Robotica (2016) volume 34, pp. 118–134. © Cambridge University Press 2014
doi:10.1017/S0263574714000939

Simulation-based fast collision detection for scaled polyhedral objects in motion by exploiting analytical contact equations

Jing-Sin Liu†*, Wen-Hua Pan‡, Wen-Yang Ku§, Y.-H. Tsao¶ and Y.-Z. Chang¶

†*Institute of Information Science, Academia Sinica, Nangang, Taipei, Taiwan 115, Republic of China*
‡*Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei, Taiwan 105, Republic of China*
§*Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada*
¶*Department of Mechanical Engineering, Chang Gung University, Tao-Yuan, Taiwan 333, Republic of China*

(Accepted March 14, 2014. First published online: August 8, 2014)

SUMMARY
Based on the results of the study of convex object motion[1] (J. Hopcroft and G. Wilfong, "Motion of objects in contact," *Int. J. Robot. Res.*, **4**(4), 32–46 (1986)), this paper addresses the problem of exact collision detection of a pair of scaled convex polyhedra in relative motion, and determines the contact conditions of tangential contact features, arbitrary relative motion involving translation and rotation, and uniform scaling of the objects about a fixed point. We propose a new concept of the decision curve based on analytical contact equations that characterize a continuum of scaling factors (or a single scaling factor), which ensures that a pair of objects undergoing a scaling transformation will maintain the same tangential contact feature pair (or make instantaneous tangential contact feature transitions). We propose a reliable simulation-based approach to construct the decision curve by hybridizing analytical contact equations and conventional collision detection method, called the Fast Collision Detection Method (FCDM). This method can determine whether two scaled objects will make contact at specific tangential contact features (vertices, edges, or faces) under particular uniform scaling factors and after distinctive relative motion with better accuracy and less computational time than the existing collision detection methods. Finally, we demonstrate our approach for solving motion design in simple assembly/disassembly problems.

KEYWORDS: Collision detection; Contact; Scaling; Relative motion; Assembly.

Notations:
GJK: Gilbert–Johnson–Keerthi distance algorithm of two convex sets[17]
Pi: A polyhedral object, $i = 1, 2$
s_i: Seed point of a polyhedron Pi, $i = 1, 2$
ρ_i: Scaling factor of a polyhedron Pi
ρ_i^l, ρ_i^u: Minimum and maximum allowable values of the scaling of object Pi
$I_i = [\rho_i^l, \rho_i^u]$ of polyhedron Pi, $i = 1, 2$
Δ_1: Horizontal spacing of discretiztion
Δ_2: Vertical spacing of discretiztion
L: Decision curve

* Corresponding author. E-mail: liu@iis.sinica.edu.tw

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

1. Introduction

A collision detection task in robotics, animation, computer graphics, Computer-Aided Design (CAD)/Computer-Aided Manufacturing (CAM) involves checking for intersections between two objects. The objects can be polyhedral or non-polyhedral, rigid or deformable, static or moving. Both approximate and exact methods have been well studied for detecting collisions between general or convex rigid polyhedra,[5,6] arbitrary or deforming polyhedra, moving ellipsoids,[9–11] and in virtual environments or graphics.[12–15,29] In particular, the determination of collision status of two rigid and convex polyhedra in motion at discrete time instants or in continuous time have already been extensively studied. Distance measures related to collision status of two objects are separation distance and penetration distance. Separation distance is the closest Euclidean distance between two separate objects. Penetration distance is the distance measure quantifying the amount of interpenetration between two overlapping objects, or the minimum translational distance required to separate the two. Penetration distance has many applications in motion planning, design, animation, and simulations.[29] It includes calculating the minimum translational distance, directional separation or penetration distances, defined as the minimum distance along a specific direction that a convex polyhedron translates to be in external contact with another convex polyhedron.[2–4,27,29] Efficient algorithms were proposed to compute the defined distances. These distance-computation algorithms are classified as feature-based and simplex-based algorithms. Feature-based algorithms, such as the Lin–Canny closest features algorithm, use features (vertices, edges, and faces) to disjoint the polyhedral objects and compute the distance between them. Simplex-based algorithms, on the other hand, regard polyhedral objects as a set of points and calculate the distance between the points. The most widely used simplex-based distance computation method is called (enhanced) the Gilbert–Johnson–Keerthi (GJK) distance algorithm.[2,3,5,17] More recently, the distance computation has been extended to a directional contact range,[28] defined as the upper and lower bounds of distances in a given line within which two convex polyhedra are in collision. An algorithm resolving directional contact range based on the Minkowski difference is proposed.

Scaling transformation is an important elementary operation of affine transformation.[2] The growth distance function[16] that quantifies the extent of penetration is defined via the identical scaling of a pair of convex polyhedra. The technique measures how much two convex polyhedra must be grown or shrunk identically to bring them into external contact. To accommodate undesired contact or collision resulting from scaling, this paper studies the exact, as opposed to approximate, detection of the contact status and the contact type for convex polyhedral objects undergoing scaling transformation, i.e., scaling the size of the objects and thus changing the regions occupied by the objects, while the object shapes remain unchanged.

Hopcroft and Wilfong[1] (p. 45) proposed one of the earliest methods for relating contact and motion. They showed that their method could be generalized to relative motion comprising translation and rotation and any kind of continuous deformation of objects, such that the objects remain convex and the deformation has a fixed point.

In this respect, our work can be regarded as a realization of their generalization to a pair of scaled convex polyhedra. Generally, the collision status of a pair of scaled convex polyhedral objects can be detected by repeatedly applying one of the two collision detection methods: feature-based algorithms and simplex-based algorithms.[3] However, for this particular subclass of the general collision detection problem, the whole detection process could be sped up by taking into account the characteristics of the underlying geometry or motion. We propose a decision curve-based method, called the Fast Contact Detection Method (FCDM), which efficiently establishes the correlations between scaling pairs, tangential contact features, and relative fine motion conditions. In addition, we consider and answer the following inverse question: Under what conditions can we derive the desirable mating configurations for convex polyhedral parts? The conditions include tangential contact features, relative motion, and scaling pairs. We also demonstrate that FCDM can be applied in a simple and flexible way to solve different motion design problems, such as those related to assembly and tolerance problems,[18,19] of convex/non-convex/hollow polyhedra objects.

Therefore, FCDM can be a useful tool that allows designers to systematically examine several possible solutions to motion design problems when the tangential contact features, relative motion, and scaling pairs of the objects are the main concerns. FCDM can be applied to manufacturing/assembly, CAD/CAM, computer-aided geometric design, interactive 3D virtual reality applications, animation, computer graphics, and other fields involving motion and scaling.

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

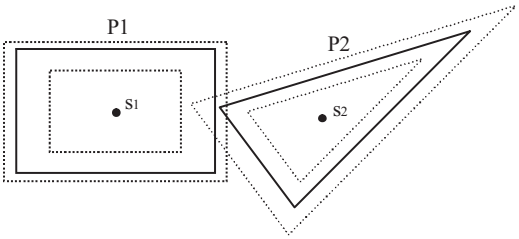


Fig. 1. Determination of exact contact status for different scaled configurations of a pair of convex polyhedra undergoing scaling transformation relative to seedpoints without shape change.

The remainder of this paper is organized as follows. In Section 2, we introduce the concept of a contact decision curve and provide illustrated examples. In Section 3, we present our FCDM for exact detection of collisions between scaled objects and demonstrate its comparative efficacy. Section 4 presents two examples to illustrate the use of FCDM as a systematic and flexible motion design tool; and Section 5 contains some concluding remarks.

2. Contact Decision Curve

Collision detection of a pair of convex polyhedra undergoing scaling transformation is stated formally in this section. Figure 1 shows the determination of contact for a pair of convex polyhedra at different scaled configurations. We present the concept of the contact decision curve, which is crucial to our simulation method. Basically, the decision curve is a curve on the plane of scaling factors that characterizes all the scaling pairs (in fact, a continuum) that cause two scaled convex polyhedra to remain in contact externally.

