Fast and Reliable Contact Computations for Grasp Planning

Young J. Kim[1], Min Tang[1], Zhixing Xue[2] and Dinesh Manocha[3]

[1]Ewha Womans University, Seoul, Korea
[2]FZI, Karlsruhe, Germany
[3]University of North Carolina at Chapel Hill, U.S.A.

1. Introduction

Grasp planning is a key problem in robotics. One of the goals is to find suitable forces and torques for an object through contact points for a robotic hand to grasp that object. At a broad level, there are two classes of grasp planning algorithms that are used to determine the relationship between the contact points and finger joint positions. The forward methods simulate the motion of the hand and finger closing and involves use of collision detection methods to calculate the contact points. On the other hand, the backward methods first locate the contact points on the object surface and compute the feasible finger joint positions using inverse kinematics on the articulated model of the finger. The main drawback of the backward methods is that collisions between the hand and the environment during the inverse kinematics computation and finding collision-free configurations can be non-trivial. On the other hand, the forward method can easily find grasp configurations without any collisions with the environment. In earlier work, one of the major issues with respect to forward grasp planning methods was finding the contact points for multi-fingered robotic hands, as it was regarded as a time consuming computation and susceptible to robustness issues.

The problem of collision detection has been extensively studied in robotics, computer graphics and simulations over the past three decades. In particular, detecting collisions or contacts between static solid models is considered as a very well studied problem in the literature, and optimized algorithms and robust implementations are available. An extensive survey on the field can be found in [7]. When an object moves along some continuous trajectory, a conventional way of detecting a collision against obstacles is to sample the trajectory and check for collisions at discrete intervals, also known as *discrete collision detection (DCD)*. However, without high sampling rates, the DCD method may miss a collision.

More recently, collision checking techniques based on *continuous collision detection (CCD)* have been developed to cope up with the collision miss problem for objects under motion. In this setting, the motion trajectory of an object is taken into account, and if a collision does occur during the trajectory, the first time of collision (or contact) is calculated between the moving object and obstacles.

In this extended abstract, we give a brief survey of our recent work on devising efficient CCD algorithms for rigid, articulated models as well as deformable models. We also illustrate how we can successfully integrate the CCD algorithms into robotic grasping, so that the grasping algorithm can be very effective in terms of contact computations.

2. Continuous Collision Detection

Given a model $\mathcal{A}$ at the initial and final configurations in space, our CCD algorithm first interpolates the two configurations with a linear motion trajectory in the configuration space. Then, the CCD algorithm detects whether a collision occurs during the entire motion.

If a collision does occur, our algorithm computes the first time of collision of $\mathcal{A}$ against all other obstacles in space.

We start this section by explaining a basic idea of our CCD algorithm based on conservative advancement (CA) for rigid, convex models and then extends it to articulated and deformable models later.

2.1. Rigid Models

Conservative advancement (CA) is a simple algorithm to calculate the first time of contact τ between two convex objects $\mathcal{A}$ and $\mathcal{B}$. The CA calculates a tight lower bound of τ by repeatedly advancing $\mathcal{A}$ toward an obstacle $\mathcal{B}$ by the time step size of Δt_i while avoiding collisions [1][2]. Here, Δt_i is calculated based on the minimum distance between $\mathcal{A}$ at time t and $\mathcal{B}$, $d(\mathcal{A}(t),\mathcal{B})$, and an upper bound μ of the motion of $\mathcal{A}$ projected onto the direction of $d(\mathcal{A}(t),\mathcal{B})$ (also see Fig. 1):

$$\Delta t_i \leq \frac{d(\mathcal{A}(t),\mathcal{B})}{\mu} \qquad (1)$$

The above procedure is iterated until $d(\mathcal{A}(t),\mathcal{B}) \leq \epsilon$. Then, the time of contact is $\tau = \sum \Delta t_i$.

