Decentralized Algorithms for Multi-Robot Manipulation via Caging

Guilherme A. S. Pereira
Mario F. M. Campos
VERLab–Vision and Robotics Laboratory,
DCC, Universidade Federal de Minas Gerais
Belo Horizonte, MG 31270-010 Brazil
gpereira@dcc.ufmg.br
mario@dcc.ufmg.br

Vijay Kumar
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104-6228, USA
kumar@grasp.cis.upenn.edu

Abstract

In this paper we address the problem of transporting objects with multiple mobile robots using the concept of "object closure". In contrast to other manipulation techniques that are typically derived from form or force closure constraints, object closure requires the less stringent condition that the object be trapped or caged by the robots. Our basic goal in this paper is to develop decentralized control policies for a group of robots to move toward a goal position while maintaining a condition of object closure. We present experimental results that show polygonal mobile robots controlled using visual feedback, transporting a convex polygonal object in an obstacle free environment toward a prescribed goal.

KEY WORDS—manipulation, caging, object closure, cooperative robots, decentralized control

1. Introduction

Object manipulation with mobile robots has been extensively discussed in the literature. Most approaches use the notions of force and form closure to perform the manipulation of relatively large objects (Ota, Miyata, and Arai 1995; Kosuge and Oosumi 1996; Rus 1997; Sugar and Kumar 1998). "Force closure" is a condition that implies that the grasp can resist any external force applied to the object. "Form closure" can be viewed as the condition guaranteeing force closure, without requiring the contacts to be frictional. In general, robots are the agents that induce contacts with the object, and are the only source of grasp forces. However, when external forces acting on the object, such as gravity and friction, are used together with contact forces to produce force closure, we have a situation of "conditional force closure". Several research groups have used conditional closure to transport an object by pushing it from an initial position to a goal (Mataric, Nilsson, and Simsarian 1995, Lynch 1996).

In contrast to these approaches, as shown in Figure 1, "object closure" requires the less stringent condition that the object be trapped or caged by the robots. (Our use of the concept of caging is slightly different from the definition in Rimon and Blake (1996), and hence the new term object closure.) In other words, although the object may have some freedom to move, it cannot be completely removed (Davidson and Blake 1998; Wang and Kumar 2002). Because a caging operation requires a relatively low degree of precision in relative positions and orientations, manipulation strategies based on caging operations are potentially more robust than, for example, approaches relying on force closure.

Caging was first introduced in Rimon and Blake (1996) for non-convex objects and two fingered gripers.[1] Other papers addressing variations on this basic theme are Davidson and Blake (1998), Sudsang and Ponce (1998, 2000), Sudsang et al. (1999), and Wang and Kumar (2002). Broadly speaking, our work may be considered closest to the work by Sudsang and Ponce (2000). They develop a centralized algorithm for moving three robots with circular geometry in an object manipulation task.

Our basic goal in this paper is to develop decentralized control policies for a group of mobile robots to move toward a goal position while maintaining the object closure condition. We assume that only local information is available to the robots, such as relative position and orientation of their near-

The International Journal of Robotics Research
Vol. 23, No. 7–8, July–August 2004, pp. 783-795,
DOI: 10.1177/0278364904045477
©2004 Sage Publications

1. E. Rimon credits K. Goldberg, University of California, Berkeley with the basic idea.

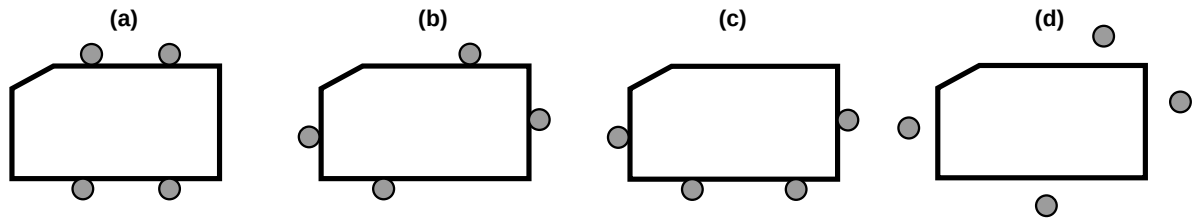

Fig. 1. Four technics of manipulation: (a) force closure (robots pressing the object); (b) form closure; (c) conditional closure (robots pushing the object up); (d) object closure or caging. Notice that in (d) the robots do not necessarily touch the object, which has some freedom to move but cannot be removed from the robots' formation.

est neighbors. Each robot knows the object shape, but does not have a dynamic model of the object. Further, each robot has approximate (and possibly infrequently updated) information about the object's orientation. Unlike previous work (Sudsang and Ponce 2000; Wang and Kumar 2002), we do not require the robots to be circular. However, we do introduce a number of simplifying assumptions to enable real-time implementation. Further, our interest is in transporting the object from an initial position toward a goal position in $\mathbb{R}^2$. We do not address the problem of precisely positioning and orienting the object in the plane.

The remainder of this paper is organized as follows. We first discuss the mathematical modeling of the object and the robots. In Section 3 we present our definition of object closure, and several key necessary conditions for establishing and maintaining object closure. In Section 4 we describe our approach to cooperative control, and in Section 5 we analyze its complexity. We briefly present results from experiments in Section 6. Finally, the main points of the paper and directions for future work are presented in Section 7.

2. Mathematical Modeling

Consider a planar world, $\mathcal{W} = \mathbb{R}^2$, occupied by a convex, polygonal object $\mathcal{O}$, and a group of n convex, polygonal robots. The ith robot R_i is described by the convex set $\mathcal{A}_i(q_i) \in \mathcal{W}$, where $q_i = (x_i, y_i, \theta_i)$ denotes the configuration of R_i. The configuration of the object is described by the coordinates $q = (x, y, \theta)$. We will use $\mathcal{C}_{\mathcal{R}_i}$ to denote the configuration space of a robot, while $\mathcal{C}$ will denote the configuration space for the object $\mathcal{O}$.

Convex robots and objects are represented by an intersection of m half-planes derived from the equations for each edge. The edge from (x_j, y_j) to (x_{j+1}, y_{j+1})[2] is given by $f_j(x, y) = a_j x + b_j y + c_j$ and $f_j(x, y) < 0$ for all points in the interior of the polygon.

If robot positions and orientations are held fixed, the region in the configuration space that corresponds to an interpenetration between the object $\mathcal{O}$ and the robot i is

$$\mathcal{C}_{obj\_i} = \{q \in \mathcal{C} \mid \text{interior}(\mathcal{A}_i(q_i) \cap \mathcal{O}(q) \neq \emptyset)\}, \quad (1)$$

2. $j+1$ is replaced by 1 for $j = m$ and $j - 1$ by m for $j = 1$.

where $\mathcal{O}(q)$ is the representation of $\mathcal{O}$ in the configuration q. This is the "configuration space object", analogous to the configuration space obstacle defined in the motion planning literature (Latombe 1991).

It is well known that, for a planar world, $\mathcal{C}_{obj\_i}$ can be represented as a three-dimensional solid. Slices through this solid yield polygonal cross-sections, each representing the configuration space for a constant orientation. There is an efficient method for computing each slice (a specific object orientation) of this solid in the case of convex polygonal objects and robots (Latombe 1991). The boundary of $\mathcal{C}_{obj\_i}$ is constructed with the edges of the robot and the object. It is well known that the running time of the algorithm is $O(l + m)$, where l is the number of edges of the robot and m is the number of edges of the object.

3. Object Closure

3.1. Definition

Before we proceed further, we will make three assumptions in this section.

ASSUMPTION A1. All robots are holonomic and identical in terms of geometry, and in terms of capabilities and constraints related to sensing, control, and mobility.

ASSUMPTION A2. All robots are point robots—$\mathcal{A}_i(q_i) = q_i = (x_i, y_i)$.

ASSUMPTION A3. The manipulated object cannot rotate—the coordinates of the object are given by $q = (x, y)$ and $\mathcal{C} \subset \mathbb{R}^2$.

Assumptions A2 and A3 make it easier to explain the basic ideas and will be relaxed in the next sections.

Figure 2 shows the boundary of $\mathcal{C}_{obj\_i}$ for a five-sided polygonal object and the point robot R_i.

