2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Autonomous Continuum Grasping

Jinglin Li, Zhou Teng, and Jing Xiao
Department of Computer Science
University of North Carolina at Charlotte

Apoorva Kapadia, Alan Bartow, and Ian Walker
Department of Electrical & Computer Engineering
Clemson University

***Abstract*— A continuum manipulator, such as a multi-section trunk/tentacle robot, is promising for deft manipulation of a wide range of objects of different shapes and sizes. Given an object, a continuum manipulator tries to grasp it by wrapping tightly around it. Autonomous grasping requires real-time determination of whether an object can be grasped after it is identified, and if so, the feasible whole-arm wrapping around configurations of the robot to grasp it, which we call *grasping configurations*, as well as the path leading to a grasping configuration. In this paper, we describe the process for autonomous grasping from object detection to executing the grasping motion and achieving force-closure grasps, with a focus on a general analysis of all possible types of planar grasping configurations of a three-section continuum manipulator. We further provide conditions for existence of solutions and describe how to find a valid grasping configuration and the associated path automatically if one exists. Experimental results with the OctArm manipulator validate our approach, and shows that the entire process to determine an autonomous grasping operation, which includes automatic detection of the target object and determination of a grasping configuration and a path to the grasping configuration that avoids obstacles, can take just a small fraction of a second. Once a grasping configuration is reached, the manipulator can lift the object stably, i.e., a force-closure grasp can be achieved.**

I. Introduction

Continuum manipulators [10] are usually defined to be those featuring continuous backbone structures, inspired by invertebrate structures found in nature, such as octopus arms [6] and elephant trunks [1]. The OctArm manipulator [2] is such a continuum manipulator. Unlike a conventional manipulator in the form of an articulated arm with a gripper, there is no divide between the "arm" and "hand/gripper" for a continuum manipulator to manipulate objects, and thus, finding a grasping configuration, where a continuum robot wraps around an object, cannot be decomposed into two separate problems of finding an end-effector location and then solving the inverse kinematics, which is well studied. Autonomous grasping with a continuum manipulator is less studied. Recently some methods have been introduced for finding grasping configurations [3], [4], [7], [12] for a continuum manipulator, given the target object, but the whole process of autonomous grasping from object identification to force-closure grasps have not been conducted experimentally.

In this paper, we describe how autonomous grasping can be achieved using an OctArm manipulator on a table top to perform 2.5 dimensional movements. Target objects also sit on the table top and are automatically detected. Based on the detection results, all valid planar grasping configurations and the associated paths are automatically generated for the robot to execute. Finally the robot lifts the object after grasping to show force-closure grasps. Fig. 1 shows the experimental set up with a table top OctArm, overseen by an overhead Microsoft Kinect. In section II we first introduce the manipulator model, the target object model as the result of automatic detection, and the corresponding planar grasping configurations. In sections III to V, we classify planar grasping configurations for the table-top OctArm and describe conditions of existence of such grasping configurations, as well as how to find the associated paths for grasping. In section VI, we present the experimental results of autonomous grasping, and in section VII, we conclude the paper and discuss future research.

Fig. 1. Experimental set up with a table top OctArm, overseen by an overhead Microsoft Kinect

II. Manipulator, Objects, and Grasping Configurations

We introduce the manipulator structure, parameters, target object model based on automatic detection, and characteristics of "grasping" configurations in the following.

A. OctArm manipulator

The OctArm robot is a concatenation of three constant-curvature sections. A *configuration* of the OctArm can be expressed by the controllable variables as $[\kappa_1, s_1, \phi_1, \kappa_2, s_2, \phi_2, \kappa_3, s_3, \phi_3]^T$, which are degrees of freedom that can be *directly changed* by the OctArm actuators [2]; κ_i and s_i, i=1, 2, 3, are the (constant) curvature and length of section i respectively, and ϕ_i is the rotation angle from the plane of section i-1 to that of section i. Note that κ_i can be either negative, zero, or positive.

978-1-4673-6358-7/13/$31.00 ©2013 IEEE

With the above controllable variables, an OctArm manipulator can change shape by bending, extending, contracting, and twisting each (circular) section. For each section i, its frame is set up as shown in Figure 2(a). Denote the two end points of section i as the *base* point p_{i-1} and the *tip* point p_i. The base of section i is the tip of section $i-1$. The section i's frame is formed at p_{i-1} with the z_i axis tangent to the section i's curve at p_{i-1}. Note that the circle center of section i, p_{ic}, always lies on the x_i axis, with $\mp 1/\kappa_i$ being the x coordinate in the i-th frame. Note also that p_{ic} lies on the positive x_i axis if $\kappa_i < 0$ and on the negative x_i axis if $\kappa_i > 0$. When $\kappa_i = 0$, section i is a straight-line segment starting from the origin p_{i-1} and along the z_i axis. Given the position of the base point p_{i-1}, and the κ_i, s_i, and ϕ_i values, the position of the tip point p_i of section i can be computed [2].

The base of the robot is set at p_0 with z_0 axis tangent to section 1's curve, see Figure 2(b). Note also that two adjacent sections i-1 and i are connected *tangentially* at the connection point p_{i-1}, i.e., the two sections share the same tangent at p_{i-1}.

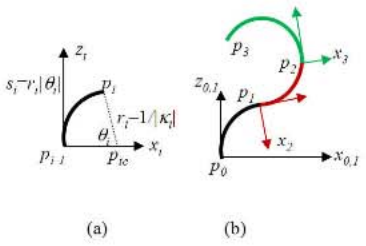


Fig. 2. An OctArm manipulator: (a) one section and its frame; (b) a planar arm configuration

Note that because of its mechanical structure, section 1 of the OctArm can bend either along the $+z_0$ axis or the $-z_0$ axis but *not* both. See Figure 3. If ϕ_i=0 for all the sections, i.e., there is no twisting from one section to the next, the OctArm is planar with all sections on the same plane – see Figure 2(b).