2.1. Uniform scaling

For each convex polyhedron, uniform deformation of the vertex coordinates assumes that any point on the object undergoes the same fraction of stretching or compression displacement relative to a fixed but arbitrary interior point, called a seedpoint s_i, such that

$$\tilde{p}_i = \rho_i p_i, \tag{1}$$

where ρ_i is the allowable scaling value within a user specified interval, and $\tilde{p}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i) - s_i$ is the displacement of the point on each polyhedral object undergoing scaling transformation ρ_i.[16] Let ρ_i^l, ρ_i^u be the respective minimum and maximum allowable values of the scaling of an object Pi such that scaling $\rho_i \in I_i = [\rho_i^l, \rho_i^u],\ i = 1,\ 2$ is the interval of the given upper and lower bounds of scaling factors.

Throughout the paper, $(P_1(\rho_1), P_2(\rho_2))$ denotes a pair of objects after applying the scaling pair (ρ_1, ρ_2) to the object pair (P_1, P_2) in its original (unscaled: $\rho_1 = \rho_2 = 1$) configuration. The relative configuration of a pair of scaled convex polyhedra $(P_1(\rho_1), P_2(\rho_2))$, whose original configuration (relative position and orientation) is specified, can also be represented uniquely and interchangeably by the scaling pair (ρ_1, ρ_2), i.e., a point on the plane of scaling factors.

2.2. Contact decision curve

A point of contact between two convex polyhedra can be represented by a pair of features, one from each object. There are three basic types of contacts[20]: face–vertex (TypeFV), edge–edge (TypeEE), and vertex–edge (TypeVE) contacts. In each type, a continuum of scaling pairs (ρ_1, ρ_2) that makes the objects maintain the same contact feature, or a single scaling pair that yields an instantaneous transition of the tangential contact features, forms a linear relationship on the plane of scaling factors[1] such that

$$\rho_2 = a\rho_1 + b, \tag{2}$$

[1] Please refer to Appendix for analytical derivations.

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

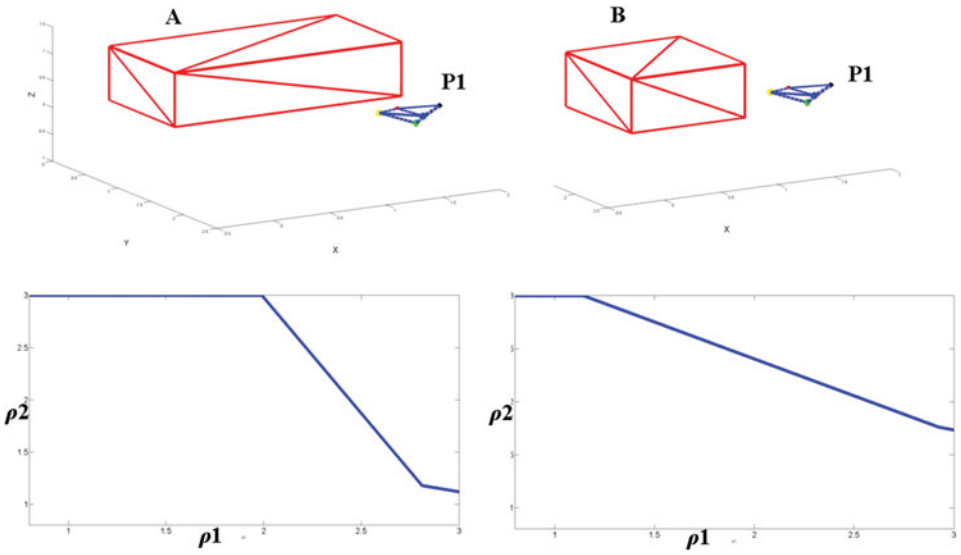


Fig. 2. Illustration of construction of decision curves for two pairs of convex polyhedra used for Example 3 for $(\rho_1, \rho_2) \in [0.8, 3]^2$.

where a and b are constants that depend on the original/initial relative configuration and the tangential contact features pair respectively. That is, if P_1 is scaled by a fraction ρ_1, then P_2 should be scaled by the fraction ρ_2 determined by Eq. (2) to maintain external contact with the specific tangential contact features pair. As (ρ_1, ρ_2) varies over all of the allowable ranges, denoted by $I_1 \times I_2$, a piecewise linear curve with each line segment represented by Eq. (2) on the plane of scaling factors, called a *Contact Decision Curve*, is constructed for each tangential contact features pair. The junction (switching point) is the single scaling pair in which the tangential contact features will change instantaneously. A switching point is often coincident with the discontinuity of contact force/torque,[21] and the objects remain in contact; however, the feature may be changed as the decision curve is traversed, i.e., as the contact scaling pairs change. Fig. 2 shows the construction of the piecewise-linear decision curve for two pairs of convex polyhedra with the scaling factors.

2.3. Overlapping and non-overlapping sub-regions

The piecewise linear curve $L = \cup_1^m \mathrm{SEG}_i$ comprises m line segments, each of which is described by the linear equation

$$\mathrm{SEG}_i : \rho_2 = a_i \rho_1 + b_i, \quad i = 1, \ldots, m,$$

which separates $I_1 \times I_2$ into an exact overlapping sub-region and an exact non-overlapping sub-region. Therefore, by checking the scaling pair (ρ_1, ρ_2) above or below the curve L, we can derive the overlap status without calculating the nearest distance or closest feature, which is necessary in conventional collision detection methods.[4] It is robust even for extreme case of very low scaling that could cause technical difficulty in numerical implementation of GJK.

Note that different initial configurations of the same pair of convex polyhedral objects have different decision curves. This is because a change in the initial configuration (e.g., by relative motion) causes a change in the decision curve, which could be efficiently updated from original decision curve.

2.4. Advantage of the decision curve in contact determination

When the polygon of scaling factors bounded by $I_1 \times I_2$ is defined, it is straightforward to check the collision status of the two objects in any scaling pair (ρ_1, ρ_2) by checking (ρ_1, ρ_2) relative to the decision curve L if the decision curve is given. The exact overlapping sub-region is above L and the exact non-overlapping sub-region is below L.[22] This allows to determine the collision status of two objects in any uniform scaling pair[4] without calculating the nearest distance or the closest feature. Furthermore, if we want to calculate a decision curve between a convex object and a non-convex object, we decompose the non-convex polyhedral object into a few convex polyhedral parts

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

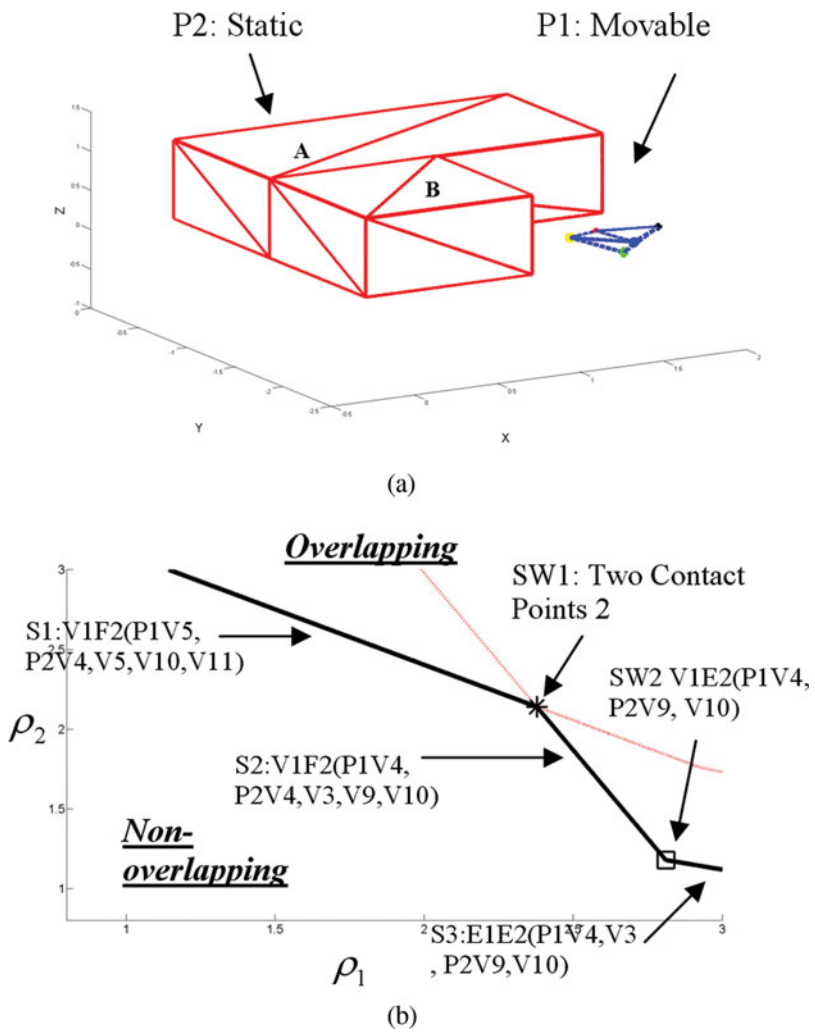


Fig. 3. Construction of contact decision curve. (a) Example I. A non-convex polyhedron and a convex polyhedron. (b) The combined decision curve from Fig. 2. $(\rho_1, \rho_2) \in [0.8, 3]^2$, SW1: *, SW2:□.

to compute the decision curve of each decomposed part and original convex object. Combining all decision curves, we can then define the whole decision curve between the convex object and the non-convex object. Moreover, the set of decomposed parts must follow the grouping principle of "Mutually Exclusive and Collectively Exhaustive."