The union of $\mathcal{C}_{obj\_i}$ for $1 \leq i \leq n$ determines the region in $\mathcal{C}$ which cannot be occupied by the object. Then,

$$\mathcal{C}_{obj} = \bigcup_{i=1}^{n} \mathcal{C}_{obj\_i}. \quad (2)$$

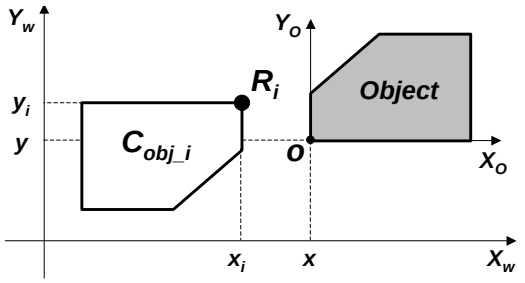

Fig. 2. $\mathcal{C}_{obj\_i}$ for a point robot R_i considering only object translations. By sliding the object around the robot, the origin, o, of the object-fixed reference frame traces out the boundaries of $\mathcal{C}_{obj\_i}$.

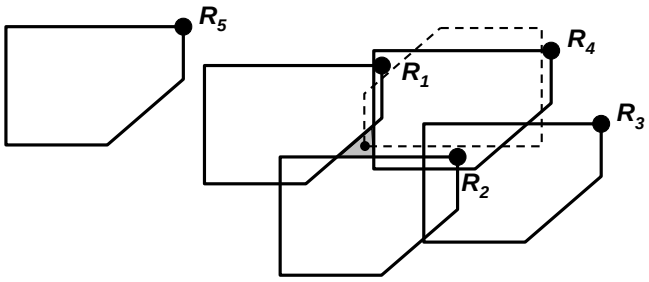

Fig. 4. Essential robots. Even with the removal of R_3 and R_5 the closure properties of the group are preserved. Thus, R_3 and R_5 are non-essential robots.

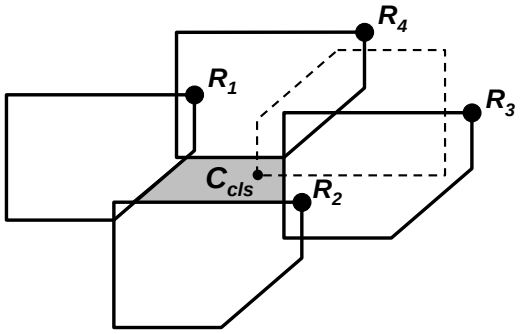

Fig. 3. Object closure. The interior (shaded gray) represents the closure configuration space, $\mathcal{C}_{cls}$, for a team of four robots. The dashed polygon represents the object. Notice that the origin of the object's reference frame is inside $\mathcal{C}_{cls}$, a compact set, indicating a condition of object closure.

Let the complement of $\mathcal{C}_{obj}$ in $\mathcal{C}$ be $\bar{\mathcal{C}}_{obj}$. When $\bar{\mathcal{C}}_{obj}$ consists of two (or more) disjoint sets, we use the term "object closure" to refer to the condition when one of these sets is compact and contains the object configuration, q. This is shown for four robots in Figure 3, where the compact set, which we refer to as the "closure configuration space" and denote by $\mathcal{C}_{cls}$, is shown shaded. Observe that the object is trapped or caged (in the terminology of Rimon and Blake (1996) when its origin is in $\mathcal{C}_{cls}$. Form closure is achieved in the limit $\|\mathcal{C}_{cls}\| \to \epsilon$, a small positive value, where $\|\cdot\|$ is a suitable measure of the set $\mathcal{C}_{cls}$ (Rimon and Burdick 1995).

We can easily relax Assumption A3 to accommodate the more general case with translations and rotations. In this case, eq. (2) remains the same, but $\mathcal{C}_{obj\_i}$ in eq. (1) is a three-dimensional solid whose cross-section, for a given angular orientation, is similar to the picture in Figure 2, and the compact subset $\mathcal{C}_{cls}$ consists of one or more three-dimensional solids whose cross-section (slice) is similar to that shown in Figure 3 (Wang and Kumar 2002).

We now define a "non-essential robot" with the help of Figure 4. In contrast to Figure 3 in which all four $\mathcal{C}_{obj\_i}$ (and therefore all four robots) are essential to construct the boundary for the closure configuration space, R_3 and R_5 are not essential for object closure in Figure 4. In a group of robots maintaining object closure, a non-essential robot, R_x, is a robot whose removal (and consequently the absence of the constraint due to $\mathcal{C}_{obj\_x}$) does not violate the state of object closure.

We now introduce a fourth assumption that allows us to establish a decentralized test for verifying object closure, which will be presented in Section 3.2.

ASSUMPTION A4. There are initially no non-essential robots in the group.

3.2. A Test for Object Closure

Checking the object closure condition involves two steps: (a) establishing the existence of $\mathcal{C}_{cls}$; (b) verifying $q \in \mathcal{C}_{cls}$. Step (a) requires obtaining state information from all robots and step (b) requires obtaining position (pose, in the more general case) of the object.

The key idea comes from Figure 3 where robots are numbered R_1 through R_n in a counterclockwise fashion. A necessary condition for object closure with no non-essential robots is that the ith robot's position satisfies $\mathcal{C}_{obj\_i-1} \cap \mathcal{C}_{obj\_i} \neq \emptyset$ and $\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_i+1} \neq \emptyset$. This condition is not sufficient. The sufficient condition involves verifying $\mathcal{C}_{cls} \neq 0$ and $q \in \mathcal{C}_{cls}$. However, this condition is necessary and sufficient for maintaining object closure once a condition of object closure is achieved. Hence we can state the following.

PROPOSITION 1. If at $t = 0$ an object is in a state of object closure with a group with no non-essential robots, a sufficient condition for maintaining object closure for $t > 0$ is $\mathcal{C}_{obj\_i-1} \cap \mathcal{C}_{obj\_i} \neq \emptyset$ and $\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_i+1} \neq \emptyset$, $1 \leq i \leq n$.

Proof. By the definition of object closure, at $t = 0$, $\bar{\mathcal{C}}_{obj}$ has at least two disjoint subsets and at least one of them, $\mathcal{C}_{cls}$, is bounded and contains the object configuration q. Since we initially do not allow non-essential robots, at $t = 0$, $\mathcal{C}_{obj}$ is necessarily connected. Further, $\mathcal{C}_{obj}$ is homeomorphic to an annulus in $\mathbb{R}^2$. In other words, there exists a continuous invertible map that maps $\mathcal{C}_{obj}$ to an annulus, with $\mathcal{C}_{cls}$ mapped to the interior of the annulus. If $\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_i+1} \neq \emptyset$, $1 \leq i \leq n$, $\mathcal{C}_{obj}$ will continue to be homeomorphic to an annulus and there are no paths from the interior of the annulus ($\mathcal{C}_{cls}$) to the exterior that do not cross the annulus. Thus, if the above condition is satisfied, the condition of object closure will be maintained. □

Note that Assumption A4 allows us to remove non-essential robots and establish sufficient conditions for object closure. Also note that the test for object closure is a decentralized test. Each robot (i) only needs information about adjacent robots ($\mathcal{C}_{obj\_i-1}$ and $\mathcal{C}_{obj\_i+1}$ to be precise) and the graph describing intersections of $\mathcal{C}_{obj\_i}$ is a Hamiltonian cycle. Thus, while Assumption A4 is not necessary from a practical standpoint, it is critical for the test in Proposition 1, which allows each robot to test for object closure by sensing adjacent robots.

