Supplemental Material for "Type-Free Global Intersection Analysis with Linear Displacement Fields"

CHENGZHU HE, Xiamen University, China
XUDONG FENG, Zhejiang University, China
ANJUN CHEN, Xiamen University, China
DAN SONG, Tianjin University, China
SHIHUI GUO, Xiamen University, China
KUI WU, Independent Researcher, United States of America

Additional Key Words and Phrases: Untangling, collision response, computational geometry, numerical optimization

1 Computational Pipeline and Time Integration

The simulation pipeline alternates between geometry-dependent constraint construction and an implicit Newton update. In particular, ray-casting responses are rebuilt from the current iterate rather than frozen over the complete time step. Each Newton iteration evaluates candidate intersection contours, performs batched ray casting and cluster culling, converts valid hits into untangling stencils, and solves the resulting block linear system using block-Jacobi preconditioned conjugate gradients. Algorithm 1 details the complete end-to-end execution of a physical simulation step with our integrated untangling framework.

2 Discrete Intersections and Contour Topology

2.1 Edge–Face Detection

Let $e = (\boldsymbol{x}_0, \boldsymbol{x}_1)$ be an edge and $f = (\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2)$ a nonincident triangle. A discrete EF event is represented by an edge parameter t and triangle barycentric coordinates $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$ satisfying

$$(1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1 = \sum_{a=0}^{2} \beta_a \boldsymbol{y}_a, \qquad \sum_{a=0}^{2} \beta_a = 1 \tag{1}$$

Broad-phase candidates sharing a vertex are rejected before this solve, as are body pairs excluded by the simulation model. EF events are produced by two mechanisms. The first evaluates the segment–triangle test of Eq. (1) on the GPU in single-precision arithmetic; a miss is final and produces no event. A candidate returned by this test is retained only when it clears a relative grazing margin $\epsilon_{\mathrm{g}} = 10^{-4}$ in three independent senses,

$$\min_{a\in\{0,1\}} |d_a| \;\geq\; \epsilon_{\mathrm{g}}\, h_f, \qquad \min_{a\in\{0,1,2\}} \beta_a \;\geq\; \epsilon_{\mathrm{g}}, \qquad \min(t, 1-t) \;\geq\; \epsilon_{\mathrm{g}}, \tag{2}$$

where $d_a = \boldsymbol{n}_f^{\mathsf{T}}(\boldsymbol{x}_a - \boldsymbol{y}_0)/\|\boldsymbol{n}_f\|$ are the signed distances of the two edge endpoints to the face plane and $h_f = \sqrt{\|\boldsymbol{n}_f\|}$ is the face characteristic length; the first margin is a length measured relative to the face scale, while the latter two are dimensionless parameter margins.

Authors' Contact Information: Chengzhu He, Xiamen University, Xiamen, China, chengzhuhe@stu.xmu.edu.cn; Xudong Feng, Zhejiang University, Hangzhou, China, xudongfeng18@gmail.com; Anjun Chen, Xiamen University, Xiamen, China, anjunchen@xmu.edu.cn; Dan Song, Tianjin University, Tianjin, China, dan.song@tju.edu.cn; Shihui Guo, Xiamen University, Xiamen, China, guoshihui@xmu.edu.cn; Kui Wu, Independent Researcher, Los Angeles, CA, United States of America, walker.kui.wu@gmail.com.

An event that violates any margin is a *grazing contact*: the edge meets the face only tangentially—touching the face plane, lying on the face boundary, or intersecting at an edge endpoint—rather than crossing the triangle interior. Such contacts arise spuriously from near-coplanar adjacent geometry and are discarded before contour construction. The second mechanism lifts effectively coincident contacts into events, so that intersections that meet at primitive boundaries rather than crossing through interiors still reach contour construction: a vertex coincident with a triangle contributes one event for each edge incident to that vertex, and two coincident edges contribute one event for each triangle incident to either edge, with the barycentric coordinates determined by the contact weights. By construction, such an event places the intersection point on the boundary of one of its two primitives. The stored physical record contains $(e, f, t, \boldsymbol{\beta}, \boldsymbol{p})$ with $\boldsymbol{p}$ given by Eq. (1).

