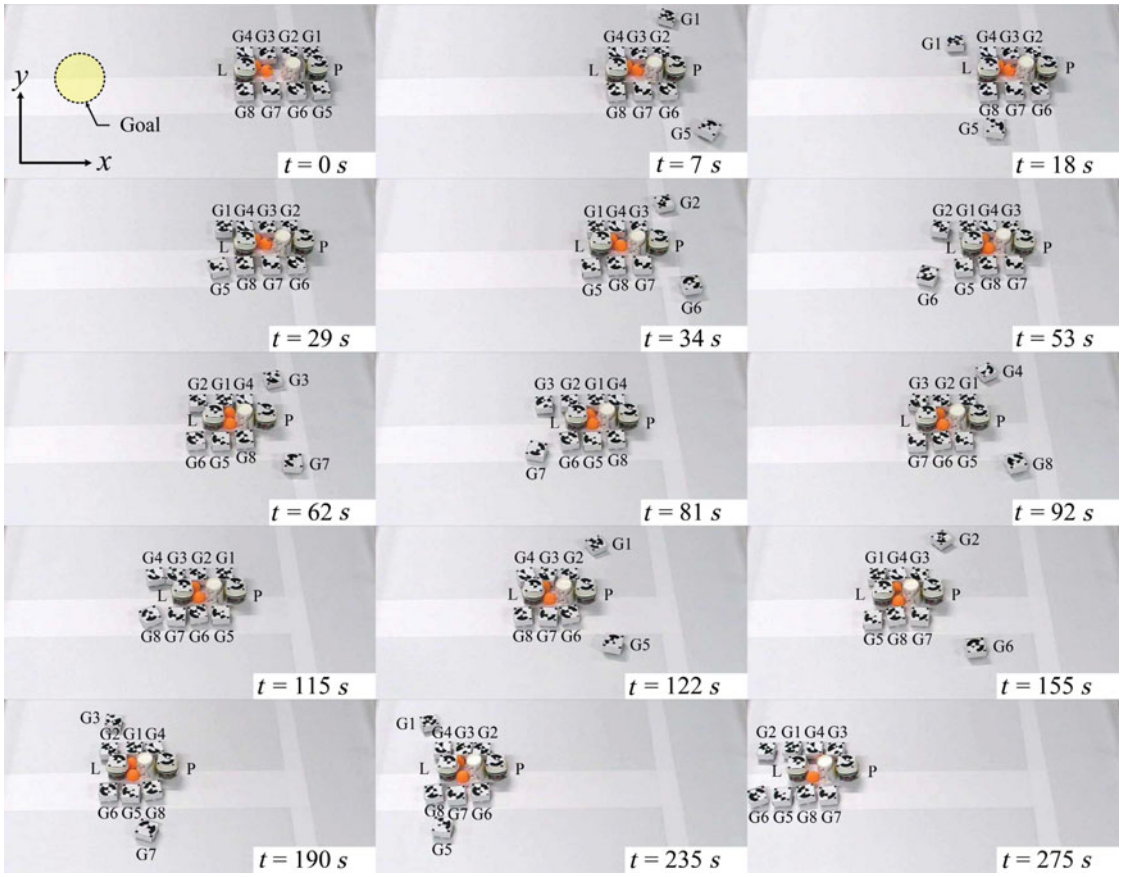


Fig. 17. Multi-object transportation with a straight path. Three objects, including rolling objects, were transported successfully in 275 s.

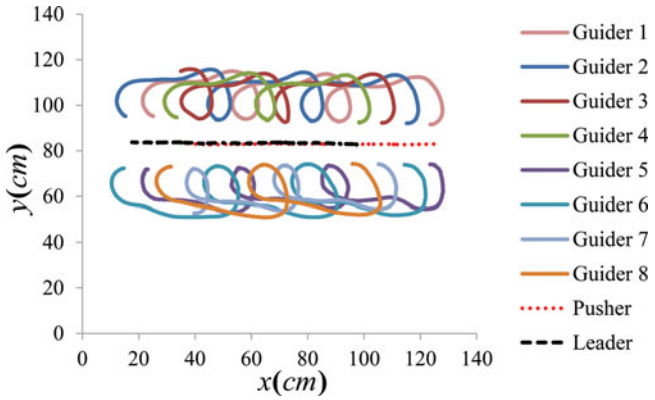


Fig. 18. The trajectories of the robots during the transportation of multiple small objects with a straight path.