We now explain how to derive the algebraic equations for object closure. For a generic robot R_k with neighbor R_i, we define $\mathcal{I}_i$ to be the subset of the robot configuration space that represents the intersection between $\mathcal{C}_{obj\_i}$ and $\mathcal{C}_{obj\_k}$:

$$\mathcal{I}_i = \{q_k \in \mathcal{C}_{R_k} \mid \mathcal{C}_{obj\_i}(q_i) \cap \mathcal{C}_{obj\_k}(q_k) \neq \emptyset\}.$$

Note that $\mathcal{C}_{obj\_i}(q_i)$ and $\mathcal{C}_{obj\_k}(q_k)$ are identical polygons, which introduces a symmetry in the form of $\mathcal{I}_i$. Further, it can be observed that

$$\mathcal{C}_{obj\_i} \cap \mathcal{C}_{obj\_k} \neq \emptyset \Leftrightarrow (q_k \in \mathcal{I}_i \wedge q_i \in \mathcal{I}_k).$$

Thus, the object closure conditions for each robot, which can be rewritten as $q_i \in \mathcal{I}_{i-1}$ and $q_i \in \mathcal{I}_{i+1}$ (see Figure 5(a)), are represented as a set of inequality constraints of the form $g_j(q_{i-1}, q_i) \leq 0$ or $g_j(q_i, q_{i+1}) \leq 0$, where g_j are the functions that delimit $\mathcal{I}_{i-1}$ or $\mathcal{I}_{i+1}$, respectively. $\mathcal{I}_{i-1}$ ($\mathcal{I}_{i+1}$) is a $2m$-sided polygon defined by $2m$ algebraic constraints, each linear in q_{i-1} and q_i (q_i and q_{i+1}). Since each polygon has up to $2m$ sides, the number of constraints for each robot is $4m$. For the situation we are considering, where the robots are points and the object cannot rotate, the boundary of $\mathcal{C}_{obj\_i}$ consists of the edges of $\mathcal{O}$ but ordered in a different way (see Figure 2). Then, each $\mathcal{I}_i$, which depends on $\mathcal{C}_{obj\_i-1}$ and $\mathcal{C}_{obj\_i+1}$, is bounded by two sets of edges, each taken from the object's polygonal description (refer to the algorithm presented in Latombe 1991 for proofs). In other words, $\mathcal{I}_{i-1}$ is given by functions $g_j(q_{i-1}, q_i)$, while $\mathcal{I}_{i+1}$ is given by another set of functions $g_j(q_i, q_{i+1})$, and each function is directly derived from the functions $f_i(x, y)$ used to describe the object. Thus, define Γ_i to be region of R_i's configuration bounded by a subset of the constraints $g_j(q_{i-1}, q_i) \leq 0$, and $g_j(q_i, q_{i+1}) \leq 0$, $1 \leq j \leq 2m$:

$$\Gamma_i = \mathcal{I}_{i-1} \cap \mathcal{I}_{i+1}.$$

An example of Γ_i can be seen in Figure 5(b).

We can now rewrite Proposition 1 as follows.

PROPOSITION 2. If at $t = 0$ an object is in a state of object closure with a group of non-essential robots, a sufficient condition for maintaining object closure at $t > 0$ is $q_i \in \Gamma_i$, $1 \leq i \leq n$.

3.3. Introducing Rotations

Thus far, we have ignored rotations. In reality, since the robots will collide with and bump against the object, the object can rotate. Even if object closure is guaranteed for a given object orientation, a small rotation followed by a translation may cause the object to "escape" from the robots' formation.

Our approach to incorporate rotations is to establish guarantees for object closure under the worst-case rotation. Because the object has no actuators, its maximum velocity is limited by the maximum velocity of the robots. Thus, if the object orientation at any instant is estimated to be θ_o, the orientation in the ensuing interval ΔT must be in the interval, $[\theta_{min}, \theta_{max}]$, where $\theta_{min} = \theta_o - \Delta T\, \omega_{max}$, $\theta_{max} = \theta_o + \Delta T\, \omega_{max}$, and ω_{max} is the (estimated) maximum object's angular velocity. Let $\mathcal{J}_i$ be defined as

$$\mathcal{J}_i = \bigcap_{\theta=\theta_{min}}^{\theta_{max}} \mathcal{I}_i(\theta),$$

where $\mathcal{I}_i(\theta)$ is $\mathcal{I}_i$ computed for an object orientation θ. Following the previous methodology, the conditions that guarantee object closure for all $\theta \in [\theta_{min}, \theta_{max}]$ are $q_i \in \mathcal{J}_{i-1}$ and $q_i \in \mathcal{J}_{i+1}$.

Since $\mathcal{C}_{obj\_i}$ is represented by the same polygon for every robot, the shape of $\mathcal{I}_i(\theta)$ is independent of the object orientation. As θ changes, $\mathcal{I}_i(\theta)$ is obtained by simply rotating $\mathcal{I}_i(\theta_o)$ around R_i. The intersection set $\mathcal{J}_i$ can be constructed as shown in Figure 6. The shaded area represents the configuration space where q_{i-1} and q_{i+1} must be in order to guarantee object closure for object orientations between θ_{min} and θ_{max}. It is bounded by circular arcs and the sides of $\mathcal{I}_i(\theta_{min})$ and $\mathcal{I}_i(\theta_{max})$. Notice that the set of inequality constraints, $g_j(q_{i-1}, q_i)$ (or $g_j(q_i, q_{i+1})$), may now be quadratic. However, the set $\mathcal{J}_i$ is still convex. From a practical standpoint, this set-valued approach for modeling the uncertainty in orientation allows us to be robust to errors in pose estimation.

3.4. Working with Polygonal Robots

The main challenge of working with polygonal robots is a practical one: the computation of $\mathcal{C}_{obj\_i}$ in real time. It is necessary to track changes in robots orientations and calculate the

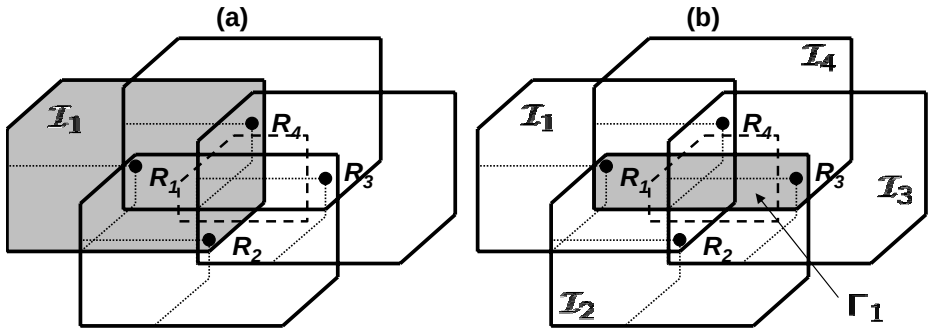

Fig. 5. Object closure is achieved if each robot i is inside Γ_i. The shaded areas represent (a) $\mathcal{I}_1$ and (b) Γ_1. Also, observe that $\Gamma_1 = \Gamma_3$ and $\Gamma_2 = \Gamma_4$.

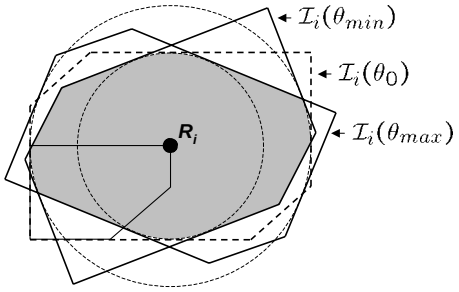

Fig. 6. $\mathcal{J}_i$ for the object in Figure 2 with $\Delta T\ \omega_{max} = 20°$.

shape of $\mathcal{C}_{obj\_i}$ in real time. Differences in the shapes of $\mathcal{C}_{obj\_i}$ complicate the intersection computations required to delineate $\mathcal{I}_i$ and $\mathcal{J}_i$. We pursue an alternative approach that lends itself to real-time implementation, one that involves deriving a sufficient condition for object closure. We define

$$\mathcal{C}_{OBJ\_k} = \mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0),$$

where $\mathcal{C}_{obj\_i}(0)$ is the configuration space object for the point robot located in the origin of the world reference frame and $\oplus$ is the Minkowski sum operator. This is the configuration space object for the polygonal robot R_k. We use $\mathcal{C}_{obj\_i}(q)$ to denote the configuration space object for a point robot at q:

$$\mathcal{C}_{obj\_i}(q) = \{q\} \oplus \mathcal{C}_{obj\_i}(0).$$

Notice that $\mathcal{C}_{OBJ\_k}$ can be constructed by the union of infinite $\mathcal{C}_{obj\_i}(q)$:

$$\mathcal{C}_{OBJ\_k} = \bigcup_{q \in \mathcal{A}_k(q_k)} \mathcal{C}_{obj\_i}(q).$$

Thus, we can write the following.

