

Motion Control for Cylindrical Objects in Microscope's View Using a Projection Method—I: Collision Detection and Detach Control

Dengpeng Xing, Fangfang Liu, Song Liu, and De Xu, *Senior Member, IEEE*

Abstract—This paper investigates collision detection between cylindrical or cylinder-enveloped components when mutual blocking occurs in the view of microscopes and detach control to separate objects if contact occurs. We use a projection method to convert two microscopic views to contours on a projection plane and to detect high-dimensional collision by studying the projection's relationships in low dimension. Eleven cases are totally categorized and 12 parameters are constructed for detection on the basis of relative postures and positions. Furthermore, we present a detach controller to handle with object contact according to collision status, objects' main axis, and the computed contact position. Simulations and experiments are carried out to demonstrate the validity of the proposed method.

Index Terms—Collision detection, detach control, dimensional reduction.

I. INTRODUCTION

REFERRING to detecting the intersection of two or more objects, collision detection is often associated with its use in computer animation, physically based modeling, geometric modeling, robotic path planning, etc. Depending on objects' material rigidity, shape linearity, dimensionality, motion, and so on, different types of collision detection methods can be classified in a wide scope and each has drawn wide attention [1].

In physical simulation, collision detection simulates the object interactions considering contact restrictions and modeling impact dynamics, and fast, accurate, and robust proximity queries are needed in complex physics engines [2]. An

interactive algorithm based on axis-aligned bounding boxes (AABBs) is presented for deformable polygonal objects to detect pairwise primitive-level intersections [3]. To perform continuous collision detection between deforming volume meshes, a culling algorithm is presented in [4] including a continuous separating axis test to eliminate redundant elementary tests. Offline precomputation and runtime queries are used in [5] to approximate the penetration depth between nonconvex models. In video game engine, the limited computing time is divided among several tasks, and its collision detection uses relative primitive algorithms. Binary space partitioning trees provide a viable, efficient, and simple detection method treating characters as points, and according to the separating axis theorem, no collision between two convex shapes is ascertained if there exists an axis along which their projections do not overlap [6]. Some packages are developed for collision detection, e.g., SWIFT [7] and CAMA [8], and simple shapes, e.g., triangle, are commonly used to model the objects, which raises a tradeoff between accuracy and efficiency.

In robotics, collision detection and avoidance is an essential problem for path planning to figure out possible collision status and contact points. Implementation of algorithms and their efficiency in practical settings are the main focus in this domain, and convex polyhedra are selected as objects or complex cases are simplified into convex pieces. In [9], a collision cone approach is used to determine collision between movable objects modeled by general quadric surfaces, and avoidance strategies are generated based on exact collision status. Another method based on sequential convex optimization penalizing collisions with a hinge loss [10] is used to plan robotic collision-free trajectories from naive straight-line initializations that might be in collision. To address collision detection, a sampling-based motion planner is introduced in [11] based on bounding volume hierarchies in workspace-time space. It implements an AABB data structure for trajectory collision queries without constraints on object's shapes or motions. Based on collision detection zones, a distributed controller is designed in [12] for motion tracking of a group of ships. Probabilistic collision checking is employed for sampling-based motion planning in robotics [13], and collision avoidance is also used in motion planning of two robotic fish [14].

Precision assembly uses microscopes and sensitive force sensors as its visual and haptic feedback and manipulates multiple small components with size ranging from millimeters to

Manuscript received June 14, 2016; revised September 8, 2016; December 4, 2016, and December 28, 2016; accepted January 27, 2017. Date of publication March 2, 2017; date of current version June 9, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61305115, Grant 61673382, Grant 61227804, Grant 61421004, and Grant 51405486.

D. Xing, F. Liu, and D. Xu are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China, and also with the School of Computer and Control Engineering, University of the Chinese Academy of Sciences, Beijing 101408, China (e-mail: dengpeng.xing@ia.ac.cn; fangfang.liu@ia.ac.cn; de.xu@ia.ac.cn).

S. Liu is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China, and with the School of Computer and Control Engineering, University of the Chinese Academy of Sciences, Beijing 100049, China, and also with the City University of Hong Kong, Kowloon, Hong Kong (e-mail: sliu@outlook.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIE.2017.2677363

micrometers [15], [16]. Possessing the property of achieving high precision, it has consequently drawn wide attention in recent years, and substantial achievements have been reported in various mechanisms and different kinds of tasks. A slice micropart is assembled into a groupware on an automated mechanism in [17] employing a laser triangulation measurement instrument and microscopic vision to acquire object's attitude. In [18], an automatic assembly platform is proposed to realize the alignment of multiple holes on two objects and 3-D assembly. Inclined precision assembly is investigated in [19] employing a platform with multiple manipulators and three microscopes.

Of the current works, two issues remain open: accurate fast collision detection and its implementation in experiment; and adequate response in the occurrence of collision. Aiming at these, this paper proposes a projection method for collision detection of cylindrical objects, as many complicated convex models can be segmented to multiple cylinders. We project objects onto an appropriate plane and detect 3-D collision status by investigating planar contours, which leads to 11 cases due to the dimensionality reduction and twelve parameters for quick detection. A detach controller is then designed to react properly for collision by generating separation force based on the contact location. We have implemented simulations and experiments for verification and comparisons.

Compared with [20], improvements in this paper include: 1) two more cases are added to complete the classification, and for some complicated cases, the computation efficiency is further enhanced; 2) a detach controller is proposed for contact status, and consequently, contact location is computed if exists; and 3) and extensive simulations and experiments are implemented for verification.

II. PROJECTION APPROACH FOR COLLISION DETECTION

The main idea of this approach is that we only need to consider the collision issue under the plane including one object's end surface. No intersection between the planar contours of the two objects projected onto this plane is the sufficient and necessary condition that the convex objects have no collision in three dimensions.

A. Projection Computation

Let b represent the component held by the upper gripper and object a lies on the lower platform. The object surface consists of shell and disk. The shell is the cylindrical surface and the disk is the end surface, whose intersection is the circle border. Project the two objects onto a plane that is perpendicular to one object, e.g., we pick a , as shown in Fig. 1. This plane named as projection plane includes the disk of object a and may intersect with b 's main axis, labeled as C_1^b . The projection of object a is then a circle. Label unite vectors \mathbf{d}^a and \mathbf{d}^b as the direction of the main axes of objects a and b , D^a and D^b as their diameters, respectively, and ΔC_0^{b-a} as the vector pointing from C_0^a to C_0^b , computed using the image Jacobian matrix and the relative image distance. Picking a unit vector \mathbf{e}_v satisfying $\mathbf{e}_v \cdot \mathbf{d}^a = 0$,

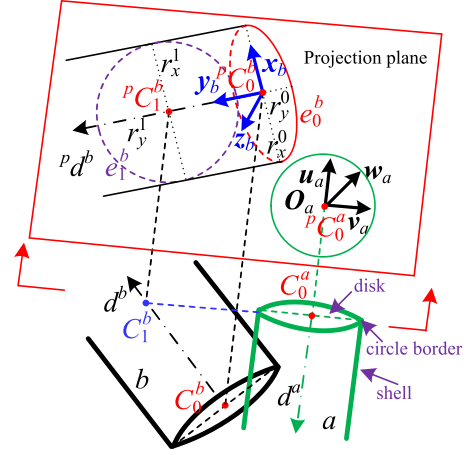


Fig. 1. Objects' projections onto the plane perpendicular to one object.

the rotational matrix yields

$${}^p\mathbf{R} = [\mathbf{e}_v \times \mathbf{d}^a \quad \mathbf{e}_v \quad \mathbf{d}^a]^T \quad (1)$$

where \times means cross product and ${}^p\mathbf{R}$ transforms from the world coordinates to $O_a - u_a v_a w_a$. Projecting the main axis of object b leads to

$${}^p\mathbf{d}^b = {}^p\mathbf{R} \frac{\mathbf{d}^a \times \mathbf{d}^b \times \mathbf{d}^a}{\|\mathbf{d}^a \times \mathbf{d}^b \times \mathbf{d}^a\|_2} \quad (2)$$

where ${}^p\mathbf{d}^b$ means the projection of object b 's main axis expressed at the coordinates of O_a . The points ${}^pC_0^a$ and ${}^pC_0^b$ correspond to the projections of end centers C_0^a and C_0^b , and the projection of the vector between two end centers yields

$$\Delta {}^pC_0^{b-a} = {}^p\mathbf{R} (\mathbf{d}^a \times \Delta C_0^{b-a} \times \mathbf{d}^a) \quad (3)$$

where $\Delta {}^pC_0^{b-a}$ represents the vector from ${}^pC_0^a$ to ${}^pC_0^b$.