2.2 From EF Events to Intersection Contours

Each retained event $z_i = (e_i, f_i, \boldsymbol{p}_i)$ is a node of an EF graph. Let $e_i \sim_E e_j$ mean that the two intersecting edges are adjacent on their mesh, and let $e \subset f$ denote edge–triangle incidence. Two events are connected when the intersection path continues across either the edge side or the face side,

$$(i,j) \in \mathcal{A}_{\mathrm{EF}} \iff \big(f_i = f_j \;\wedge\; e_i \sim_E e_j\big) \;\vee\; \big(e_i \subset f_j \;\wedge\; e_j \subset f_i\big) \tag{3}$$

The first case advances between adjacent edges while remaining on one intersected face; the second crosses the local intersection by exchanging which primitive supplies the edge and which supplies the face. When the incidence traversal encounters a topological loop vertex, the implementation appends a LoopVertex placeholder whose edge and face are both collapsed to that vertex and connects it to the corresponding physical EF node. These placeholders close the EF graph for later topology processing but carry zero response stiffness. Boundary endpoints terminate a branch, and flood-filling the extended graph $\mathcal{G}_{\mathrm{EF}} = (\{z_i\}, \mathcal{A}_{\mathrm{EF}})$ produces raw discrete intersection contours. This construction uses only primitive incidence and therefore does not assume a manifold neighborhood.

We then apply a pairwise degeneracy filter whose retention rule couples each event to its adjacency neighborhood. Event z_i is *boundary-degenerate* when either its edge parameter or one of its face barycentric coordinates lies within $\epsilon_{\mathrm{deg}} = 10^{-4}$ of a primitive boundary,

$$\mathrm{degen}(z_i) \iff \min(t_i, 1-t_i) < \epsilon_{\mathrm{deg}} \;\vee\; \min_{a\in\{0,1,2\}} \beta_{i,a} < \epsilon_{\mathrm{deg}}, \tag{4}$$

Algorithm 1: Full Simulation Time Step with Untangling Resolution

Input: Current state $(\boldsymbol{q}^n, \boldsymbol{v}^n)$, time step h, mass matrix $\boldsymbol{M}$, external forces $\boldsymbol{f}_{\text{ext}}$
Output: Updated state $(\boldsymbol{q}^{n+1}, \boldsymbol{v}^{n+1})$
`// 1. Inertial prediction`
$\widetilde{\boldsymbol{q}} \leftarrow \boldsymbol{q}^n + h\boldsymbol{v}^n + h^2\boldsymbol{M}^{-1}\boldsymbol{f}_{\text{ext}}$; $\quad \boldsymbol{q}^{(0)} \leftarrow \widetilde{\boldsymbol{q}}$;
for *Newton iteration* $k = 0, 1, \dots, K_{\max}$ **do**
  $\boldsymbol{x} \leftarrow \text{Lift}(\boldsymbol{q}^{(k)})$ via Eq. (48);
  `// 2. Collision & Intersection Detection`
  Update GPU LBVH over current geometry $\boldsymbol{x}$;
  Detect proximity pairs $\mathcal{P}_{\text{prox}}$ and discrete EF pairs $\mathcal{Z}_{\text{EF}}$;
  `// 3. Untangling Resolution (Ours)`
  if $\mathcal{Z}_{\text{EF}} \neq \emptyset$ **then**
    Extract connected contours $\{C\}$ and seeds $\mathcal{A}_s$ (Eq. (3));
    foreach *contour* C **do**
      Compute multi-source distance fields (ℓ_s, g_s) (Algorithm 2);
      Generate candidate directions $\mathcal{D}_0$ from $\boldsymbol{S}_w$ (Sec. 3);
      Perform batched ray casting and cluster culling (Algorithm 4);
      Apply contour-side candidate restriction and validity filtering;
      Refine winner on $\mathbb{S}^2$ via projected Newton (Eq. (46));
    end
    Resolve multi-contour conflicts via Greedy Maximum WIS;
    Convert active hits $\mathcal{S}^\star$ to penalty stencils $\mathcal{R}$;
  else
    $\mathcal{R} \leftarrow \emptyset$;
  end
  `// 4. Global Gradient and Hessian Assembly`
  Assemble inertia, elasticity, bending, and ABD orthogonality (Sec. 7);
  Assemble proximity barrier forces for pairs not in untangling regions (Sec. 4);
  foreach *untangling stencil* $h \in \mathcal{R}$ **do**
    $\boldsymbol{g} \mathrel{+}= \kappa \bar{A}_h C_h \boldsymbol{J}_h^\top$; $\quad \boldsymbol{H} \mathrel{+}= \kappa \bar{A}_h \boldsymbol{J}_h^\top \boldsymbol{J}_h$;
  end
  `// 5. Linear Solve & Convergence Check`
  if $\|\boldsymbol{g}\|_\infty < \epsilon_{\text{tol}}$ ***and*** $\mathcal{R} = \emptyset$ **then**
    break;
  end
  Solve $\boldsymbol{H}\Delta\boldsymbol{q} = -\boldsymbol{g}$ using GPU Block-Jacobi PCG;
  `// 6. Filtered Line Search & State Update`
  Step $\alpha \in (0, 1]$ via backtracking subject to CCD bound (Eq. (20));
  $\boldsymbol{q}^{(k+1)} \leftarrow \boldsymbol{q}^{(k)} + \alpha\Delta\boldsymbol{q}$;
end
$\boldsymbol{q}^{n+1} \leftarrow \boldsymbol{q}^{(k)}$; $\quad \boldsymbol{v}^{n+1} \leftarrow (\boldsymbol{q}^{n+1} - \boldsymbol{q}^n)/h$;
return $(\boldsymbol{q}^{n+1}, \boldsymbol{v}^{n+1})$;