PROPOSITION 3. If $q_a \in \mathcal{A}_k(q_k)$ and $q_b \in \mathcal{A}_l(q_l)$ are the closest pair of points between R_k and R_l, then $\mathcal{C}_{obj\_i}(q_a) \cap \mathcal{C}_{obj\_i}(q_b) \neq \emptyset$ implies

$$(\mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0)) \cap (\mathcal{A}_l(q_l) \oplus \mathcal{C}_{obj\_i}(0)) \neq \emptyset.$$

Proof. Observe that $\mathcal{C}_{obj\_i}(q_a) = \{q_a\} \oplus \mathcal{C}_{obj\_i}(0)$ and $\{q_a\} \oplus \mathcal{C}_{obj\_i}(0) \subset \mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0)$, since $q_a \in \mathcal{A}_k(q_k)$. Also $\mathcal{C}_{obj\_i}(q_b) = \{q_b\} \oplus \mathcal{C}_{obj\_i}(0)$ and $\{q_b\} \oplus \mathcal{C}_{obj\_i}(0) \subset \mathcal{A}_l(q_l) \oplus \mathcal{C}_{obj\_i}(0)$, since $q_b \in \mathcal{A}_l(q_l)$. Because $(\{q_a\} \oplus \mathcal{C}_{obj\_i}(0)) \cap (\{q_b\} \oplus \mathcal{C}_{obj\_i}(0)) \neq \emptyset$, $(\mathcal{A}_k(q_k) \oplus \mathcal{C}_{obj\_i}(0)) \cap (\mathcal{A}_l(q_l) \oplus \mathcal{C}_{obj\_i}(0)) \neq \emptyset$. □

Using the closest pair of points as reference point robots for our computations leads us to a conservative but simple test for object closure for polygonal robots. Since $\mathcal{C}_{obj\_i}$ of a point robot can be computed off-line, the on-line computation is limited to the translation of this set to the location of the virtual point robots. This computation is illustrated in Figure 7.

3.5. Circular Objects and Robots

So far, we have consider only polygonal and point entities. Circular robots can be considered as a special case of point robots. Observe that the same methodology proposed for point robots can be directly applied if the object is grown by the size of the robots.

When circular objects are considered we can easily improve the efficiency of our methodology. In the case of point robots, because C_{obj_i} is a cylinder in the configuration space (constant for all orientations), the test for object closure reduces to a comparison between the diameter of the object with the distance between the robots. In the same way, the test when polygonal robots are considered reduces to simply checking the distance between the closest pair of points between two robots. This allows an exact solution for testing object closure (in contrast to the conservative one in the case of polygonal object in Section 3.4).

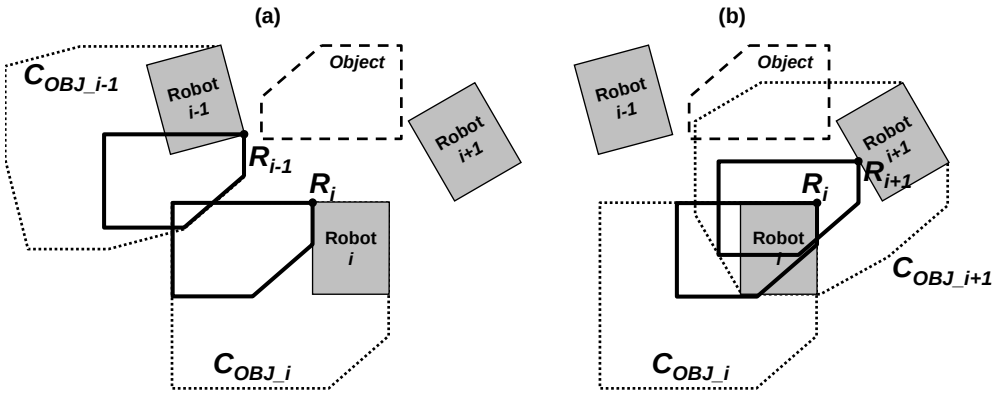

Fig. 7. Robot i checks closure (a) using the imaginary point robots, R_i and R_{i-1} (left), and (b) using a different set of point robots, R_i and R_{i+1} (right). The dotted polygons represent $\mathcal{C}_{OBJ\_k}$.

4. Control

Motivated by the sensors in our experimental test-bed, we assume each robot is able to sense the relative position and orientation of other robots in its field of view. Because all robots can be instrumented or tagged (for example, with color markers) this is a reasonable assumption. Further, each robot has a model of the object geometry and is able to estimate the position and orientation of the object. As it may not be possible to instrument all objects in the environment, we assume that these estimates have greater errors and may suffer from greater latency and slower update rates. Finally, we assume each robot has information about the goal destination for the object, q_{goal}, and its position (configuration) relative to this goal.

Our control system is decentralized and implemented using a set of reactive controllers. Each robot switches between the controllers as shown in Figure 8. The switches are governed by the activation of constraints that depend on the relative positioning of a robot with respect to its neighbors and the robots' estimate of the object orientation.

Recall from Proposition 2 that object closure constraints for R_i are defined by inequalities, $g_j(q_{i-1}, q_i) \leq 0$ or $g_j(q_i, q_{i+1}) \leq 0$. We consider the jth constraint to be active when $g_j = \delta_1$, where δ_1 is a small negative number that can be thought of as a threshold. In addition to ensuring $q_i \in \Gamma_i$, it is also necessary to ensure the robots do not try to cluster together thus crushing the object. From a practical standpoint, although the object may be rigid and immune to damage, this "clustering behavior" will cause large contact forces and jamming due to friction. To avoid this, we introduce a new set of constraints that prevent a robot from being very close to its neighbors: $g_j \geq \delta_2$, where $\delta_2 < \delta_1 < 0$. This defines a "safe" configuration space for each q_i where the object is caged but jamming is avoided. Practically, the set of constraints $\delta_2 \leq g_j \leq \delta_1$ define two polygons with the same shape as Γ_i but with different sizes. The reactive controllers and the sequential composition of these controllers are shown in Figure 8.

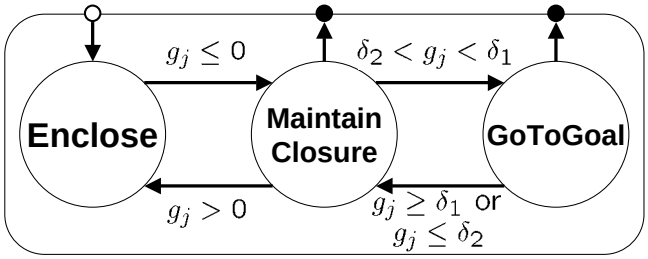

Fig. 8. The switched control system with three modes for multi-robot manipulation. δ_1 and δ_2 are thresholds for activating the transition between modes.

In this section, we consider a simple kinematic model for each robot. For the ith robot, the dynamical model is given by

$$\dot{q}_i = u_i,$$

where $q_i = (x_i, y_i)$. We will assume each robot has a potential function $\phi(q)$ with a unique minimum at q_{goal} which is presumably derived from a knowledge of the obstacles and the goal destination for the object. Further, we will assume each robot knows this potential function, with a reference input given by

$$u_T = -\nabla\phi(q), \tag{3}$$

where ∇ is the gradient operator.

We will denote the constraints due to robot R_{i-1}, which have the form $g_j(q_{i-1}, q_i) \leq 0$, by g_j^l and those due to robot R_{i+1}, which have the form $g_j(q_i, q_{i+1}) \leq 0$ by g_j^r.

In the Enclose mode, each robot tries to initially achieve object closure. The control input in this mode is

$$u_i = -k_1 \left(a\nabla g_j^r + b\nabla g_p^l\right), \tag{4}$$

where ∇g_y^x is a unit vector along the gradient of the constraint defined by

$$\nabla g_y^x = \frac{\partial g_y^x/\partial q_i}{\|\partial g_y^x/\partial q_i\|}.$$

∇g_j^l is due to robot $i-1$ and ∇g_p^r is due to robot $i+1$. The variables a and b can each be -1, 0, or 1. When $g_j \leq \delta_2$, the value -1 is assigned. When $\delta_1 > g_j > \delta_2$, the value 0 is assigned. When $g_j \geq \delta_1$, the value 1 is assigned. Since the gradient vectors are normalized, the positive constant k_1 determines the robot velocity.

It is necessary to make two remarks about eq. (4). First, it is possible that more than one constraint may be active between a pair of robots. In such a case, we simply choose j and g_j^r (and similarly p and g_p^l) to be the one corresponding to the closest constraint boundary. If there are two constraints whose boundaries are equally close, we must replace ∇g_y^x with the generalized gradient. Secondly, this equation is only valid for situations where the robots are close to achieving object closure, i.e. situations where with a small motion the robots would achieve this condition. Achieving object closure requires global knowledge about the object and it is difficult to establish guarantees with decentralized approaches, except in simple cases such as with point robots and circular objects. A discussion of such strategies and their limitations is provided in Song and Kumar (2002).