The point C_1^b does not exist if \mathbf{d}^b is perpendicular to \mathbf{d}^a and may appear otherwise, which can be distinguished by

$$\Delta_1 = \mathbf{d}^a \cdot \mathbf{d}^b \quad (4)$$

where $\Delta_1 \neq 0$ indicates nonperpendicularity.

B. Nonperpendicular Cases

If \mathbf{d}^b is not perpendicular to \mathbf{d}^a , the point C_1^b is computed

$$\Delta {}^pC_1^{b-a} = \Delta {}^pC_0^{b-a} + \tan \theta (\Delta C_0^{b-a} \cdot \mathbf{d}^a) {}^p\mathbf{d}^b \quad (5)$$

where $\Delta {}^pC_1^{b-a} = {}^p\mathbf{R}(C_1^b - C_0^a)$ represents the vector pointing from C_0^a to C_1^b expressed in the coordinates of O_a , and θ is the acute angle between the two main axes. The second item on the right-hand side of (5) computes the length of $C_1^b - C_0^a$ on the projection plane according to the projected distance of C_0^b .

If the object b is totally above the projection plane, those models have no contact, which can be distinguished by

$$\Delta_2 = - \left(\Delta C_0^{b-a} + \frac{D^b \mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b}{2\|\mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b\|_2} \right) \cdot \mathbf{d}^a \quad (6)$$

where $\mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b$ is a vector on the object b 's disk and perpendicular to the vector that is parallel with the projection plane.

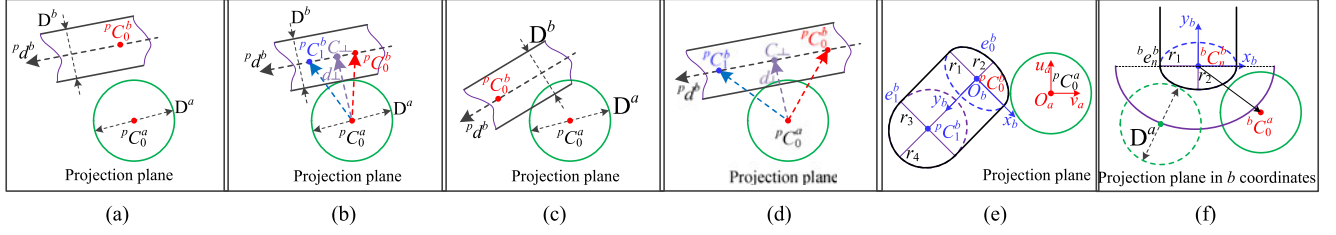


Fig. 2. Cases of crossed projected contours. (a) Case 3. (b) Case 4. (c) Crossed contours. (d) Case 5. (e) Truncation view. (f) In b 's coordinates.

The content in the bracket of the above equation describes the position of the lowest point on object b 's disk relative to C_0^a along the projection axis.

Case 1: $\Delta_2 > 0$. The object b is upper than a from the view-point of projection axis and noncollision is confirmed.

Case 2: $\Delta_2 = 0$. The disk other than the shell of object b intersects with the projection plane. Whether the two end surfaces contact is determined by

$$\Delta_3 = \begin{cases} \|\Delta C_0^{b-a}\|_2 - \frac{D^a + D^b}{2}, & \Delta C_0^{b-a} \cdot \mathbf{d}^a = 0 \\ \|\Delta C_0^{b-a} + \frac{D^b \mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b}{2\|\mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b\|_2}\|_2 - \frac{D^a}{2}, & \text{otherwise} \end{cases} \quad (7)$$

where if $\Delta C_0^{b-a} \cdot \mathbf{d}^a$ is zero, the two objects' main axes are parallel, and then, the above parameter compares the two end centers' distance and the sum of two radii; otherwise, Δ_3 indicates whether the lowest point on b 's disk locates inside a 's contour or not. If $\Delta_3 > 0$, no contact is approved; otherwise, collision happens and the contact point locates at

$$\mathbf{x}_c = \begin{cases} \Delta C_0^{b-a} + \frac{D^b \Delta C_0^{b-a}}{2\|\Delta C_0^{b-a}\|_2}, & \Delta C_0^{b-a} \cdot \mathbf{d}^a = 0 \\ \Delta C_0^{b-a} + \frac{D^b \mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b}{2\|\mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b\|_2}, & \text{otherwise} \end{cases} \quad (8)$$

where \mathbf{x}_c is the position of the contact point relative to C_0^a .

The following explores $\Delta_2 < 0$. The diameter of object a 's projected circle is still D^a and b 's contour width is D^b . Three possible relative positions exist in the contours, depending on the cross points between the projected contours or extensions. This is determined by

$$\Delta_4 = \|\Delta^p C_0^{b-a} \times {}^p \mathbf{d}^b\|_2 - \frac{D^a + D^b}{2} \quad (9)$$

where Δ_4 is the contours' shortest distance in the plane.

Case 3: $\Delta_4 > 0$. The main axes have sufficient displacement, and as a result, no collision happens, as shown in Fig. 2(a).

Case 4: $\Delta_4 = 0$. One intersection exists between the projected shapes or extension, as shown in Fig. 2(b). Draw a line pointing from ${}^p C_0^a$ and perpendicular to ${}^p \mathbf{d}^b$, and label this cross point as C_\perp . The vector from ${}^p C_0^a$ to C_\perp is

$$\Delta C_\perp = C_\perp - {}^p C_0^a = {}^p \mathbf{d}^b \times \Delta^p C_0^{b-a} \times {}^p \mathbf{d}^b. \quad (10)$$

A collision judgment parameter yields

$$\Delta_5 = (\Delta C_\perp \times \Delta^p C_0^{b-a}) \cdot (\Delta C_\perp \times \Delta^p C_1^{b-a}) \quad (11)$$

where Δ_5 indicates the location of C_\perp . If $\Delta_5 > 0$, C_\perp is outside of the line segment ${}^p C_0^a {}^p C_1^a$, which ensures no collision; otherwise, C_\perp lies within the line segment and contact occurs.

If the collision point exists, its position is expressed as

$$\mathbf{x}_c = \Delta C_0^{b-a} + \frac{(\Delta C_\perp - \Delta^p C_0^{b-a}) \cdot {}^p \mathbf{d}^b}{\sin \theta} \mathbf{d}^b - {}^p \mathbf{R}^{-1} \times \left(\Delta C_\perp - \frac{D^a \Delta C_\perp}{2\|\Delta C_\perp\|_2} \right) \quad (12)$$

where the second item on the right-hand side restores the vector along \mathbf{d}^b on the basis of its length on the projection plane, and the third part of this equation represents the vector from the point on \mathbf{d}^b whose projection is C_\perp to the collision point.

Below discusses $\Delta_4 < 0$, where the projected contours or extensions have multiple cross points, as shown in Fig. 2(c).

Case 5: $\Delta_5 \leq 0$. Penetration is reflected between objects and collision happens, as shown in Fig. 2(d), and the contact point's location relative to a end center can be computed by (12).

For the cases satisfying $\Delta_5 > 0$, a focus on the projected shape between ${}^p C_0^b$ and ${}^p C_1^b$ is needed. The object b 's disk is projected as an ellipse, labeled as e_0^b in Fig. 2(e). This ellipse's major axis is perpendicular to the main axis projection whose radius is $r_x^0 = \frac{D^b}{2}$, and its minor axis is along ${}^p \mathbf{d}^b$ with a radius of $r_y^0 = \frac{D^b \cos \theta}{2}$. The object b 's cross section that includes point C_1^b and is perpendicular to \mathbf{d}^a is also projected as an ellipse, labeled as e_1^b . This ellipse's minor axis is perpendicular to the main axis projection whose radius is $r_x^1 = \frac{D^b}{2}$, and its major axis is along ${}^p \mathbf{d}^b$ with a radius of $r_y^1 = \frac{D^b}{2 \cos \theta}$. The projected contour of object b is composed of a half part of these ellipses and two parallel connection lines, as the black curve in Fig. 2(e).