6.2. Experimental results of the proposed technique

We conducted on three types of practical experiments to verify a diverse range of applications, as follows: multiple small-object transportation with a straight path, multiple small-object transportation with a bending path, and multiple large-object transportation using multiple leader and pusher robots.

6.2.1. Small-object transportation with a straight path. The first experiment involved small-object transportation with a straight path, as shown in Fig. 17. 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 eight 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.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

Table VIII. The summary of the experimental results according to localization errors.

Standard deviation (cm) (Gaussian noise with zero-mean)	Successful trials	Completion time (s)	
		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

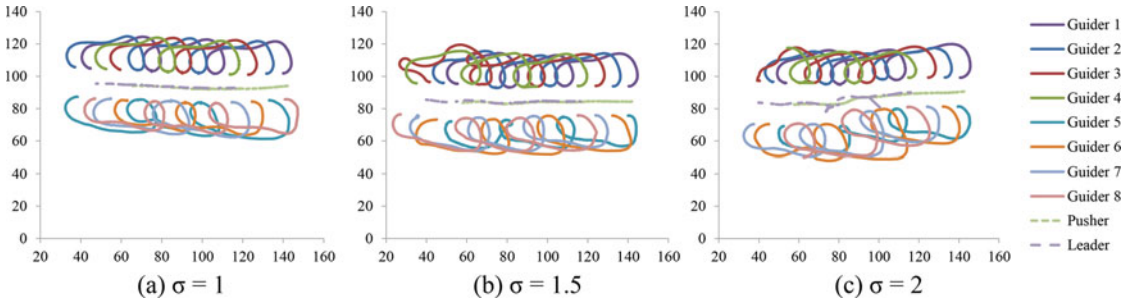


Fig. 19. The trajectories of robots according to localization errors with zero-mean Gaussian noise. The standard deviation (σ) has an effect on the trajectories of robots and the success rate of transportation. (a) $\sigma = 1$. (b) $\sigma = 1.5$. (c) $\sigma = 2$.

Two ping-pong balls and a paper cup, as noted earlier, were used as the manipulated objects. The initial position 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 s, as shown in Fig. 17. At the same time, two tail robots (G1 and G5) followed each row in the following-in-contact state for 18 s. The moving guider robots (G1 and G5) detected the head robot (G4 and G8) within the maximum sensing range ($s_{\max} = 18$cm) at 18 s, which triggered a change to the lining-up state. The guider robots generate a virtual electric dipole field ($\varphi = 0^\circ$) 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. 17, at 34 s. 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 s and the average travel distance of the eight guider robots was 278 cm. The trajectories of the robots are illustrated in Fig. 18.

The above experiment was conducted in our position tracking system which has the localization errors as described in Table VII. 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 VIII and Fig. 19 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. 19. The guider robots showed unnecessary motion in the following-in-contact state and they could not stand in a line in the lining-up state, as shown in Fig. 19(c).

6.2.2. Small-object transportation with a bending path. The second experiment involved small-object transportation with a bending path. In this experiment, the initial position and goal position of the

