Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Adaptive Spatio-Temporal Refinement for Robust Triangle-SDF Continuous Collision Detection

Jong-Hyun Kim

jonghyunkim@inha.ac.kr

Inha University

Article

Keywords:

Posted Date: October 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11216743/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Adaptive Spatio-Temporal Refinement for Robust Triangle-SDF Continuous Collision Detection

Jong-Hyun Kim[1,*]

[1]College of AI and Convergence (Department of Artificial Intelligence, Design Technology), Graduate School of Electrical and Computer Engineering, Inha University, Michuhol-gu, Incheon 22212, South Korea
[*]corresponding author, jonghyunkim@inha.ac.kr

ABSTRACT

In this paper, we propose a robust adaptive spatio-temporal refinement method for continuous collision detection between triangle meshes and signed distance fields (SDFs). Existing Triangle–SDF continuous collision detection (CCD) methods can rapidly search for the time of impact (TOI) using *Frank–Wolfe with Golden Section Search (FWGSS)*; however, they may miss the earliest collision or compute an inaccurate TOI in the presence of complex rotational motion, high relative velocities, multiple temporal local minima, or abrupt changes in the SDF gradient. To address these limitations, our method uses the collision candidate computed by the original FWGSS as an initial solution and evaluates an SDF-based conservative lower bound together with spatial and temporal uncertainties for each spatio-temporal cell defined by a triangle region and a time interval. Cells identified as collision-free by the conservative lower-bound criterion are immediately pruned, while uncertain cells are selectively subdivided in either the triangle domain or the temporal domain by comparing their spatial and temporal uncertainties. Spatial refinement employs centroid-based triangle subdivision, whereas temporal refinement adaptively narrows the time interval to search for an earlier collision candidate. By avoiding uniform subdivision over the entire spatial and temporal domains and concentrating additional computation only on regions that require further examination, the proposed method improves collision-detection robustness while limiting the additional computational cost. The method supports both translational and rotational rigid-body motion and is combined with a bounding sphere hierarchy to perform efficient narrow-phase collision detection for complex triangle meshes. We further compare the proposed method with discrete collision detection, a conventional Frank–Wolfe-based CCD method, and FWGSS through a variety of experiments, including complex scenes involving simultaneous interactions among multiple rigid bodies. The results demonstrate that the proposed adaptive spatio-temporal refinement provides more reliable Triangle–SDF CCD for high-speed motion and challenging collision configurations while retaining the efficient search characteristics of the existing method.

Introduction

Collision detection is a fundamental computational component in physics-based animation, rigid and deformable body simulation, virtual environments, robotics, and interactive graphics systems. In particular, when rapidly moving or rotating objects are simulated with relatively large time steps, frame-based discrete collision detection (DCD) may miss collisions between two objects, which can cause objects to penetrate each other or pass directly through thin geometry, resulting in tunneling artifacts. To address this problem, continuous collision detection (CCD) considers object motion over the entire time interval and computes the first collision time, or time of impact (TOI), enabling more robust collision handling. A signed distance field (SDF) directly provides the signed distance from an arbitrary spatial position to the object surface and offers the advantage that the contact normal can be readily obtained from its gradient.

These properties allow SDFs to represent complex geometry efficiently and have made them effective for collision detection with triangle meshes. Existing Triangle–SDF DCD methods search for contact points inside a triangle by minimizing the SDF value in the triangle's barycentric coordinate space[1]. However, because these methods test collisions only at specific time instants, they can miss collisions under fast motion or with thin SDF geometry. More recently, Pelletier-Guénette et al. extended the Triangle–SDF collision problem to the temporal domain and proposed a continuous collision detection method based on *Frank–Wolfe with Golden Section Search (FWGSS)*[2]. Among recent methods, FWGSS is the most directly comparable Triangle–SDF CCD approach to our formulation because it operates on the same triangle–SDF collision representation and directly computes a continuous TOI. Accordingly, we use FWGSS as the primary recent baseline throughout our evaluation. This method alternately optimizes the triangle barycentric coordinates and the temporal variable to search for the earliest collision time and provides improved robustness over DCD for fast motion and complex rotations. After the first contact, it also performs centroid-based adaptive triangle subdivision to generate multiple simultaneous contact points within a large triangle and uses a bounding sphere hierarchy to reduce unnecessary narrow-phase tests. However, FWGSS is inherently an

iterative optimization method based on gradients and line searches, and its search result can be influenced by local structures in a complex spatio-temporal distance landscape.

This issue becomes particularly relevant when an object crosses an SDF boundary multiple times within a single time step, undergoes high-angular-velocity rotational motion, or when a large triangle passes through complex SDF geometry or regions with abrupt changes in the SDF gradient. Under these conditions, the spatio-temporal search domain can become highly complex. The existing method progressively shrinks the temporal upper bound using penetrations detected during the current optimization, but it does not directly evaluate whether the entire spatio-temporal domain before the candidate TOI has been sufficiently explored. Therefore, for more complex motions, it may be necessary to additionally explore regions that could contain an earlier collision candidate.

To address this issue, we propose a Triangle–SDF CCD method based on *adaptive spatio-temporal refinement*. The original FWGSS is used as an initial TOI candidate finder, and the triangle–time domain before the candidate is represented as spatio-temporal cells. Each cell is first tested using an SDF-based conservative lower bound, and the remaining cells are selectively subdivided only along the direction with larger spatial or temporal uncertainty. Thus, computation is concentrated on regions with unresolved collision status rather than uniformly subdividing the entire spatio-temporal domain.

The proposed method can be combined with the rigid-body motion model, bounding sphere hierarchy, and adaptive contact generation of the original FWGSS. In multi-rigid-body scenes using the same mesh, a single body-local SDF and acceleration structure are shared, and Triangle–SDF queries are performed through relative transformations to reduce additional preprocessing and memory costs. The main contributions of this paper are as follows.

- We propose an *adaptive spatio-temporal refinement framework* for examining the spatio-temporal region preceding the candidate collision time computed by FWGSS.
- We independently evaluate the spatial and temporal uncertainty of each spatio-temporal cell and propose a *direction-aware adaptive subdivision strategy* that selectively subdivides the direction with larger uncertainty.
- We present an efficient pruning method based on a conservative lower-bound test using the SDF distance and the spatial extent of a cell, which removes collision-free regions early and applies additional FWGSS searches only where necessary.
- We quantitatively compare DCD, Frank–Wolfe, the original FWGSS, and the proposed method in scenes involving fast translational motion, high-angular-velocity rotation, diverse SDF shapes, and interactions among multiple rigid bodies, evaluating accuracy, robustness, and computational performance.

Problem Statement

We consider the problem of detecting the earliest collision between a triangle mesh moving over the time interval $[t_0, t_1]$ and a signed distance field (SDF). Let the three vertices of a triangle at time t be $\mathbf{a}_t$, $\mathbf{b}_t$, and $\mathbf{c}_t$. An arbitrary point $\mathbf{x}_t$ inside the triangle can be expressed using barycentric coordinates (u, v, w) as follows.

$$\mathbf{x}_t = u\mathbf{a}_t + v\mathbf{b}_t + w\mathbf{c}_t, \qquad u, v, w \geq 0, \qquad u + v + w = 1. \tag{1}$$

The SDF $\phi(\mathbf{x})$ is 0 on the object surface, positive outside, and negative inside. Therefore, the minimum signed distance between the triangle and the SDF at time t is defined as

$$d(t) = \min_{\substack{u,v,w \geq 0 \\ u+v+w=1}} \phi\left(u\mathbf{a}_t + v\mathbf{b}_t + w\mathbf{c}_t\right) \tag{2}$$

The objective of Triangle–SDF continuous collision detection is to compute the first collision time, or time of impact (TOI), at which $d(t)$ first becomes less than or equal to 0 within the given time interval.

$$t_{\text{hit}} = \min\left\{t \in [t_0, t_1] \mid d(t) \leq 0\right\}. \tag{3}$$

This problem becomes particularly difficult under fast translational motion or high-angular-velocity rotational motion. If a triangle passes multiple SDF features or repeatedly crosses an SDF boundary within a single time step, $d(t)$ may contain multiple temporal local minima. In addition, for a large triangle or a complex SDF, multiple spatial collision candidates may exist simultaneously. Therefore, local optimization with a limited number of iterations may fail to robustly identify the earliest collision candidate. Accordingly, this work focuses on selectively re-exploring unresolved parts of the spatio-temporal domain before the candidate TOI.

Related Work

Continuous Collision Detection

Continuous collision detection (CCD), unlike discrete collision detection (DCD), which tests intersections only at discrete time instants, considers continuous object motion over a given time interval and computes the first collision time (time of impact, TOI).

Early rigid-body CCD studies proposed methods that directly considered polyhedral motion for efficient first-contact detection [3], an important topic in physics-based animation.

Brochu et al. proposed a CCD method for obtaining geometrically exact predicates even in floating-point environments [4], while Tang et al. improved the accuracy and computational efficiency of triangle-based CCD using sign classification of the Bernstein basis[5]. More recently, substantial effort has been devoted to systematically evaluating and guaranteeing the numerical robustness of CCD.

Wang et al. introduced a large-scale CCD benchmark and an inclusion-based algorithm and systematically analyzed numerical failures and degeneracies in existing CCD algorithms [6]. The same group later proposed an exact root-parity computation for cubic polynomials arising in CCD, reducing dependence on numerical parameters[7]. Zhang et al. formulated collision detection between curved shapes and curved paths as a sum-of-squares formulation for general curved surfaces and paths and thereby extended the range of collision detection[8]. Chen et al. proposed a time-dependent inclusion formulation for CCD between parametric surfaces and more efficiently restricted the temporal interval in high-dimensional spatio-temporal search [9]. These studies demonstrate the importance of conservative spatio-temporal bounds and efficient search strategies for accurate CCD.