Since $\Delta_5 > 0$, the object a 's projected circle is closer to the ellipse than the connection lines. A rough judgment is given as

$$\Delta_6 = \begin{cases} \|\Delta^p C_1^{b-a}\|_2 - r_y^1 - \frac{D^a}{2}, & \xi \geq 0 \\ \|\Delta^p C_0^{b-a}\|_2 - r_x^0 - \frac{D^a}{2}, & \xi < 0 \end{cases} \quad (13)$$

where Δ_6 includes a rough safety region and $\xi = \|\Delta^p C_0^{b-a}\|_2 - \|\Delta^p C_1^{b-a}\|_2$ is an indicator to show which ellipse center is nearer to ${}^p C_0^a$. Δ_6 checks if the distance between a 's projected contour and ellipse circle is larger than its major radius.

Case 6: $\Delta_6 > 0$. Adequate distance lies between the two objects and results in no collision.

A further investigation is needed if $\Delta_6 \leq 0$. The projections of object b 's disk and cross section have the same radius in the direction perpendicular to ${}^p \mathbf{d}^b$, while different along ${}^p \mathbf{d}^b$. The projected shape may change if the out half of the ellipse closer to the point ${}^p C_0^a$ is violated by the other ellipse. This violation

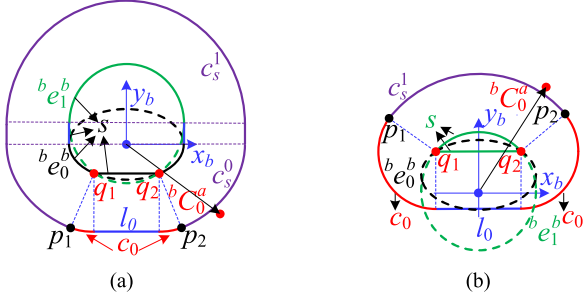


Fig. 3. Collision curve for the instances of ellipse intersect. (a) $\Delta C_0^{b-a} \cdot \mathbf{d}^a \geq 0$. (b) $\Delta C_0^{b-a} \cdot \mathbf{d}^a < 0$.

can be reflected by a parameter

$$\Delta_7 = \left(\Delta C_0^{b-a} - \frac{D^b \mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b}{2 \|\mathbf{d}^b \times \mathbf{d}^a \times \mathbf{d}^b\|_2} \right) \cdot \mathbf{d}^a \quad (14)$$

where Δ_7 reflects the uppermost point on b 's disk along \mathbf{d}^a that is above/below the projection plane. Those ellipses violate if b 's end surface has a part that surpasses the projection plane.

Case 7: $\Delta_7 \geq 0$. No violation mentioned above exists. Between ${}^p C_0^b$ and ${}^p C_1^b$, the closer point to ${}^p C_0^a$ is picked as the reference for computation, and for generalization, we use superscript or subscript n to correlate with the ellipse at whose center we set the origin of a new coordinate frame and f to represent the other one

$$(n, f) = \begin{cases} (1, 0), & (\Delta C_0^{b-a} \cdot \mathbf{d}^a < 0) \cup (\xi > 0 \cap \Delta_7 \geq 0) \\ & \cup (\Delta C_0^{b-a} \cdot \mathbf{d}^a \geq 0 \cap \xi \geq 0) \\ (0, 1), & \text{otherwise.} \end{cases} \quad (15)$$

n is set as 1 in three conditions: $\xi > 0 \cap \Delta_7 \geq 0$ means ${}^p C_1^b$ is closer with unviolated ellipses; $\Delta C_0^{b-a} \cdot \mathbf{d}^a < 0$, as shown in Fig. 3(b), means ${}^b e_1^b$ is the closer violated ellipse; and $\Delta C_0^{b-a} \cdot \mathbf{d}^a \geq 0 \cap \xi \geq 0$, as shown in Fig. 3(a).

Transform from $O_a - u_a v_a w_a$ to the coordinates of object b , where origin locates at ${}^p C_n^b$, \mathbf{y}_b is along ${}^p \mathbf{d}^b$, and \mathbf{z}_b is parallel with \mathbf{w}_a , as shown in Fig. 3(f). The homogeneous transformation matrix yields

$${}^a T_b = \begin{bmatrix} {}^a R_b & \Delta^p C_n^{b-a} \\ 0 & 1 \end{bmatrix} \quad (16)$$

where ${}^a R_b = [{}^p \mathbf{d}^b \times \mathbf{w}_a \quad {}^p \mathbf{d}^b \quad \mathbf{w}_a]$ is the rotation matrix mapping from the coordinates of object b to O_a

In the new coordinates, the ellipse equation in the projection plane yields

$${}^b e_n^b : [x_e \quad y_e \quad z_e]^T = [r_x^n \cos \alpha \quad r_y^n \sin \alpha \quad 0]^T \quad (17)$$

where r_x^n is the radius of the axis perpendicular to the main axis projection and r_y^n is the radius of the other axis, $\alpha + n\pi \in [\pi, 2\pi)$, which is determined that the upper ellipse is used for ${}^b e_1^b$ and the lower for ${}^b e_0^b$. For any point on this ellipse, the center position of the circle with a diameter of D^a that is tangent to

the ellipse at this point satisfies

$$\begin{bmatrix} (r_x^n)^2 y_e & - (r_y^n)^2 x_e \\ x_s - x_e & y_s - y_e \end{bmatrix} \begin{bmatrix} x_s - x_e \\ y_s - y_e \end{bmatrix} = \begin{bmatrix} 0 \\ (\frac{D^a}{2})^2 \end{bmatrix} \quad (18)$$

where $\mathbf{x}_s = [x_s, y_s, 0]^T$ is the point location with $\frac{D^a}{2}$ distance to the target ellipse. Combining the above two equations leads to

$$c_s : \mathbf{x}_s = \begin{bmatrix} r_x^n \cos \alpha + \frac{D^a r_y^n \cos \alpha}{2 \sqrt{(r_x^n \sin \alpha)^2 + (r_y^n \cos \alpha)^2}} \\ r_y^n \sin \alpha + \frac{D^a r_x^n \sin \alpha}{2 \sqrt{(r_x^n \sin \alpha)^2 + (r_y^n \cos \alpha)^2}} \\ 0 \end{bmatrix} \quad (19)$$

where c_s describes a collision curve with $\frac{D^a}{2}$ safety distance to the ellipse, as shown in Fig. 2(f).

In the new coordinates, we obtain the following result:

$$\Delta^b C_i^{a-b} = -{}^a R_b^{-1} \Delta^p C_i^{b-a} \quad (20)$$

where $i = 0, 1$. The line $\Delta^b C_n^{a-b}$ intersects with the collision curve at a point whose position is computed by

$$\mathbf{x}_s^n \times \Delta^b C_n^{a-b} = 0 \quad (21)$$

where $\mathbf{x}_s^n = [x_s^n, y_s^n, 0]^T$ represents the cross point between c_s and $\Delta^b C_n^{a-b}$, which can be solved by combining (19)–(21). Δ_8 is used to reflect the relative location of this cross point

$$\Delta_8 = (\Delta^b C_n^{a-b} - \mathbf{x}_s^n) \cdot \Delta^b C_n^{a-b}. \quad (22)$$

$\Delta_8 > 0$ represents \mathbf{x}_s^n lying between ${}^p C_0^a$ and ${}^p C_n^b$, which ensures safety; otherwise, collision happens.