and *strictly interior* otherwise. A strictly interior event is always retained; a boundary-degenerate event is retained exactly when at least one of its neighbors is strictly interior,

$$\text{keep}(z_i) \iff \neg\,\text{degen}(z_i) \;\vee\; \exists\, j : \; (i, j) \in \mathcal{A}_{\text{EF}} \;\wedge\; \neg\,\text{degen}(z_j), \tag{5}$$

and is discarded only when every neighbor is itself boundary-degenerate. Retaining the degenerate events adjacent to interior regions keeps a contour connected where its crossing grazes a primitive boundary. Because Eq. (2) already rejects every event that Eq. (4) would flag, boundary-degenerate events originate exclusively from the proximity mechanism. When no strictly interior event exists, the filter retains the complete event set: a *pure boundary intersection complex*, in which distinct surface sheets meet at vertex–edge or edge–edge incidence rather than crossing through triangle interiors, is a genuine intersection state, and discarding it would report an intersection-free configuration that the untangling stage would never see. Only the edge and face barycentric coordinates of an event participate in this decision; neither Euclidean hit distance nor plane-crossing depth does.

2.3 Contour-Seeded Distance Fields

The adaptive k-ring candidate restriction and the support validity filters both rely on two per-vertex scalar fields defined on each logical side's vertex graph $\mathcal{G}_V = (\mathcal{V}_s, \mathcal{E}_s)$. Let $\mathcal{A}_s$ be the set of contour-boundary seed vertices on side $s \in \{\text{s}, \text{d}\}$, and let $\mathcal{N}(v) = \{u \mid (u, v) \in \mathcal{E}_s\}$ denote the mesh vertex neighbors of v. Both fields are evaluated as multi-source shortest-path traversals seeded at $\mathcal{A}_s$ and restricted to the vertex range of the object owning side s (Algorithm 2).

Discrete hop distance. The topological hop distance $\ell_s(v) \in \mathbb{N}$ counts unweighted graph edges from v to the nearest seed:

$$\ell_s(v) = \min_{a \in \mathcal{A}_s} d_{\text{hop}}(a, v), \qquad d_{\text{hop}}(a, v) = \min_{\text{paths } a \to v} |\text{edges}| \tag{6}$$

Because every edge has uniform unit weight, multi-source breadth-first search settles each vertex exactly once upon its first arrival, running in linear time $O(|\mathcal{V}_s| + |\mathcal{E}_s|)$.