CCD also plays an important role not only in collision detection but also in penetration-free simulation.

Li et al. introduced incremental potential contact (IPC), which integrates collision and elastic deformation within a unified variational framework to achieve intersection- and inversion-free simulation[10]. This framework was later extended to objects of different codimensions, including thin shells, rods, and particles, through C-IPC[11]. Whereas these methods include physical contact resolution after collision, our work focuses on the first-contact detection stage of Triangle–SDF CCD that can be applied independently of such simulation frameworks.

SDF-Based Collision Detection

A signed distance field (SDF) represents the signed shortest distance between a spatial point and an object surface and has been widely used in collision detection because it efficiently supports inside/outside tests and surface-normal evaluation.

Jones et al. provided a comprehensive survey of the construction, representation, and applications of three-dimensional distance fields [12], while Koschier et al. combined high-order polynomial approximation with spatial subdivision to improve accuracy and memory efficiency[13]. More recently, Pujol and Chica used an error-controlled adaptive grid and proposed an SDF-query acceleration method that concentrates computational resources in high-error regions rather than uniformly storing a high-resolution grid[14].

Robustly handling continuous surface primitives rather than point sampling remains an important issue in SDF-based collision detection. Xu and Barbič used CCD between points and SDFs to handle 6-DoF haptic contact for complex rigid-body geometry[15]. Macklin et al. extended this problem to the triangle domain and proposed a local-optimization-based Triangle–SDF DCD method[1]. Their method employs projected gradient descent, Frank–Wolfe, and golden-section search to produce more robust contact points than simple vertex sampling.

Recent studies have also investigated the use of SDFs as more general collision representations. Liu et al. proposed a real-time SDF–SDF collision detection method that uses interval computation and SDF gradients to locate intersection regions[16]. Wang et al. represented the swept volume of an arbitrarily moving shape implicitly as an SDF and applied it to continuous collision-free trajectory generation[17]. Montaut et al. further combined acceleration strategies with the GJK algorithm, widely used in conventional convex collision detection, to improve the convergence speed of distance and collision queries[18]. These studies show that both efficient collision queries and the combination of geometric representations with optimization strategies are important for real-time collision processing. SDF representations are also widely used in learning-based implicit geometry. DeepSDF demonstrated that complex three-dimensional geometry can be represented by learning a continuous signed distance function as a neural implicit representation[19]. Zesch et al. represented contact regions between triangle primitives using a six-dimensional neural collision field and proposed a learning-based collision representation that replaces traditional contact-point sampling [20]. These methods demonstrate the expressive power of implicit representations, but differ in objective and formulation from geometry-based Triangle–SDF CCD methods, such as ours, that directly compute the first collision time.

Triangle–SDF and Spatio-Temporal Collision Optimization

The work most directly related to ours is *Real-Time Triangle-SDF Continuous Collision Detection*, proposed by Pelletier-Guénette et al.[2]. This method extends the Triangle–SDF local optimization of Macklin et al. to the temporal domain and uses

Frank–Wolfe with Golden Section Search (FWGSS) to alternately optimize the triangle barycentric coordinates and temporal variable. When penetration is detected at the current iterate, the corresponding time is set as a new upper bound, excluding later times from the search interval, and spatial and temporal line searches are alternated to converge toward the first collision time. To handle cases in which a single triangle simultaneously contacts multiple SDF features at the first TOI, the method also performs centroid-based adaptive triangle subdivision to generate supplemental contact points. However, this spatial subdivision is mainly used *at the already computed TOI* to generate additional contact-manifold samples, rather than to adaptively subdivide the entire spatio-temporal domain before the candidate TOI to search for an earlier collision candidate. Our work differs from the original FWGSS in this respect. Specifically, while using the FWGSS result as an efficient initial TOI candidate, we partition the triangle–time domain before the candidate into multiple spatio-temporal cells and evaluate the spatial and temporal uncertainty of each cell to selectively refine only the required direction.

More recently, Nielsen and Erleben formulated CCD between two complex SDFs as alternating spatial-temporal optimization [21]. This method moves sparse surface samples along the tangent space of the SDF while alternating spatial and temporal optimization to directly compute the SDF–SDF TOI. This work is closely related to ours in that it accelerates CCD for complex implicit geometry through spatio-temporal optimization.

However, its underlying collision representation is *SDF–SDF*, and sparse surface points are used as optimization variables. In contrast, our method retains the *triangle–SDF* formulation and directly refines cells defined by the triangle spatial domain and temporal interval, selectively subdividing the direction with larger spatial or temporal uncertainty. Because the SDF–SDF formulation requires a different collision representation, optimization variables, and preprocessing pipeline, a direct numerical comparison would not provide an equivalent Triangle–SDF setting. We therefore discuss this method as closely related recent work rather than use it as a direct experimental baseline. Recent work has also continued to address contact handling for triangle primitives themselves. Kikuchi and Kanai unified various DCD contacts among vertices, edges, and triangles within a primitive-first formulation to handle complex self- and inter-object collisions[22]. Chen et al. computed shortest paths from the interior of an already self-intersecting mesh to its boundary to recover complex deformable contact[23]. Erleben systematically analyzed how the quality of mesh-based contact points and the accuracy of contact normals affect simulation results[24]. These studies show that collision detection requires not only intersection tests but also accurate contact locations and high-quality contact manifolds.

Adaptive Spatio-Temporal Triangle–SDF CCD

Overview

The proposed method uses the initial TOI candidate provided by FWGSS as an upper bound and adaptively re-explores only the preceding region. The overall procedure consists of five stages. First, FWGSS computes t_c; if no hit is found, we set $t_c = t_1$. Next, candidate triangles are extracted using the BSH over $[t_0, t_c]$. For each candidate, a cell of the form $\mathscr{T}_s \times [t_a, t_b]$ is constructed and collision-free cells are removed using a conservative lower bound, after which the remaining cells are selectively subdivided in either the spatial or temporal direction. Restricted FWGSS is applied only to uncertain cells that can no longer be subdivided to search for an earlier TOI. Thus, the proposed method follows a coarse-to-fine structure that combines fast initial collision-candidate search with selective refinement in the triangle–time domain. Rather than uniformly subdividing the entire search domain, it first eliminates collision-free regions using a conservative lower bound and concentrates additional computation only on cells whose collision status remains unresolved.

The overall procedure of the proposed adaptive spatio-temporal refinement framework is summarized in Figure 1.

Rigid Triangle Motion

Let the physical duration of one simulation step be ΔT, and define normalized time as $t \in [0,1]$. Linear velocity and world-space angular velocity are assumed constant within a simulation step. Let the initial center position of the rigid body be $\mathbf{g}_0$, its linear velocity be $\mathbf{v}$, and its angular velocity be ω. The center then moves as $\mathbf{g}(t) = \mathbf{g}_0 + t\Delta T\mathbf{v}$. Let the initial orientation be $\mathbf{R}_0$, and for $\|\omega\| > 0$, let $\hat{\omega} = \omega/\|\omega\|$. Then the position of a body-local point $\mathbf{q}$ at time t is given by

$$\mathbf{x}(\mathbf{q},t) = \mathbf{g}(t) + \mathbf{R}(\hat{\omega}, \|\omega\|\Delta T t)\,\mathbf{R}_0\mathbf{q}, \tag{4}$$

where $\mathbf{R}(\hat{\omega},\theta)$ denotes an axis–angle rotation. When $\|\omega\| = 0$, the rotational term is set to the identity transformation.

The instantaneous velocity of the point, $\mathbf{v}_x(t) = \mathbf{v} + \omega \times (\mathbf{x}(\mathbf{q},t) - \mathbf{g}(t))$, is used in the temporal search of FWGSS.

FWGSS Candidate Generation

The proposed method uses the original FWGSS[2] as the initial candidate finder without modification.

A point on a triangle is represented using the barycentric coordinates defined in Equation 1 and is denoted by $\beta = (u,v,w)$. In practice, we use the effective field that incorporates the collision margin δ, $\bar{\phi}(\mathbf{x}) = \phi(\mathbf{x}) - \delta$.