In order to illustrate the combination of decision curves of decomposed convex polyhedra, we present Figs. 3 and 4, each of which has a pair of polyhedral objects that are separated initially.

Notations in Figs. 3 and 4 are as follows:

SW denotes switching point.

V, E, and F denote vertex, edge, and face respectively.

S denotes segment.

V1F2 (P1V5,P2V4,V5,V10,V11) denotes a vertex–face contact, which is vertex V1 of P1 in contact with face composed by vertices V4,V5,V10,V11 of P2.

V1E2 (P1V4,P2V9,V10) denotes a VE contact, which is V4 of P1 in contact with edge V9V10 of P2.

E1E2 (P1V4V3,P2V9V10) denotes an EE contact, which is edge V4V3 of P1 in contact with edge V9V10 of P2.

The first example in Fig. 3 is a continuation of Fig. 2. Figure 3(a) shows a non-convex P2 that we decomposed into the two convex parts (A, B) shown in Fig. 2. We obtain the decision curve of Fig. 3(b) by combining the decision curve of each part depicted in Fig. 2. Given the same value of ρ_1, if ρ_2 has more than one value, then the smallest one will be the exact scaling value. The region above

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

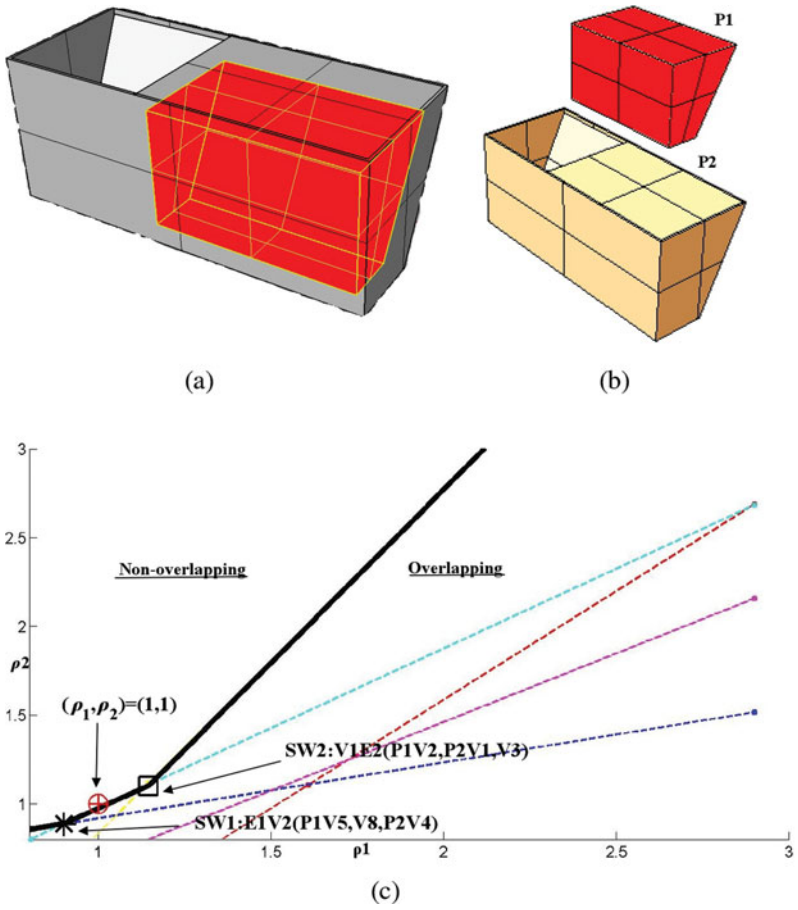


Fig. 4. Construction of contact decision curve (a) Example II: A convex object inside a hollow box. (b) Two objects. (c) The decision curve (black). $(\rho_1, \rho_2) \in [0.8, 3]^2$. SW1: *, SW2:□, $(\rho_1, \rho_2) = (1, 1)$:O, which is above the decision curve.

the decision curve is the overlapping area because its overlapping area covers all others, therefore the smallest scaling value of ρ_2 should be the exact value.

Figure 4(a) shows the other example, a more complex example in which a convex object is placed inside a hollow box. To compute the decision curve, we apply the process used for a non-convex object in the same way as illustrated in Fig. 3, i.e., decomposed the hollow box into six convex parts, then combined those six decision curves. However, if there is more than one value of ρ_2 for the same value of ρ_1, then the largest one will be the exact scaling value, since the region below the decision curve is the overlapping area for the hollow object (Fig. 4(b)). Therefore, the concept of decision curve allows us to exactly determine the collision status of two convex/non-convex/hollow polyhedra objects in any uniform scaling pair easily.

2.5. Relative motion

Without loss of generality, it is assumed that one object (Polyhedron 1) in a scaling pair is stationary and the other (Polyhedron 2) is movable. A motion can be defined by specifying its type/direction/limit[20,23] based on the following categories.

Motion type: It is assumed that relative motion is a combination of translation and rotation; therefore following are the types of motion: translation along a straight line, rotation about a given axis, and composite motion. The last type includes a translation followed by a rotation, a rotation followed by a translation, and a screw motion (defined as a translation followed by a rotation with the screw axis vector, the screw angle, and the screw pitch[24]). In fact, any type of relative motion in different representations (e.g., quaternion, rational motion . . . etc.) could be adopted to interpolate the new position of a movable object.

Direction: The axis of motion in a global/local coordinate frame. It is given by a direction vector.

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

Limit: The limit of allowable motion, which ensures that the process will terminate after a finite number of iterations.

Value: In any application, if there is a specific value within the motion limit that can satisfy the requirement, then that value will be recorded.

The relative motion can be represented with the notation "MotionType (Direction = Value)". For example, Trans($Z = 0.5$) represents a translation of 0.5 units along the Z-axis [0 0 1]; and Rot($X = 0.5$) represents a rotation of 0.5° around the X-axis[1 0 0]. Screw([1 0 1]axis = (5, 0.125)) represents a screw motion, a screw angle of 5° around the axis vector [1 0 1], and a screw pitch of 0.125 units along the screw axis [1 0 1].

3. Fast Contact Detection Method

Our objective is to develop an efficient and accurate automated process using only geometric data (vertex, edge, and face data) as input, and/or relative motion conditions, to generate a piecewise linear decision curve that characterizes the set of all contact scaling pairs and the loci of all contact points. We call the proposed simulation-based method as FCDM.

3.1. Conventional collision detection method

Calculating the distance between two objects is straightforward with conventional collision detection algorithms (i.e., feature-based algorithms and simplex-based algorithms). However, if we want to determine which scaling pair would let the two objects just contact at one point, we must apply an exhaustive search approach, such as the bisection method.

First, we choose a resolution of ρ_1 and ρ_2 that uniformly discretizes the rectangle $I_1 \times I_2$ in the plane of scaling factors into a finite number of rectangular grids with horizontal spacing Δ_1 and vertical spacing Δ_2. For a given $\rho_1 \in I_1$, the bisection method can normally be used to find the corresponding $\rho_2 \in I_2$ so that two polyhedral objects just contact. Therefore, the number of scaling pairs, n, on the decision curve depends on the resolution of ρ_1. The detected number of grid points, k, of ρ_2 that must be checked by the bisection method is resolution-dependent,

$$\begin{aligned} n &= \text{ceil}\left(\frac{\text{search range of } \rho_1}{\text{resolution of } \rho_1}\right) = \text{ceil}\left(\frac{I_1}{\Delta_1}\right), \\ 2^{k'} &= \left(\frac{\text{search range of } \rho_2}{\text{resolution of } \rho_2}\right) = \frac{I_2}{\Delta_2}, \\ k &= \text{ceil}(k'), \\ &\quad \text{ceil}() : \text{Ceiling function that maps a real number} \\ &\quad \text{to the smallest following integer.} \end{aligned} \tag{3}$$

Overall, the bisection method needs to perform $n \times k$ collision detection steps.