Geodesic distance. The geodesic distance $g_s(v) \in \mathbb{R}_{\geq 0}$ accumulates material-space edge lengths along the shortest path:

$$g_s(v) = \min_{a \in \mathcal{A}_s} d_{\text{geo}}(a, v), \qquad d_{\text{geo}}(a, v) = \min_{\text{paths } a \to v} \sum_{(u,w) \in \text{path}} \|\bar{\boldsymbol{x}}_u - \bar{\boldsymbol{x}}_w\| \tag{7}$$

Multi-source Dijkstra traversal uses a min-priority queue keyed by tentative distance and simultaneously records the *nearest seed* $\pi_s(v) = \arg\min_{a \in \mathcal{A}_s} d_{\text{geo}}(a, v)$ by propagating the originating seed identifier along the shortest-path tree.

Role independence and cross-side statistics. The two logical sides are evaluated independently from their respective boundary sets $\mathcal{A}_s$, so source and destination adaptive radii $k_{\text{s}}^{(m)}$ and $k_{\text{d}}^{(m)}$ evolve asynchronously. The per-side maxima

$$H_s = \max_{v \in \mathcal{V}_s} \ell_s(v), \qquad G_s = \max_{v \in \mathcal{V}_s} g_s(v) \tag{8}$$

serve as the mesh hop and geodesic diameters used to clamp the adaptive expansion ($k_{s,\max} = H_s + 1$). Additionally, querying seed distances against the opposite side's fields yields the reference values $\ell_{\bar{s}}(\pi_s(v))$ (where $\bar{s} \in \{\text{s}, \text{d}\} \setminus \{s\}$ denotes the complementary logical side) used by our support validity filters.

Algorithm 2: Contour-Seeded Distance Fields and Statistics

Input: Mesh 1-skeleton $\mathcal{G}_V = (\mathcal{V}_s, \mathcal{E}_s)$, rest nodal positions $\bar{\boldsymbol{x}}$, contour seed set $\mathcal{A}_s$
Output: Hop field ℓ_s, rest geodesic field g_s, nearest-seed map π_s, mesh diameters (H_s, G_s)

Initialize $\ell_s(v) \leftarrow \infty$, $g_s(v) \leftarrow \infty$, $\pi_s(v) \leftarrow$ null for all $v \in \mathcal{V}_s$;
`// 1. Multi-source BFS for discrete hop distance`
FIFO queue $Q \leftarrow \emptyset$;
foreach $a \in \mathcal{A}_s$ **do**
  $\ell_s(a) \leftarrow 0$; Push a into Q;
while $Q \neq \emptyset$ **do**
  Pop $u \leftarrow \mathrm{head}(Q)$;
  foreach $v \in \mathcal{N}(u)$ **do**
    if $\ell_s(v) = \infty$ **then**
      $\ell_s(v) \leftarrow \ell_s(u) + 1$; Push v into Q;
`// 2. Multi-source Dijkstra for rest geodesic distance`
Priority queue $PQ \leftarrow \emptyset$;
foreach $a \in \mathcal{A}_s$ **do**
  $g_s(a) \leftarrow 0$; $\pi_s(a) \leftarrow a$; Insert $(a, 0)$ into PQ;
while $PQ \neq \emptyset$ **do**
  Pop (u, d) with minimum distance d from PQ;
  if $d \leq g_s(u)$ **then**
    foreach $v \in \mathcal{N}(u)$ **do**
      $w_{uv} \leftarrow \|\bar{x}_u - \bar{x}_v\|$;
      if $g_s(u) + w_{uv} < g_s(v)$ **then**
        $g_s(v) \leftarrow g_s(u) + w_{uv}$; $\pi_s(v) \leftarrow \pi_s(u)$;
        Insert or decrease key $(v, g_s(v))$ in PQ;
`// 3. Mesh diameter and normalizer extraction`
$H_s \leftarrow \max_{v \in \mathcal{V}_s} \ell_s(v)$; $G_s \leftarrow \max_{v \in \mathcal{V}_s} g_s(v)$;
return $(\ell_s, g_s, \pi_s, H_s, G_s)$;

Why two fields. Hop distance provides a tessellation-invariant topological radius: k-rings of fixed k cover a comparable combinatorial neighborhood regardless of local refinement. Geodesic distance provides a metric-aware scale: our spatial leakage metric normalizes intrinsic distances by $k_s^{(0)} G_s / H_s$ to obtain a resolution-independent threshold. Using rest-space rather than world-space edge lengths keeps both distance fields invariant across Newton iterations within a simulation step, ensuring candidate domains and validity predicates do not drift during deformation.