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

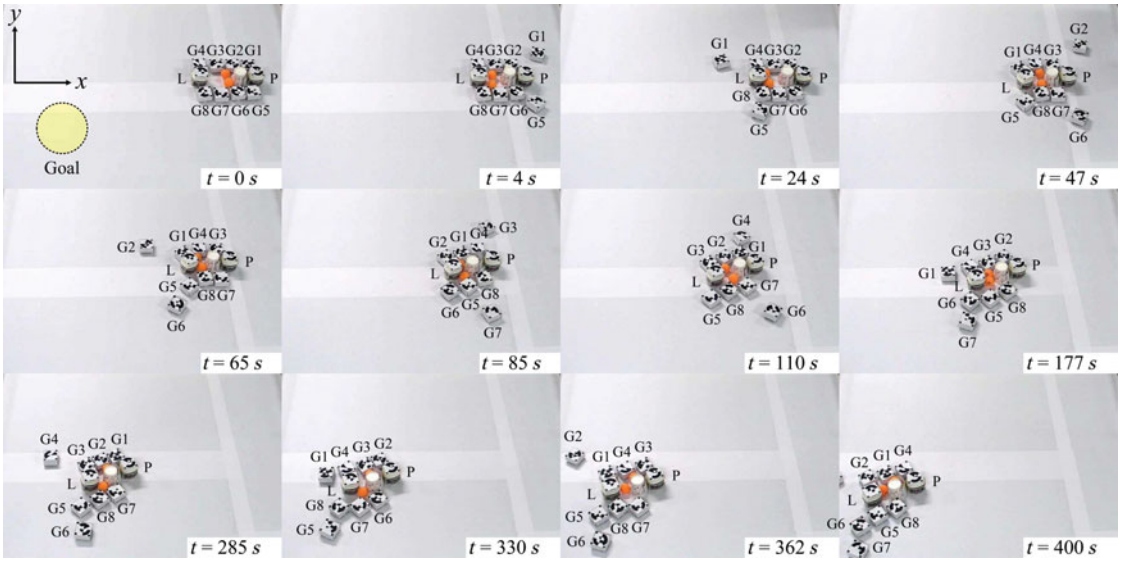


Fig. 20. Multi-object transportation with a bending path. Three objects, including rolling objects, were transported successfully in 400 s.

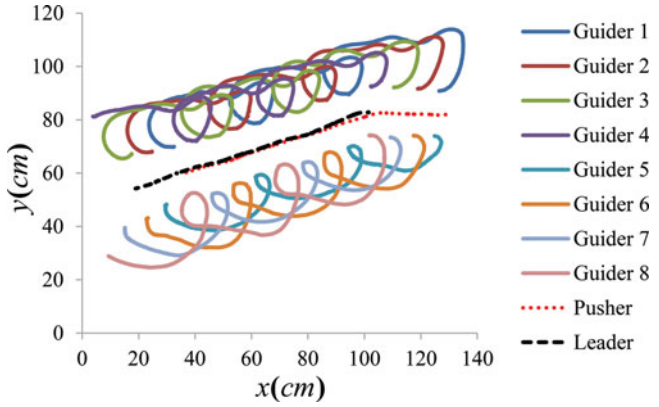


Fig. 21. The trajectories of the robots in the transportation experiment with a bending path ($\varphi = 30^\circ$).

objects were (120, 80) cm and (30, 50) cm, respectively. The basic transportation process was identical to the previous object transportation test with a straight path, except for the rotational direction 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. 20 at 47 s. The virtual electric dipole field was generated using (4), and the rotational angle φ was 30°. The pusher robot pushed objects in the bending direction because the parallel line 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 s and the average travel distance of the eight guider robots was 479.4 cm. The trajectories of the robots are illustrated in Fig. 21.

6.2.3. Large-object transportation. The last test assessed large-object transportation. For the large-object transportation test, multiple leader and pusher robots were used. 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, 10 Elisa-3 robots were used for the 10 guider robots (G1–G10). The initial position 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. For example, two pusher robots share the orders of the left