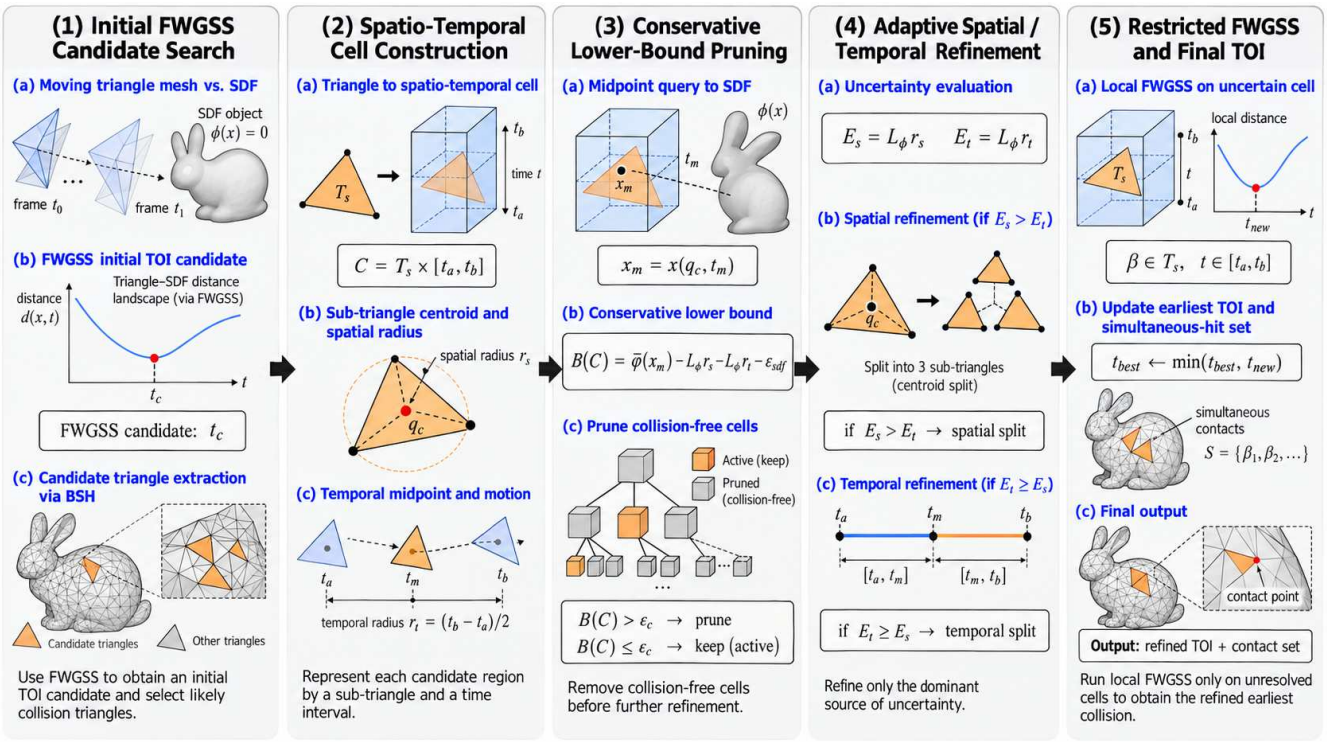

Figure 1. Overview of the proposed adaptive spatio-temporal Triangle–SDF CCD pipeline. (1) An initial time-of-impact (TOI) candidate is first obtained by FWGSS, and likely collision triangles are selected through the bounding sphere hierarchy (BSH). (2) Each candidate region is represented as a spatio-temporal cell composed of a sub-triangle and a time interval. (3) A conservative lower bound is evaluated at the temporal midpoint to prune collision-free cells. (4) For unresolved cells, the method compares spatial and temporal uncertainty and selectively performs either spatial or temporal refinement. (5) Restricted FWGSS is then applied only to the remaining uncertain cells to refine the earliest TOI and construct the simultaneous contact set.

FWGSS alternates spatial Frank–Wolfe updates and temporal golden-section searches, and shrinks the temporal upper bound when penetration is detected. We leave this baseline update unchanged and use the t_c returned by FWGSS not as the final TOI but as the initial upper bound for refinement. The margin and iteration tolerance used in the experiments are summarized in Table 1.

Spatio-Temporal Cell

Once FWGSS returns a candidate TOI t_c, we define the search region preceding the candidate as the following spatio-temporal cell:

$$\mathscr{C} = \mathscr{T}_s \times [t_a, t_b], \qquad t_0 \le t_a < t_b \le t_c \tag{5}$$

Here, $\mathscr{T}_s$ is either the full original triangle or a sub-triangle generated by adaptive subdivision.

Let the three barycentric vertices of $\mathscr{T}_s$ be $\beta_0, \beta_1, \beta_2$, and define the centroid as $\beta_c = (\beta_0 + \beta_1 + \beta_2)/3$. The corresponding body-local point is denoted by $\mathbf{q}_c$. The spatial extent of the sub-triangle is defined as the maximum distance from the centroid to its three vertices.

$$r_s = \max_{i \in \{0,1,2\}} \|\mathbf{q}_i - \mathbf{q}_c\|. \tag{6}$$

Because a rigid transformation preserves distances, r_s can be used over the entire time interval as a bound on the spatial deviation of any point inside the sub-triangle from the centroid trajectory.

Temporal Motion Bound

Let the temporal midpoint and half interval of a cell be $t_m = (t_a + t_b)/2$ and $h_t = (t_b - t_a)/2$. The translational displacement bound is $r_{\text{trans}} = \|\mathbf{v}\| \Delta T h_t$, and if the midpoint rotation amount is $\theta = \|\omega\| \Delta T h_t$, the rotational displacement bound of the

centroid is computed using the chord length as

$$r_{\text{rot}} = \begin{cases} 2\|\mathbf{q}_c\| \sin\left(\dfrac{\theta}{2}\right), & \theta < \pi, \\ 2\|\mathbf{q}_c\|, & \theta \geq \pi \end{cases} \tag{7}$$

Therefore, the temporal motion radius of the cell is

$$r_t = r_{\text{trans}} + r_{\text{rot}} \tag{8}$$

This bound allows the potential distance over the entire cell to be evaluated using a single representative SDF query at the temporal midpoint.

Conservative SDF Lower Bound

Assume that the SDF satisfies a Lipschitz constant L_ϕ,

$$|\phi(\mathbf{x}) - \phi(\mathbf{y})| \leq L_\phi \|\mathbf{x} - \mathbf{y}\| \tag{9}$$

where an ideal Euclidean SDF has $L_\phi = 1$. Let the centroid position at the temporal midpoint be $\mathbf{x}_m = \mathbf{x}(\mathbf{q}_c, t_m)$. Every point inside the cell then lies within at most $r_s + r_t$ of $\mathbf{x}_m$. We therefore define the lower bound of the signed distance inside the cell as

$$B(\mathscr{C}) = \bar{\phi}(\mathbf{x}_m) - L_\phi r_s - L_\phi r_t - \varepsilon_{\text{sdf}} \tag{10}$$

where ε_{sdf} accounts for the representation error of a discretized SDF. For this bound to be conservative, L_ϕ must be a valid Lipschitz upper bound for the SDF representation and ε_{sdf} must include the required approximation error. If $B(\mathscr{C}) > \varepsilon_c$, then $\bar{\phi}(\mathbf{x}) > 0$ throughout the cell, and the cell is classified as collision-free and pruned. For exact analytical SDFs, we use $L_\phi = 1$ and $\varepsilon_{\text{sdf}} = 0$; the numerical tolerance is listed in Table 1.

Spatial and Temporal Uncertainty

For a cell that cannot be eliminated by the lower-bound test, we define the spatial and temporal uncertainties as

$$E_s = L_\phi r_s, \qquad E_t = L_\phi r_t \tag{11}$$

The two terms defined above represent the spatial and temporal contributions to the lower bound in Equation 10, respectively. When both subdivisions are feasible, the refinement direction is selected as

$$\mathscr{R}(\mathscr{C}) = \begin{cases} \text{spatial subdivision}, & E_s > E_t, \\ \text{temporal subdivision}, & E_t \geq E_s, \end{cases} \tag{12}$$

Thus, only the direction that contributes more strongly to the uncertainty of the current lower bound is preferentially subdivided.

Adaptive Spatial and Temporal Refinement

If $E_s > E_t$, the current domain $\mathscr{T}_s$ is divided into three parts about its centroid. Letting the cell centroid be $\beta_g = \beta_c$, the three child domains are

$$\begin{aligned} \mathscr{T}_A &= (\beta_1, \beta_2, \beta_g), \\ \mathscr{T}_B &= (\beta_0, \beta_g, \beta_2), \\ \mathscr{T}_C &= (\beta_0, \beta_1, \beta_g) \end{aligned} \tag{13}$$

defined as above, and each child retains the same time interval $[t_a, t_b]$. Conversely, if $E_t \geq E_s$, the triangle domain is retained and only the time interval is bisected at the midpoint t_m.

$$\mathscr{C} \to \{\mathscr{T}_s \times [t_a, t_m], \mathscr{T}_s \times [t_m, t_b]\} \tag{14}$$

Temporal subdivision halves the interval width, thereby reducing both the translational and rotational motion bounds. Therefore, according to Equation 12, each cell is preferentially subdivided along the direction associated with the larger uncertainty, either spatial extent or temporal motion.

Terminal Cell and Restricted FWGSS

Spatial and temporal splitting stop according to the maximum-depth and minimum-size conditions in Table 1, while N_{cell} separately limits the number of cells that may be visited per triangle. A cell that cannot be eliminated by the lower-bound test but cannot be subdivided further in either direction because of the resolution criteria is defined as a *terminal uncertain cell*. For a terminal uncertain cell, the entire domain is not searched again. Instead, local FWGSS is applied only over the restricted domain $\beta \in \mathscr{T}_s$, $t \in [t_a, t_b]$. If an earlier hit is found, t_{best} is updated, and the upper bound of subsequently processed cells is clamped as $t_b \leftarrow \min(t_b, t_{\text{best}})$. Thus, as earlier hits are discovered, the subsequent search interval also shrinks. Handling of cases in which the cell budget is exhausted first is described separately in Section 'Termination and Resolution-Bounded Certification'.

Adaptive Spatio-Temporal Search Algorithm

Algorithm 1 summarizes the complete adaptive spatio-temporal search for a single triangle.