In the MaintainClosure mode, a robot tries to maintain object closure while navigating toward the goal. The control input for this state is

$$u_i = -k_1 \left(a\nabla g_j^r + b\nabla g_p^l\right) + k_2\, u_T, \tag{5}$$

where k_2 is a positive constant and u_T is given by eq. (3).

In the GoToGoal mode, the robots move towards the goal without any reference to the constraints. This mode has the following input:

$$u_i = k_2\, u_T. \tag{6}$$

Thus, each robot follows the reference input given by u_T. Observe that the controller (5) reduces to eq. (6) when $a = b = 0$. However, if any constraint $g_j \geq \delta_1$ or $g_j \leq \delta_2$ ($a \neq 0$ or $b \neq 0$) is violated, the controller switches to the MaintainClosure mode in eq. (5). We will prove that this controller guarantees that the condition of object closure is maintained.

Before we do that, we will need to make an important observation about the constraints $g_j(q_i, q_k)$. Let us consider, for illustrative purposes, the special case of object translations and point robots.

LEMMA 1. For the special case of object translations and point robots, a generic robot R_k induces a constraint $g_j(q_i, q_k)$ in R_i of the form $a_j(x_i - x_k) + b_j(y_i - y_k) + c_j \leq 0$, which is linear in q_i and q_k. Further, for each $g_j(q_i, q_k)$ there is another constraint induced in R_k by R_i, $g_p(q_k, q_i)$, such that

$$\frac{\partial g_j}{\partial q_i} = -\frac{\partial g_p}{\partial q_k}. \tag{7}$$

Proof. In order to prove this lemma we will refer to the algorithm for computing the bounds of the intersection between a movable (robot) and a fixed (obstacle) polygonal region in the configuration space proposed originally in Lozano-Pérez (1983) and presented in Latombe (1991). By this algorithm, the edges of the intersection are the edges of the fixed polygon and the negated edges (edges with direction opposite to the original ones) of the movable polygon, ordered by their normals.

Consider a generic edge of the object (the movable polygon in our case) given by $A_j x + B_j y + C_j = 0$ in the world-fixed coordinate system. This is transformed into the jth edge of C_{obj_i} as

$$a_j(x - x_i) + b_j(y - y_i) + c_j = 0,$$

where $a_j = -A_j$, $b_j = -B_j$ and $c_j = C_j$. A similar equation represents the jth edge of C_{obj_k}:

$$a_j(x - x_k) + b_j(y - y_k) + c_j = 0.$$

The boundary of $\mathcal{I}_i$ is obtained by fixing $\mathcal{C}_{obj\_i}$ and determining its intersection with $\mathcal{C}_{obj\_k}$ (the movable polygon). Thus, since both $\mathcal{C}_{obj\_i}$ and $\mathcal{C}_{obj\_k}$ are identical polygons, $\mathcal{I}_i$, which is centered at R_i, contains for each edge of $\mathcal{C}_{obj\_i}$, two edges with line equations of the form

$$a_j(x - x_i - d_{xr}) + b_j(y - y_i - d_{yr}) + c_j = 0, \tag{8}$$

and

$$-a_j(x - x_i - d_{xs}) - b_j(y - y_i - d_{ys}) + c_j = 0, \tag{9}$$

where d_{xr}, d_{yr}, d_{xs}, and d_{ys} are constant offsets that depend on the dimensions of $\mathcal{C}_{obj\_i}$. Observe that these two equations represent parallel lines; $\mathcal{I}_i$ contains parallel edges even if the object does not have any. Moreover, for every edge of $\mathcal{I}_i$, there is another edge that is parallel to it, lending symmetry to the shape of $\mathcal{I}_i$, independently of the object shape. Since the bounds of $\mathcal{I}_i$ represent constraints for R_k, we may write each constraint based on eqs. (8) and (9) as a function of $q_k = (x_k, y_k)$

$$a_j(x_k - x_i) + b_j(y_k - y_i) + c_r \leq 0, \tag{10}$$

and

$$-a_j(x_k - x_i) - b_j(y_k - y_i) + c_s \leq 0, \tag{11}$$

where the offsets are now included in c_x. The same observation can be made for $\mathcal{I}_k$ and q_i, yielding the constraints:

$$a_j(x_i - x_k) + b_j(y_i - y_k) + c_r \leq 0, \tag{12}$$

and

$$-a_j(x_i - x_k) - b_j(y_i - y_k) + c_s \leq 0. \tag{13}$$

We may say that eqs. (10) and (11) represent the rth and sth constraints, $g_r(q_k, q_i) \leq 0$ and $g_s(q_k, q_i) \leq 0$, respectively, for R_k. Similarly, eqs. (12) and (13) represent $g_r(q_i, q_k) \leq 0$ and $g_s(q_i, q_k) \leq 0$ for R_i. Now, observe by eqs. (10) and (13) that

$$\frac{\partial g_r(q_k, q_i)}{\partial q_i} = -\frac{\partial g_s(q_i, q_k)}{\partial q_k},$$

and by eqs. (12) and (11) that

$$\frac{\partial g_s(q_k, q_i)}{\partial q_i} = -\frac{\partial g_r(q_i, q_k)}{\partial q_k}.$$

□

From the form of eqs. (10) and (11), and as is evident from the proof of Lemma 1, the following result can be easily proven.

LEMMA 2. For each constraint $g_j(q_i, q_k)$ induced by R_k on R_i, of the form $a_j(x_i - x_k) + b_j(y_i - y_k) + c_j \leq 0$,

$$\frac{\partial g_j}{\partial q_i} = -\frac{\partial g_j}{\partial q_k}. \tag{14}$$

Observe that each constraint describes a line in the world reference frame translated by the position of one of the neighbors. Since $\mathcal{I}_i$ has the same form for all robots, when a constraint, $g_j(q_i, q_k)$, is active for one robot, there is an identical constraint with opposite sign, $-g_p(q_i, q_k)$, active for one of its neighbors. Figure 9 shows a typical situation when one constraint is active for the ith robot and an identical constraint, with opposite sign, is active for one of its neighbors. This is also the case (and eq. (14) is valid) when $g_j(q_i, q_k)$ is not an equation for a straight line (as is the case when rotations are considered). We use this observation to prove that, once the robots have captured the object, the controller (5) guarantees object closure is maintained.

PROPOSITION 4. Once the robots achieve the condition of object closure, the switched control system represented by eq. (5) guarantees object closure.

Proof. We consider a generic constraint involving a generic pair of robots R_i and R_k, $g_j(q_i, q_k) \leq \delta_1$, and show that, when the constraint is active, the control input makes $\dot{g}_j(q_i, q_k) \leq 0$.[3] The time derivative of $g_j(q_i, q_k)$ is given by

$$\dot{g}_j(q_i, q_k) = \frac{\partial g_j}{\partial q_i}\dot{q}_i + \frac{\partial g_j}{\partial q_k}\dot{q}_k. \tag{15}$$

3. A similar treatment can be pursued for the constraints $\dot{g}_j(q_i, q_k) \geq 0$ becoming active.

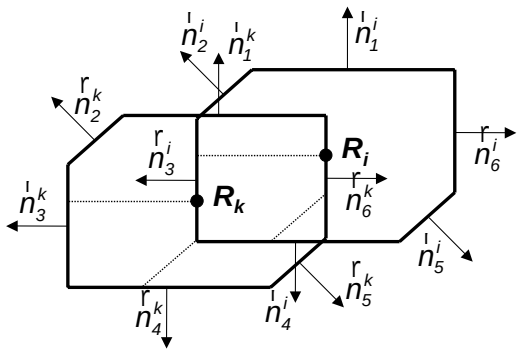

Fig. 9. An active constraint for robot R_i ($g_6^k(q_i, q_k) = 0$) indicates the activation of an identical constraint with opposite sign for one of its neighbors ($g_3^i(q_i, q_k) = 0$). In this picture, $\delta_1 = 0$. Notice that the normal vector of the active constraint for R_i, $\vec{n}_3^i$, is equal to $-\vec{n}_6^k$, the negative of the normal vector of the active constraint for R_k.