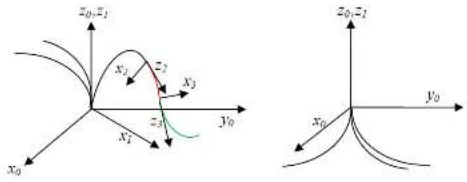


Fig. 3. Section 1 of an OctArm manipulator (in black): (a) bending along $+z_0$ axis, where sections 2 and 3 are also shown in one case; (b) bending along $-z_0$ axis

B. Object model and grasping configurations

With an overhead camera (i.e., a Microsoft Kinect), an object on a table top (along the x_0z_0-plane of the manipulator) can be detected as a polygonal mesh region on the image. Fig. 4 shows an example. A bounding circle with center c and radius r can be obtained to indicate the location and size of the object. The bounding circle is roughly the minimum bounding circle, which can be found in linear time [5]. To take advantage of the passive compliance of the OctArm for force-closure grasps (see, for example, [9] for a concise survey), the bounding circle can be made smaller by slightly cutting into the object region. However, nonprehensile manipulation tasks, such as dragging an object on the table, may not require form or force closure grasps.

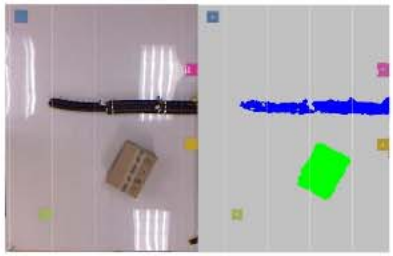
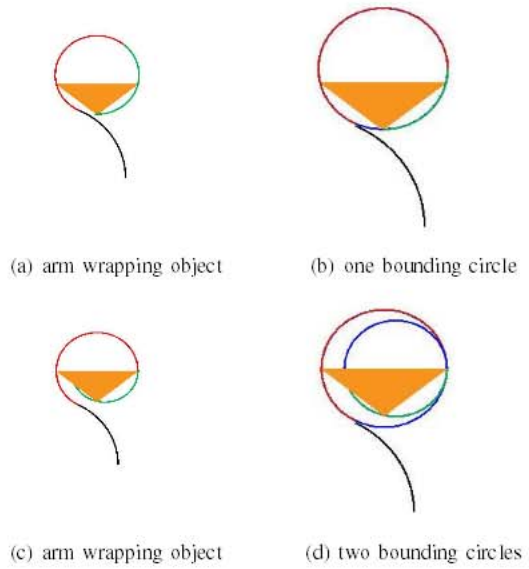


Fig. 4. An example object in the manipulator workspace (left) as detected by the camera (right)

Either section 3 of the OctArm or both section 2 and section 3 of the OctArm can be used to grasp an object in a planar configuration (by wrapping around the bounding circle of the object) with arm sections all on the x_0z_0-plane of the manipulator.

If both section 2 and section 3 of the OcrArm are used to "grasp" the object, either the two sections wrap around the bounding circle of an object polygon together, or two circles, one bounding and the other partially bounding, can be used to provide a tighter wrap. Figure 5 shows an example, where the OctArm, with its three sections colored in black, red, and green respectively, wraps around a triangular object. The triangular object is wrapped around based on either one bounding circle (i.e., its circumcircle with three contact points)– see Figure 5(a) and (b), or its minimum bounding circle and a partial bounding circle that are internally tangent (which result in four contact points between the arm sections and the object) – see Figure 5(c) and (d). The case with two bounding (and partially bounding) circles creates a tighter wrap with smaller circles and more contact points. In the Appendix, we show how to find such two circles to produce a tighter grasp for a given (sensed) polygonal object region in general.

For convenience, we name the bounding and partially bounding circles of an object polygon found above the *object circles* in the rest of the paper. We now define a *grasping configuration* as an OctArm configuration such that either section 3 or both sections 2 and 3 of the OctArm wrap around the object circle(s) in a planar configuration.



(a) arm wrapping object (b) one bounding circle

(c) arm wrapping object (d) two bounding circles

Fig. 5. A triangular object wrapped around by the OctArm with its three sections colored in black, red, and green respectively. (a) and (b): based on the circumcircle; (c) and (d) based on two tangent circles bounding (and partially bounding) the object

III. Classification of Grasping Configurations

Since each OctArm section is part of a circle, we denote the circle corresponding to section i as $\mathcal{C}_i$. In the rest of this paper, we draw $\mathcal{C}_1$ black, $\mathcal{C}_2$ red, and $\mathcal{C}_3$ green.

Since the sections are connected tangentially, the following condition is necessary for the existence of a grasping configuration for a planar OctArm, given either one object circle cir or two (internally tangent) object circles cir_1 and cir_2:

- $\mathcal{C}_1$ is tangent to $\mathcal{C}_2$, and one of the following holds:
  - $\mathcal{C}_2$ is tangent to $\mathcal{C}_3$, and $\mathcal{C}_3$ coincides with cir (i.e., section 3 wraps around the object);
  - $\mathcal{C}_2$ and $\mathcal{C}_3$ both coincide with cir (i.e., both sections 2 and 3 wrap around the object);
  - $\mathcal{C}_2$ and $\mathcal{C}_3$ coincide with either cir_1 and cir_2 respectively or cir_2 and cir_1 respectively.

We further classify the relations among section circles and object circles into ten cases as shown in Table I, satisfying the above conditions.

In the next two sections, we describe how to find grasping configurations in the ten cases in Table I in two steps:

1: find *circle solutions*, i.e., solve for OctArm section circles for each case;

2: find feasible grasping configurations, i.e., for each circle solution, check if the values of planar OctArm variables κ_i and s_i satisfy the respective value ranges of the OctArm.