Mesh-Level Candidate Selection

For every triangle in a complex triangle mesh, applying Algorithm 1 would substantially increase the computational cost. Therefore, the proposed method directly reuses the BSH broad phase of the original FWGSS. Each BSH node has a local-space bounding sphere $\mathscr{B}_j = (\mathbf{c}_j, r_j)$ that moves under rigid transformation. To conservatively account for the deviation of a nonlinear rotational trajectory from the sphere's endpoint chord, we enlarge the radius using

$$p_{\text{nl}} = \left\| \mathbf{c}(t_m) - \frac{\mathbf{c}(t_a) + \mathbf{c}(t_b)}{2} \right\|, \qquad r_j^* = r_j + p_{\text{nl}} + \delta \tag{15}$$

To preserve conservativeness, if the angular displacement over a broad-phase time interval exceeds π, the interval is first subdivided, and the nonlinear padding of Equation 15 is applied within each sub-interval. BSH nodes with earlier potential contact times are processed first in the priority queue, and candidate triangles are likewise sorted by potential contact time. In the current implementation, for a static SDF scene, up to 12 triangles with the earliest potential contact times are selected for spatio-temporal refinement. The primary triangle returned by the original FWGSS is always included in this candidate set. If the number of candidates exceeds the configured budget and some triangles cannot be processed, the completion flag is set to false to indicate that a limited search budget was used. The complete mesh-level procedure is given in Algorithm 2.

Moving Rigid Bodies and Shared SDF

When multiple instances of the same rigid mesh collide with one another, it is unnecessary to generate a separate SDF for each object. In our implementation, a single body-local SDF is generated and shared by all rigid instances.

Let the source body be A and the target body be B. A source point is transformed into target-local coordinates as follows.

$$\mathbf{x}_{A\to B}(t) = \mathbf{R}_B^T(t)\,(\mathbf{x}_A(t) - \mathbf{g}_B(t)) \tag{16}$$

The collision query is then evaluated as $\phi_B(\mathbf{x}_{A\to B}(t))$. Therefore, the target SDF does not need to be regenerated in world space. The relative translation is bounded by $r_{\text{trans}}^{AB} = \|\mathbf{v}_A \Delta T_A - \mathbf{v}_B \Delta T_B\| h_t$. The angular displacements of the source and target are $\theta_A = \|\omega_A\| \Delta T_A h_t$ and $\theta_B = \|\omega_B\| \Delta T_B h_t$, respectively. The two rotational contributions $r_{\omega A}$ and $r_{\omega B}$ are computed by applying the chord bound of Equation 7 to the radii $\|\mathbf{q}_c\|$ and the midpoint target-local radius $\|\mathbf{x}_{A\to B}(t_m)\|$, respectively. The temporal radius of the relative-motion cell is therefore

$$r_t^{AB} = r_{\text{trans}}^{AB} + r_{\omega A} + r_{\omega B}. \tag{17}$$

In our implementation, a trilinear grid SDF generated once from the Bunny mesh is shared by all Bunny instances. During SDF construction, the target mesh is assumed to be a closed mesh with consistent orientation.

The default grid resolution is 40, and padding of 0.32 m is added outside the model boundary. Assuming that the uniform grid consists of 1-Lipschitz distance samples, the axis-aligned difference between neighboring grid samples is at most the grid spacing. Therefore, each partial derivative of the trilinear interpolant is bounded by 1, and a conservative bound $\|\nabla\phi\|_2 \le \sqrt{3}$ can be used for the Euclidean gradient norm. Accordingly, we use $L_\phi = \sqrt{3}$ for the grid SDF, which is a bound on the interpolated field itself. In our grid-SDF experiments, the interpolated field itself is used as the collision geometry, and the additional representation error is therefore set to $\varepsilon_{\text{sdf}} = 0$. If certification with respect to the original triangle-mesh surface is required, the maximum geometric approximation error between the mesh and the grid SDF must be separately estimated and included in ε_{sdf}.

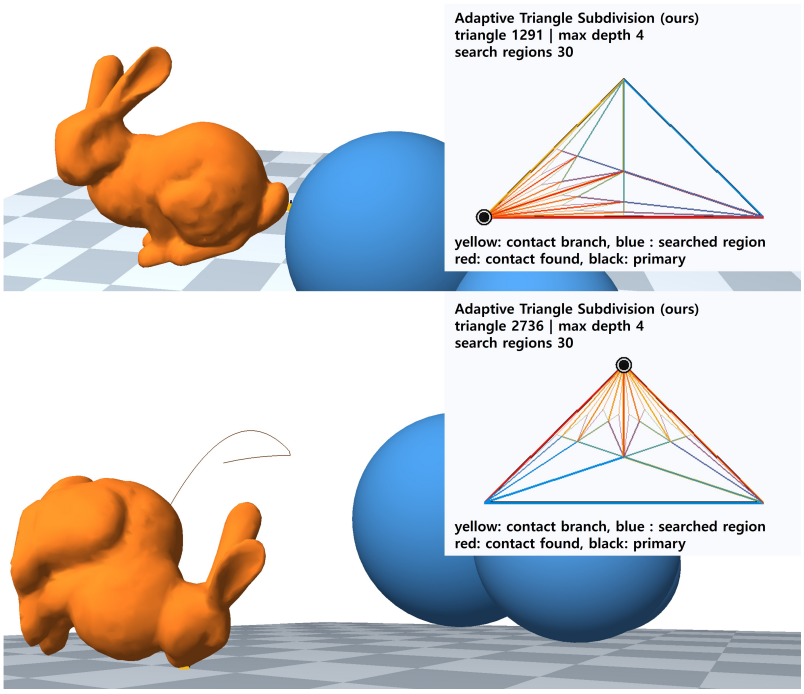

Figure 2. Examples of post-TOI adaptive contact subdivision. For each example, the left image shows the collision configuration, and the right inset visualizes the centroid-based subdivision used to identify supplemental simultaneous contacts after the primary TOI has been determined. Yellow denotes the contact branch, blue indicates searched regions, red indicates detected supplemental contacts, and black marks the primary contact.

Contact Generation

Spatio-temporal refinement is used only for the earliest-TOI search. After the final TOI has been determined, supplemental simultaneous contacts are generated using the centroid-based adaptive subdivision of the original Triangle–SDF method [2]. DCD is applied in sub-regions that do not contain the primary contact, and only regions in which additional contacts are found are recursively subdivided. The number of contacts and subdivision depth used in the current implementation are listed in Table 1. This post-TOI contact subdivision differs in purpose from the pre-TOI refinement in Section 'Adaptive Spatial and Temporal Refinement', where the former enriches the contact manifold and the latter subdivides the collision search domain before the candidate TOI.

Figure 2 visualizes representative post-TOI contact-subdivision trees. The left column shows the collision configurations, while the right column shows centroid-based triangle subdivision for selected triangles after the primary TOI has already been determined. Yellow denotes the contact branch, blue denotes searched regions, red indicates detected supplemental contacts, and black marks the primary contact. The visualization therefore illustrates contact-manifold enrichment and should be distinguished from the pre-TOI spatio-temporal refinement used to recover an earlier TOI.

Parameter Configuration

The main parameters used in the current implementation are summarized in Table 1. All scenes use meters (m) and seconds (s) as the base units, with a physics time step of $1/60\,\mathrm{s}$.

For scenes in which multiple rigid bodies interact simultaneously, pairwise queries are limited using the overrides in Table 2. Parameters not listed in the table use the settings given in Table 1.

Termination and Resolution-Bounded Certification

In this paper, "certification" means that an individual cell satisfying $B(\mathscr{C}) > \varepsilon_c$ can be conservatively determined to be collision-free under valid Lipschitz and SDF error bounds. The actual implementation uses finite depth, a minimum cell size, N_{cell}, and a mesh-level N_{tri} budget, and therefore does not subdivide all uncertain regions indefinitely.

An uncertain cell that reaches the resolution limit is handled by restricted FWGSS and is marked with $complete = false$. If N_{cell} is exhausted first, or if there are candidates exceeding N_{tri}, some regions remain unprocessed, and thus $complete = false$. In contrast, $complete = true$ indicates that, within the specified candidate domain, lower-bound pruning was completed without budget truncation or terminal FWGSS fallback, assuming a conservative broad phase and valid SDF bounds.

Parameter	Value	Description
ΔT	$1/60\,\mathrm{s}$	Physics time step
δ	$1.0\times10^{-3}\,\mathrm{m}$	Collision margin
N_{FWGSS}	56	Maximum FWGSS iterations
N_{GSS}	56	Maximum GSS iterations
$\varepsilon_{\mathrm{GSS}}$	1.0×10^{-8}	GSS tolerance
ε_t	1.0×10^{-7}	Temporal tolerance
$\varepsilon_{\mathrm{sim}}$	2.0×10^{-5}	Simultaneous TOI tolerance
ε_x	1.0×10^{-7}	Spatial tolerance
$\ell_t^{\max}$	7	Maximum temporal depth
$\ell_s^{\max}$	3	Maximum spatial depth
N_{cell}	768	Maximum cells per triangle
N_{tri}	12	Maximum refinement candidates
$\Delta t_{\min}$	1.25×10^{-2}	Minimum normalized interval
$r_s^{\min}$	$4.0\times10^{-3}\,\mathrm{m}$	Minimum spatial radius
$A_{\min}$	$1.0\times10^{-9}\,\mathrm{m}^2$	Minimum sub-triangle area
L_ϕ	1.0	Analytic SDF Lipschitz bound
$\varepsilon_{\mathrm{sdf}}$	0.0	Additional SDF error bound
ε_c	1.0×10^{-9}	Certification slack
d_{contact}	4	Contact subdivision depth
N_{contact}	3	Contacts per triangle

Table 1. Parameters used in the current implementation.