For the ith robot, if $g_j(q_i, q_k)$ is active, then for the kth robot, $g_p(q_i, q_k)$ is also active. Without loss of generality, let R_k be the left neighbor of R_i. In the control law (5), $\nabla g_j^l = \nabla g_j^k$ for R_i and $\nabla g_j^r = \nabla g_p^i = -\nabla g_j^k$ for R_k. Let ∇g_α be the term associated with the constraint induced by the other neighbor (right) of R_i and let ∇g_β be the term associated with the constraint induced by the other neighbor (left) of R_k. Substituting for $\dot{q}_i$ and $\dot{q}_k$ in eq. (15) from eq. (5), the time derivative of $g_j(q_i, q_k)$ is given by

$$\begin{aligned}
\dot{g}_j(q_i, q_k) =& \frac{\partial g_j}{\partial q_i} \cdot \left[-k_1 \left(a\nabla g_\alpha + \nabla g_j^k\right) + k_2 u_T\right] \\
&+ \frac{\partial g_j}{\partial q_k} \cdot \left[-k_1 \left(-\nabla g_j^k + b\nabla g_\beta\right) + k_2 u_T\right] \\
=& \left\| \frac{\partial g_j}{\partial q_i} \right\| \nabla g_j^k \cdot \left[-k_1 \left(a\nabla g_\alpha + \nabla g_j^k\right) + k_2 u_T\right] \\
&- \left\| \frac{\partial g_j}{\partial q_i} \right\| \nabla g_j^k \cdot \left[-k_1 \left(-\nabla g_j^k + b\nabla g_\beta\right) + k_2 u_T\right] \\
=& \left\| \frac{\partial g_j}{\partial q_i} \right\| \left[-k_1 \left(\nabla g_j^k \cdot \nabla g_j^k + a\nabla g_j^k \cdot \nabla g_\alpha\right)\right. \\
&+ k_2 \nabla g_j^k \cdot u_T \\
&- k_1 \left((-\nabla g_j^k) \cdot (-\nabla g_j^k) + b(-\nabla g_j^k) \cdot \nabla g_\beta\right) \\
&\left. + k_2 (-\nabla g_j^k) \cdot u_T\right] \\
=& -k_1 \left\| \frac{\partial g_j}{\partial q_i} \right\| \left(2\nabla g_j^k \cdot \nabla g_j^k + a\nabla g_j^k \cdot \nabla g_\alpha\right. \\
&\left. -b\nabla g_j^k \cdot \nabla g_\beta\right) .
\end{aligned}$$

We denote by θ_a the angle between the unit vectors ∇g_j^k and ∇g_α, and by θ_b the angle between the unit vectors ∇g_j^k and ∇g_β. The expression for $\dot{g}_j(q_i, q_k)$ becomes

$$\begin{aligned}\dot{g}_j(q_i, q_k) = &- k_1 \left\| \frac{\partial g_j}{\partial q_i} \right\| \left(2\|\nabla g_j^k\|^2 + a\|\nabla g_j^k\| \, \|\nabla g_\alpha\| \cos\theta_a \right. \\ &\left. - b\|\nabla g_j^k\| \, \|\nabla g_\beta\| \cos\theta_b\right) \\ = &- k_1 \left\| \frac{\partial g_j}{\partial q_i} \right\| (2 + a\cos\theta_a - b\cos\theta_b) \leq 0.\end{aligned}$$

Since $-1 \leq a\cos\theta_a \leq 1$ and $-1 \leq b\cos\theta_b \leq 1$, for all $a, b \in \{-1, 0, 1\}$, then $\dot{g}_j(q_i, q_k) \leq 0$. Therefore, given the initial conditions, $g_j(q_i, q_k) \leq 0$, for all $1 \leq i \leq n$, $k \in \{q-1, q+1\}$ and $1 \leq j \leq 4m$, and the fact that the derivatives $\dot{g}_j(q_i, q_k)$ are strictly smaller than 0 when the jth constraint is active, the proposition is proved. □

It should be noted that the controller in eq. (5) for the MaintainClosure mode makes the multi-robot system a switched system. This is because an attempt to decrease the value of an active constraint may result in another constraint becoming active, which in turn will result in a change in the right-hand side of eq. (5). Even if each instance of the control law (5) results in a desirable outcome, the performance of the switched system may result in undesirable consequences (Liberzon and Morse 1999). Because the system has a discontinuous right-hand side, it is necessary to consider Filippov (1988) solutions for the switched system in order to analyze solutions along constraint boundaries. This analysis is beyond the scope of the paper. However, we note that extensive experimentation and numerical simulations with polygonal and circular objects have shown that the system is not plagued by chattering behavior.

Using the same methodology presented above we can also prove that when a constraint $g_j(q_i, q_k) = \delta_2$, the control law in the MaintainClosure mode maintains the condition $g_j(q_i, q_k) \geq 0$. Thus, if the robots are in the MaintainClosure mode, they either stay in this mode while moving toward the goal, or they switch to the GoToGoal mode.

It is also important to show that even when the robots are in the MaintainClosure mode trying to preserve the constraints, the whole team (including the object) moves toward the goal. In order to show this, we define the group position, $\bar{q}$, and group velocity, $\dot{\bar{q}}$, respectively as follows:

$$\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i, \qquad \dot{\bar{q}} = \frac{1}{n}\sum_{i=1}^{n} \dot{q}_i.$$

We will now show, when the robots are either in the MaintainClosure mode or the GoToGoal mode, the group velocity is always parallel to u_T.

PROPOSITION 5. If all the robots are in a state of object closure, the controllers in eqs. (5) and (6) guarantee that the group velocity is in the direction of u_T.

Proof. We define v_T to be a unit vector perpendicular to u_T. We need to prove that (a) $u_T \cdot \dot{\bar{q}} > 0$ and (b) $v_T \cdot \dot{\bar{q}} = 0$. Given the control law (5) we can write

$$u_T \cdot \dot{q}_i = -k_1 u_T \cdot \left(a_i \nabla g_j^{i-1} + b_i \nabla g_j^{i+1}\right) + k_2 \|u_T\|^2,$$

and therefore

$$\sum_{i=1}^{n} u_T \cdot \dot{q}_i = -\sum_{i=1}^{n} k_1 u_T \cdot \left(a_i \nabla g_j^{i-1} + b_i \nabla g_j^{i+1}\right) + \sum_{i=1}^{n} k_2 \|u_T\|^2$$

$$u_T \cdot \sum_{i=1}^{n} \dot{q}_i = -k_1 u_T \cdot \sum_{i=1}^{n} \left(a_i \nabla g_j^{i-1} + b_i \nabla g_j^{i+1}\right) + n k_2 \|u_T\|^2. \tag{16}$$

For each active constraint with gradient ∇g_j^k, there is another identical constraint with gradient $-\nabla g_j^k$ as discussed before. Thus, the summation on the right-hand side of eq. (16) is zero and we can rewrite this equation as

$$u_T \cdot \sum_{i=1}^{n} \dot{q}_i = n k_2 \|u_T\|^2.$$

Since k_2 is a positive constant,

$$u_T \cdot \dot{\bar{q}} = k_2 \|u_T\|^2 > 0.$$

In the same way, we can write

$$v_T \cdot \dot{q}_i = -k_1 v_T \cdot \left(a_i \nabla g_j^{i-1} + b_i \nabla g_j^{i+1}\right) + k_2 v_T \cdot u_T,$$

and, since v_T is perpendicular to u_T,

$$v_T \cdot \sum_{i=1}^{n} \dot{q}_i = -k_1 v_T \cdot \sum_{i=1}^{n} \left(a_i \nabla g_j^{i-1} + b_i \nabla g_j^{i+1}\right).$$

Because the summation on the right-hand side is zero,

$$v_T \cdot \dot{\bar{q}} = 0.$$

Therefore, since the group velocity is in the direction of u_T, the ensemble follows the reference input toward the goal. □

It is more difficult to prove that the control law (4) in the Enclose mode leads to a condition of object closure. The main difficulties come from the assumption related to non-essential robots and the book-keeping associated with numbering the robots so that the robots are numbered sequentially in the counterclockwise direction. It is worth noticing that simple potential field controllers, like that presented in Song and Kumar (2002) have the attractive property of symmetrically distributing the robots around the object and producing initial conditions that are favorable for the Enclose mode.

It is also natural to ask if the kinematic model can be extended to non-holonomic robots. For non-holonomic robots, u_i, which is a two-dimensional vector, can be used as a set-point for controllers that take in account the non-holonomic constraints. An example of such an approach is shown in Esposito and Kumar (2002). This is a direction of future research.