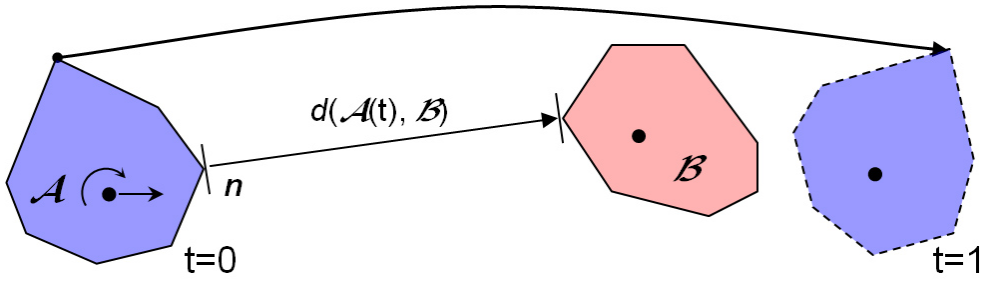


Figure 1. Conservative Advancement for Convex Models.

Eq. 1 works only for convex objects. However, in [3], we have extended the CA method to polygon-soup models by using bounding volume hierarchies (BVHs) based on swept sphere volumes (SSV) [4]. Since these SSVs are convex object, Eq. 1 is still applicable; the distance between SSVs can be easily obtained based on [4], and in [3], we have presented an efficient formula to compute the motion bounds for SSVs.

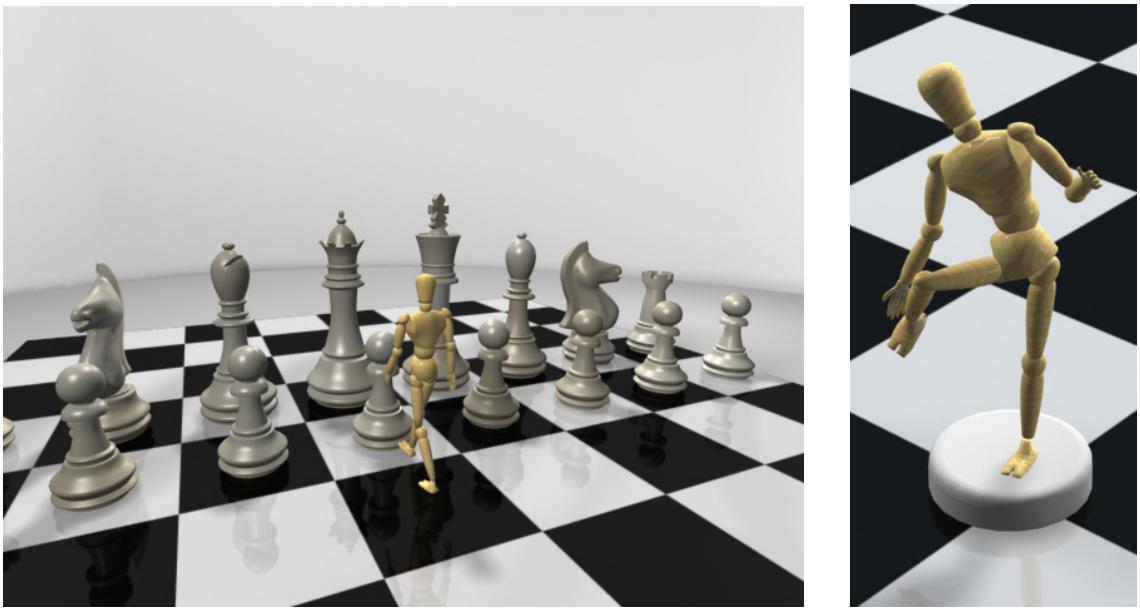
Moreover, in [3], we described a novel scheme, known as *controlled conservative advancement (C^{2}A)*, that automatically selects the depth of BVH traversal to speed up the CA computation. The main idea is that during the first few CA iterations, we do not compute the closest distance exactly, thus the depth of BVH traversal is adjusted to a small value. However, toward the final CA iterations, we compute exact distance utilizing the full BVH. In practice, our CCD algorithm takes 2.8 msec for rigid models consisting of 70K triangles.