IV. Circle Solutions

We can further classify the ten cases in Table I into cases with infinite circle solutions, which are cases (1)–(4), and cases with finite circle solutions, which are cases (5)–(10). We now describe how to find their solutions in turn.

TABLE I

Ten cases of all possible relations among circles of arm sections and the object

Case	#Obj. Cir.	Circle Relations
(1)	1	$\mathcal{C}_2$ externally tangent to both $\mathcal{C}_1$ and $\mathcal{C}_3$ $\mathcal{C}_3$ coincides cir
(2)	1	$\mathcal{C}_2$ externally tangent to $\mathcal{C}_1$ $\mathcal{C}_2$ contains and internally tangent to $\mathcal{C}_3$ $\mathcal{C}_3$ coincides cir
(3)	1	$\mathcal{C}_2$ internally tangent to $\mathcal{C}_1$ $\mathcal{C}_3$ coincides cir and externally tangent to $\mathcal{C}_2$
(4)	1	$\mathcal{C}_2$ internally tangent to both $\mathcal{C}_1$ and $\mathcal{C}_3$ $\mathcal{C}_3$ is inside $\mathcal{C}_2$ and coincides cir
(5)	1	$\mathcal{C}_1$ externally tangent to $\mathcal{C}_2$ $\mathcal{C}_2$ and $\mathcal{C}_3$ coincides cir
(6)	1	$\mathcal{C}_1$ internally tangent to and contains $\mathcal{C}_2$ $\mathcal{C}_2$ and $\mathcal{C}_3$ coincides cir
(7)	2	$\mathcal{C}_1$ externally tangent to $\mathcal{C}_2$ $\mathcal{C}_2$, $\mathcal{C}_3$ coincide cir_1, cir_2 respectively
(8)	2	$\mathcal{C}_1$ externally tangent to $\mathcal{C}_2$ $\mathcal{C}_2$,$\mathcal{C}_3$ coincide cir_2, cir_1 respectively
(9)	2	$\mathcal{C}_1$ internally tangent to and contains $\mathcal{C}_2$ $\mathcal{C}_2$, $\mathcal{C}_3$ coincide cir_1, cir_2 respectively
(10)	2	$\mathcal{C}_1$ internally tangent to and contains $\mathcal{C}_2$ $\mathcal{C}_2$, $\mathcal{C}_3$ coincide cir_2, cir_1 respectively

A. Cases of infinite circle solutions

Given object circle cir with center (x_c, z_c) and radius r, there are at most two possible solutions in each of cases (1)–(4) for a given κ_1 and κ_2, if they satisfy:

$$\frac{1}{|\kappa_1|} + \frac{2}{|\kappa_2|} + r \geq \sqrt{(x_c + \frac{1}{\kappa_1})^2 + z_c{}^2} \qquad (1)$$

Condition 1 describes the requirement that the total length of the OctArm is enough for section 3 to (partially) wrap around cir.

Figure 6 shows two example solutions for each case for $\kappa_1 < 0$, corresponding to the center of $\mathcal{C}_1$ on the positive x_1 axis. Similarly, for a given $\kappa_1 > 0$ (corresponding to the center of $\mathcal{C}_1$ on the negative x_1 axis), there are two possible solutions, and Figure 7 shows the solutions for case (1). Note that in both Figure 6 and Figure 7, $\mathcal{C}_3$ (or cir) is the same green one at the same location.

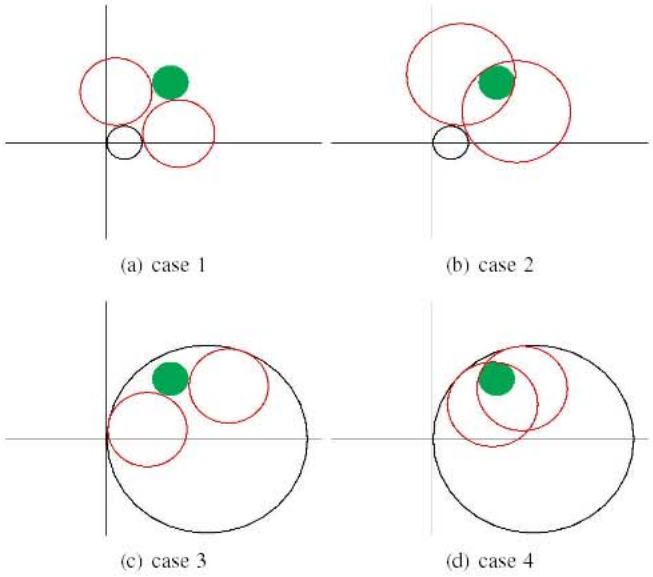
Now we describe, for each of cases (1)–(4), given κ_1 and κ_2 satisfying equation (1), how to find the centers of $\mathcal{C}_2$ corresponding to the two solutions respectively. Let $r_1 = 1/|\kappa_1|$ and $r_2 = 1/|\kappa_2|$. Let

$$d_{12} = \begin{cases} r_1 + r_2 & \text{cases (1) and (2)} \\ r_1 - r_2 & \text{cases (3) and (4)} \end{cases} \qquad (2)$$

$$d_{23} = \begin{cases} r_2 + r & \text{cases (1) and (3)} \\ r_2 - r & \text{cases (2) and (4)} \end{cases} \qquad (3)$$

$$d_{13} = \sqrt{(x_c + \frac{1}{\kappa_1})^2 + z_c{}^2} \qquad (4)$$

d_{12}, d_{23}, and d_{13} are lengths of the three sides of the triangle formed by the centers of $\mathcal{C}_1$, $\mathcal{C}_2$, and cir (which is also $\mathcal{C}_3$).