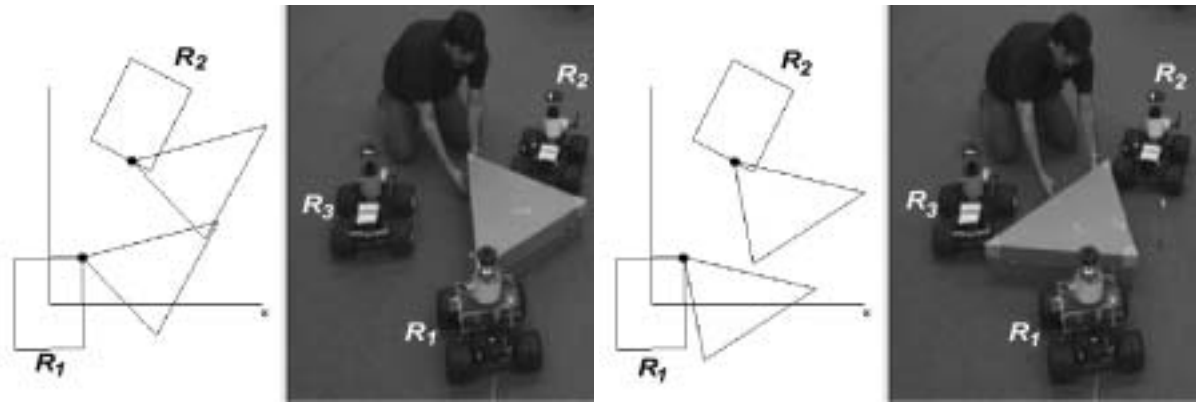

Fig. 10. Three robots caging a triangular object (see also Extension 1). R_1's computations of $\mathcal{C}_{obj\_1}$ and $\mathcal{C}_{obj\_2}$ for the imaginary point robots located at the closest pair of points are shown. The overlap (left) indicates the object is constrained for this specific orientation, and the lack of overlap (right) shows that object closure is not maintained for this slice of the configuration space.

5. Computational Complexity

Considering that $\mathcal{I}_i$ (or $\mathcal{J}_i$ when rotations are considered) can be computed off-line, the decentralized control algorithm requires on-line computation for: (a) determining the object orientation θ and the two neighbors' positions; (b) computing $\mathcal{I}_i(\theta)$ by rotating $\mathcal{I}_i$; (c) translating $\mathcal{I}_i(\theta)$ to q_{i-1} and q_{i+1} and computing $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$ respectively; (d) verifying if there are active constraints; (e) computing the control signals according to eqs. (4), (5), or (6).

The estimation of θ, q_{i-1}, and q_{i+1} is not addressed here but it is important to mention that, in the case of polygonal robots, an $O(l)$ algorithm (Ponamgi, Manocha, and Lin 1997) needs to be used to determine the closest pair of points between the robots. Details of experimental implementation are included in Pereira et al. (2002). Since $\mathcal{I}_i$ ($\mathcal{J}_i$) is defined by up to $2m$ functions, each robot needs to compute up to $2m$ rotations and $4m$ translations in order to compute $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$ ($\mathcal{J}_{i-1}$ and $\mathcal{J}_{i+1}$). The determination of the active constraints, if there are any, can be done by evaluating the $4m$ inequalities that define Γ_i. Observe, however, that Γ_i do not need to be computed explicitly since the cost of computing this region is higher than evaluating all constraints for $\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$. The computation of the control laws can be done in constant time. Therefore, the cost of the algorithm is $O(m + l)$, and is independent of the number of robots in the group.

6. Experiments

Our mobile robots are car-like platforms equipped with omnidirectional cameras as their only sensor. Although we have performed experiments with teams of up to five mobile robots, we report here experiments with three and four robots. The communication among the robots relies on IEEE 802.11b networking. To facilitate the visual processing, each team member and the goal position are marked with different colors. Because each robot has only one camera we use communication between robots and cooperative sensing for (a) localization with respect to each other and (b) estimating the pose of the object (Pereira et al. 2002). The communication is only used for multi-eyed stereo algorithms and not for control or decision making. Ground truth information is obtained from a calibrated overhead camera.

Figure 10 illustrates the test for object closure performed by Robot 1 (R_1). R_1 estimates the position of its neighbor R_2, as well as the orientation of the object. It then computes $\mathcal{C}_{obj\_1}$ and $\mathcal{C}_{obj\_2}$ based on its estimate of the pair of closest points, one on R_1 and one on R_2. The snapshot on the left shows overlap and therefore a positive test for object closure. The snapshot on the right shows a situation in which the object can actually escape. A similar test (not shown in the picture) needs to be performed with robot R_3. Extension 1 shows a video with the test. It shows that this test can be performed in real time with a Pentium III 850 MHz.

In Figures 11 and 12, we show experimental results with three robots, R_1, R_2, and R_3, transporting a triangular box toward a goal position. Data collected from the overhead camera are shown for typical experimental runs. Figure 11 (Extension 2) shows a situation where robots R_2 and R_3 start in the Enclose mode but then change their control behaviors in order to perform the task. In Figure 12 (Extension 3), the actual $\mathcal{C}_{OBJ\_i}$ for the rectangular robot geometry is overlaid on the experimental data. Note, however, that the robots do not use $\mathcal{C}_{OBJ\_i}$ for maintaining object closure, but instead they work with the virtual point robot model explained in Section 3.4. The object can be seen to be caged in each of the three snapshots shown. A close-up of the robots during the task is shown

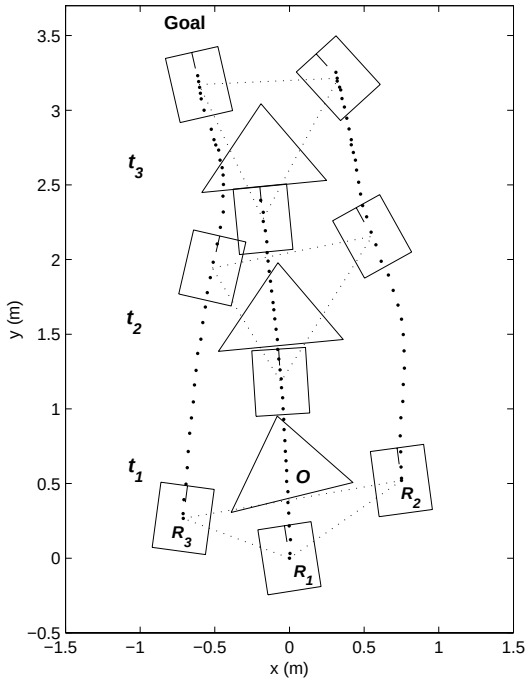

Fig. 11. Object transportation: t_1, R_2 and R_3 are in the Enclose mode (see Figure 8) trying to achieve object closure; t_2, object closure constraints are satisfied, R_2 and R_3 are in the MaintainClosure mode; t_3, the robots are in the GoToGoal mode. R_1 is in the GoToGoal mode in all three snapshots (see also Extension 2).

in Extension 4. In this movie it is clear that the robots switch between the modes of the controller in order to maintain the object closure condition.

Extension 5 shows an experimental trial where four robots are caging a holonomic robotic platform (Nomad XR4000). This is an extension of the manipulation problem where the enclosed object (a robot) is actively controlled and thus not passive. However, for all practical purposes, the caged robot is an object with unmodeled dynamics. As mentioned in Section 3.5, the circular shape of this robot reduces the caging test to a simple comparison between the diameter of the Nomad and the distance between the robots' closest pair of points. In this experiment, the Nomad is running a simple infrared-based obstacle avoidance controller that treats the surrounding robots as obstacles. Thus, it can be viewed as an adversary that is trying to escape by violating the object closure condition. The extension shows that robots can successfully maintain object closure without any knowledge of the adversary's strategy or its dynamics.

7. Concluding Remarks

We have presented algorithms for manipulating objects with multiple mobile robots combining the paradigms of pushing and caging. We have defined the concept of object closure, a condition that ensures the objects are caged during manipulation. The main contributions of the paper are: (a) an algorithm that enables each robot to independently verify the condition of object closure; (b) a decentralized control algorithm that enables each robot to move while maintaining object closure.