2.2. Articulated Models

One can trivially extend the CA for rigid models to articulated models by treating the individual link in the articulated model as an independent rigid body. However, there are two issues for this naive approach. First of all, some of these links may not be colliding

during the motion. These are redundant links. Secondly, some links may be collided later than other links. These links are also redundant since we are interested in the first time of contact. We address the first problem by computing the tight bounding volume of a moving link based on the Taylor models and culling away redundant links. The second problem can be alleviated by sorting the time of contacts based on their estimates. The first technique is referred to as spatial culling and the latter as temporal culling [5].

In practice, our CCD algorithm can find the first time of contact between links (self-collision) as well as between a link and obstacles in 2.23 msec for an articulated model consisting of 20K triangles and 15 links against an environment consisting of 101K triangles, as shown in Fig. 2.



(a) Walk (b) Exercise

Figure 2. CCD for Articulated Models. For an articulated model consisting of 15 links and 20K triangles, and a chessboard environment consisting of 101K triangles, our CCD algorithm takes 2.23 ms (a) and 1.21ms (b).

2.3. Deformable Models

CA also can be used for deformable models based on local advancement [6]. The basic idea is still based on Eq. 1, but the underlying interpolatory motion is not in SE(3) any more. In this case, we use a linear line segment connecting deforming vertices in work space. The rest of points on the deforming model are linearly interpolated. Now, Eq.1 for deformable models boils down to computing the motion bound for primitives, feature pairs and bounding volumes. In [6], we have presented an efficient method to calculate these bounds. In practice, our CCD algorithm takes 792 msec for deformable models consisting of 252K triangles.

3. Application to Grasp Planning

Our CCD algorithms have been integrated into the widely used grasp simulator GraspIt! [8] to plan feasible grasps. The Schunk Anthropomorphic Hand with 17 joints is used for our

experiment and modeled as an articulated model with 18 links with 17 DOFs. The four identical fingers as a separate kinematic chain are attached to the palm. In the simulation, the hand as a predefined shape moves from a starting position along an approach direction towards the object. At the hand pose, where the first contact point is detected by our CCD algorithm, the finger closing process is performed. The finger joints are moved from the positions defined in the preshape to their maximal bending angles. As the CCD algorithm finds only the first contact point during this motion, multiple CCD calls are needed to find all the possible contact points. We move each finger separately to close the object. After a contact point is detected by the CCD algorithm, the finger link in contact is marked as "stopped". The rest of the finger links in the same kinematic chain as the colliding link are moved at the next continuous collision checking. After all the finger links are marked as "stopped", the object is enclosed by the hand. The detected contact points are used to build a 6D grasp wrench space to compute the maximum external disturbance force that the grasp can resist, which is evaluated as the grasp quality. Grasping forces are computed for stable grasps to grasp the object firmly.

The motions of hand movement and finger closing of the grasp planning are known a priori [9]. In the approach phase, the fingers in grasp preshape do not move, whereas the hand translates along an approach direction towards the object. In finger closing phase, the hand poses does not change. The finger joints are moved to the maximum bending angles. Based on this observation, we have improved the performance of contact computation for grasp planning. In more detail, since both the hand motion relative to the starting position and the finger motion relative to the palm are fixed, we can precompute the bounding box of the swept volume with respect to the motion. In case of hand motion, the swept volume of each finger link for the translational motion are computed. In case of finger closing, the swept volumes are recursively generated. The volume of the link is rotated by every affecting joints, so that the resulted swept volume represents the volume that can be reached by this finger link. These bounding boxes are shown in Fig. 3. If the object does not collide with these bounding boxes, it is unreachable for the hand and thus no further collision checking is performed.