3.2. Analytical method

Alternatively, for contact maintenance/transition, we can derive ρ_1 and ρ_2 explicitly based on Eq. (2) by considering the geometric data of two size-changeable convex polyhedral objects that have different contact statuses. This analytical method can find the contact scaling pair and its associated tangential contact feature, e.g., (F,V), (E,E), or (V,E). However, the method cannot verify whether there is contact because real objects are solid, while it only considers vertices, edges, and faces of a solid object.

3.3. Fast contact detection method

The steps of FCDM are shown in Fig. 5. By combining an analytical method and a conventional collision detection method, FCDM overcomes the drawback of the former method and avoids the large amount of collision checking required by the latter method.

For two convex polyhedra comprising α and β vertices respectively, the proposed analytical method calculates the following combinations:

$$(F, V) : C_3^{\alpha} \times C_1^{\beta}, \quad (E, E) : C_2^{\alpha} \times C_2^{\beta}, \quad (V, E) : C_1^{\alpha} \times C_2^{\beta}. \tag{4}$$

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

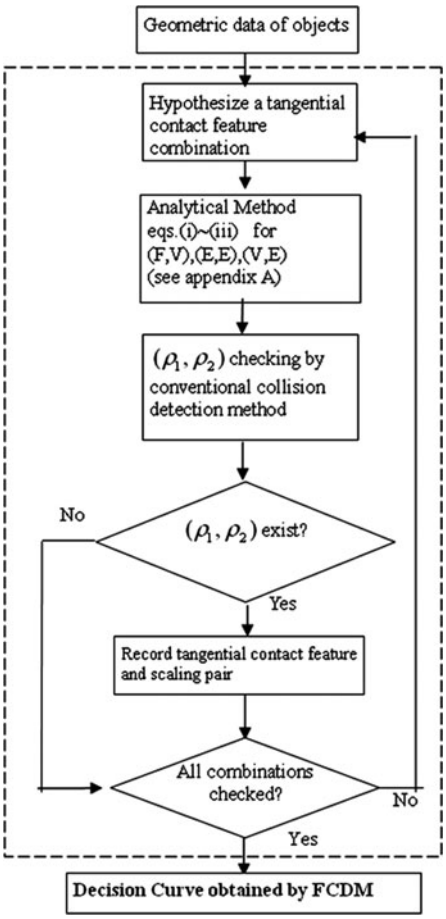


Fig. 5. Flowchart of FCDM.

Hence, the scaling pairs (ρ_1, ρ_2) for each combination can be determined. Then we test the feasibility of an iterative hypothesize-and-check procedure for scaling pairs of all combinations by applying analytical equations. The basic iterative search is performed as follows. A possible scaling pair of a tangential contact feature combination is selected, and the conventional collision detection algorithm (in this paper GJK) quickly verifies whether the candidate pair is a contact scaling pair. Therefore, the total number of repeat processes under the conventional collision detection method is based on the number of combinations, which will be much smaller than the number required under the bisection method. The hypothesis-and-check procedure is iterated until all combinations have been checked. All line segments can be constructed in this manner, and the intersections of the lines are the switching points.

3.4. Demonstrated computational advantages of FCDM

The computational complexity of building a decision curve depends on the time required for collision detection during the construction process. The analytical method can only identify a possible contact scaling pair and its associated tangential contact features. Meanwhile, conventional collision detection algorithms can easily determine whether two objects will collide when the scaling pair is known. However, if the user wants to know which scaling pair would make the two objects collide, it is necessary to employ an exhaustive search approach. As most exhaustive search approaches are based on the bisection method, which is resolution-dependent, the computation time complexity $O(n \times k)$ will increase if results that are more accurate are required, as shown in Eq. (3).

By contrast, FCDM identifies possible contact scaling pairs directly by analyzing the line segments (i.e., a continuum of contact scaling pairs) that make up the decision curve, not by considering every pair on the curve. The computation time required by FCDM to build a decision curve only depends on the number of vertices of the objects, as per Eq. (4), and it is independent of the resolution and the

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

search area. Furthermore, the resolution only depends on the programming language (C++, Matlab® . . . etc.), and it is usually less than 10^{-4} in our working examples.

Figure 6 and Table I show the empirical comparison of the time required for collision detection by FCDM and the bisection method. For a rational polyhedron object, α and β are about $4 \sim 2 \times 10^1$; therefore, points F1 $\sim$ F4 indicate the FCDM results of different (α, β) pairs, from the simplest condition F1(4,4) to the most complex F4(20,20). Lines B1 $\sim$ B4 depict the relationship between the time needed for collision detection and the resolution under the exhaustive search approach, where B1 represents the result of the search area $I_1 \times I_2 = [0, 1]^2$; B2 represents $I_1 \times I_2 = [0, 2]^2$; B3 represents $I_1 \times I_2 = [0, 3]^2$, and B4 represents $I_1 \times I_2 = [0, 4]^2$.

From Fig. 6, it is clear that FCDM only depends on the number of vertices of the objects, whereas the bisection method depends on the resolution and the search area. Considering the most complex condition of FCDM, i.e., $(\alpha, \beta) = (20, 20)$ (point F4 in Fig. 6), the time required for collision detection operations is 3.8×10^3. However, if we apply the bisection method to the simplest state, i.e., the unit search area $I_1 \times I_2 = [0, 1]^2$ (line F1 in Fig. 6), and the same number of collision detection operations, then the resolution would be $\Delta_1 = \Delta_2 = 2.5 \times 10^{-3}$. Hence, FCDM achieves a better resolution (10^{-4} vs. 2.5×10^{-3}). In addition, under the bisection based on the same resolution ($I_1 \times I_2 = [0, 1]^2$, $\Delta_1 = \Delta_2 = 10^{-4}$), the number of collision detection operations is 1.4×10^5, which is much higher than under the FCDM method (1.4×10^5 vs. 3.8×10^3).

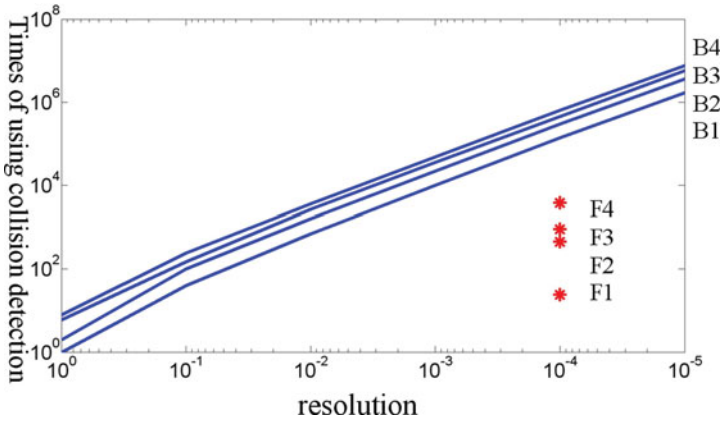


Fig. 6. The number of collision detection operations when building the decision curve versus the resolution of the curve. Computation time of FCDM does not depend on the resolution of discretization of scaling factors, thus allowing as accurate the determination of collision status as needed. B1 $\sim$ B4: conventional bisection method. F1 $\sim$ F4: FCDM.

Table I. Comparison of the number of collision detection operations required to build a decision curve (also see Fig.6).