Parameter	Pairwise value
N_{tri}	1
N_{cell}	96
$\ell_t^{\max}$	5
$\ell_s^{\max}$	2
$\Delta t_{\min}$	0.025
L_ϕ	$\sqrt{3}$

Table 2. Pairwise overrides relative to Table 1.

Therefore, the overall method does not provide strict global certification; instead, it provides *resolution-bounded spatio-temporal certification*. The trade-off between accuracy and computational cost is controlled by budgets such as $\ell_t^{\max}$, $\ell_s^{\max}$, N_{cell}, and N_{tri}.

Computational Complexity

Let N_c denote the number of spatio-temporal cells visited for a single triangle, and let the number of terminal uncertain cells be N_u and the cost of one FWGSS evaluation be C_{FWGSS}. The cost of the proposed refinement can then be approximated as

$$O\left(N_c C_\phi + N_u C_{\mathrm{FWGSS}}\right) \tag{18}$$

where C_ϕ denotes the cost of one SDF evaluation. Because collision-free cells are removed using only a midpoint SDF query and bound evaluation, the number of restricted FWGSS calls, N_u, directly affects the total cost.

In the implementation, N_c is limited by N_{cell}, and at the mesh level, at most N_{tri} candidates are refined. Therefore, the additional computation is determined by the number of local candidates and unresolved cells that pass the broad phase rather than by the total number of triangles in the mesh. The smaller budget for pairwise queries is given in Table 2.

Experimental Results

Experimental Setup

The experiments were conducted using a fixed physics time step of $1/60\,\mathrm{s}$. Performance was measured on a Windows 11 desktop equipped with an AMD Ryzen 9 9950X3D CPU and 32 GB RAM, and all CCD timings were measured using a single CPU thread in a 64-bit Release build. OpenGL rendering and collision response times were excluded. Unless otherwise specified, the collision margin and refinement budget used the values in Table 1.

Method	Miss (%)	False Pos. (%)	TOI Error (μs)	Dist. Error (mm)	Complete (%)
DCD	14.92	0.08	N/A	N/A	N/A
FW	4.18	1.86	7.31	1.42	N/A
FWGSS	0.96	0.91	0.54	0.44	N/A
Ours	0.18	0.91	0.21	0.19	97.6

Table 3. Overall collision-detection accuracy over 50,000 trajectory queries. Miss and false-positive rates are evaluated over reference-hit and reference-no-hit queries, respectively. TOI and stopping-distance errors are reported only for methods that compute a continuous TOI.

For the accuracy and ablation analyses, 50,000 triangle–SDF trajectory queries were used. Of these, 25,000 were constructed to contain a collision in the reference, and the remaining queries were collision-free. The triangle size, initial pose, translational velocity, and angular velocity were varied using a fixed random seed, with the angular velocity range set to 0–24 rad/s. The SDFs were analytic spheres, boxes, and tori, with scale and relative pose varied to include smooth, grazing, and fast-crossing conditions. The denominators for the miss rate and false-positive rate are the reference-hit and reference-no-hit queries, respectively.

The reference TOI was computed offline using the same collision margin δ. First, during a dense temporal sweep with an interval of 1.0×10^{-6} s, the minimum effective SDF inside the triangle was evaluated with a high spatial iteration budget to identify the first contact interval, and the interval was refined using bisection/local search. Therefore, the reference is not a strict symbolic ground truth but a high-resolution numerical reference for experimental comparison.

The difficult-motion and single-body runtime experiments used the Stanford Bunny consisting of 4,968 triangles. Runtime was measured by repeating the same query sequence five times after warm-up and reporting the per-query mean and 95th percentile. In the multi-rigid-body experiments, multiple instances shared the same Bunny mesh and body-local grid SDF and used the relative-motion formulation of Section 'Moving Rigid Bodies and Shared SDF'. The grid resolution and pairwise budget followed the settings in Section 'Moving Rigid Bodies and Shared SDF' and Table 2.

The compared methods were end-of-step DCD, temporally modified Frank–Wolfe (FW), FWGSS[2], and the proposed method (Ours). FWGSS serves as the primary baseline because it is the most directly comparable recent Triangle–SDF CCD method and shares the collision representation and continuous-TOI formulation targeted by our refinement framework. DCD and FW are included as additional baselines to show the progression from discrete testing to continuous optimization. The same SDF, collision margin, trajectory, and broad-phase configuration were applied to all methods.

Overall Accuracy and Robustness

In the benchmark of Section 'Experimental Setup', we compared the missed collision rate, false-positive rate, TOI error, and stopping-distance error. For FW, FWGSS, and Ours, which return continuous TOIs, the physical-time TOI error was defined as

$$e_{\mathrm{TOI}} = \Delta T \, |t_{\mathrm{m}} - t_{\mathrm{ref}}| \tag{19}$$

and reported in μs. Because DCD does not return a TOI within the simulation step, it was excluded from the TOI and stopping-distance error comparisons.

In Table 3, the miss rates of DCD, FW, and FWGSS were 14.92%, 4.18%, and 0.96%, respectively, whereas that of Ours was 0.18%. The mean TOI error and stopping-distance error of Ours were 0.21 μs and 0.19 mm, respectively. Because the initial FWGSS hit is retained, the false-positive rate is the same as that of FWGSS at 0.91%. A total of 97.6% of all queries returned $complete = true$.

Robustness with respect to angular velocity is shown in Figure 3. At 24 rad/s, the miss rates of FWGSS and Ours were 2.80% and 0.65%, respectively. As rotational speed increases, the temporal motion bound becomes larger; however, the actual split direction is determined by comparing E_s and E_t for each cell, so spatial refinement can also occur.

Difficult Motion Cases

Fast Translation / Tunneling

In this scene, the Stanford Bunny consisting of 4,968 triangles was configured to rapidly cross the SDF boundary within a single simulation step. The blue Bunny in Figure 4 shows the current configuration, while the black, red, and green wireframes indicate the TOI poses of the reference, FWGSS, and Ours, respectively. DCD failed to detect the collision and therefore no pose is shown for it.

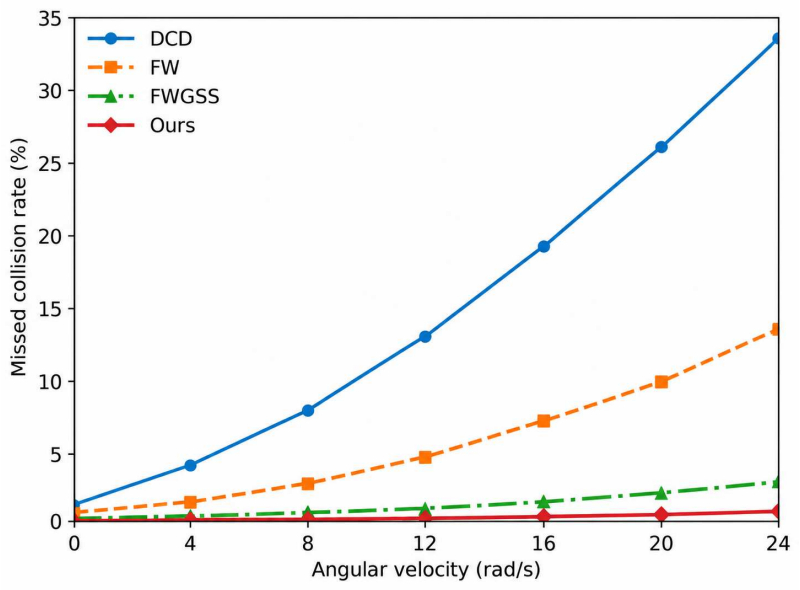

Figure 3. Missed collision rate with increasing angular velocity. The proposed method maintains a low miss rate as rotational motion becomes more challenging, while DCD, FW, and FWGSS show progressively larger degradation.

Method	Avg. time (ms)	95th perc. (ms)	Relative cost
DCD	0.21	0.34	0.31×
FW	0.39	0.62	0.57×
FWGSS	0.68	1.05	1.00×
Ours	0.96	1.47	1.41×

Table 4. Computational performance of the compared collision detection methods. The reported values represent CCD query time only and exclude rendering and collision response.

The reference TOI was 0.679064. DCD returned `MISS`, FW returned 0.721846, and FWGSS returned 0.690368, whereas Ours returned 0.679064, matching the reference. In this case, the delayed TOI was recovered by re-searching the unresolved interval preceding the candidate.

High Angular Velocity

This scene was configured so that a rapidly rotating Stanford Bunny passes between torus and sphere SDF features. Because different parts of the mesh follow different trajectories under rotation, this creates a case in which a single optimization trajectory may have difficulty finding the earliest contact. In Figure 5, the black, red, and green wireframes indicate the TOI configurations of the reference, FWGSS, and Ours, respectively. Because Ours and the reference return the same TOI, the two wireframes almost completely overlap.

The reference TOI was 0.393990. DCD detected the collision only at $t = 1.000000$, while FW and FWGSS returned 0.566214 and 0.498505, respectively. Ours returned 0.393990, recovering the reference earliest TOI. This query terminated within the specified budget in a `resolution-bounded` state.

Computational Performance

Table 4 compares CCD query time excluding rendering and collision response. The average time of Ours was 0.96 ms, approximately 41% higher than the 0.68 ms of FWGSS, and the 95th percentiles were 1.47 ms and 1.05 ms, respectively.

Scene-wise runtimes are shown in Figure 6. For regular motion, the difference between FWGSS and Ours is small, whereas in fast translation and high angular velocity, unresolved cells and restricted local FWGSS calls increase, resulting in greater additional cost.

The average runtime breakdown of the proposed method is summarized in Table 5. The initial FWGSS candidate search accounts for 70.8% of the total time, while lower-bound evaluation, cell management, and restricted local FWGSS account for the remaining cost.