2.4 GIA Classification

Algorithm 3 states the contour routing used by our GIA baseline. Let B be the number of boundary endpoints, L the number of loop vertices, and $\sigma(b)$ the propagated logical side at endpoint b. The algorithm exposes why multiple loop vertices or atypical boundary valence fall outside the canonical table.

Algorithm 3: Contour-type classification

Input: Connected EF contour C with boundary endpoints ∂C, loop vertices ΛC, and endpoint side labels σ
Output: One of Closed, Eight, LL, BB/II, BI/BI, BLI, Cross, or Undefined

$B \leftarrow |\partial C|$; $L \leftarrow |\Lambda C|$;
if $B = 0$ **then**
  if $L = 0$ **then**
    return Closed;
  else if $L = 1$ **then**
    return Eight;
  else if $L = 2$ **then**
    return LL;
else if $L = 0$ *and* $B = 2$ **then**
  Let b_0, b_1 be the two boundary endpoints;
  if $\sigma(b_0) = \sigma(b_1)$ **then**
    return BB/II;
  else
    return BI/BI;
else if $L = 1$ *and* $B = 1$ **then**
  return BLI;
else if $L = 1$ *and* $B = 2$ **then**
  return Cross;
return Undefined;

2.5 Intersection Contour Minimization Baseline

ICM minimizes the discrete contour length $\mathcal{L}(C) = \sum_i \ell_i$. Consider EF event i with edge $e_i = (\boldsymbol{x}_0, \boldsymbol{x}_1)$ and intersected face $f_i = (\boldsymbol{y}_0, \boldsymbol{y}_1, \boldsymbol{y}_2)$. Define

$$\boldsymbol{e}_i = \boldsymbol{x}_1 - \boldsymbol{x}_0, \qquad \widehat{\boldsymbol{n}}_i = \frac{(\boldsymbol{y}_1 - \boldsymbol{y}_0) \times (\boldsymbol{y}_2 - \boldsymbol{y}_0)}{\|(\boldsymbol{y}_1 - \boldsymbol{y}_0) \times (\boldsymbol{y}_2 - \boldsymbol{y}_0)\|}. \tag{9}$$

The edge adjacency list supplies every face B_{ij} incident on e_i and its vertex $\boldsymbol{z}_{ij}$ opposite the edge. Let $(\boldsymbol{b}_{ij,0}, \boldsymbol{b}_{ij,1}, \boldsymbol{b}_{ij,2})$ be the ordered coordinates of B_{ij}. Its unit normal and its inward edge co-normal are computed as

$$\widehat{\boldsymbol{b}}_{ij} = \frac{(\boldsymbol{b}_{ij,1} - \boldsymbol{b}_{ij,0}) \times (\boldsymbol{b}_{ij,2} - \boldsymbol{b}_{ij,0})}{\|(\boldsymbol{b}_{ij,1} - \boldsymbol{b}_{ij,0}) \times (\boldsymbol{b}_{ij,2} - \boldsymbol{b}_{ij,0})\|}, \qquad \widehat{\boldsymbol{m}}_{ij} = \frac{\boldsymbol{d}_{ij} - (\boldsymbol{d}_{ij}^\mathsf{T} \widehat{\boldsymbol{e}}_i) \widehat{\boldsymbol{e}}_i}{\|\boldsymbol{d}_{ij} - (\boldsymbol{d}_{ij}^\mathsf{T} \widehat{\boldsymbol{e}}_i) \widehat{\boldsymbol{e}}_i\|}, \tag{10}$$

where $\widehat{\boldsymbol{e}}_i = \boldsymbol{e}_i / \|\boldsymbol{e}_i\|$ and $\boldsymbol{d}_{ij} = \boldsymbol{z}_{ij} - (\boldsymbol{x}_0 + \boldsymbol{x}_1)/2$. The oriented contour tangent within B_{ij} is then

$$\widetilde{\boldsymbol{r}}_{ij} = \widehat{\boldsymbol{n}}_i \times \widehat{\boldsymbol{b}}_{ij}, \qquad \boldsymbol{r}_{ij} \in \left\{ \pm \frac{\widetilde{\boldsymbol{r}}_{ij}}{\|\widetilde{\boldsymbol{r}}_{ij}\|} \right\}, \qquad \boldsymbol{r}_{ij}^\mathsf{T} \widehat{\boldsymbol{m}}_{ij} \geq 0. \tag{11}$$