We have intensively used the aforementioned CCD algorithm to find feasible grasps for more than 100 household objects [10] using different multi-fingered robotic hands. Some results of single-handed grasping, bimanual grasping, pick-and-place in a complex environment are shown in Fig. 4. On average, our modified GraspIt! algorithm with the CCD algorithms is about five time faster than the original algorithm with discrete collision detection.

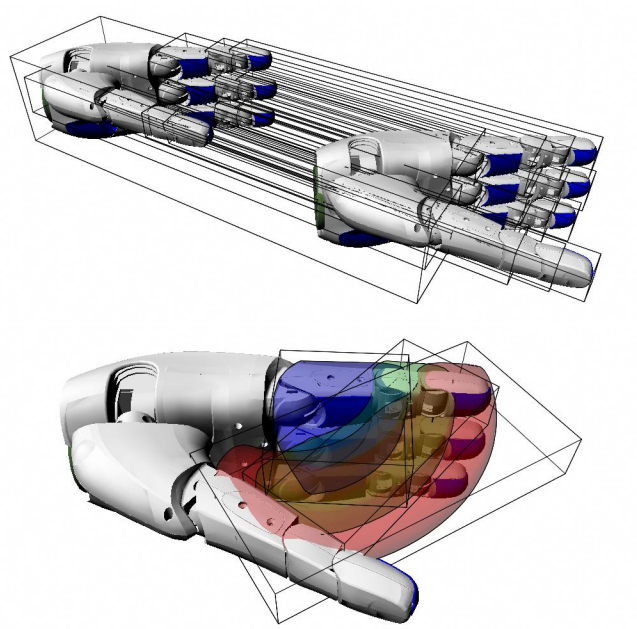
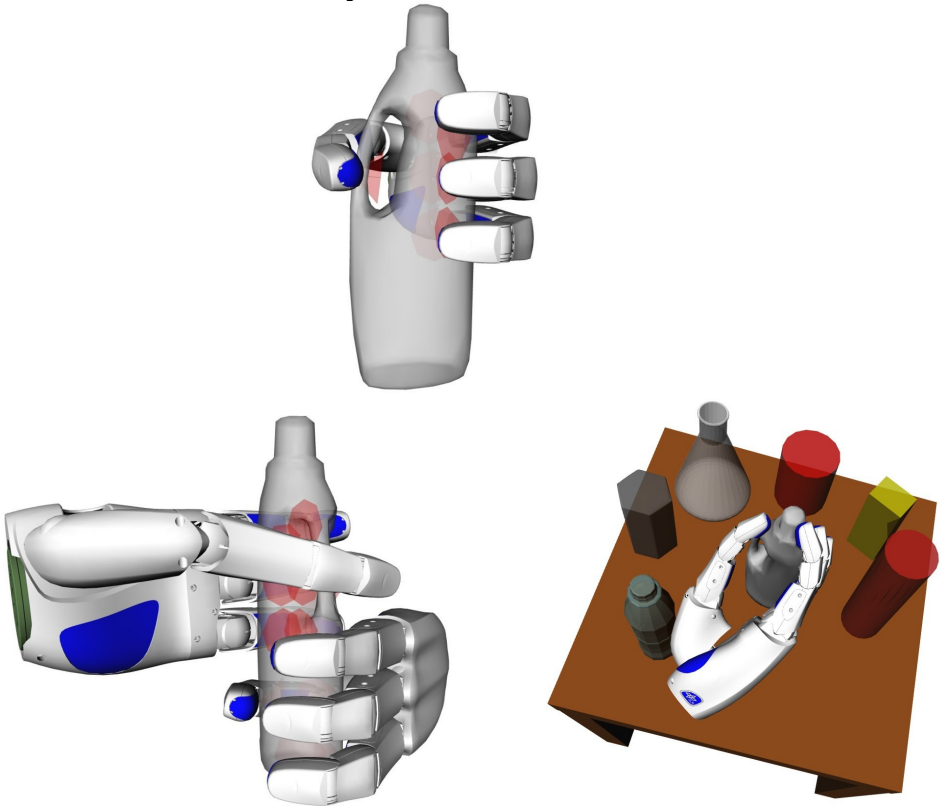


Figure 3. Bounding Boxes of Swept Volumes for Finger Links. These bounding boxes are used to cull the objects that the hand can not reach.