(a) case 1 (b) case 2

(c) case 3 (d) case 4

Fig. 6. Two possible types of circle solutions for cases (1)–(4), for $\kappa_1 < 0$

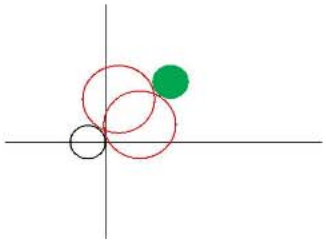


Fig. 7. Two possible types of circle solutions for case (1), for $\kappa_1 > 0$

Figure 8 shows an example, where p_{ic} is the center of $\mathcal{C}_i$. Let R be the radius of the circumcircle of the triangle, and let:

$$D_1 = d_{12} + d_{23} + d_{13}$$
$$D_2 = d_{23} + d_{13} - d_{12}$$
$$D_3 = d_{13} + d_{12} - d_{23}$$
$$D_4 = d_{12} + d_{23} - d_{13}$$

then R satisfies:

$$R = \frac{d_{12}d_{23}d_{13}}{\sqrt{D_1D_2D_3D_4}} \quad (5)$$

Let α be the counterclockwise angle from the x_1 axis to the vector $\mathbf{p}_c - \mathbf{p}_{1c}$ (see Figure 8). Thus,

$$\alpha = atan2(z_c, x_c + \frac{1}{\kappa_1}). \quad (6)$$

Based on the Laws of Cosine and Sine, the angle β of the triangle, as shown in Figure 8, can be uniquely determined, satisfying

$$cos\beta = \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}d_{13}} \quad (7)$$

and

$$sin\beta = \frac{d_{23}}{2R} \quad (8)$$

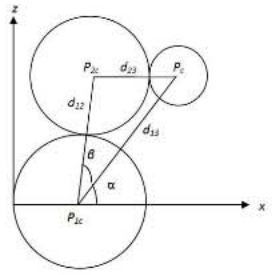
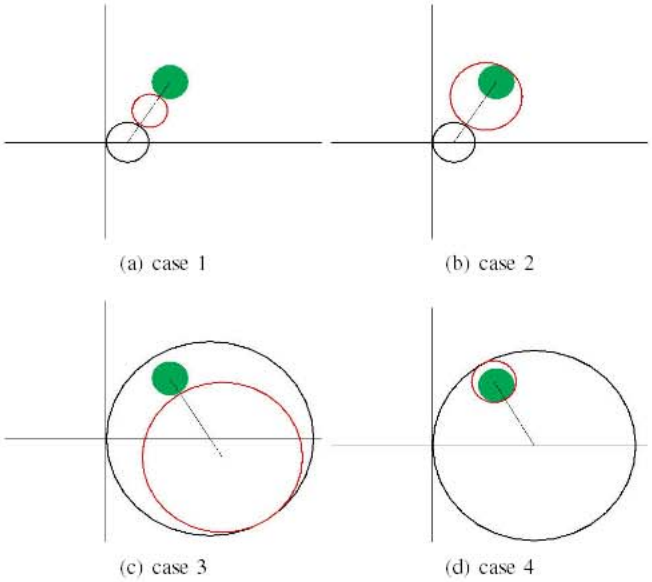


Fig. 8. The triangle connecting three circle centers

With α and β known, the two solutions of p_{2c} (the center of $\mathcal{C}_2$) can be obtained:

$$x_{2c} = -\frac{1}{\kappa_1} + d_{12}cos(\alpha \pm \beta), z_{2c} = d_{12}sin(\alpha \pm \beta). \quad (9)$$

When $\beta = 0$, i.e., the three circle centers are on the same line, there is a unique solution for κ_2 and p_{2c} in each of cases (1)–(4). Figure 9 illustrates those solutions for each case, again for the same cir as in Figures 6 and 7.



(a) case 1 (b) case 2

(c) case 3 (d) case 4

Fig. 9. Circle solutions where the three centers are on the same line

Since the planar OctArm's workspace is mostly the 1st and 2nd quadrants of its base xz frame, we assume cir can only be in the 1st and 2nd quadrants. If cir crosses the positive z axis, then case (4) is not possible so that its corresponding solutions should be ruled out.

Although there are at most a pair of possible solutions for specific κ_1 and κ_2 satisfying equation (1) for a given cir, since κ_1 and κ_2 are variables for the OctArm robot and can change values continuously in their respective intervals: $\kappa_i \in [\kappa_{i,-max}, \kappa_{i,max}]$, which includes $\kappa_i = 0$, i=1,2, there can be infinite pairs of solutions for each of the cases (1)–(4).

We now classify the possible infinite circle solutions into 8 types, determined by the two types of solutions for each

of the four cases (1)–(4). The second part of the attached movie clip shows possible circle solutions via animation.

There is one additional possible circle solution for each case, when the three circle centers are on the same line (Figure 9).

Note that for each case of Figure 6, because the section 1 of the OctArm can only bend along one direction of z_0 axis (Figure 3), each circle solution corresponds to only one possible configuration of section 1 and section 2 of the OctArm (while section 3 can have different lengths, i.e., s_3 values). Note also that among the two possible solutions in each case of Figure 6, the solution on the right requires longer lengths for section 1 and section 2.

B. Cases of finite solutions

Let us first consider cases (5) and (6). If cir is inside the 1st quadrant of the robot base x_0z_0 frame, in either case (5) and case (6), the solution is unique as derived below. Since $\mathcal{C}_1$ is tangent to cir, the following holds:

$$(r_1 - x_c)^2 + z_c^2 = (r_1 \pm r)^2 \tag{10}$$

which yields

$$r_1 = \frac{x_c^2 + z_c^2 - r^2}{2(x_c \pm r)} \tag{11}$$

where "+" is for case (5) and "−" is for case (6). Thus $|\kappa_1| = 1/r_1$ is solved. Because cir is in the first quadrant, $\mathcal{C}_1$ has to be in the first quadrant, and thus its center is on the positive x axis with coordinates $(r_1, 0)$, and $\kappa_1 = -1/r_1$.