(a) Number of collision detection operations by Conventional collision detection method with the bisection method

	Search Area			
Resolution	$[0,1]^2$	$[0,2]^2$	$[0,3]^2$	$[0,4]^2$
1	1	2	6	8
10^{-1}	4.0×10^1	1.0×10^2	1.5×10^2	2.4×10^2
10^{-2}	7.0×10^2	1.6×10^3	2.7×10^3	3.6×10^3
10^{-3}	1.0×10^4	2.2×10^4	3.6×10^4	4.8×10^4
10^{-4}	1.4×10^5	3.0×10^5	4.5×10^5	6.4×10^5
10^{-5}	1.7×10^6	3.6×10^6	5.7×10^6	7.6×10^6

(b) Number of collision detection operations by FCDM. F: vertices of polyhedra ($\alpha \times \beta$)

	Vertices			
Resolution	F1:(4× 4)	F2:(10× 10)	F3:(10× 20)	F4:(20× 20)
10^{-4}	2.4×10^1	4.5×10^2	9.0×10^2	3.8×10^3

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

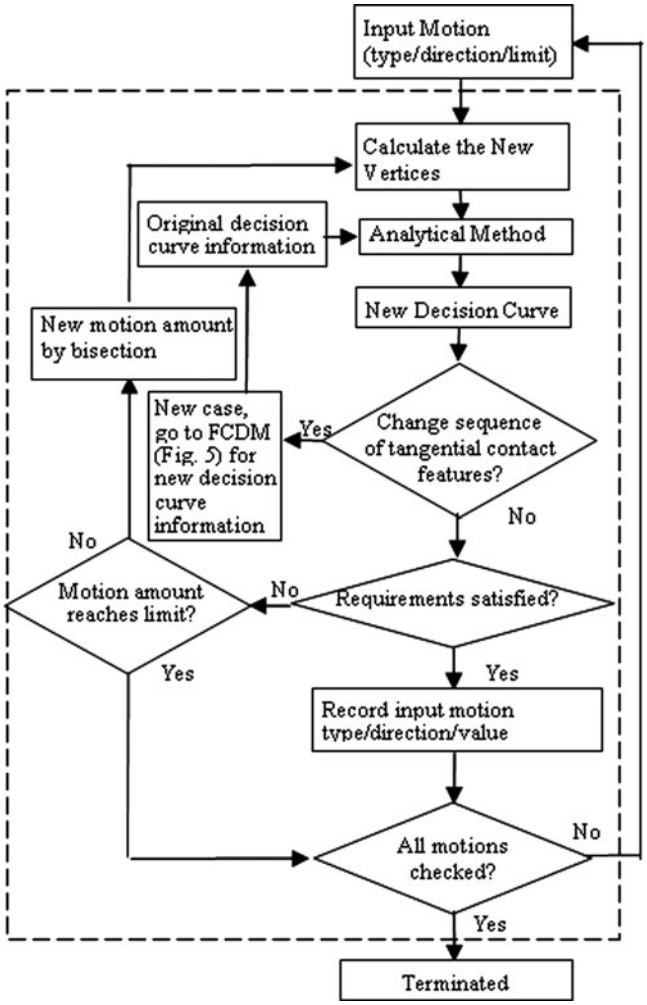


Fig. 7. Flowchart of FCDM with relative motion.

In the unit search area, given the same number of collision detection operations, FCDM yields a (three orders of magnitude) better resolution than the bisection method; and given the same resolution, FCDM performs fewer (by three orders of magnitude) collision detection operations than the bisection method. Hence, FCDM is more efficient in building a decision curve than the conventional collision detection method implemented by the resolution-dependent bisection method in practical use.

4. Motion Design for Contact

The speedup and improved accuracy of FCDM for collision detection between scaled convex polyhedra are significant for applications where collision detection accuracy is particular importance, thus its potential usefulness as a motion design tool is of interest that is more practical. There is not a single set of motion specs (type, direction, limit, value) that will provide an accurate solution to the industrial design problems in which tangential contact features, relative motion, and scaling of objects are the key concerns. In this section, we show that FCDM is a flexible and useful tool that allows designers to systematically examine several possible solutions. In order to determine the relative motion of objects required by a specific motion design problem, we present a method for constructing a family of new decision curves after any type of relative motion. The flowchart of the method is shown in Fig. 7. If the sequence of tangential contact features is the same as that on the original decision curve, the motion is deemed within the coherent motion range. However, if the motion is outside the range, the sequence of tangential contact features will change, and we have to recalculate the features through FCDM (in Fig. 5) for the whole new decision curve. A designer only needs to modify the desired requirements based on the flowchart in Fig. 7 to solve different

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

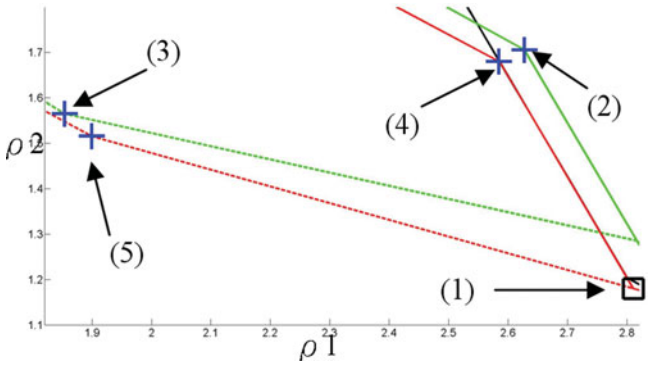


Fig. 8. Example I. □: original SW2; +: new SW V1E2 after motion (1) given the original decision curve of nominal design; new decision curve after motion; (2) RZ$= -0.8874°$; (3) RZ $= -3.4218°$; (4) TX $= -0.0528$; (5) TX $= -0.1597$.

design problems. For example, during a manufacturing process, if the expected tangential contact feature of two objects does not appear on the original decision curve, then FCDM can find a suitable relative motion condition that reorients two objects so that they contact with the expected feature. In order to demonstrate the efficacy of our approach, we consider two examples of Figs. 3 and 4 in the following. We have already addressed the precise assembly problems of two convex objects in previous works.[22] Here, we extend our method to solve the non-convex and hollow objects assembly problem. Example I involves assembling a convex object with a non-convex polyhedral object; and Example II comprises one convex polyhedral object inside a hollow polyhedral object. The objects can be decomposed into a few convex polyhedral parts.

A change in the objects motion type, initial configurations, and geometrical variation can result in different design problems. Based on the flowchart in Fig. 7, the solution to the motion design problem (what conditions would yield the desired mating configuration) should focus on the area outside the coherent motion range when confirming whether the new tangential contact features satisfy the user's requirement. The search requirements of motion design can thus be defined as follows:

Goal: Find the corresponding motions that would yield the new tangential contact feature required by the user.

Search target: The corresponding motion, including the type, direction, and value.

For this application, rather than specifying the dimensions precisely, the user instead uses a pair of approximate polyhedra. FCDM is easier to change the dimensions and the needed rigid body motion to reorient the objects to achieve the desired contact feature.

4.1. Example I: One convex and one non-convex polyhedral objects

For the non-convex object depicted in Example I, we assume that V1E2 (P1V5,P2V3,V7) is the desired new tangential contact feature for mating the two polyhedra. For this type of problem, there are several motion conditions that can achieve the goal. The user can choose the most suitable relative motion directly based on the scaling pair requirement. The following are two examples of motion conditions: (i) a rotation around the Z-axis, with a relative motion range Rot($Z = -0.8874° \sim -3.4218°$); and (ii) a translation along the X-axis, with Trans($X = -0.0528 \sim -0.1597$). The corresponding scaling pairs are derived from the range $\rho_1 = 1.8540$ (i.e., Rot($Z = -3.4218°$)) to $\rho_1 = 2.6275$ (i.e., Rot($Z = -0.8874°$)). The new induced decision curves are then constructed as shown in Fig. 8.