If collides, ${}^p C_0^a$ is on the collision curve c_s , i.e., $\mathbf{x}_s^n = \Delta^b C_n^{a-b}$. Substituting it into (19), the contact point \mathbf{x}_s^c is acquired, which lies on e_n^b , and its relative position to C_0^a yields

$$\mathbf{x}_c = {}^p R^{-1} {}^a R_b (\mathbf{x}_s^c - \Delta^b C_n^{a-b}) + (1 - n) \Delta^p C_s^c \quad (23)$$

where $\Delta^p C_s^c = (\Delta C_0^{b-a} \cdot \mathbf{d}^a + \mathbf{x}_s^c \cdot {}^p \mathbf{y}_b \tan \theta) \mathbf{d}^a$ is the projection distance of the contact point relative to a ' end center, and $\mathbf{x}_s^c \cdot {}^p \mathbf{y}_b \tan \theta$ computes the relative projection distance error between the contact point and b 's end center. The last part of the above equation only emerges when $n = 0$, since the point on ${}^b e_1^b$ has zero projection distance.

Case 8: $\Delta_7 < 0$. The two ellipses violate. Label $\zeta = (\Delta^p C_1^{b-a} - \Delta^p C_0^{b-a}) \cdot {}^p \mathbf{d}^b$ to reflect ${}^p C_1^b$ that is above or below ${}^p C_0^b$ in b 's coordinates. Combining ellipse equations leads to the positions of the two intersections, labeled as q_1 and q_2

$$\mathbf{q}_i = \left[\frac{r_x^n}{2} \cos \alpha_i \quad \frac{r_y^n}{2} \sin \alpha_i \quad 0 \right]^T, \quad i = 1, 2 \quad (24)$$

and its corresponding point on the collision curve c_s is represented as \mathbf{p}_i , which is acquired by substituting \mathbf{q}_i into (19).

If $\Delta C_0^{b-a} \cdot \mathbf{d}^a \geq 0$, the intersections locate at the lower ellipses and b 's projection constitutes of ${}^b e_1^b$, two connection lines, a part of ${}^b e_0^b$, and the line $\overline{q_1 q_2}$, which is \mathbf{S} , as shown in Fig. 3(a). The collision curve has three parts: c_s^1 surrounding ${}^b e_1^b$, two connection lines, and a combination of c_s^0 above \mathbf{p}_i , the line l_0 , and two arcs with $\frac{D^a}{2}$ distance to \mathbf{q}_i . Here, we do not discuss the instance of $\Delta C_0^{b-a} \cdot \mathbf{d}^a \in [0, \zeta]$ since it can be distinguished by Δ_4 . If $\Delta C_0^{b-a} \cdot \mathbf{d}^a < 0$, the intersections

locate at the upper ellipses and the projection shape of object b consists of the upper ellipse of ${}^b e_1^b$ and the line $\overline{q_1 q_2}$, as shown in Fig. 3(b). Its collision curve is the shape surrounded by c_s^1 above p_i , the line l_0 , and two arcs.

A judgment parameter for violation instance is

$$\Delta_9 = \begin{cases} -\Delta^b C_n^{a-q_i} \cdot \mathbf{y}_b - \frac{D^a}{2}, & (\|\Delta^b C_n^{a-b} \cdot \mathbf{x}_b\| \leq \|\mathbf{q}_i \cdot \mathbf{x}_b\|) \\ & \cap (\Delta^b C_n^{a-q_i} \cdot \mathbf{y}_b < 0) \\ \|\Delta^b C_n^{a-q_i}\|_2 - \frac{D^a}{2}, & [(-1)^{i+1} (\mathbf{v}_i \times \Delta^b C_n^{a-q_i}) \cdot \mathbf{z}_b > 0] \\ & \cap [(-1)^{i+1} (\Delta^b C_n^{a-q_i} \times \mathbf{y}_b) \cdot \mathbf{z}_b < 0] \\ \Delta_8, & \text{otherwise} \end{cases} \quad (25)$$

where $\Delta^b C_n^{a-q_i} = (\Delta^b C_n^{a-b} - \mathbf{q}_i)$ is the vector pointing from q_i to ${}^b C_0^a$ and $\mathbf{v}_i = (\mathbf{p}_i - \mathbf{q}_i)$. In this equation, the first row corresponds to a 's center point below the line l_0 ; in the middle row, ${}^b C_0^a$ is closer to the arc c_0 than the other collision curve; and the last row indicates that Δ_8 applies for other $\Delta^b C_0^a$.

If $\Delta_9 > 0$, no collision occurs; otherwise, contact happens, with the contact point expressed in b coordinates as

$$\mathbf{x}_c^b = \begin{cases} \frac{D^a}{2} \mathbf{y}_b, & (\|\Delta^b C_n^{a-b} \cdot \mathbf{x}_b\| \leq \|\mathbf{q}_i^n \cdot \mathbf{x}_b\|) \\ & \cap (\Delta^b C_n^{a-q_i^n} \cdot \mathbf{y}_b < 0) \\ -\Delta^b C_n^{a-q_i^n}, & [(-1)^{i+1} (\mathbf{v}_i \times \Delta^b C_n^{a-q_i^n}) \cdot \mathbf{z}_b > 0] \\ & \cap [(-1)^{i+1} (\Delta^b C_n^{a-q_i^n} \times \mathbf{y}_b) \cdot \mathbf{z}_b < 0] \\ \mathbf{x}_s^c - \Delta^b C_n^{a-b}, & \text{otherwise.} \end{cases} \quad (26)$$

In the world coordinates, the contact point location yields (27), shown at the bottom of the page, where the first row depicts the contact point locates on ${}^b e_0^b$ in Fig. 3(a), and $\Delta \mathbf{p}_s^c$ is added to address the projection distance.

C. Perpendicular Instances

If \mathbf{d}^b is perpendicular to \mathbf{d}^a , i.e., $\Delta_1 = 0$, the projection of object b is a rectangle, as shown in Fig. 4(a). Noncontact is concluded with a sufficient distance between objects, which is described by

$$\Delta_{10} = -C_0^{b-a} \cdot \mathbf{d}^a - \frac{D^b}{2} \quad (28)$$

which computes the projection distance of b 's lowest point.

Case 9: $\Delta_{10} > 0$. The object b has enough displacement above a and safety is guaranteed.

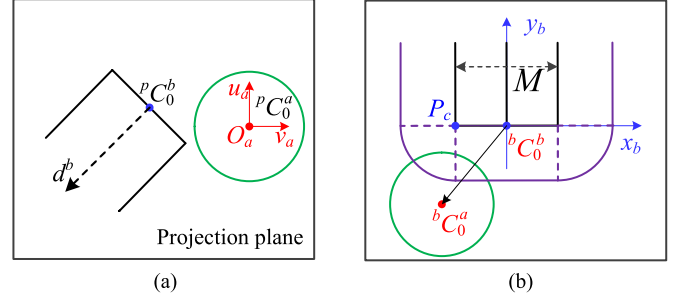


Fig. 4. Projecting objects for perpendicular cases. (a) Projection. (b) In b 's coordinates.

Case 10: $\Delta_{10} = 0$. The lowest points of object b along \mathbf{d}^a constitute of a line, and we consider the relative position between this line and a 's projected circle, with a parameter of

$$\Delta_{11} = \begin{cases} -1, & [(\|\Delta \mathbf{C}_\perp\|_2 \leq \frac{D^a}{2}) \cap (\Delta^p C_0^{b-a} \cdot \mathbf{d}^b < 0)] \\ 1, & \text{otherwise.} \end{cases} \quad (29)$$

$\Delta_{11} > 0$ is the requirement for noncollision and a negative Δ_{11} leads to object contact, with the contact position yielding

$$\mathbf{x}_c = \begin{cases} {}^p \mathbf{R}^{-1} \Delta^p C_0^{b-a}, & \Delta^p C_0^{b-a} \cdot \mathbf{d}^b > 0 \\ {}^p \mathbf{R}^{-1} \Delta \mathbf{C}_\perp, & \text{otherwise.} \end{cases} \quad (30)$$

Case 11: $\Delta_{10} < 0$. We need to investigate the projected contour relationship. C_0^b may be above the projection plane if $C_0^{b-a} \cdot \mathbf{d}^a \in (-\frac{D^b}{2}, 0]$ and the intersection between object b and the projection plane is a rectangle, with a width of

$$M = \begin{cases} 2\sqrt{\left(\frac{D^b}{2}\right)^2 - (\mathbf{d}^a \cdot C_0^{b-a})^2}, & \Delta_{10} \in \left[-\frac{D^b}{2}, 0\right) \\ D^b, & \Delta_{10} < -\frac{D^b}{2}. \end{cases} \quad (31)$$