Its edge-motion correction and the event contribution are

$$\boldsymbol{g}_{ij} = \boldsymbol{r}_{ij} - 2 \frac{\boldsymbol{e}_i^\mathsf{T} \boldsymbol{r}_{ij}}{\boldsymbol{e}_i^\mathsf{T} \widehat{\boldsymbol{n}}_i} \widehat{\boldsymbol{n}}_i, \qquad \boldsymbol{g}_i = \sum_{B_{ij} \ni e_i} \boldsymbol{g}_{ij}. \tag{12}$$

The global scheme applies the saturation

$$S(\boldsymbol{g}) = h_0 \frac{\boldsymbol{g}}{\sqrt{\|\boldsymbol{g}\|^2 + g_0^2}}, \qquad \boldsymbol{G}_C = \sum_{i \in C} s_i S(\boldsymbol{g}_i), \qquad \boldsymbol{H}_C = S(\boldsymbol{G}_C), \tag{13}$$

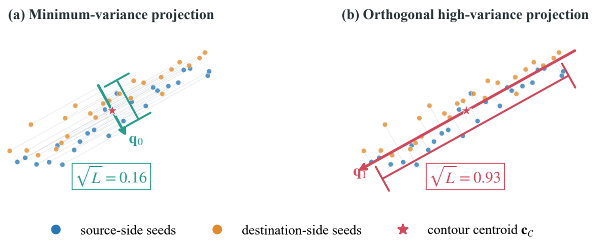

Fig. 1. Shortest-projection intuition. The smallest-eigenvalue direction $\boldsymbol{q}_0$ minimizes the quadratic projected thickness in Eq. (14); an orthogonal direction spans a substantially longer interval.

where $s_i \in \{-1, 1\}$ aligns the propagated logical edge side of event i with the contour root. In the discretization, $\ell_i = \frac{1}{2}\sum_{j\in\mathcal{N}(i)} \|\boldsymbol{p}_j - \boldsymbol{p}_i\|$ is the half-sum of distances to contour-adjacent EF events. We use $h_0 = 0.01\,\mathrm{m}$, $g_0 = 1$, and per-event stiffness $k_i = 10^4\ell_i$; LoopVertex events have zero stiffness. The applied sign decreases $\mathcal{L}$.

3 Scatter-Direction Derivation

Let $\mathcal{A} = \mathcal{A}_0 \cup \mathcal{A}_1$ be the contour-support vertices, $A_v > 0$ their area weights, and $\boldsymbol{c}_C$ the contour centroid. For a unit direction $\boldsymbol{r}$, the weighted squared projected thickness is

$$L(\boldsymbol{r}) = \sum_{v\in\mathcal{A}} A_v\big(\boldsymbol{r}^{\mathsf{T}}(\boldsymbol{x}_v - \boldsymbol{c}_C)\big)^2 = \boldsymbol{r}^{\mathsf{T}} S_w \boldsymbol{r}, \qquad \|\boldsymbol{r}\| = 1 \tag{14}$$

Here $S_w = \sum_v A_v(\boldsymbol{x}_v - \boldsymbol{c}_C)(\boldsymbol{x}_v - \boldsymbol{c}_C)^{\mathsf{T}}$. Let $\lambda_0 \le \lambda_1 \le \lambda_2$ be its eigenvalues with orthonormal eigenvectors $\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2$. Expanding $\boldsymbol{r} = \sum_i \alpha_i \boldsymbol{q}_i$ gives

$$L(\boldsymbol{r}) = \sum_{i=0}^{2} \lambda_i \alpha_i^2 \ge \lambda_0 \sum_{i=0}^{2} \alpha_i^2 = \lambda_0 = L(\boldsymbol{q}_0) \tag{15}$$

Thus $\pm\boldsymbol{q}_0$ minimize the area-weighted root-mean-square thickness of the observed contour support and form the minimum-variance axis of S_w.