4.2. Example II: One convex and one hollow polyhedral objects

For the hollow box in Example II, we demonstrate another important application, namely, planning a path to remove the scalable object from the scalable box with an opening. In order to solve this narrow-passage problem, the original scaling pair $(\rho_1, \rho_2) = (1, 1)$ must be below the new decision curve after motion, i.e., in the non-overlapping area. Collision-free paths could be generated to remove the object, and Fig. 9 depicts one such path that ensures that collision-free scaling pairs exist at each designed waypoint on the path, and that each waypoint represents a feasible transition of spatial motion. At each waypoint, if we use GJK rather than the decision curve against the scaled configurations, a number of collision checks must be performed. By contrast, the feasibility of a path

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

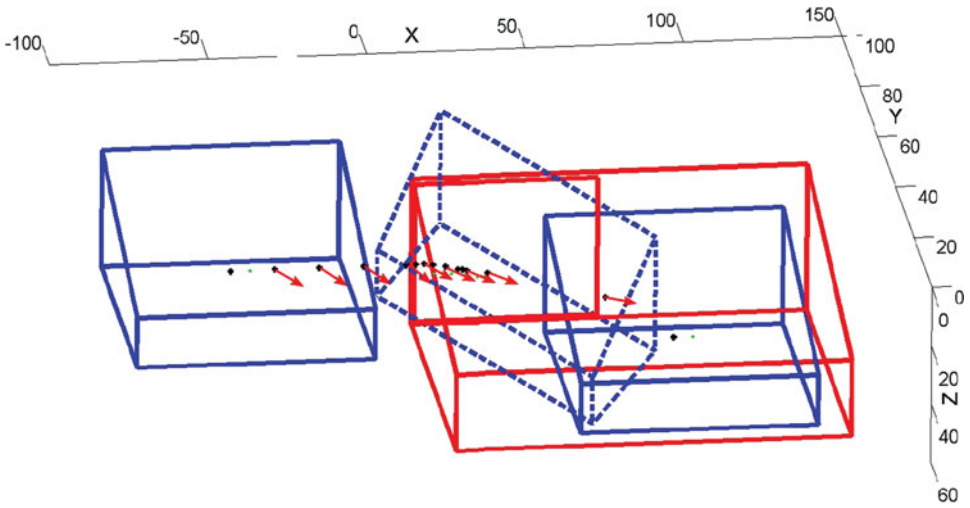


Fig. 9. Example II. The path for removing a convex polyhedron from a box with an opening. Each black point represents the loci of center-of-gravity of the convex object during removal from the inside of the box, and the arrow indicates the angle of rotation.

resulting from changes to one waypoint or a portion of waypoints is easy to perform in the plane of scaling factors using the decision curve.

The performed simulations of the tangential contact feature problem for non-convex objects and the path planning problem for hollow objects demonstrate that FCDM can deal with the (grown or shrunk) uncertainty of scaling factors, the new expected tangential contact features, and the corresponding motion designs. We deduce that our approach to detecting the collision of two scaled convex polyhedra can provide highly accurate result with less computational time, even with smaller deviations of scaling pairs or contact changes. Moreover, using FCDM, the user could easily find many possible solutions meeting the specified requirements, including the solutions unlikely to be found using conventional methods that rely on perturb of a known configuration.

5. Conclusions

In this paper, we extend the Hopcroft and Wilfong's[1] concepts on the contact and motion of rigid objects, and present a simulation-enabled method for efficiently detecting collisions between a pair of scaled convex polyhedra objects undergoing uniform scaling transformation and rigid body motion. We have devised a fast and accurate simulation method called FCDM based on building a piecewise linear curve – a decision curve – by using the analytical equation for contact maintenance/transition relations on the plane of scaling factors. Comparative simulations show that FCDM not only speeds ups the determination of the collision status and contact conditions for a pair of scaled convex polyhedra but is also very robust in that the method is less sensitive to wide variations in the relative object size. We provide two examples of demonstrating the usefulness of FCDM to designers to systematically examine a spectrum of possible solutions to solve practical problems in which the accuracy of collision detection algorithm is of particular importance, such as toleranced assembly problems when the size, motion, and contact of the objects are the main concerns.

Acknowledgments

The authors wish to thank Professor Masatomo Inui, Ibaraki University, Japan for many helpful discussions about tolerance assembly. The reviewers and editor-in-chief Prof. Greg Chirikjian helped improving the presentation of the paper. This work was supported by the National Science Council of the Republic of China under contracts NSC 94-2212-E-001-001, NSC 95-2212-E-001-001, and NSC 96-2221-E-001-018-MY2.

References

1. J. Hopcroft and G. Wilfong, "Motion of objects in contact," *Int. J. Robot. Res.*, **4**(4), 32–46 (1986).
2. G. van den Bergen, *Collision Detection in Interactive 3D Environments*, Morgan Kaufmann Series in Interactive 3D Technology (Morgan Kaufmann, Burlington, MA, 2004).

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

3. C. Ericson, *Real-Time Collision Detection* (Elsevier, Philadelphia, PA, 2005).
4. M. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," *Proceedings of the IMA Conference on Mathematics of Surfaces*, Vol. 1 (1998) pp. 602–608.
5. S. Cameron, "Enhancing GJK: Computing Minimum and Penetration Distance Between Convex Polyhedra," *Proceedings of the IEEE International Conference on Robotics and Automation* (1997) pp. 3112–3117.
6. A. Akgunduz, P. Banerjee and S. Mehrotra, "A linear programming solution for exact collision detection," *J. Comput. Inform. Sci. Eng.* **5**, 48–55 (2005).
7. Y. Kitamura, A. Smith, H. Takemura and F. Kishino, "A real-time algorithm for accurate collision detection for deformable polyhedral objects," *Presence* **7**(1), 36–52 (1997).
8. P. Jimenez, F. Thomas and C. Torras, "Collision Detection Algorithms for Motion Planning," **In**: *Robot Motion Planning and Control*, Lecture Notes in Control and Information Science, Vol. 229 (J.-P. Laumond ed.) Chapter 6. (1998) pp. 305-343.
9. W. Wang, Y. K. Choi, B. Chan, M. S. Kim and J. Wang, "Efficient collision detection for moving ellipsoids using separating planes," *Computing* **72**, 235–246 (2004).
10. E. Rimon and S. P. Boyd, "Obstacle collision detection using best ellipsoid fit," *J. Intell. Robot. Syst.* **18**, 105–126 (1997).
11. M.-Y. Ju, J.-S. Liu, S.-P. Shiang, Y.-R. Chien, K.-S. Hwang and W.-C Lee, "Fast and accurate collision detection based on enclosed ellipsoid," *Robotica* **19**, 381–394 (2001).
12. P. Jimenez, F. Thomas and C. Torras, "3D collision detection: A survey," *Comput. Graph.* **25**(2), 269–285 (2001).
13. C. Fares and Y. Hamam, "Collision Detection for Rigid Bodies: A State of the Art Review," *15th International Conference on Computer Graphics and Applications (GraphiCon'2005)*, Russia (2005).
14. M. C. Lin, "Efficient Collision Detection for Animation and Robotics," *PhD Thesis* (University of California, Berkeley, 1993).
15. K. Forbes, "Motion Curves: A Versatile Representation for Motion Data," *MS Thesis* (Graduate Department of Computer Science, University of Toronto, Crescent Toronto, Ontario, Canada, 2005).
16. C. J. Ong and E. G. Gilbert, "Growth distances: New measures for object separation and penetration," *IEEE Trans. Robot. Autom.* **12**(6), 888–903 (1996).
17. E. G. Gilbert, D. W. Johnson and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Trans. Robot. Autom.* **4**, 193–203 (1988).
18. M. Inui, M. Miura and F. Kimura, "Positioning conditions of parts with tolerance in an assembly," *Proceedings of the IEEE International Conference on Robotics and Automation* (1996) pp. 2202–2207.
19. M. Inui, "Tolerance Analysis of Mechanisms Using Dynamic, Simulation Method," *Nihon Kikai Gakkai Kanto Shibu. Seimitsu Kogakkai Ibaragi Koenkai Koen Ronbunshu*, Vol. 1999 (The Japan Society of Mechanical Engineers Kanto Branch, Japan, 1999) pp. 119–120.
20. Steven M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, UK, 2006).
21. J. Xiao and X. Ji, "Automatic generation of high-level contact space," *Int. J. Robot. Res.* **20**(7), 584–606 (2001).
22. W.-H. Pan, J.-S. Liu and W.-Y. Ku, "Fast collision detection method for the scaled convex polyhedral objects with relative motion," *Proceedings of the IEEE International Symposium on Assembly and Manufacturing*, Ann Arbor, Michigan (2007) pp. 184–190.
23. F. Yang and M. M. Marefat, "Achieving desired contact state transitions of polyhedral objects with compliant motions," *Proceedings of the IEEE International Conference on Robotics and Automation* (2005) pp. 4206–4211.
24. L.-W. Tsai, *Robot Analysis* (John Wiley, New York, 1999).
25. B. B. Goeree, E. D. Fasse and M. M. Marefat, "Determining feasible contact state of pairs of spatial polyhedra," *Proceedings of the IEEE International Conference on Robotics and Automation* (2000) pp. 1396–1401.
26. B. B. Goeree, E. D. Fasse and M. M. Marefat, "Verifying contact hypotheses of planar polyhedral objects using penetration growth distance," *Robot. Comput. Integr. Manuf.* **17**, 233–246 (2001).
27. S.-M. Hong, J.-H. Yeo and H.-W. Park, "A fast procedure for computing incremental growth distance," *Robotica***18**, 429–441(2000).
28. Y.-K. Choi, X. Li, F. Rong, W. Wang and S. Cameron, "Determining the directional contact range of two convex polyhedra," *Comput. Aided Des.* **42**, 27–35 (2010).
29. C. Je, M. Tang, Y. Lee, M. Lee and Y. J. Kim, "PolyDepth: Real-time penetration depth computation using iterative contact-space projection," *ACM Trans. Graph.* **31**(1) Article 5, 1–14 (Jan. 2012).