If cir is inside the 2nd quadrant of the robot base x_0z_0 frame, i.e., $x_c < 0$, in either case (5) and case (6), the solution is also unique, and the derivation can be done similarly, based on the following equation:

$$(-r_1 - x_c)^2 + z_c^2 = (r_1 \pm r)^2 \tag{12}$$

The result is:

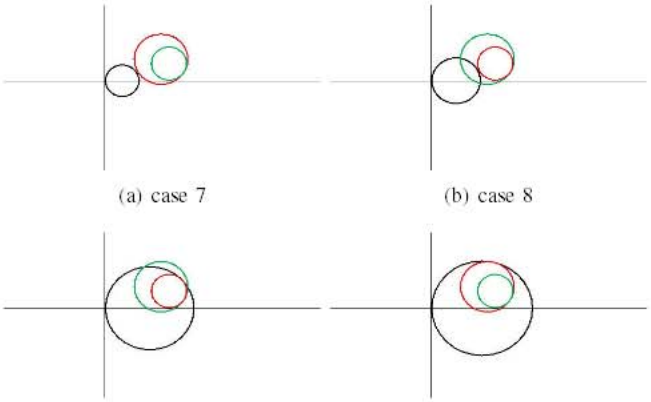
$$r_1 = \frac{x_c^2 + z_c^2 - r^2}{2(-x_c \pm r)} \tag{13}$$

where "+" is for case (5) and "−" is for case (6). Note that $r_1 > 0$, since $x_c < 0$. $\mathcal{C}_1$ has to be in the 2nd quadrant, its center is on the negative x axis with coordinates $(-r_1, 0)$, and $\kappa_1 = 1/r_1$.

If cir crosses the z_0 axis, i.e., in two quadrants, then case (6) is not possible. There are two solutions for case (5), corresponding to the center of $\mathcal{C}_1$ on the positive and negative x axis respectively. The two solutions are expressed in equation (11) and equation (13) with the "+" sign.

For cases (7)–(10), the solution(s) of $\mathcal{C}_1$ can be obtained similarly as those for case (5) and case (6) above, where the radius r_{o1} of cir_1 will be used in equation (11) and equation (13), with "+" for case (7) and "−" for case (9), and the radius r_{o2} of cir_2 will be used in equation (11) and equation (13), with "+" for case (8) and "−" for case (10).

Figure 10 shows example solutions for cases (7)–(10). Note that even though $\mathcal{C}_1$ cuts through the larger object circle before touching $\mathcal{C}_2$ in cases (8) and (9), they still represent possible cases since an arbitrary object does not fill its bounding circle entirely.



(a) case 7 (b) case 8

(c) case 9 (d) case 10

Fig. 10. Circle solutions with two given object circles

C. Tangent points between circles

In the above, we described how to find the unknown centers of section circles in tangent. The tangent point between any two tangent circles can be easily found, given the two circle centers. Let p_i, $i = 1, 2$, be the tangent point between $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$. p_i is in fact the end point of section i of the OctArm.

For cases (1)–(4) and cases (7)–(10), the position vector of p_i can be obtained as

$$\mathbf{p}_i = \frac{r_i}{r_i \pm r_{i+1}}(\mathbf{p}_{(i+1)c} - \mathbf{p}_{ic}) + \mathbf{p}_{ic}$$

where "+" and "−" signs are for externally and internally tangent circles respectively. Note that r_3, the radius for $\mathcal{C}_3$, is the same as the radius of the object circle that $\mathcal{C}_3$ coincides, and $\mathbf{p}_{3c}$ is the same as the position of the center of the object circle. In cases (7)–(10), r_2 and $\mathbf{p}_{2c}$ are also determined by an object circle.

For cases (5) and (6), there is only one tangent point p_1, with the position vector:

$$\mathbf{p}_1 = (\frac{r_1}{r_1 \pm r})(\mathbf{p}_c - \mathbf{p}_{1c}) + \mathbf{p}_{1c}$$

where "+" and "−" signs are for case (5) and case (6) respectively.

V. Existence of Grasping Configurations and Associated Paths

Since the OctArm parameters κ_i and s_i for each section i have finite ranges of valid values: $[\kappa_{i,-max}, \kappa_{i,max}]$, which includes $\kappa_i = 0$, and $[s_{i,min}, s_{i,max}]$, with $s_{i,min} > 0$, the solutions of section circles described above will result in valid grasping configurations if the corresponding parameter values are within those valid ranges.

As noted above, p_i is the end point of section i of the OctArm. Let θ_i be the angle from the x_i axis to the x_{i+1} axis (see Figure 2). Thus, the length s_i of section i satisfies $s_i = r_i|\theta_i|$. Knowing the position of p_i in the i-th frame, θ_i can be obtained easily via inverse kinematics [8].