Fig. 4(b) shows the collision (purple) curve, which is $\frac{D^a}{2}$ away from b 's projected contour. The collision parameter is

$$\Delta_{12} = \begin{cases} \|\Delta^b C_0^{a-b} \cdot \mathbf{x}_b\| - \frac{D^a}{2} - \frac{M}{2}, & \Delta^b C_0^{a-b} \cdot \mathbf{y}_b \geq 0 \\ & (\Delta^b C_0^{a-b} \cdot \mathbf{y}_b < 0) \cap \\ -\Delta^b C_0^{a-b} \cdot \mathbf{y}_b - \frac{D^a}{2}, & (\|\Delta^b C_0^{a-b} \cdot \mathbf{x}_b\| \leq \frac{M}{2}) \\ \|\Delta^b C_0^{a-b} - P_c\|_2 - \frac{D^a}{2}, & \text{otherwise} \end{cases} \quad (32)$$

where $P_c = \frac{M}{2} \text{sign}(\Delta^b C_0^{a-b} \cdot \mathbf{x}_b) \mathbf{x}_b$ is the corner point on the side of ${}^b C_0^a$. $\Delta_{12} > 0$ is the condition to ensure no collision. If

$$\mathbf{x}_c = \begin{cases} {}^p \mathbf{R}^{-1} \mathbf{R}_b \mathbf{x}_c^b + \Delta \mathbf{p}_s^c, & (\Delta C_0^{b-a} \cdot \mathbf{d}^a \geq 0) \cap (\xi < 0) \cap \\ & (\mathbf{v}_1 \times \Delta^b C_0^{a-q_1^0}) \cdot (\mathbf{v}_2 \times \Delta^b C_0^{a-q_2^0}) > 0 \\ {}^p \mathbf{R}^{-1} \mathbf{R}_b \mathbf{x}_c^b, & \text{otherwise} \end{cases} \quad (27)$$

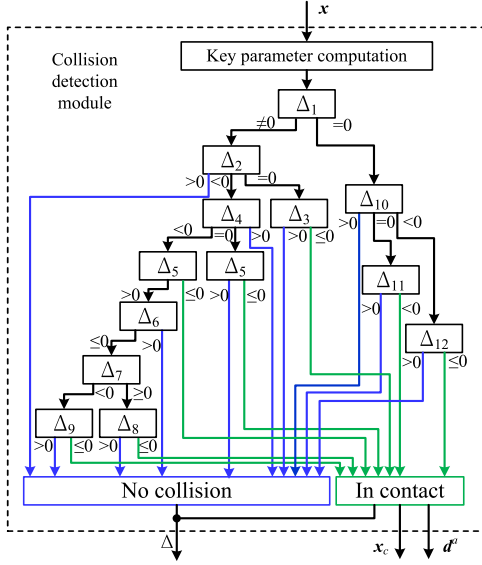


Fig. 5. Detection process of the projection approach.

collision happens, the contact position in b coordinates yields

$$\mathbf{x}_c^{b'} = \begin{cases} -\frac{D^a}{2} \text{sign}(\Delta^b \mathbf{C}_0^{a-b} \cdot \mathbf{x}_b) \mathbf{x}_b, & \Delta^b \mathbf{C}_0^{a-b} \cdot \mathbf{y}_b \geq 0 \\ \frac{D^a}{2} \mathbf{y}_b, & (\Delta^b \mathbf{C}_0^{a-b} \cdot \mathbf{y}_b < 0) \cap \\ & (\|\Delta^b \mathbf{C}_0^{a-b} \cdot \mathbf{x}_b\| \leq \frac{M}{2}) \\ \frac{D^a (P_c - \Delta^b \mathbf{C}_0^{a-b})}{2\|\mathbf{P}_c - \Delta^b \mathbf{C}_0^{a-b}\|_2}, & \text{otherwise.} \end{cases} \quad (33)$$

The contact point in the world coordinates results in

$$\mathbf{x}_c = {}^p \mathbf{R}^{-1a} \mathbf{R}_b \mathbf{x}_c^{b'}. \quad (34)$$

D. Detection Procedure

Fig. 5 shows the detection process for the projection approach, in which progressive relationships among judgment parameters are clear. Based on the current state \mathbf{x} , key parameters are computed, such as \mathbf{d}^a , \mathbf{d}^b , \mathbf{C}_0^{b-a} , etc. According to Δ_1 , the decision mechanism is divided into two branches: Δ_2 – Δ_9 serve for the nonperpendicular instances, and the others are responsible for the cases of perpendicular objects. From the practical point of view, it is a high possibility to employ only Δ_2 – Δ_6 for detection, since components are usually prohibited to stay too close for the sake of safety. For those complicated examples, further investigation is carried out by using Δ_7 – Δ_9 . A collision parameter Δ indicating the detection result outputs, together with \mathbf{x}_c and \mathbf{d}^a if contact happens.

This method can also be extended to detect the collisions among multiple cylindrical objects. For N objects, select the end surface of one object as the projection plane and project the other $N - 1$ objects onto this plane. After $N - 1$ computations, the collision states between the selected object and the others are computed according to the relationships of their planar contours.

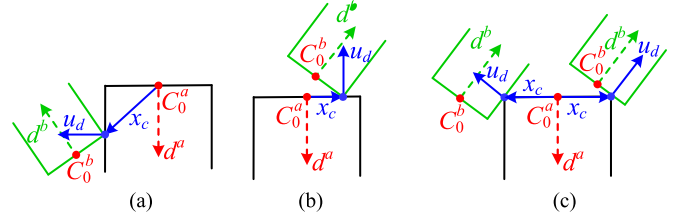


Fig. 6. Detach force direction corresponding to different contact locations. (a) $\mathbf{x}_c \cdot \mathbf{d}^a > 0$. (b) $\|\mathbf{x}_c\|_2 < \frac{D^a}{2}$. (c) Other contacts.

Continuing this projection and calculation, N objects' collision is detected in $\frac{N \times (N-1)}{2}$ computations.

III. DETACH CONTROL

Object collision is commonly forbidden in motion control, but due to misoperation or perturbations, collision may occur and force response must be immediately generated to detach objects in order to prevent damage. This detach control is usually the last protection for safe operation, and its appropriate design can enhance the robustness of the controller. This detach force is determined by the contact point and state error: the contact location specifies the force direction and the state error influences the force magnitude. Adding state error into detach force is based on the idea: if the goal is near a contact state, a small detach force may lead to a trajectory convergence to this goal, and if the desired state is far away, large force is generated to separate objects with substantial distance.

The detach force is designed as a piecewise function

$$\mathbf{u}_d = \begin{cases} 0, & \Delta > 0 \\ \varepsilon_d \|\Delta \mathbf{x}_d\|_2 \frac{\mathbf{d}^a \times \mathbf{x}_c \times \mathbf{d}^a}{\|\mathbf{d}^a \times \mathbf{x}_c \times \mathbf{d}^a\|_2}, & (\Delta \leq 0) \cap (\mathbf{x}_c \cdot \mathbf{d}^a > 0) \\ -\varepsilon_d \|\Delta \mathbf{x}_d\|_2 \mathbf{d}^a, & (\Delta \leq 0) \cap (\|\mathbf{x}_c\|_2 < \frac{D^a}{2}) \\ \varepsilon_d \|\Delta \mathbf{x}_d\|_2 \vec{c}, & \text{otherwise} \end{cases} \quad (35)$$

where \mathbf{u}_d is the detach force, $\Delta \mathbf{x}_d = \mathbf{x} - \mathbf{x}_d$ is the state error, \mathbf{x} and \mathbf{x}_d are, respectively, the current motion state and the desired state which is usually determined in advance, e.g., the well-aligned state before assembly, Δ reflects the collision status: $\Delta > 0$ means noncollision and a nonpositive Δ indicates object contact, and ε_d is a scaling parameter, which is less if more confidence on safety of future motion is gained. If the contact point is on the shell, as shown in Fig. 6(a), the detach direction is normal to the shell. If the contact point locates on the disk, as shown in Fig. 6(b), the detach force is opposite to \mathbf{d}^a . If the contact point belongs to a 's circle border, as shown in Fig. 6(c), the detach force is along the direction \vec{c} , which is parallel with or perpendicular to the projection of \mathbf{d}^b onto the plane that is composed by \mathbf{d}^a and \mathbf{x}_c .