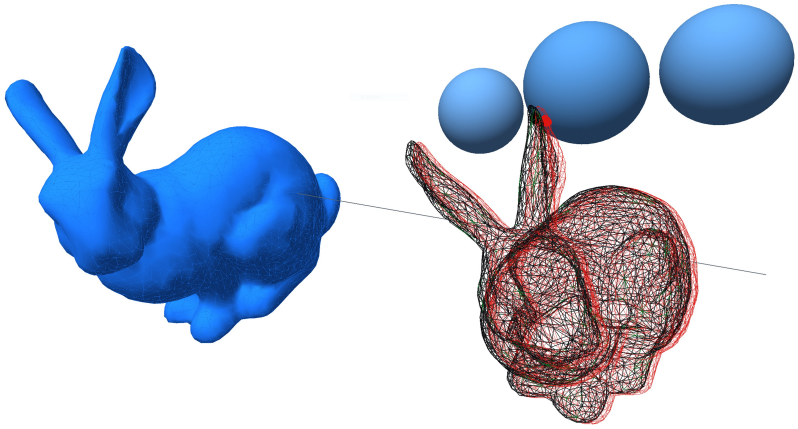

Figure 4. Fast translation / tunneling case using the full Stanford Bunny mesh. The shaded blue bunny shows the current configuration, while the wireframe bunnies indicate the time-of-impact (TOI) poses returned by different methods. The black wireframe denotes the reference TOI, the red wireframe denotes FWGSS, and the green wireframe denotes the proposed method. DCD misses the collision entirely, whereas the proposed method recovers the reference earliest TOI.

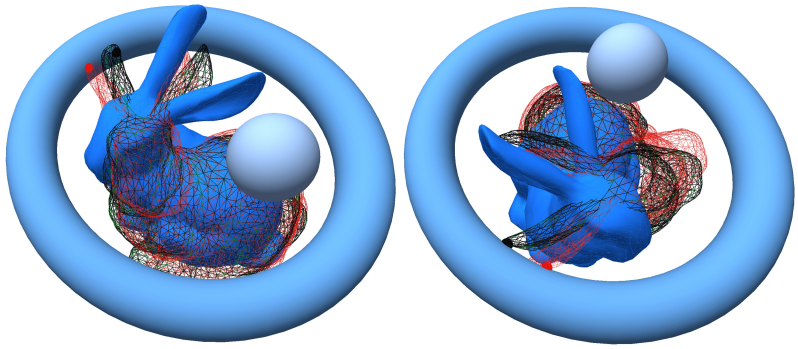

Figure 5. High-angular-velocity collision case using the full Stanford Bunny mesh, shown at two representative time instants. The shaded blue bunny represents the current rigid-body configuration, while the wireframe bunnies indicate the TOI configurations obtained by the reference solution, FWGSS, and the proposed method. The black wireframe denotes the reference TOI, the red wireframe denotes FWGSS, and the green wireframe denotes the proposed method. Because the proposed TOI coincides with the reference TOI, the green and black wireframes largely overlap.

Adaptive Refinement Analysis

Table 6 summarizes the refinement counts observed in the difficult-motion queries. The counts are cumulative values over all candidate triangles selected for refinement within each mesh-level query.

In Fast Translation, temporal and spatial splits occurred at similar frequencies, whereas both types of splits increased substantially in High Angular Velocity. This indicates that the split direction is not determined solely by motion type; rather, the spatial extent and temporal motion bound of each candidate cell jointly determine the refinement pattern. Because pruned cells are removed without additional optimization, restricted FWGSS is concentrated on terminal cells that remain unresolved by the lower-bound test. As specified in Algorithm 1, each terminal uncertain cell invokes one restricted FWGSS search, so these two counts are reported together in Table 6.

Ablation Study

To isolate the contributions of conservative lower-bound pruning, spatial refinement, temporal refinement, and uncertainty-based splitting, we performed an ablation study using the same trajectory set and cell budget. The Cells column in Table 7 denotes the average number of visited cells per triangle query.

Without lower-bound pruning, the budget is also consumed by collision-free cells, increasing the average number of cells and runtime, while the miss rate also slightly increases under the finite budget. Temporal Only and Spatial Only have lower computational cost but reduced accuracy because they cannot handle spatial or temporal uncertainty, respectively. Uniform Refinement recovers accuracy close to that of the Full Method but requires more cells and runtime. Therefore, uncertainty-

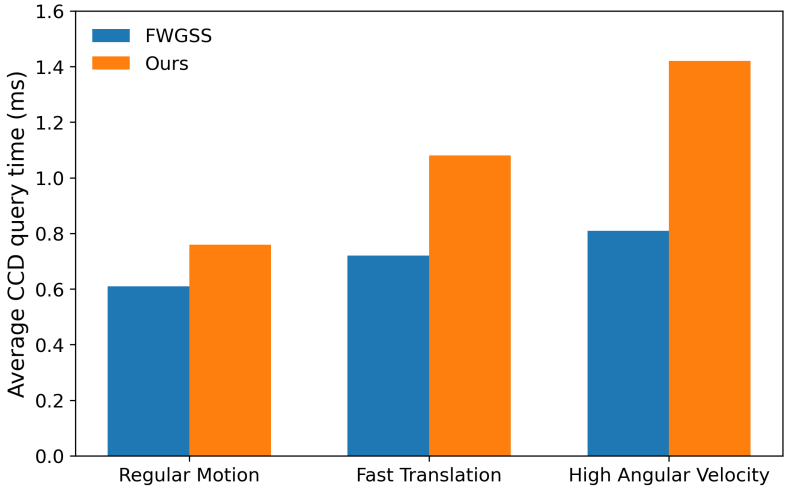

Figure 6. Average CCD query time for FWGSS and the proposed method under different motion configurations. Difficult motions require additional spatio-temporal refinement and therefore larger overhead.

Stage	Time (ms)	Fraction
Initial FWGSS candidate search	0.68	70.8%
Lower-bound evaluation / pruning	0.09	9.4%
Cell refinement and management	0.07	7.3%
Restricted local FWGSS	0.12	12.5%
Total	0.96	100.0%

Table 5. Average runtime breakdown of the proposed method.

directed splitting and pruning contribute not only to accuracy but also to allocating the given budget to regions where it is needed.

Multi-Rigid-Body Scalability

Two scenes were used in which multiple identical Stanford Bunny models move with independent poses and velocities. Bunny–Bunny collisions are handled using a shared body-local SDF and the relative-motion formulation of Section 'Moving Rigid Bodies and Shared SDF', and the narrow phase is performed only for pairs that pass the broad phase.

Figure 7(a) shows successive collisions among independently moving bodies, while Figure 7(b) shows a stress test in which the bodies converge toward the center and form dense contact. The average detector times for 4, 8, 12, and 16 bodies are shown in Figure 8. For both methods, the cost increases with the number of active pairs, while Ours incurs consistent additional overhead over FWGSS because of pairwise refinement.

Discussion

Accuracy–Efficiency Trade-off

The proposed method retains the candidate generation of FWGSS while applying additional computation only to unresolved regions preceding the candidate. Therefore, the overhead is not fixed but depends on the number of cells remaining after pruning and the number of restricted FWGSS calls. The increased computation for difficult motion results from examining a larger search domain to recover the earliest hit.

This structure has the advantage that accuracy and cost can be directly controlled through the refinement budget. Increasing the depth, N_{cell}, and N_{tri} allows more unresolved regions to be processed, but also increases runtime. Conversely, a smaller budget is advantageous for interactive workloads, but may increase the number of queries with `complete=false`. Therefore, the goal of the proposed method is not to provide a detector that is always faster than FWGSS, but to allocate additional computation only to queries that require it.

Scene	T-splits	S-splits	Pruned	Terminal / Local FWGSS
Fast Translation / Tunneling	120	119	211	158
High Angular Velocity	625	657	767	1182

Table 6. Adaptive refinement statistics aggregated over all refined candidate triangles within each mesh-level query. The final column reports terminal uncertain cells, each of which invokes one restricted local FWGSS search.

Configuration	Miss (%)	TOI err. (μs)	Time (ms)	Cells
w/o LB Pruning	0.24	0.27	1.43	742
Temporal Only	0.44	0.39	0.87	471
Spatial Only	0.71	0.62	0.83	454
Uniform Refinement	0.22	0.25	1.31	701
Full Method	0.18	0.21	0.96	512

Table 7. Ablation study of the proposed adaptive spatio-temporal refinement. TOI error is measured with respect to the reference earliest TOI.

Implications of Adaptive Refinement

Neither spatial nor temporal refinement is always dominant for a particular motion class. Translational and angular velocities increase E_t, whereas triangle size and local domain extent directly affect E_s. Across the full mesh, these two factors vary from triangle to triangle, so different split directions may be selected even within the same scene. This is consistent with Table 6, where both spatial and temporal splits occur in the fast translation and high-angular-velocity cases.

By combining lower-bound pruning with uncertainty-directed splitting, refinement is concentrated on cells for which collision cannot be excluded. From this perspective, rather than being a separate optimizer that replaces FWGSS, the proposed method selectively examines the possibility of earlier collisions using the existing candidate as an upper bound and can be viewed as a verification/refinement layer.

Limitations

First, because finite refinement depths and the N_{cell} and N_{tri} budgets are used, the method does not provide strict global certification. If restricted FWGSS is invoked for a terminal uncertain cell or budget truncation occurs, the query is recorded as `complete=false`. Thus, the certification in this work is defined with respect to the configured resolution and search budget.