If κ_i is within its range $[\kappa_{i,-max}, \kappa_{i,max}]$, the following constraints must be satisfied so that at least one circle

solution in each different case yields a valid grasping configuration:

- The size of object polygon is appropriate:

If object is in the 1st quadrant of the base x_0z_0 frame,

$$0 < \frac{1}{r} \leq \begin{cases} \kappa_{3,max} & \text{cases (2), (3)} \\ |\kappa_{3,-max}| & \text{cases (1), (4)} \\ min(\kappa_{2,max}, \kappa_{3,max}) & \text{case (5)} \\ min(|\kappa_{2,-max}|, |\kappa_{3,-max}|) & \text{case (6)} \end{cases}$$

$$0 < \frac{1}{r_{o1}} \leq \begin{cases} \kappa_{2,max} & \text{case (7)} \\ |\kappa_{2,-max}| & \text{case (9)} \\ \kappa_{3,max} & \text{case (8)} \\ |\kappa_{3,-max}| & \text{case (10)} \end{cases}$$

$$0 < \frac{1}{r_{o2}} \leq \begin{cases} \kappa_{3,max} & \text{case (7)} \\ |\kappa_{3,-max}| & \text{case (9)} \\ \kappa_{2,max} & \text{case (8)} \\ |\kappa_{2,-max}| & \text{case (10)} \end{cases}$$

If object is in the 2nd quadrant of the base x_0z_0 frame,

$$0 < \frac{1}{r} \leq \begin{cases} |\kappa_{3,-max}| & \text{cases (2), (3)} \\ \kappa_{3,max} & \text{cases (1), (4)} \\ min(|\kappa_{2,-max}|, |\kappa_{3,-max}|) & \text{case (5)} \\ min(\kappa_{2,max}, \kappa_{3,max}) & \text{case (6)} \end{cases}$$

$$0 < \frac{1}{r_{o1}} \leq \begin{cases} |\kappa_{2,-max}| & \text{case (7)} \\ \kappa_{2,max} & \text{case (9)} \\ |\kappa_{3,-max}| & \text{case (8)} \\ \kappa_{3,max} & \text{case (10)} \end{cases}$$

$$0 < \frac{1}{r_{o2}} \leq \begin{cases} |\kappa_{3,-max}| & \text{case (7)} \\ \kappa_{3,max} & \text{case (9)} \\ |\kappa_{2,-max}| & \text{case (8)} \\ \kappa_{2,max} & \text{case (10)} \end{cases}$$

If object crosses the base $+z_0$ axis,

$$0 < \frac{1}{r} \leq \begin{cases} max(|\kappa_{3,-max}|, \kappa_{3,max}) & \text{cases (1)–(3)} \\ max(min(|\kappa_{2,-max}|, |\kappa_{3,-max}|), \\ \quad min(\kappa_{2,max}, \kappa_{3,max})) & \text{case (5)} \end{cases}$$

$$0 < \frac{1}{r_{o1}} \leq \begin{cases} max(|\kappa_{2,-max}|, \kappa_{2,max}) & \text{case (7)} \\ max(|\kappa_{3,-max}|, \kappa_{3,max}) & \text{case (8)} \end{cases}$$

$$0 < \frac{1}{r_{o2}} \leq \begin{cases} max(|\kappa_{3,-max}|, \kappa_{3,max}) & \text{cases (7)} \\ max(|\kappa_{2,-max}|, \kappa_{2,max}) & \text{case (8)} \end{cases}$$

The object polygon perimeter has to satisfy the following:

$$a\pi r \leq \begin{cases} s_{3,max} & \text{cases (1)–(4)} \\ s_{2,max} + s_{3,max} & \text{cases (5), (6)} \end{cases}$$

$$a\pi r_{o1} \leq \begin{cases} s_{2,max} & \text{cases (7), (9)} \\ s_{3,max} & \text{cases (8), (10)} \end{cases}$$

$$a\pi r_{o2} \leq \begin{cases} s_{2,max} & \text{cases (8), (10)} \\ s_{3,max} & \text{cases (7), (9)} \end{cases}$$

where $a \in (0, 2]$ is a coefficient determining how much the object circle has to be wrapped. Its value depends on the shape, size, and material characteristics of the target object, as well as on the task of manipulation. For example, the task of pulling an object could require a smaller a than that of picking up the object.

- Object is reachable:

$$s_{1,min} \leq r_1\theta_1 \leq s_{1,max} \quad \text{for all cases} \tag{14}$$

$$s_{2,min} \leq r_2\theta_2 \leq s_{2,max} \quad \text{cases (1)–(4)} \tag{15}$$

Based on the above, all valid grasping configurations can be found automatically given either (a) cir with center position and radius, or (b) cir_1 and cir_2 with their centers and radii.

For each valid grasping configuration, the next task is to find a valid and preferrable optimized path for the manipulator to reach the grasping configuration. This is the task of path planning for the continuum manipulator, and we use the real-time algorithm in [12] to find a near-optimal path for the OctArm to reach the grasping configuration for the target object while avoiding other objects.

VI. Experimental Validations

We have tested our methods for achieving autonomous grasping with the OctArm in the experimental set up (shown in Fig. 1). The OctArm is pneumatically actuated with pressure regulators connected to each part of the inflatable base, middle, and tip sections. The pressure regulator signals are computed based on a MATLAB/Simulink block diagram, which is connected to the arm via a Quanser data acquisition system.

The parameters of the OctArm have the following ranges:

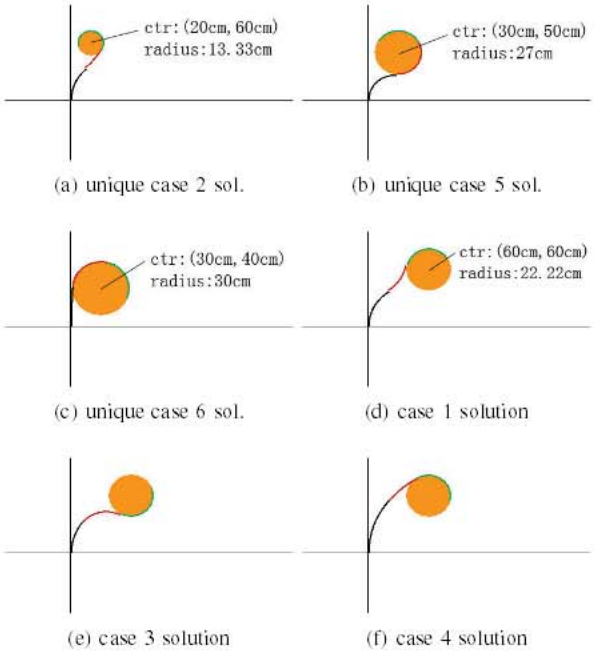
- $\kappa_1 \in [-0.0189, 0.0228]$ (1/cm), $s_1 \in [28, 42]$ (cm),
- $\kappa_2 \in [0.0327, 0.0379]$ (1/cm), $s_2 \in [26.5, 44]$ (cm),
- $\kappa_3 \in [-0.045, 0.0808]$ (1/cm), $s_3 \in [32.5, 53.5]$ (cm).