To acquire \vec{c} , compose a plane involving \mathbf{x}_c and \mathbf{d}^a , and project the object b 's main axis onto the new plane

$${}^c \mathbf{d}^b = \frac{(\mathbf{x}_c \times \mathbf{d}^a) \times \mathbf{d}^b \times (\mathbf{x}_c \times \mathbf{d}^a)}{\|(\mathbf{x}_c \times \mathbf{d}^a) \times \mathbf{d}^b \times (\mathbf{x}_c \times \mathbf{d}^a)\|_2} \quad (36)$$

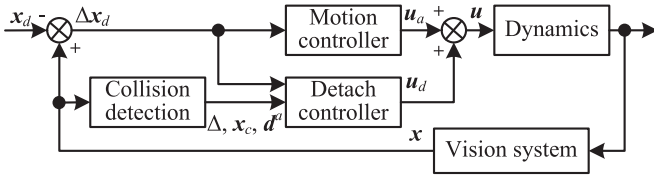


Fig. 7. Diagram of detach control.

where ${}^c d^b$ is the projection of d^b onto the plane including x_c and d^a . The reaction direction is determined according to ${}^c d^b$

$$\vec{c} = \begin{cases} {}^c d^b, & {}^c d^b \cdot x_c > 0 \\ x_c \times d^a \times {}^c d^b, & \text{otherwise} \end{cases} \quad (37)$$

where if an acute angle is between ${}^c d^b$ and x_c , ${}^c d^b$ is directly chosen as the force direction, while the perpendicular vector to ${}^c d^b$ is picked to detach objects.

The force controller includes the detach force when the collision happens and the attraction force to approach the desired state

$$u = u_a + u_d \quad (38)$$

where u is the force applied and u_a is the attraction force, which can be, e.g., a PD controller to reach the desired states.

Fig. 7 shows the control diagram to handle collisions. Collision is detected according to current objects' states, and the collision state Δ , main axis of the reference object d^a , and position of the contact point x_c if exists feed into the detach controller, where the detach force is generated in terms of contact location and motion state error. The sum of the detach force and the driven force applies to the robot to achieve expected manipulation. A high-gain motion controller may erase the detach effort and result in continuous collisions. Therefore, its parameters should be selected to render the motion controller a dumping property, and overshoot should be avoided.

IV. SIMULATIONS AND EXPERIMENTS

A. Simulation

The cameras i and j locate at the places perpendicular to the y -axis and the x -axis, respectively, and the diameters of objects a and b are 4 and 5 mm, respectively. As a comparison, we acquire the collision status using the following optimization equations:

$$f(x) = \|(x - x_0^a) \times d^a\|_2 - \frac{D^a}{2} \quad (39)$$

where x is a point on b 's surface or inside b , x_0^a is the end center of object a , and the optimization problem is written as a constraint nonlinear one

$$\begin{aligned} & \min f(x), \\ & \text{s.t.} \begin{cases} \|(x - x_0^b) \times d^b\|_2 - \frac{D^b}{2} \leq 0 \\ 0 \leq (x - x_0^b) \cdot d^b \leq \lambda^b \\ 0 \leq (x - x_0^a) \cdot d^a \leq \lambda^a \end{cases} \end{aligned} \quad (40)$$

where x_0^b is the end center of object b , and λ^a and λ^b are non-negative parameters to illustrate the length of the corresponding

objects in interest region. This optimization attempts to find the closest point x^* , which belongs to b and is not above a 's disk, to a . If $f_{\min} > 0$, safety is ensured; otherwise, contact happens, with its location at

$$x_c = x^* - x_0^a. \quad (41)$$

We run the program on a computer with Intel i3 3.30-GHz processor. Table I shows the parameters observed in each camera and their computation results using the proposed method, the compared optimization method, and the SWIFT method. The parameters Δu_i , Δv_i , and Δu_j are the position distances in the image of cameras i and j , where u and v mean the horizontal and vertical distances. The parameter φ stands for the angle between object's axis direction and the horizontal axis in image, where the subscript corresponds to the camera number and the superscript means the object. SNOPT is used for optimization to locate the closest point on object b . Fig. 8 displays the projections of some typical instances for collision detection. In cases 1 and 9, object b is totally above a and then has no contour on the projection plane. That is why, there is no showup of these two cases. Computational cost is chosen as the index to compare between the methods since plenty of examples show that the two approaches conclude identical collision status and contact point location given the same object states. The optimization approach computes 3-D relations, which takes more time and costs about 33 ms to assert contact status, while in projection, it only needs to consider the planar contours and, therefore, uses much less time—about 0.2 ms commonly and 0.4 ms for the cases that need to use iterative algorithms to compute the closest points between circle and ellipse. In each observed posture and position, optimization spends different computational time since it starts at the same initialization, the point C_0^b , but ends at different results. Compared with 3-D optimization, the projection method presents about 0.6% of computation cost on average. We also compare with SWIFT whose object fitting model employs 7200 vertices in order to acquire accurate results. As shown in Table I, SWIFT computes very fast, about 5 ms on average. Compared with it, the proposed method is about of 4% computation cost. Overall, the projection method shows great advantages in computation, suitable for fast and accurate operation.

A continuous motion is simulated driving object b from a given initial state to the desired posture and position. The initial state of object b is randomly selected as of position $[\Delta u_i, \Delta v_i, \Delta u_j]^T = [-3, -2, 4]^T$ mm and of posture vector of its main axis $[-1, 4, 1]^T$, and the desired state is of position $[\Delta u_i, \Delta v_i, \Delta u_j]^T = [5, -1, -0.5]^T$ mm and of posture vector of its main axis $[1, -3, 1]^T$. The PD controller is employed to drive the motor, and its parameters are set as $k_{pp} = 0.3\mathbf{I}_{5 \times 5}$ and $k_{pd} = 0.002\mathbf{I}_{5 \times 5}$. The detach scaling parameter is chosen as $\varepsilon_d = 10$. Fig. 10 shows the initial and desired states for two objects viewed from top, where the black and the blue contours stand for the initial and desired states of object b , respectively. The green-dotted line is moving object b directly to the desired state, along which collision with object a will obviously occur; therefore, the detach controller applies to separate them. The magenta-dotted curve is the trajectory of C_0^b employing the

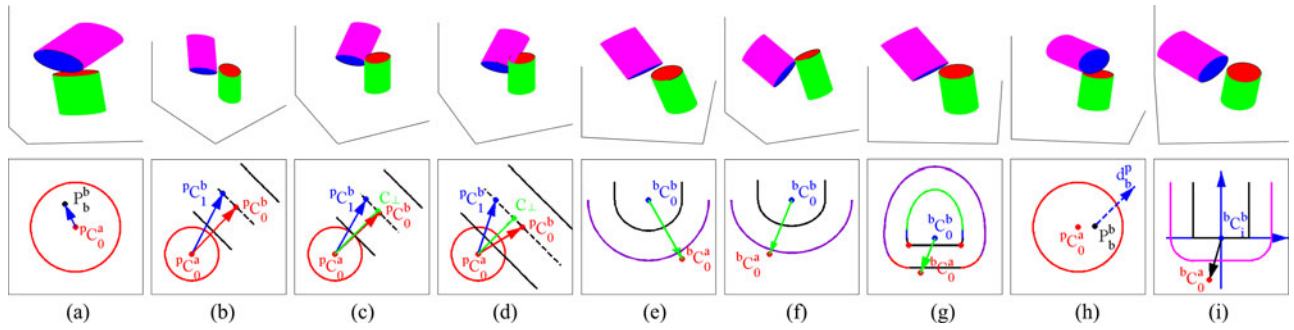


Fig. 8. Projections of nine typical instances for contact detection, where the upper row is the 3-D models, and the lower row is the projection illustration. (a) Case 2. (b) Case 3. (c) Case 4. (d) Case 5. (e) Case 6. (f) Case 7. (g) Case 8. (h) Case 10. (i) Case 11.