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

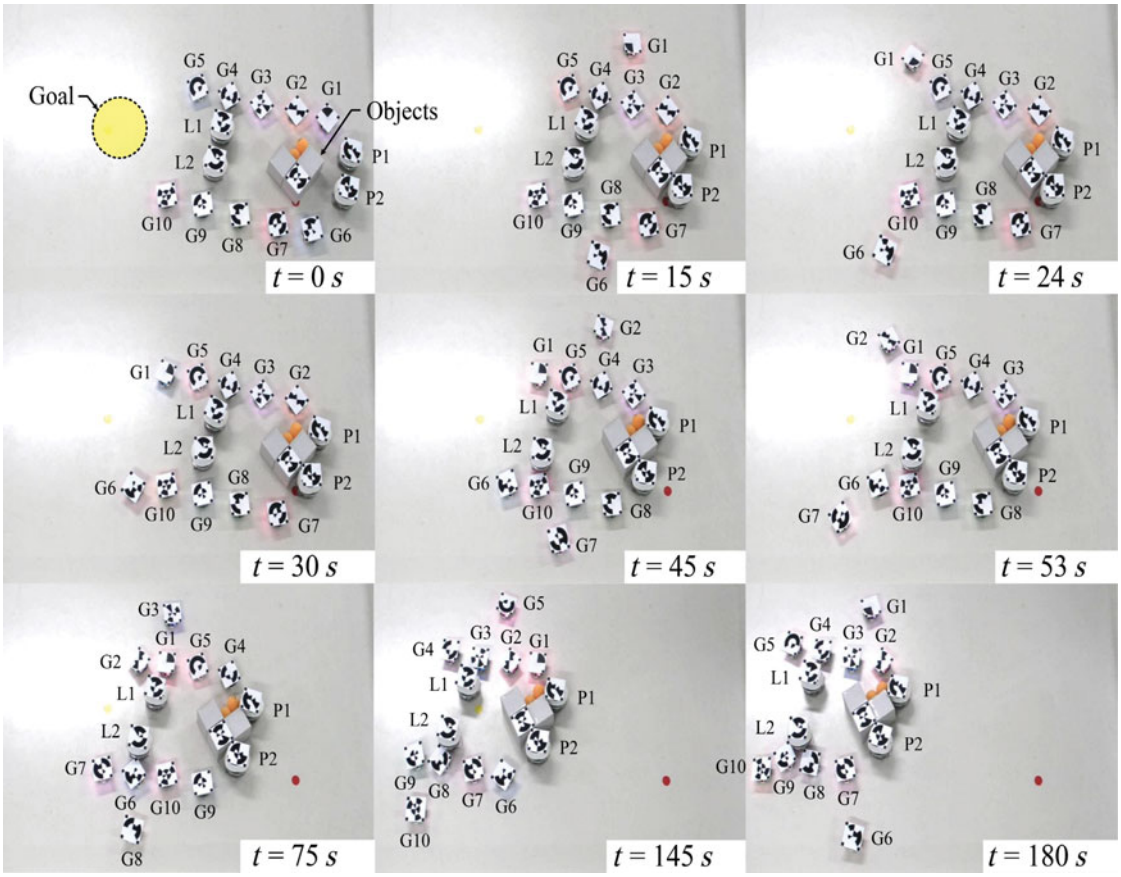


Fig. 22. Large-object transportation using multiple pusher robots and leader robots. Three objects, including large or rolling objects, were transported successfully in 180 s.

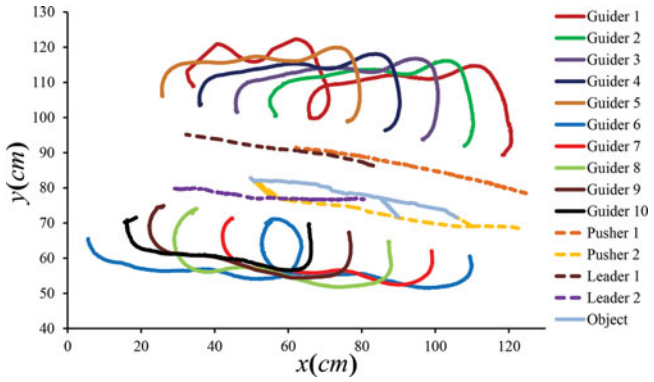


Fig. 23. The trajectories of the robots during large-object transportation.

and right guider robots through the local communication. In Fig. 22 at 0 s, 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 travel time for large-object transportation was 180 s. The trajectories of the robots are illustrated in Fig. 23.

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

7. Conclusion

This paper proposes a decentralized cooperative robot team for object transportation using parallel line formation with circular 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 line formation using a virtual electric dipole field to guide the objects. The main contribution of the paper is that multiple robots create an extended parallel line toward the goal using cyclic motion and transport objects using the finite state machine of each robot. Multiple objects can 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 are also satisfied. The proposed technique can be applied to various real object transportation applications, such as garbage collectors, foraging and transport vehicles.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2013R1A2A1A05005547), in part by the Brain Korea 21 Plus Project, in part by ASRI.