Appendix: Proof of Eq. (3)

There are six distinct types of tangential contact features between two convex polyhedra[21]: vertex–vertex (TypeVV), vertex–edge (TypeVE), vertex–face (TypeVF), edge–edge (TypeEE, which can be further classified as edge–edge cross and edge–edge touch), edge–face (TypeEF), and face–face (TypeFF). Among them, TypeVF and edge–edge cross are single-point contact features; hence they are least constrained kinematically.[25,26] The TypeVE and TypeEF contact features are generic transitions between two consecutive contacts, i.e., TypeVF and non-collinear TypeFF contacts.[27]

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

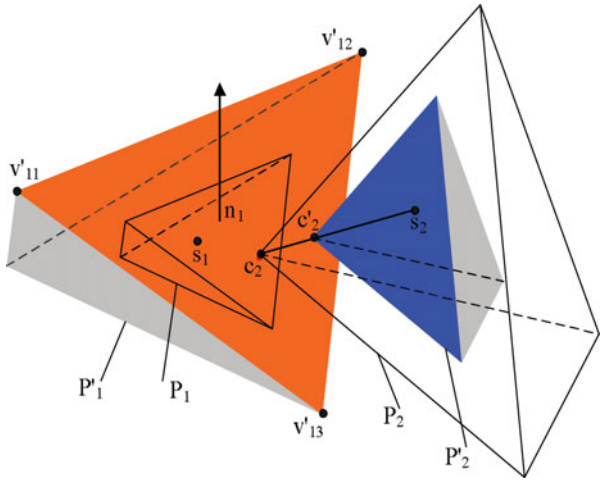


Fig. A1. Face–vertex contact of a pair of convex polyhedra before and after (colored) uniform scaling transformation.

The TypeVE contact feature often acts as the switching point for tangential contact feature changes.

Notations:

v_{ij}: The jth vertex points of polyhedron Pi, $i = 1, 2$
$\vec{n}_1$: The unit normal vector of the contact face of polyhedron P1
c_i: The contact point (vertex or projective point of s_i on the contact edge) of polyhedron Pi,
$c_i' = c_i + \rho_i \overrightarrow{d_i}$: the scaled position of c_i,
$\overrightarrow{d_i} = c_i - s_i$: the scaling direction of polyhedron Pi
dot (.,.): dot product
$D(x, y) = \|x - y\|$: The distance between the entities x and y (a point, an edge, or a face)

A1. Contact equations
(i) TypeFV contact

In Fig. A1, let v_{11}, v_{12}, v_{13} be the three vertices of the contact face F_1 of polyhedron 1, and let $\vec{n}_1$ be its unit outward normal. Then,

$$F_1 = \vec{n}_1 \cdot (p - v_{11}) = n_{1x}x + n_{1y}y + n_{1z}z + d_{v11} = 0$$

is scaled to

$$\begin{aligned} F_1' &= \vec{n}_1 \cdot (v - v_{11}') \\ &= n_{1x}x + n_{1y}y + n_{1z}z + d_{v11} \pm D(F_1, F_1') \\ &= 0, \end{aligned}$$

where the "± " sign denotes the expansion or shrinkage of F_1 as v_{11} is scaled to $v_{11}' = v_{11} + D(F_1, F_1')\vec{n}_1$.
We consider the plus sign (i.e., the enlarged polyhedron 1) in the following derivation.
Note that

$$\begin{aligned} D(F_1, F_1') &= (\rho_1 - 1)D(s_1, F_1), \\ \rho_1 &= D(s_1, F_1')/D(s_1, F_1), \\ \rho_2 &= D(s_2, c_2')/D(s_2, c_2) = t|d_2|/D(s_2, c_2). \end{aligned}$$

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

By substituting c_2' into F_1' and rearranging the equation, we get $c_2' = \rho_2(c_2 - s_2) + s_2$, and c_2' on F_1' when TypeFV is the contact feature. Therefore, $n_{1x}c_{2x}' + n_{1y}c_{2y}' + n_{1z}c_{2z}' + d_{a'1} = 0$, c_2' and a_1' can be rewritten in terms of ρ_1,ρ_2 as

$$\begin{aligned}&\rho_2 \text{dot}(\vec{n}_1, (c_2 - s_2)) + (n_{1x}s_{2x} + n_{1y}s_{2y} + n_{1z}s_{2z})\\ &= \rho_1 \times \text{dot}(\vec{n}_1, (v_{11} - s_1)) + (n_{1x}s_{1x} + n_{1y}s_{1y} + n_{1z}s_{1z}).\end{aligned}$$

As a result,

$$\rho_2 = \frac{\rho_1 \text{dot}(\vec{n}_1, (v_{11} - s_1)) + \text{dot}(\vec{n}_1, (s_1 - s_2))}{\text{dot}(\vec{n}_1, (c_2 - s_2))}.$$

Therefore, ρ_1,ρ_2 are related linearly.

Furthermore, the new contact point (vertex) is the scaled vertex on $P_2(\rho_2)$, $c_2' = c_2 + \rho_2\vec{d}_2$. The locus of the contact points is a line segment with tangent $\vec{d}_2$.

(ii) TypeEE contact

Let v_{11}, v_{12} be the end vertices of edge L_1, and let v_{21}, v_{22} be the vertices of edge L_2. Then,

$$\begin{aligned}L_1(t) &= v_{11} + t(v_{12} - v_{11}), t \in [0, 1]\,,\\ L_2(s) &= v_{21} + s(v_{22} - v_{21}), s \in [0, 1]\,.\end{aligned}$$

After scaling,

$$\begin{aligned}L_1'(t) &= v_{11}' + te_1', e_1' = v_{12}' - v_{11}',\\ L_2'(s) &= v_{21}' + se_2', e_2' = v_{22}' - v_{21}'.\end{aligned}$$

L_1', L_2' only intersect at one point if s, t exist and there is only one appropriate ρ_2 for the given ρ_1, i.e., $v_{11}' + te_1', e_1' = v_{21}' + se_2'$, or

$$\underbrace{\begin{bmatrix}-(v_{12} - v_{11} & v_{21} - s_2 & v_{22} - v_{21}\end{bmatrix}}_{A}\begin{bmatrix}t\\ \rho_2\\ s\rho_2\end{bmatrix} = s_1 - s_2 + \rho_1(v_{11} - s_1).$$

For notational simplicity, we define the following notations:

$$\begin{aligned}\det[VS] &\equiv \det[-(v_{12} - v_{11}), (s_1 - s_2), (v_{22} - v_{21})],\\ \det[VV] &\equiv \det[-(v_{12} - v_{11}), (v_{11} - s_1), (v_{22} - v_{21})],\\ \det[A] &\equiv \det[-(v_{12} - v_{11}), (v_{21} - s_2), (v_{22} - v_{21})],\\ \text{dot}(nvs) &\equiv \text{dot}(\vec{n}_1, (v_{11} - s_1)),\\ \text{dot}(nvs) &\equiv \text{dot}(\vec{n}_1, (s_1 - s_2)),\\ \text{dot}(ncs) &\equiv \text{dot}(\vec{n}_1, (c_2 - s_2)).\end{aligned}$$

There are two distinct cases.

(a) Edge–edge cross: If matrix A is non-singular and a unique intersection exists, then $\rho_2 = \frac{\det[\,-(v_{12} - v_{11})\ s_1 - s_2 + \rho_1(v_{11} - s_1)\ v_{22} - v_{21}\,]}{\det(A)}$, which shows that ρ_1, ρ_2 are linearly related because of the multilinearity of the determinant.

(b) Edge–edge touch: $\det(A) = 0$, and the two edges $v_{12} - v_{11}$, $v_{22} - v_{21}$ overlap, so the number of intersection points is infinite.

(iii) TypeVE contact

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

TypeVE acts as an instantaneous contact when tangential contact features change. There are three possible transition scenarios.