TABLE I
OBSERVED PARAMETERS AND CORRESPONDING COMPUTATION RESULTS

Observed postures and positions							Corresponding cases		Computation time (ms)		
$\varphi_i^a (^\circ)$	$\varphi_j^a (^\circ)$	$\varphi_i^b (^\circ)$	$\varphi_j^b (^\circ)$	$\Delta u_i (\mu\text{m})$	$\Delta v_i (\mu\text{m})$	$\Delta u_j (\mu\text{m})$	cases	contact?	projection	optimization	SWIFT
174.3	174.3	45	45	-1500	2150	0	case 1	No	0.17	34.5	4.9
174.3	174.3	45	45	-1500	2095	0	case 2	Yes	0.18	11.2	5.1
180	180	45	45	-3300	-1000	3500	case 3	No	0.17	28.0	5.0
180	180	45	45	-3300	-1000	3064	case 4	Yes	0.18	72.4	4.7
180	180	45	45	-3300	-2000	2000	case 5	Yes	0.18	20.2	5.3
-174.3	163.3	16.7	5.7	4000	-2500	3500	case 6	No	0.18	30.1	5.6
-174.3	163.3	45	45	2000	-3000	3000	case 7	No	0.42	56.5	5.4
-174.3	-174.3	45	45	1200	-1100	3000	case 8	No	0.22	30.4	4.9
90	90	-90	90	-2000	3000	-2000	case 9	No	0.15	25.6	5.2
-174.3	174.3	-78.7	78.7	-500	2555	-200	case 10	No	0.17	31.1	4.9
-174.3	174.3	-78.7	78.7	3500	-1000	2000	case 11	No	0.17	23.7	5.5

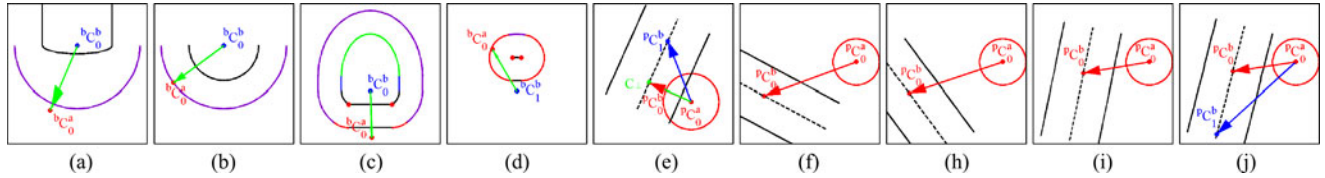


Fig. 9. Projection pictures at nine times of a continual motion. (a) 0 s. (b) 2.2 s. (c) 3 s. (d) 7.8 s. (e) 13.6 s. (f) 20 s. (g) 30 s. (h) 40 s. (i) 50 s.

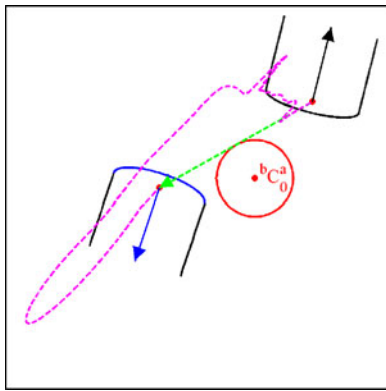


Fig. 10. States and trajectory viewed from top for the simulated continuous motion.

detach controller. Fig. 9 shows nine states of objects. During the motion, there are seven collisions in total and two of them

at times of 2.2 and 7.8 s. Detach force is immediately activated when the contact occurs in order to quickly separate objects, and this force only lasts for a very short period since the previous controller resumes after no collision is detected. This reveals the effectiveness of collision detection accuracy of the proposed approach and the detach force control for separation.

B. Experiment

We also carry out experiments on a precision assembly platform, which consists of six robot arms and three microscopes, as shown in Fig. 11. The camera i is GC2450 with a resolution of 2448×2050 and the camera j is PointGrey 50S5M-C whose resolution is 2448×2048 . Their cell size is $3.45 \mu\text{m}$. The upper manipulator is equipped with three Sugar KGW06030-G, whose translational resolution is $\pm 0.5 \mu\text{m}$, and two manual tilt adjustment Sigma KKD-25C. The lower manipulator constitutes Micos ES-100 with movement errors within $0.1 \mu\text{m}$ to realize vertical elevation, and KGW06050-L, KGW06075-

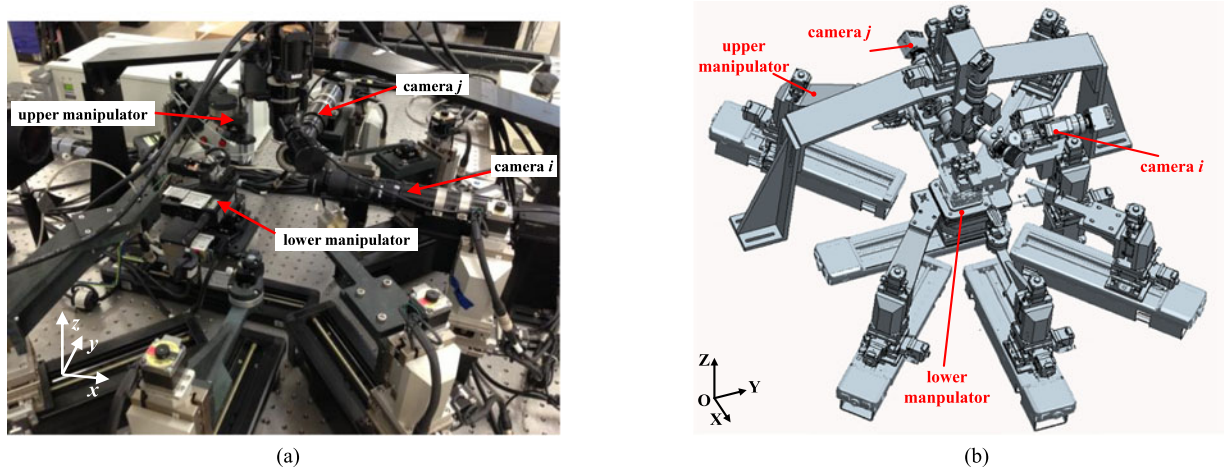


Fig. 11. Experimental platform for precision assembly with three microscopes and six robot arms. (a) Platform. (b) CAD model.

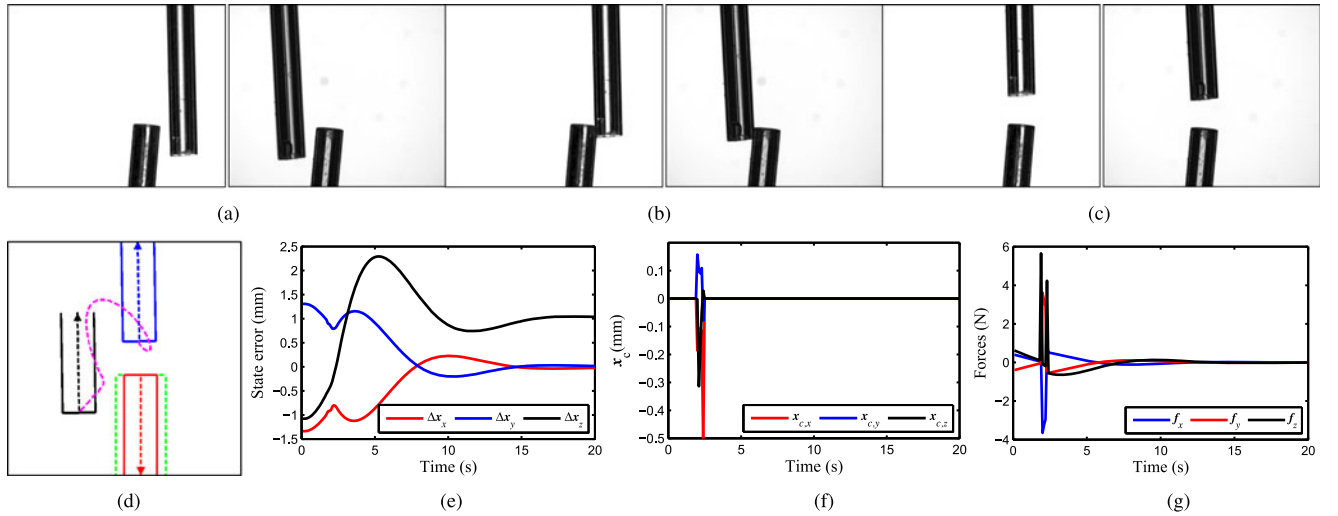


Fig. 12. Experiment results. (a) Initial states observed in two horizontal cameras. (b) One of the collision states where b 's diameter is enlarged. (c) Final states for two objects. (d) Trajectory and object relative position viewed from the direction that is perpendicular to d^a and the error vector between desired state and initial state. (e) State error versus step time. (f) Contact point locations when exist. (g) Forces generated by the proposed controller.