There are two main advantages of our approach. The decentralized algorithms mainly rely on the robots' ability to estimate the positions of their neighbors. Because robots are easily instrumented (in our case, this is done by tagging them with colored collars), this is relative easy even in an unstructured environment. Therefore, our methodology is potentially scalable for larger groups of robots operating in unstructured environments. Secondly, our algorithms do not rely on exact estimates of the position and orientation of the manipulated object. Therefore, they are robust to errors in pose estimation.

The main limitations of the algorithms used here include: (a) the assumption of convex shapes; (b) the overapproximation that is involved in verifying object closure when rotations are present; (c) the use of the virtual point robots which yields conservative, sufficient conditions for maintaining object closure. All these assumptions yield conservative results with associated degradation in performance. For example, ensuring object closure with concave objects is often simpler than is

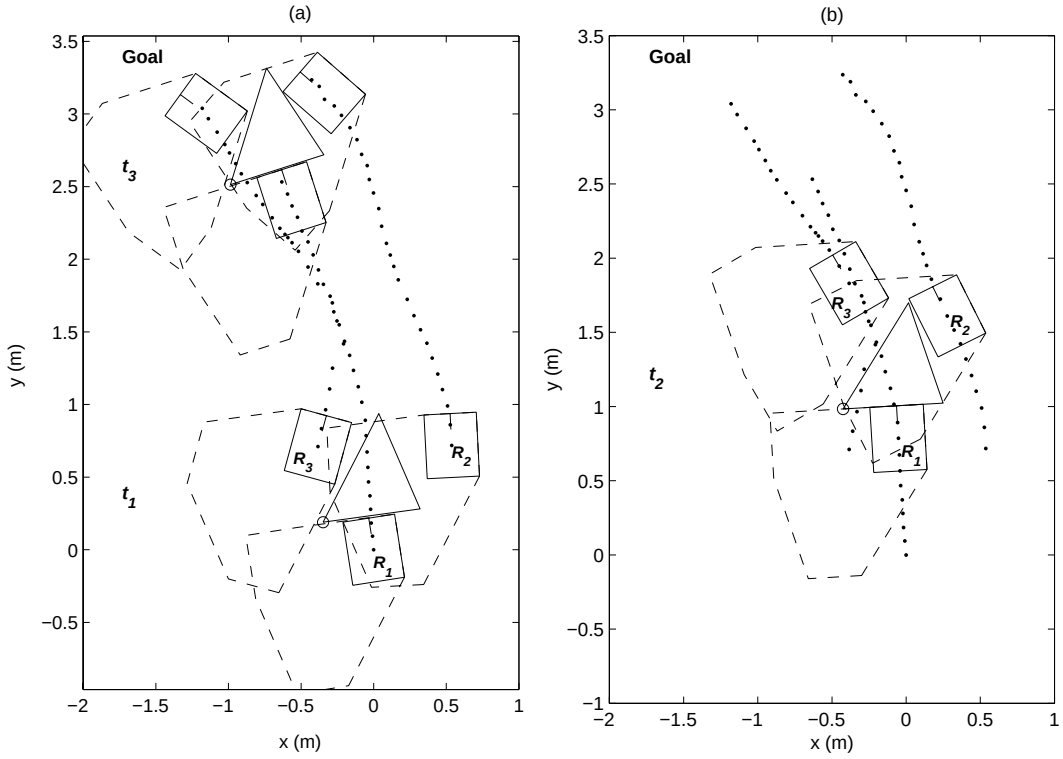

Fig. 12. The actual $\mathcal{C}_{OBJ\_i}$ (dashed polygons) for each robot. The origin of the object (∘) is always inside $\mathcal{C}_{cls}$ (the compact set delimited by the three $\mathcal{C}_{OBJ\_i}$) indicating an object closure condition. (a) Initial and final configurations; (b) an intermediate configuration (see also Extension 3).

the case for convex objects. However, these assumptions and overapproximations enable real-time performance and decentralized decision making with guarantees, and are important from a practical standpoint.

There are several important directions for future work. First, it is necessary to explicitly model the non-holonomic behavior of the robots. The work in Esposito and Kumar (2002) provides a starting point in this direction. Secondly, we do not specifically consider algorithms for acquiring the object and establishing object closure (the Enclose mode) here. Wang and Kumar (2002) and Song and Kumar (2002) provide some approaches to this, with guarantees for small teams of three or four robots. There are challenges in designing decentralized policies that scale up to large numbers of robots. One of the key steps here is to remove the assumption related to non-essential robots. Finally, we do not address the precise positioning and orienting of the object. By varying the threshold δ_2, we can obtain tighter tolerances on the object position relative to the robots. However, it is also essential to plan trajectories for the individual robots, instead of simply prescribing a common feedforward control signal u_T. The work in Sudsang and Ponce (2000) provides a starting point in this direction.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Test for object closure based on the object's orientation computed in real time.
2	Video	Three robots switching modes to achieve object closure.
3	Video	Three robots caging a triangular object.
4	Video	Close-up of the robots in the manipulation task.
5	Video	Four robots "shepherding" a circular holonomic robot. The circular robot is running a simple obstacle avoidance algorithm.

Acknowledgments

We gratefully acknowledge the support of CNPq-Brazil grants 200765/01-9 and 300212/99-2, AFOSR grant no. F49620-01-1-0382, NSF grant no. CDS-97-03220, NSF grant no. IIS-0083420, and DARPA ITO MARS Program, grant no. 130-1303-4-534328-xxxx-2000-0000.

References

Davidson, C., and Blake, A. 1998. Caging planar objects with a three-finger one-parameter gripper. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, pp. 2722–2727.

Esposito, J. M., and Kumar, V. 2002. A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 1691–1696.

Filippov, A. 1988. *Differential Equations With Discontinuous Right-Hand Sides*, Kluwer Academic, Dordrecht.

Kosuge, K., and Oosumi, T. 1996. Decentralized control of multiple robots handling an object. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Osaka, Japan, November 4–8, pp. 318–323.

Latombe, J.-C. 1991. *Robot Motion Planning*, Kluwer Academic, Dordrecht.

Liberzon, D., and Morse, A.S. 1999. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine* 19(5):59–70.

Lozano-Pérez, T. 1983. Spatial planning: a configuration space approach. *IEEE Transactions on Computers* 32(2):108–120.

Lynch, K. M. 1996. Stable pushing: mechanics, controllability, and planning. *International Journal of Robotics Research* 15(6):533–556.

Mataric, M., Nilsson, M., and Simsarian, K. 1995. Cooperative multi-robot box-pushing. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pittsburgh, PA, pp. 556–561.

Ota, J., Miyata, N., and Arai, T. 1995. Transferring and regrasping a large object by cooperation of multiple mobile robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Pittsburgh, PA, pp. 543–548.

Pereira, G. A. S., Kumar, V., Spletzer, J., Taylor, C. J., and Campos, M. F. M. 2002. Cooperative transport of planar objects by multiple mobile robots using object closure. *Experimental Robotics VIII*, B. Siciliano and P. Dario, editors, Springer-Verlag, Berlin, pp. 275–284.

Ponamgi, M. K., Manocha, D., and Lin, M. C. 1997. Incremental algorithms for collision detection between polygonal models. *IEEE Transactions on Visualization and Computer Graphics* 3(1):51–64.

Rimon, E., and Blake, A. 1996. Caging 2D bodies by one parameter, two-fingered gripping systems. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, MN, pp. 1458–1464.

Rimon, E., and Burdick, J. 1995. A configuration space analysis of bodies in contact. Part I: first-order mobility. *Mechanism and Machine Theory* 30(6):897–912.

Rus, D. 1997. Coordinated manipulation of objects in a plane. *Algorithmica* 19(1/2):129–147.

Song, P., and Kumar, V. 2002. A potential field-based approach to multi-robot manipulation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 1217–1222.

Sudsang, A., and Ponce, J. 1998. On grasping and manipulating polygonal objects with disc-shaped robots in the plane. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, pp. 2740–2746.

Sudsang, A., and Ponce, J. 2000. A new approach to motion planning for disc-shaped robots manipulating a polygonal object in the plane. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 24–28, pp. 1068–1075.

Sudsang, A., Ponce, J., Hyman, M., and Kriegman, D. J. 1999. On manipulating polygonal objects three 2-DoF robots in the plane. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, pp. 2227–2234.

Sugar, T., and Kumar, V. 1998. Decentralized control of cooperating mobile manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, pp. 2916–2921.

Wang, Z., and Kumar, V. 2002. Object closure and manipulation by multiple cooperative mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 394–399.