It should be noted that the OctArm limits for section curvatures and section lengths are dependent on each other and a minimum or maximum curvature cannot be achieved along with minimum or maximum length, and vice versa.

Figure 11 shows the valid grasping configurations found for four objects by our analytical approach and based on the above OctArm parameter ranges: (a)–(c) have only unique OctArm solutions respectively, and the types of solutions are indicated; (d)–(f) show a cir that has three OctArm solutions of different types. Clearly, given a cir, the types and number of valid grasping configurations vary, but it is possible to have any type of grasping configuration among cases (1)–(6).

Similarly, for two object circles cir_1 and cir_2, the types and number of valid grasping configurations vary, but a valid one can be of any of the cases (7)–(10).

Figure 12 shows the autonomous execution of the OctArm to reach a grasping configuration, grasp, and lift an object while avoiding another object. The target object was a box wrapped in a foam so that it has a large size and irregular shape, which was difficult for a conventional gripper to pick up. The objects were first detected automatically based on the RGB-D data obtained from the overhead Kinect camera, next the grasping configuration was automatically generated, and finally the path connecting the initial configuration of the arm



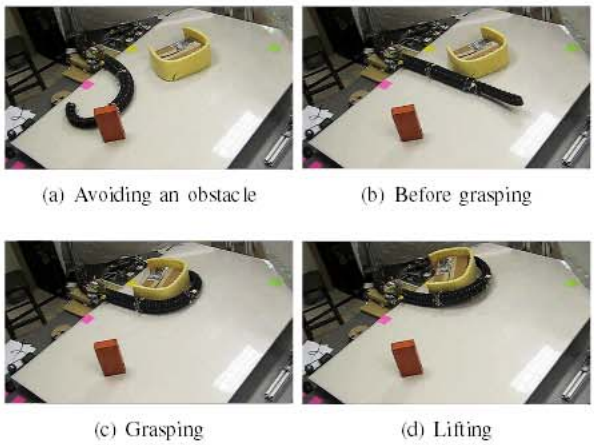
(a) unique case 2 sol. (b) unique case 5 sol.

(c) unique case 6 sol. (d) case 1 solution

(e) case 3 solution (f) case 4 solution

Fig. 11. Grasping configurations: (a)–(c) show the unique solutions for three different object circles respectively, and (d)–(f) show the three solutions for the one object circle specified in (d).

to the grasping configuration was automatically generated, which also avoids the other object. The final lift of the object (see Figure 13) shows that the grasp was a stable and force-closure grasp. Figure 14 shows the OctArm automatically avoiding two objects before getting ready to reach a grasping configuration for one of the objects (i.e., the blue one).



(a) Avoiding an obstacle (b) Before grasping

(c) Grasping (d) Lifting

Fig. 12. Snapshots of the OctArm automatically grasping and lifting an object while avoiding an obstacle

The attached video shows three operation cases. Case (1) is the autonomous grasping and lifting operation of the object in Figure 13, case (2) is the autonomous obstacle avoidance operation as shown in Figure 14, and case (3) is the autonomous operation of grasping and lifting an object while avoiding another object, as shown in Figure 12. The object In the grasping cases, it is interesting to note that the target object moved after the initial contact with the arm. The

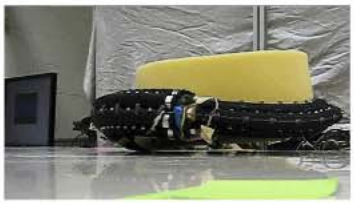
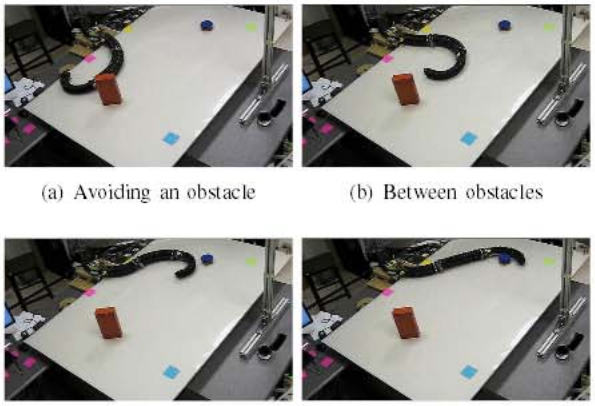


Fig. 13. Result of lifting the object after grasping



(a) Avoiding an obstacle (b) Between obstacles

(c) Avoiding the other obstacle (d) Wrapping

Fig. 14. Snapshots of the OctArm to avoid two objects before getting ready to wrap one object

ability of the manipulator to deal with such moveable objects shows a significant advantage over rigid-link manipulators with grippers, which may not be able to properly grasp an object that is not fully static. It should be noted that the object was lifted about 2 inches off the surface, though this ability varies depending on the shape, size, and weight of the object being grasped.