L, and SGSP-40yaw to rotate around its x -axis, y -axis, and z -axis, individually. Since the visual field of microscopes is quite small, the objects change to smaller ones compared with the experiments raised in [15]. Blocking may happen for such a mechanism with multiple manipulators when objects come into view sideways. The object diameters are both $985 \mu\text{m}$, but in the collision algorithm, we set D^a as 1.5 mm in considering protection of brittle components. The same controller parameters are chosen as used in simulation.

$[\Delta u_i, \Delta v_i, \Delta u_j]^T = [-1337, -1307, 1080]^T \mu\text{m}$ is set as the initial state of object b , and the desired state is $[\Delta u_i, \Delta v_i, \Delta u_j]^T = [0, 1000, 0]^T \mu\text{m}$. The postures keep unchanged with $\varphi_i^a = 91.5^\circ$, $\varphi_j^a = 93^\circ$, $\varphi_i^b = 85.1^\circ$, and $\varphi_j^b = 84.6^\circ$. The sampling time in the experiment is 0.02 s, which is limited by the frequency of the cameras. Fig. 12(a) and (c) shows the initial states and the final states in 20 s applying the proposed controller. Fig. 12(b) shows one of the collision status. This experiment is a typical alignment task before assem-

bly, and the difference lies in the lower position initialization of the upper object under which situation collision may occur. Fig. 12(d) shows the view from the direction that is perpendicular to d^a and the vector pointing from the initial state to the desired state. Blue and black contours are the initial and final states of object b , the red shape is object a , and green dotted is the object a with an enlarged diameter for the sake of protection during experiment. As can be seen, directly pushing object b to its desired state inevitably generates collision, and the proposed controller adjusts its motion according to the contact locations and presents the magenta-dashed trajectory. The detached force quickly separates two components, and object b approaches its desired state after counteracts the negative effect of the detach operation. The last three subfigures in Fig. 12 show the state errors, contact point locations if exist, and forces versus sampling time. At around 2 s, object b contacts with the enlarged region of object a , and the detach force is applied to separate two objects.

V. CONCLUSION

This paper focused on fast and accurate collision detection between cylindrical objects observed by microscopes and proposed a detach controller in the occurrence of contact. A projection approach was presented to transform two planar views to projected contours in a selected plane and to use relationships of low dimension to judge collisions in high dimension. To do that, 11 cases were classified in total according to the projected shapes and relative relations, and 12 parameters were designed to facilitate detection. In the occurrence of object collision, we proposed a detach controller for separation in terms of contact position. Simulation and experiment results proved the feasibility of the proposed methods and their efficiency was also compared. It shows that the projection method has great computation efficiency and is especially suitable for constant operation.

REFERENCES

- [1] P. Jimenez, F. Thomas, and C. Torras, "3D collision detection: A survey," *Comput. Graph.*, vol. 25, no. 2, pp. 269–285, 2001.
- [2] M. C. Lin, "Efficient collision detection for animation and robotics," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, 1993.
- [3] X. Zhang and Y. J. Kim, "Interactive collision detection for deformable models using streaming AABBs," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 2, pp. 318–329, Mar./Apr. 2007.
- [4] M. Tang, D. Manocha, S.-E. Yoon, P. Du, J.-P. Heo, and R. F. Tong, "VolCCD: Fast continuous collision culling between deforming volume meshes," *ACM Trans. Graph.*, vol. 30, no. 5, pp. 111.1–111.15, 2011.
- [5] L. He, J. Pan, D. Li, and D. Manocha, "Efficient penetration depth computation between rigid models using contact space propagation sampling," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 10–17, Jan. 2016.
- [6] D. H. Eberly, *3D Game Engine Design: A Practical Approach to Real-time Computer Graphics*. San Francisco, CA, USA: Morgan Kaufmann, 2007.
- [7] S. A. Ehmann and M. C. Lin, "Accelerated proximity queries between convex polyhedra by multi-level voronoi marching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2000, pp. 2101–2106.
- [8] M. Tang, H. Wang, L. Tang, R. Tang, and D. Manocha, "CAMA: Contact-aware matrix assembly with unified collision handling for GPU-based cloth simulation," *Comput. Graph. Forum*, vol. 35, pp. 511–521, 2016.
- [9] A. Chakravarthy and D. Ghose, "Collision cones for quadric surfaces," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1159–1166, Dec. 2011.
- [10] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1209–1230, 2014.
- [11] U. Schwesinger, R. Siegwart, and P. Furgale, "Fast collision detection through bounding volume hierarchies in workspace-time space for sampling-based motion planners," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 63–68.
- [12] K. D. Do, "Synchronization motion tracking control of multiple underactuated ships with collision avoidance," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 2976–2989, May 2016.
- [13] J. Pan and D. Manocha, "Fast probabilistic collision checking for sampling-based motion planning using locality-sensitive hashing," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1477–1496, 2016.
- [14] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1401–1410, May 2009.
- [15] D. Xing, D. Xu, F. Liu, H. Li, and Z. Zhang, "Precision assembly among multiple thin objects with various fit types," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 1, pp. 364–378, Feb. 2016.
- [16] S. Liu, D. Xu, F. Liu, D. Zhang, and Z. Zhang, "Relative pose estimation for alignment of long cylindrical components based on microscopic vision," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1388–1398, Jun. 2016.
- [17] F. Shen, W. Wu, D. Yu, D. Xu, and Z. Cao, "High-precision automated 3-D assembly with attitude adjustment performed by LMTI and vision-based control," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1777–1789, Aug. 2015.
- [18] S. Liu, D. Xu, D. Zhang, and Z. Zhang, "High precision automatic assembly based on microscopic vision and force information," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 382–393, Jan. 2016.
- [19] D. Xing, F. Liu, F. Qin, and D. Xu, "Coordinated insertion control for inclined precision assembly," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 2990–2999, May 2016.
- [20] D. Xing, D. Xu, and F. Liu, "Collision detection for cylindrical objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 4798–4803.



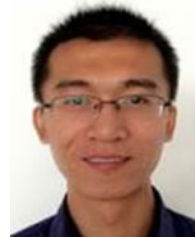
Dengpeng Xing received the B.S. degree in mechanical electronics and the M.S. degree in mechanical manufacturing and automation from Tianjin University, Tianjin, China, in 2002 and 2006, respectively, and the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010.

He is currently an Associate Professor with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include robot control and learning, precision assembly, and optimization.



Fangfang Liu received the B.S. and Ph.D. degrees in mechanical electronics from Zhejiang University, Hangzhou, China, in 2006 and 2012, respectively.

She is currently an Associate Professor with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include precision mechanical design and mechatronic control.



Song Liu received the B.S. degree in sensing technology and instrumentation from Shandong University, Jinan, China, in 2012. He is currently working toward the Ph.D. degree in control science and engineering in the Department of Mechanical and Biomedical Engineering, College of Science, City University of Hong Kong, Kowloon, Hong Kong; the Institute of Automation, Chinese Academy of Sciences, Beijing, China; and the University of the Chinese Academy of Sciences, Beijing.

His current research interests include visual measurement, visual control, visual positioning, microassembly, and robot learning.



De Xu (M'05–SM'09) received the B.S. and M.S. degrees from Shandong University of Technology, Jinan, China, in 1985 and 1990, respectively, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2001, all in control science and engineering.

Since 2001, he has been with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, where he is currently a Professor with the Research Center of Precision Sensing and Control. His research interests include

robotics and automation, in particular, the control of robots, such as visual control and intelligent control.