Second, the conservativeness of the lower bound depends on a valid Lipschitz upper bound and the SDF approximation error. For grid SDFs, a bound appropriate for the employed interpolant is required, and if guarantees with respect to the original polygonal surface are required, the mesh–grid approximation error must be included in $\varepsilon_{\mathrm{sdf}}$. The broad phase must likewise be constructed conservatively so that no potential interval is missed.

Third, for rapid rotations, large triangles, and dense multi-body contacts containing many cells near the threshold, the number of restricted FWGSS calls can increase, leading to a larger worst-case workload. If the candidate budget is too small, a triangle with an earlier contact may be excluded from refinement, so the budget should be adjusted according to the application's accuracy requirements.

Finally, this work focuses on collision detection and TOI estimation, and the stability of collision response is outside the scope of this study. Future work may consider integration with contact response, deformable geometry, and extensions to non-rigid motion.

Conclusion

In this paper, we proposed an adaptive spatio-temporal Triangle–SDF CCD method that selectively re-searches the triangle–time domain preceding the FWGSS candidate. Collision-free cells are removed using a conservative lower bound, spatial or temporal refinement is selected according to E_s and E_t, and restricted FWGSS is applied only to terminal uncertain cells.

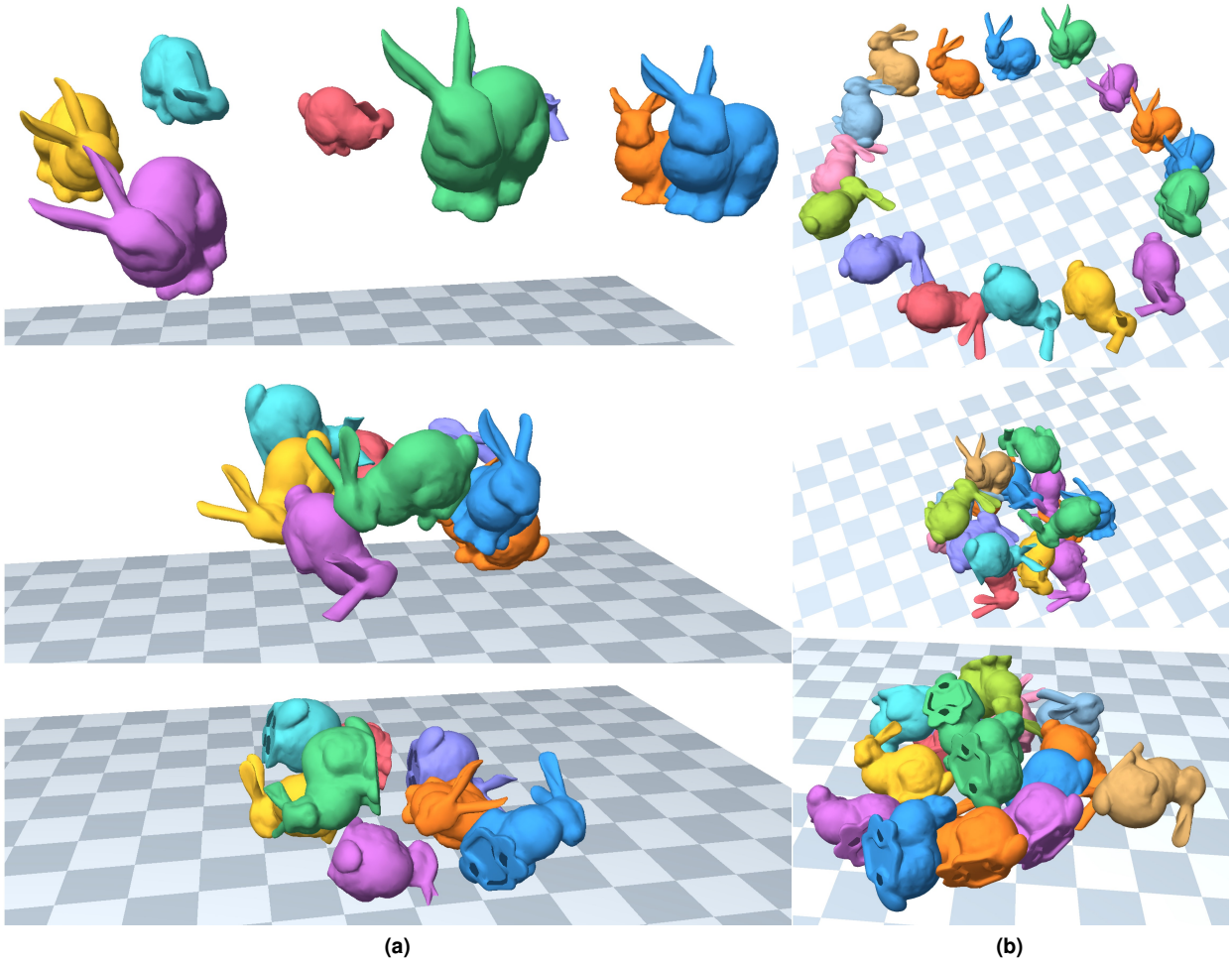

Figure 7. Multi-rigid-body collision examples using multiple Stanford Bunny models. (a) Independently moving bodies undergoing successive pairwise collisions. (b) A radial stress test in which multiple bunnies move toward the center and form a dense contact configuration.

The experiments evaluated recovery of earlier TOIs under fast translation and high angular velocity, the computational characteristics of selective refinement, ablation, and multi-rigid-body scalability. The proposed method requires additional runtime, but concentrates this cost on unresolved regions and provides resolution-bounded certification under a finite budget. Future work will investigate tighter and formally certified broad-phase bounds and extensions to deformable/non-rigid geometry.

References

1. Macklin, M. *et al.* Local optimization for robust signed distance field collision. *Proc. ACM on Comput. Graph. Interact. Tech.* **3**, DOI: 10.1145/3384538 (2020).

2. Pelletier-Guénette, J., Mercier-Aubin, A. & Andrews, S. Real-time Triangle-SDF continuous collision detection. *Proc. ACM on Comput. Graph. Interact. Tech.* **8**, DOI: 10.1145/3747862 (2025).

3. Redon, S., Kheddar, A. & Coquillart, S. Fast continuous collision detection between rigid bodies. *Comput. Graph. Forum* **21**, 279–287, DOI: 10.1111/1467-8659.t01-1-00587 (2002).

4. Brochu, T., Edwards, E. & Bridson, R. Efficient geometrically exact continuous collision detection. *ACM Transactions on Graph.* **31**, DOI: 10.1145/2185520.2185592 (2012).

5. Tang, M., Tong, R., Wang, Z. & Manocha, D. Fast and exact continuous collision detection with Bernstein sign classification. *ACM Transactions on Graph.* **33**, DOI: 10.1145/2661229.2661237 (2014).

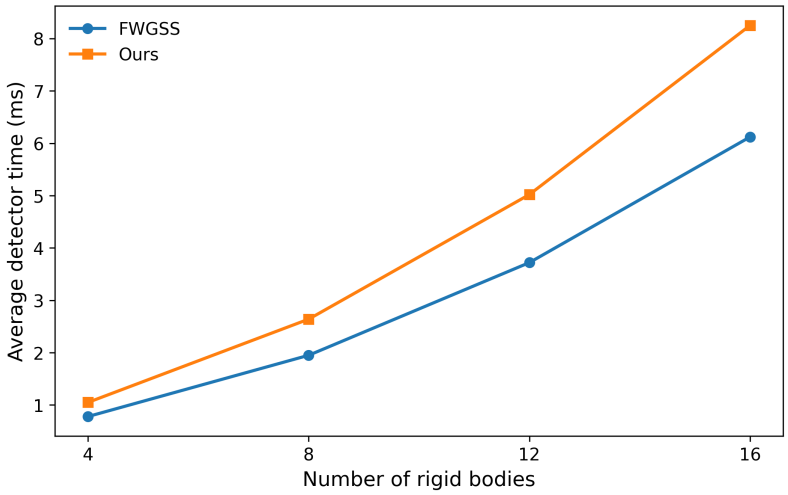

Figure 8. Average collision-detection time as the number of rigid bodies increases. Both FWGSS and the proposed method exhibit increasing cost as more potential body pairs become active. The proposed method adds pairwise refinement overhead while preserving a similar scalability trend.