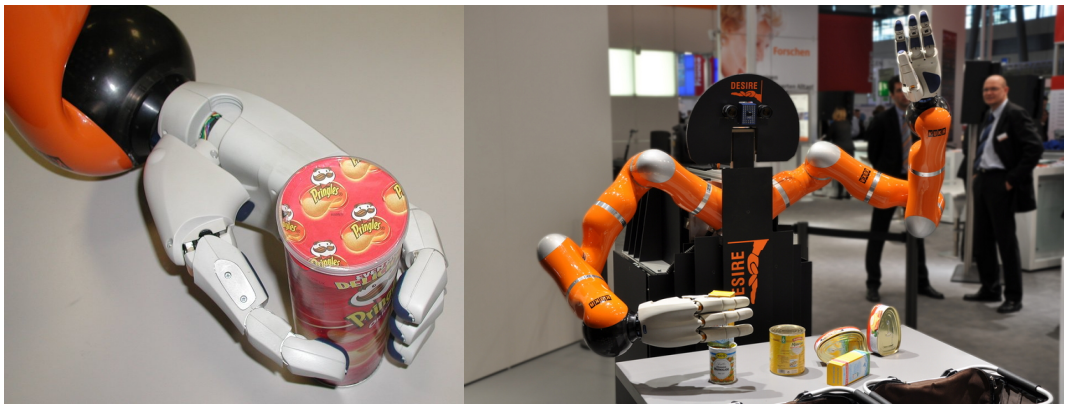


Figure 4. Virtual and Real Robotic Experimental Results of Planned Grasps with our CCD Algorithm.

4. Conclusions

We have presented efficient CCD algorithms for rigid, articulated and deformable models, and have showed how these algorithms can be used to accelerate grasp planning in terms of contact computations and stable grasps. We have measured the performance of our CCD and grasp planning algorithms applied to different objects. The integrated grasp planning algorithm considerably improves the state-of-the-art. In the future, we would like to extend our grasp planning algorithm to deformable objects including soft objects and cloth-like objects.

References

[1] M. C. Lin, "Efficient collision detection for animation and robotics," *Ph.D. dissertation*, University of California, Berkeley, CA, Dec. 1993.

[2] B. V. Mirtich, "Impulse-based dynamic simulation of rigid body systems," *Ph.D. dissertation*, University of California, Berkeley, 1996.

[3] M. Tang, Y. J. Kim, and D. Manocha, "C^{2}A: Controlled conservative advancement for continuous collision detection of polygonal models," *Proc. of IEEE Conference on Robotics and Automation*, 2009.

[4] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," Department of Computer Science, University of North Carolina, Tech. Rep. TR99-018, 1999.

[5] X. Zhang, S. Redon, M. Lee, and Y. J. Kim, "Continuous collision detection for articulated models using Taylor models and temporal culling," ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007), vol. 26, no. 3, 2007.

[6] M. Tang, Y. J. Kim, and D. Manocha, "Continuous Collision Detection for Non-rigid Contact Computations using Local Advancement," *Proc. of IEEE Conference on Robotics and Automation*, 2010.

[7] M. Lin, and D. Manocha, Collision and proximity queries. *In Handbook of Discrete and Computational Geometry*, 2003.

[8] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *Proc. of IEEE Robotics & Automation Magazine*, 2004.

[9] Z. Xue, P. Woerner, J.M. Zoellner, and R. Dillmann, "Efficient grasp planning using continuous collision detection," *Proc. of IEEE Conference on Mechatronics and Automation*, 2009.

[10] Z. Xue, A. Kasper, J.M. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," *Prof. of 14th Conference on Advanced Robotics*, 2009.