The time costs for the three cases shown in the video are presented in Table II. Clearly the combined time cost for object detection, grasping configuration generation, and path generation is very small, even with obstalce avoidance in cases (2) and (3), comparing to the execution time of the OctArm, which shows that our algorithms are suitable for real-time operation. The time cost for generating a grasping configuration is mainly the time for generating an object circle from the detected object point cloud, and the time cost to generate a grasping configuration based on the object circle takes 1 ms or less since our presented approach is analytical. The time cost for generating a path depends on the complexity of the environment: in operation case (2), there were two objects to avoid, which resulted in a higher time cost. Note that the time delay was added before lifting an object to ensure a complete grasp before lifting started.

It should be noted that the execution time of the OctArm could be reduced with closed-loop control. However, with the control of the OctArm being open loop, the slower arm speed was able to ensure that inertia would not change the desired pose of the arm with respect to the objects, especially when the arm was in close proximity of an obstacle. Significantly increasing the speed of the motion could lead to situations

TABLE II
TIME COSTS OF THE THREE OPERATIONS SHOWN IN THE ATTACHED VIDEO

Op.	Obj. Detect.	Grasp Gen.	Path Gen.	Execution	Delay
(1)	180 ms	11 ms	5 ms	28 s	4 s
(2)	180 ms	5 ms	25 ms	60 s	n/a
(3)	180 ms	11 ms	20 ms	136 s	4 s

when the inertia of the manipulator would create unintended collisions and deviations from the planned paths and cause failures of the operations.

VII. CONCLUSIONS

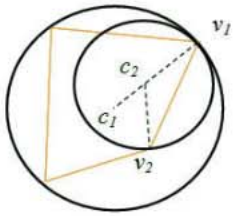
This paper presents the entire process of achieving autonomous grasping of table-top objects by a table-top OctArm manipulator under overhead vision sensing in real time. It provides a complete analysis to characterize all possible planar grasping configurations for the table-top OctArm manipulator with a fixed base for any given object in the workspace. The experimental results confirm the effectiveness of our real-time methods for autonomous grasping. Our next step is to extend experimental testing to non-planar grasping [3] to realize more flexible autonomous object manipulation by a continuum manipulator. Adding closed-loop control to the manipulator is also part of our future work.

APPENDIX

In general, given an object represented as a polygonal mesh, we can find two internally tangent bounding (and partially bounding) circles cir_1 and cir_2 of a cross section of the object, i.e., an object polygon, which are for a tighter grasp/wrap (by the OctArm) of the object, in the following way:

- Find the minimum bounding circle [5], cir_1, with center c_1 and radius r_1.
- Pick a vertex v_1 of the object polygon on cir_1 as the internal tangent point between cir_1 and a smaller circle cir_2 (to be found).
- Pick a vertex v_2 of the object polygon that is not on cir_1, and find the point c_2 on line segment $\overline{v_1c_1}$ that is equidistant to v_1 and v_2; c_2 is then the center of cir_2, and the radius r_2 of cir_2 equals to the length of $\overline{v_1c_2}$.

Figure 15 shows an example object polygon, and cir_1 and cir_2. Clearly, the choice of cir_2 is not generally unique. However, there are only a finite number of possible cir_2's, which can all be easily found by a program, repeating the above steps. For each cir_2 found, there are two possible grasping configurations of the OctArm (that has a fixed base), by having C_2 coincide either cir_1, the larger object circle, or cir_2, the smaller object circle. These two possible configurations can then be checked for force- or form-closure grasps. Figure 16 shows two possible grasping configurations regarding two object circles.



(a)

Fig. 15. An object polygon and a pair of its bounding (and partially bounding) circles for grasping

(a) (b)

Fig. 16. Two possible ways of grasping with respect to two object circles: (a) C_2 and C_3 coincide the large cir_1 and the small cir_2 respectively; (b) C_2 and C_3 coincide the small cir_2 and the large cir_1 respectively.

ACKNOWLEDGMENT

This work is supported by the US National Science Foundation under grants IIS-0904093 and IIS-0904116.

REFERENCES

[1] R. Cieslak and A. Morecki, "Elephant trunk type elastic manipulator a tool for bulk and liquid type materials transportation," *Robotica*, 17:11-16, 1999.
[2] B.A. Jones and I.D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, 22(1):43-55, Feb. 2006.
[3] J. Li and J. Xiao, "Determining Grasping Configurations for a Spatial Continuum Manipulator," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), San Francisco, Sept. 2011.
[4] J. Li and J. Xiao, "Progressive Generation of Force-Closure Grasps for an n-section Continuum Manipulator," to appear *IEEE International Conference on Robotics and Automation* (ICRA), May 2013.
[5] N. Magiddo, "Linear-Time Algorithms for Linear Programming in R^3 and Related Problems," *SIAM Journal on Comp.*, 12(4):759-776, 1983.
[6] W. McMahan, B.A. Jones, I.D. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic manipulators inspired by cephalopod limbs," *Proc. CDEN Design Conf.*, Montreal, Canada, pp. 1-10, July 2004.
[7] W. McMahan, I.D. Walker, "Octopus-Inspired Grasp-Synergies for Continuum Manipulators," *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, Bangkok, Thailand, pp. 945-950, Feb. 2009.
[8] S. Neppalli, M.A. Csencsits, B.A. Jones, and I.D. Walker, "Closed-form Inverse Kinematics for Continuum Manipulators," *Advanced Robotics*, 23:2077-2091, 2009.
[9] J. Ponce and B. Faverjon, "On Computing Three-Finger Force-Closure Grasps of Polygonal Objects," *IEEE Trans. Robot. and Automation*, 11(6):868-880, Dec. 1995.
[10] G. Robinson and J.B.C. Davies, "Continuum Robots - A State of the Art," *ICRA*, pp. 2849-2854, May 1999.
[11] D. Trivedi, C.D. Rahn, W.M. Kier, and I.D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, 5(3):99-117, Sept. 2008.
[12] J. Xiao and R. Vatcha, "Real-time Adaptive Motion Planning for a Continuum Manipulator," *IROS*, Oct. 2010.