6. Wang, B. *et al.* A large-scale benchmark and an inclusion-based algorithm for continuous collision detection. *ACM Transactions on Graph.* **40**, DOI: 10.1145/3460775 (2021).
7. Wang, B. *et al.* Fast and exact root parity for continuous collision detection. *Comput. Graph. Forum* **41**, 355–363, DOI: 10.1111/cgf.14479 (2022).
8. Zhang, P., Marschner, Z., Solomon, J. & Tamstorf, R. Sum-of-squares collision detection for curved shapes and paths. In *ACM SIGGRAPH 2023 Conference Proceedings*, DOI: 10.1145/3588432.3591507 (Association for Computing Machinery, 2023).
9. Chen, X. *et al.* A time-dependent inclusion-based method for continuous collision detection between parametric surfaces. *ACM Transactions on Graph.* **43**, DOI: 10.1145/3687960 (2024).
10. Li, M. *et al.* Incremental potential contact: Intersection- and inversion-free, large-deformation dynamics. *ACM Transactions on Graph.* **39**, DOI: 10.1145/3386569.3392425 (2020).
11. Li, M., Kaufman, D. M. & Jiang, C. Codimensional incremental potential contact. *ACM Transactions on Graph.* **40**, DOI: 10.1145/3450626.3459767 (2021).
12. Jones, M. W., Bærentzen, J. A. & Šrámek, M. 3D distance fields: A survey of techniques and applications. *IEEE Transactions on Vis. Comput. Graph.* **12**, 581–599, DOI: 10.1109/TVCG.2006.56 (2006).
13. Koschier, D., Deul, C., Brand, M. & Bender, J. An hp-adaptive discretization algorithm for signed distance field generation. *IEEE Transactions on Vis. Comput. Graph.* **23**, 2208–2221, DOI: 10.1109/TVCG.2017.2730202 (2017).
14. Pujol, E. & Chica, A. Adaptive approximation of signed distance fields through piecewise continuous interpolation. *Comput. & Graph.* **114**, 337–346, DOI: 10.1016/j.cag.2023.06.020 (2023).
15. Xu, H. & Barbič, J. 6-DoF haptic rendering using continuous collision detection between points and signed distance fields. *IEEE Transactions on Haptics* **10**, 151–161, DOI: 10.1109/TOH.2016.2613872 (2017).
16. Liu, P. *et al.* Real-time collision detection between general SDFs. *Comput. Aided Geom. Des.* **111**, 102305, DOI: 10.1016/j.cagd.2024.102305 (2024).
17. Wang, J. *et al.* Implicit swept volume SDF: Enabling continuous collision-free trajectory generation for arbitrary shapes. *ACM Transactions on Graph.* **43**, DOI: 10.1145/3658181 (2024).
18. Montaut, L., Lidec, Q. L., Petrík, V., Sivic, J. & Carpentier, J. GJK++: Leveraging acceleration methods for faster collision detection. *IEEE Transactions on Robotics* **40**, 2564–2581, DOI: 10.1109/TRO.2024.3386370 (2024).
19. Park, J. J., Florence, P., Straub, J., Newcombe, R. & Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 165–174, DOI: 10.1109/CVPR.2019.00025 (2019).

20. Zesch, R. S., Modi, V., Sueda, S. & Levin, D. I. W. Neural collision fields for triangle primitives. In *SIGGRAPH Asia 2023 Conference Papers*, DOI: 10.1145/3610548.3618225 (Association for Computing Machinery, 2023).

21. Nielsen, R. G. & Erleben, K. Alternating spatial-temporal optimization for continuous collision detection of Signed Distance Fields. *Comput. Graph. Forum* **45**, e70570, DOI: 10.1111/cgf.70570 (2026).

22. Kikuchi, T. & Kanai, T. A unified discrete collision framework for triangle primitives. *Comput. Graph. Forum* **44**, e70029, DOI: 10.1111/cgf.70029 (2025).

23. Chen, H., Diaz, E. & Yuksel, C. Shortest path to boundary for self-intersecting meshes. *ACM Transactions on Graph.* **42**, DOI: 10.1145/3592136 (2023).

24. Erleben, K. Methodology for assessing mesh-based contact point methods. *ACM Transactions on Graph.* **37**, DOI: 10.1145/3096239 (2018).

Acknowledgements (not compulsory)

The authors received no specific funding for this work.

Data Availability

The source code and result videos supporting the findings of this study are provided in the supplementary materials accompanying this article.

Funding

This work was supported by INHA UNIVERSITY Research Grant. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25474535). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2026-25548560, Artificial Intelligence Innovation Graduate School (Inha University)).

Author contributions statement

J.H.K. conceived the study, developed the methodology, implemented the system, conducted the experiments, analyzed the results, and wrote and reviewed the manuscript.

Algorithm 1 Adaptive Spatio-Temporal Refinement for a Triangle

Require: Triangle trajectory $\mathscr{T}$, SDF ϕ, $[t_0, t_1]$, optional initial FWGSS result R_{init}
Ensure: Best collision result R_{best} and completion flag $complete$

$complete \leftarrow true$
if $R_{\text{init}}.hit$ and $R_{\text{init}}.toi \leq t_1 + \varepsilon_t$ **then**
  $R_{\text{best}} \leftarrow R_{\text{init}}$
  $t_{\text{best}} \leftarrow R_{\text{init}}.toi$
else
  $R_{\text{best}} \leftarrow \emptyset$
  $t_{\text{best}} \leftarrow t_1$
end if
if $t_{\text{best}} \leq t_0 + \varepsilon_t$ **then**
  return $(R_{\text{best}}, complete)$
end if
$\mathscr{S} \leftarrow \{(\mathscr{T}, [t_0, t_{\text{best}}], 0, 0)\}$
$N_{\text{visited}} \leftarrow 0$
while $\mathscr{S} \neq \emptyset$ **do**
  if $N_{\text{visited}} \geq N_{\text{cell}}$ **then**
    $complete \leftarrow false$
    break
  end if
  $\mathscr{C} \leftarrow \textsc{Pop}(\mathscr{S})$
  $N_{\text{visited}} \leftarrow N_{\text{visited}} + 1$
  $\mathscr{C}.t_b \leftarrow \min(\mathscr{C}.t_b, t_{\text{best}})$
  if $\mathscr{C}.t_a \geq t_{\text{best}} - \varepsilon_t$ **then**
    continue
  end if
  $(B, E_s, E_t, r_s, r_t) \leftarrow \textsc{CellBound}(\mathscr{C})$
  if $B > \varepsilon_c$ **then**
    continue ▷ collision-free cell
  end if
  $can_s \leftarrow (\ell_s < \ell_s^{\max}) \wedge (r_s > r_s^{\min}) \wedge (A(\mathscr{T}_s) > A_{\min})$
  $can_t \leftarrow (\ell_t < \ell_t^{\max}) \wedge ((t_b - t_a) > \Delta t_{\min})$
  if $can_s \vee can_t$ **then**
    if $can_s \wedge (\neg can_t \vee E_s > E_t)$ **then**
      Subdivide current domain $\mathscr{T}_s$ into $\mathscr{T}_A, \mathscr{T}_B, \mathscr{T}_C$
      Push the three spatial child cells with $\ell_s + 1$
    else
      $t_m \leftarrow (t_a + t_b)/2$
      Push temporal child cells $[t_a, t_m]$ and $[t_m, t_b]$ with $\ell_t + 1$
    end if
    continue
  end if
  $complete \leftarrow false$
  $R \leftarrow \text{FWGSS}(\mathscr{T}_s, [t_a, t_b])$
  if $R.hit$ and $(\neg R_{\text{best}}.hit \vee R.toi < t_{\text{best}} - \varepsilon_t)$ **then**
    $R_{\text{best}} \leftarrow R$
    $t_{\text{best}} \leftarrow R.toi$
  end if
end while
return $(R_{\text{best}}, complete)$

Algorithm 2 Mesh-Level Adaptive Spatio-Temporal CCD

Require: Triangle mesh M, BSH H, rigid motion, SDF ϕ
Ensure: Best result R_{best}, simultaneous-hit set $\mathscr{H}_{\text{best}}$, and completion flag $complete$

$R_0 \leftarrow \text{FWGSSMESH}(M, H, \phi)$
if $R_0.hit$ **then**
    $R_{\text{best}} \leftarrow R_0$
    $t_{\text{best}} \leftarrow R_0.toi$
    $\tau_0 \leftarrow R_0.triangleId$
    $\tau_{\text{best}} \leftarrow \tau_0$
    $\mathscr{H}_{\text{best}} \leftarrow \{R_0\}$
else
    $R_{\text{best}} \leftarrow \emptyset$
    $t_{\text{best}} \leftarrow t_1$
    $\tau_0 \leftarrow \emptyset$
    $\tau_{\text{best}} \leftarrow \emptyset$
    $\mathscr{H}_{\text{best}} \leftarrow \emptyset$
end if
$complete \leftarrow true$
$\mathscr{Q} \leftarrow \text{BROADPHASECANDIDATES}(H, [t_0, t_{\text{best}}])$
Sort $\mathscr{Q}$ by potential collision time
Ensure τ_0 is included if $R_0.hit$
if $|\mathscr{Q}| > N_{\text{tri}}$ **then**
    Keep at most N_{tri} candidates, including τ_0
    $complete \leftarrow false$
end if
for each candidate triangle $\tau \in \mathscr{Q}$ **do**
    if $R_{\text{best}}.hit$ and $\tau = \tau_{\text{best}}$ **then**
        $R_{\text{init}} \leftarrow R_{\text{best}}$
    else
        $R_{\text{init}} \leftarrow \emptyset$
    end if
    $(R_\tau, complete_\tau) \leftarrow \text{ADAPTIVESPACETIME}(\tau, [t_0, t_{\text{best}}], R_{\text{init}})$
    $complete \leftarrow complete \wedge complete_\tau$
    if $R_\tau.hit$ and $R_\tau.toi < t_{\text{best}} - \varepsilon_t$ **then**
        $R_{\text{best}} \leftarrow R_\tau$
        $t_{\text{best}} \leftarrow R_\tau.toi$
        $\tau_{\text{best}} \leftarrow \tau$
        $\mathscr{H}_{\text{best}} \leftarrow \{R_\tau\}$
    else if $R_\tau.hit$ and $|R_\tau.toi - t_{\text{best}}| \le \varepsilon_{\text{sim}}$ **then**
        Add R_τ to $\mathscr{H}_{\text{best}}$
    end if
end for
if $R_{\text{best}}.hit$ **then**
    Generate supplemental contacts for all results in $\mathscr{H}_{\text{best}}$ at t_{best}
end if
return $(R_{\text{best}}, \mathscr{H}_{\text{best}}, complete)$

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Videos.zip
- Code.zip