(a) In most situations, TypeVE acts as an instantaneous transition from the TypeFV feature to the TypeEE feature or *vice versa*. In this case, we have

$$\rho_2 = \frac{\rho_1 \mathrm{dot}(\vec{n}_1, (v_{11} - s_1)) + \mathrm{dot}(\vec{n}_1, (s_1 - s_2))}{\mathrm{dot}(\vec{n}_1, (c_2 - s_2))},$$

$$\rho_2 = \frac{\det[\, -\rho_1(v_{12} - v_{11}),\ s_1 - s_2 + \rho_1(v_{11} - s_1),\ v_{22} - v_{21}\,]}{\det(-\rho_1(v_{12} - v_{11}), v_{21} - s_2, v_{22} - v_{21})}.$$

Since $a_1 = v_{11}$, and L_1', L_2' intersect at c_2', then

$$\begin{aligned} c_2' &= \rho_2(c_2 - s_2) + s_2 = \rho_2(v_{21} - s_2) + s_2 + s\rho_2(v_{22} - v_{21}) \\ &= v_{11}' + te_1' = v_{21}' + se_2' = \rho_1(v_{11} - s_1) + s_1 + t\rho_1(v_{12} - v_{11}). \end{aligned}$$

Thus, the transition occurs at the single (ρ_1, ρ_2) pair given by

$$\rho_1 = \frac{\mathrm{dot}(\vec{n}_1, (c_2 - s_2)) \times \det[VS] - \mathrm{dot}(\vec{n}_1, (s_1 - s_2)) \times \det[A]}{\mathrm{dot}(\vec{n}_1, (v_{11} - s_1)) \times \det[A] - \mathrm{dot}(\vec{n}_1, (c_2 - s_2)) \times \det[VV]},$$

$$\rho_2 = \frac{\rho_1 \times \mathrm{dot}(\vec{n}_1, (v_{11} - s_1)) + \mathrm{dot}(\vec{n}_1, (s_1 - s_2))}{\mathrm{dot}(\vec{n}_1, (c_2 - s_2))}.$$

(b) TypeVE acts as an instantaneous transition between one TypeEE cross feature and another TypeE'E'cross feature.

In this case, we have

$$\begin{cases} \rho_2 = \dfrac{\det[\, -\rho_1(v_{12} - v_{11}),\ s_1 - s_2 + \rho_1(v_{11} - s_1),\ v_{22} - v_{21}\,]}{\det[-\rho_1(v_{12} - v_{11}), v_{21} - s_2, v_{22} - v_{21}]}, \\ \rho_2 = \dfrac{\det[\, -\rho_1(v_{12} - v_{11}),\ s_1 - s_2 + \rho_1(v_{11} - s_1),\ (v_{22} - v_{21})'\,]}{\det[-\rho_1(v_{12} - v_{11}), (v_{21} - s_2)', (v_{22} - v_{21})']}. \end{cases}$$

Combining the above two equations yields

$$\begin{aligned} &\frac{\det[-(v_{12} - v_{11}), \quad (s_1 - s_2) + \rho_1(v_{11} - s_1), \quad (v_{22} - v_{21})']}{\det[-(v_{12} - v_{11}), \quad (s_1 - s_2) + \rho_1(v_{11} - s_1), \quad (v_{22} - v_{21})']} \\ &= \frac{\det[-(v_{12} - v_{11}), \quad (s_1 - s_2) + \rho_1(v_{11} - s_1), \quad (v_{22} - v_{21})']}{\det[-(v_{12} - v_{11}), \quad (v_{21} - s_2)', \quad (v_{22} - v_{21})']}. \end{aligned}$$

Thus, the transition occurs at the single (ρ_1, ρ_2) pair given by

$$(\det[VS] + \rho_1 \det[VV]) \times \det[A'] = \left(\det[VS'] + \rho_1 \det[VV']\right) \times \det[A].$$

As a result,

$$\rho_1 = \frac{\det[VS'] \times \det[A] - \det[VS] \times \det[A']}{\det[VV] \times \det[A'] - \det[VV'] \times \det[A]},$$

$$\rho_2 = \frac{\det[-\rho_1(v_{12} - v_{11}),\ s_1 - s_2 + \rho_1(v_{11} - s_1),\ (v_{22} - v_{21})]}{\det[-\rho_1(v_{12} - v_{11}),\ (v_{21} - s_2),\ (v_{22} - v_{21})]}.$$

(c) (V,E) acts as an instantaneous transition between one (F,V) feature and another (F,V) feature denoted by the (F',V') feature

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press

Denote $\vec{n}_1'$, $(v_{11} - s_1)'$, $(s_1 - s_2)'$ as vectors of TypeF′V′ contact.
Then, equating

$$\begin{cases} \rho_2 = \dfrac{\rho_1 \times \text{dot}(\vec{n}_1, (v_{11} - s_1)) + \text{dot}(\vec{n}_1, (s_1 - s_2))}{\text{dot}(, (c_2 - s_2))}, \\ \rho_2 = \dfrac{\rho_1 \times \text{dot}(\vec{n}_1', (v_{11} - s_1)') + \text{dot}(\vec{n}_1', (s_1 - s_2)')}{\text{dot}(\vec{n}_1', (c_2 - s_2))} \end{cases}$$

yields

$$\frac{\rho_1 \text{dot}(\vec{n}_1, (v_{11} - s_1)) + \text{dot}(\vec{n}_1, (s_1 - s_2))}{\text{dot}(\vec{n}_1, (c_2 - s_2))} = \frac{\rho_1 \text{dot}(\vec{n}_1', (v_{11} - s_1)') + \text{dot}(\vec{n}_1', (s_1 - s_2)')}{\text{dot}(\vec{n}_1', (c_2 - s_2))},$$

$$\text{i.e.,} \ \frac{\rho_1 \text{dot}(nvs) + \text{dot}(nss)}{\text{dot}(ncs)} = \frac{\rho_1 \text{dot}(nvs)' + \text{dot}(nss)'}{\text{dot}(ncs)'}.$$

Thus, the transition occurs at the single (ρ_1, ρ_2) pair given by

$$\rho_1 = \frac{-\text{dot}(nss) \cdot \text{dot}(ncs)' + \text{dot}(nss)' \cdot \text{dot}(ncs)}{\text{dot}(nvs) \cdot \text{dot}(ncs)' - \text{dot}(nvs)' \cdot \text{dot}(ncs)},$$

$$\rho_2 = \frac{\rho_1 \times \text{dot}(\vec{n}_1, (v_{11} - s_1)) + \text{dot}(\vec{n}_1, (s_1 - s_2))}{\text{dot}(\vec{n}_1, (c_2 - s_2))}.$$

(iv) (E,F) contact

We consider two scenarios. (1) At least one vertex of the contact edge is on the contact face. This can be reduced to the (V,F) case by setting p_2 as the contact vertex. (2) The two vertices of the contact edge are outside the contact face, but the computation and linearity of ρ_1, ρ_2 are not affected. We only need to check whether the edge is still tangent to the face.

(v) (F,F) contact

Note that (F,F) contact only occurs when the pair of the closest features are two parallel faces. This case can be reduced to the (V,F) case by the following setting. Let p_2, which is the projection point of s_2 on the contact face of polyhedron 1, be the contact vertex; and let the unit normal $\vec{n}_1$ of the contact face of polyhedron 1 be the deforming direction of polyhedron 2. As a result, the deformed point p_2' may no longer be located at F_1'; however, it does not affect the computation and linearity of (ρ_1, ρ_2). We only need to check if the two faces are still tangent to each other.

(vi) (V,V) contact

To maintain a (V,V) contact under uniform scaling transformation, let $c_1 = c_2 = c$ be a given contact vertex. The scaled contact vertex $c_1' = c_2'$, or

$$s_1 + \rho_1(c_1 - s_1) = s_2 + \rho_2(c_2 - s_2),$$

$$c = \frac{\rho_1 - 1}{\rho_1 - \rho_2} s_1 + \frac{1 - \rho_2}{\rho_1 - \rho_2} s_2.$$

Thus, $c \in \overline{s_1 s_2}$, and the contact scaling pair (ρ_1, ρ_2) is related linearly. This can only happen for two specially chosen seedpoints, rather than a generic contact case involving scaled convex polyhedral objects.

https://doi.org/10.1017/S0263574714000939 Published online by Cambridge University Press