References

1. T. Arai, E. Pagello and L. E. Parker, "Guest editorial: Advances in multirobot systems," *IEEE Trans. Robot. Autom.* **18**(5), 655–661 (2002).
2. Y. U. Cao, A. S. Fukunaga and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auton. Robots.* **4**(1), 7–27 (1997).
3. J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar and J. P. Ostrowski, "Cooperative Localization and Control for Multi-Robot Manipulation," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, HI (2001) pp. 631–636.
4. M. N. Ahmadabadi and E. Nakano, "A "constrain and move" approach to distributed object manipulation," *IEEE Trans. Robot. Autom.* **17**(2), 157–172 (2001).
5. V. Shinde, A. Dutta and A. Saxena, "Experiments on multi-agent capture of a stochastically moving object using modified projective path planning," *Robotica* **31**(2), 267–284 (2013).
6. A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robot. Res.* **14**(4), 319–334 (1999).
7. Y. H. Liu, "Qualitative test and force optimization of 3d frictional form-closure grasps using linear programming," *IEEE Trans. Robot. Autom.* **15**(1), 163–173 (1999).
8. Z. Liu, H. Kamogawa and J. Ota, "Fast grasping of unknown objects through automatic determination of the required number of mobile robots," *Adv. Robotics* **27**(6), 445–458 (2013).
9. R. Groß, E. Tuci, M. Dorigo, M. Bonani and F. Mondada, "Object Transport by Modular Robots that Self-Assemble," Proceedings of IEEE International Conference on Robotics and Automation, Los Alamitos, CA (2006) pp. 2558–2564.
10. M. J. Mataric, M. Nilsson and K. T. Simsarian, "Cooperative Multi-Robot Box-Pushing," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA (1995) pp. 556–561.
11. Q. Li and S. Payandeh, "Manipulation of convex objects via two-agent point-contact push," *Int. J. Robot. Res.* **26**(4), 377–403 (2007).
12. B. P. Gerkey and M. J. Matari´c, "Sold!: Auction methods for multi-robot coordination," *IEEE Trans. Robot. Autom.* **18**(5), 758–768 (2002).
13. G. Eoh, J. S. Choi and B. H. Lee, "Faulty robot rescue by multi-robot cooperation," *Robotica* **31**(8), 1239–1249 (2013).
14. G. A. S. Pereira, M. F. M. Campos and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *Int. J. Robot. Res.* **23**(7–8), 783–795 (2004).
15. W. Wan, R. Fukui, M. Shimosaka, T. Sato and Y. Kuniyoshi, "Grasping by Caging: A Promising Tool to Deal with Uncertainty," Proceedings of IEEE International Conference on Robotics and Automation, Saint Paul, MN (2012) pp. 5142–5149.
16. B. Donald, L. Gariepy and D. Rus, "Distributed Manipulation of Multiple Objects using Ropes," Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA (2000) pp. 450–457.
17. J. Fink, P. Cheng and V. Kumar, "Cooperative Towing with Multiple Robots," *Proceedings of the ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, New York, USA (2008) pp. 1269–1276.
18. G. Eoh, K. W. Lee, J. H. Oh, S. H. Lee and B. H. Lee, "Cooperative multiple-object towing with linked robots," *J. Commun. Comput.* **10**(3), 385–393 (2013).
19. A. Yamashita, T. Arai, J. Ota and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *IEEE Trans. Robot. Autom.* **19**(2), 223–237 (2003).

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press

20. G. Eoh, Y. Cho and B. H. Lee, "Reconfigurable Formation Control for Multi-object Transportation," Proceedings of the 12th International Conference on Distributed Autonomous Robotic Systems, Daejeon, South Korea (2014) pp. 304–305.
21. J. Ren, K. A. McIsaac and R. V. Patel, "A potential model using generalized sigmoid functions," *IEEE Trans. Syst. Man Cybern. B.* **37**(2), 477–484 (2007).
22. 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 Trans. Image Process.* **21**(4), 2130–2140 (2012).
23. D. K. Cheng, *Field and Wave Electromagnetics*, 2nd ed. (Addison-Wesley, Reading, MA, 1989).
24. T. Igarashi, Y. Kamiyama and M. Inami, "A Dipole Field for Object Delivery by Pushing on a Flat Surface," Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK (2010) pp. 5149–5119.
25. K. C. Koh and H. S. Cho, "A smooth path tracking algorithm for wheeled mobile robots with dynamic constraints," *J. Intell. Robot. Syst.* **24**(4), 367–385 (1999).
26. M. Rubenstein, A. Cornejo and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science* **345**(6198), 795–799 (2014).
27. D. Kim, K. H. Jeong and B. H. Lee, "An approach to multi-agent interactive control in an intelligent space," *Int. J. Control, Autom. Syst.* **13**(3), 697–708 (2015).

https://doi.org/10.1017/S0263574716000096 Published online by Cambridge University Press