The scatter matrix contains only geometry adjacent to the detected contour. For a deep penetration, its minimum-variance axis may not minimize the displacement across the full intersecting region; we therefore also evaluate the signed orthogonal eigenvectors and four centroid-derived directions. Every candidate is judged by its reconstructed VF/EE/FV hit set and displacement objective.

4 Collision Filtering During Untangling

Standard CCD assumes that the configuration at the beginning of a line search is collision-free. Untangling violates this premise: the accepted Newton direction is deliberately constructed to separate primitives that already intersect at the current iterate. For such a pair, a continuous trajectory begins at or inside contact and therefore necessarily reports a time of impact near zero. If this event is treated as a new collision, the CCD line search suppresses the very displacement required to remove the penetration.

Our filter is contour-local and role-aware; it does not disable collision protection between complete objects. For contour c, assign the source and destination labels

$$\rho_c^{\mathrm{s}} = 2c, \qquad \rho_c^{\mathrm{d}} = 2c + 1 = \rho_c^{\mathrm{s}} \oplus 1 \tag{16}$$

The operation $\oplus 1$ toggles the least significant bit. Each contour response assigns its source feature the first label and its destination feature the complementary label. Let $\mathcal{R}_V(v)$ be the set of labels incident on vertex v, and extend labels to edges and faces by union,

$$\mathcal{R}_E(e) = \bigcup_{v\in e} \mathcal{R}_V(v), \qquad \mathcal{R}_F(f) = \bigcup_{v\in f} \mathcal{R}_V(v) \tag{17}$$

A broad-phase VF candidate (v, f) is omitted from the CCD narrow phase precisely when its two features contain complementary labels,

$$\chi_{\mathrm{VF}}(v, f) = [\exists r \in \mathcal{R}_V(v),\ s \in \mathcal{R}_F(f) : r = s \oplus 1] \tag{18}$$

where $[\cdot]$ denotes the Iverson bracket, evaluating to 1 if the enclosed condition is true and 0 otherwise. Equation (18) covers both orientations: a source vertex against a destination face and a destination vertex against a source face. Similarly, an EE candidate (e_1, e_2) is omitted when

$$\chi_{\mathrm{EE}}(e_1, e_2) = [\exists r \in \mathcal{R}_E(e_1),\ s \in \mathcal{R}_E(e_2) : r = s \oplus 1] \tag{19}$$

Thus, only VF/FV and EE pairs connecting the two logical sides of the *same* active contour are exempted. Pairs belonging to different contours, unrelated parts of the same mesh, and ordinary inter-object contacts continue to contribute a time-of-impact bound.

Primitives in the initially invalid contour neighborhood receive the same broad-phase exclusion. This auxiliary mask handles degenerate seeds for which a complete complementary label has not yet propagated. After the broad-phase flags have been assigned, all unmarked candidates undergo the usual VF and EE narrow phase, and the line-search step is bounded by the earliest remaining impact,

$$\alpha \le \eta \min_{p:\,\chi(p)=0} \tau_p, \qquad 0 < \eta < 1 \tag{20}$$

Here τ_p is the candidate time of impact and η is the CCD safety factor.

5 BVH Traversal Bounds

For each query type, a BVH node is rejected only when its entire box cannot intersect the swept primitive; VF or EE geometry is evaluated at leaves. Candidate-region membership, self-incidence, and hop-distance restrictions are applied in addition to the geometric box tests.

5.1 VF Rays

Let a source vertex emit the segment

$$\boldsymbol{x}(t) = \boldsymbol{o} + t\boldsymbol{r}, \qquad t \in [0, \bar{t}] \tag{21}$$

Let a BVH node have axis-aligned bounds $[\boldsymbol{b}^-, \boldsymbol{b}^+]$. For every coordinate j, define the ordered slab parameters

$$t_j^- = \min\left(\frac{b_j^- - o_j}{r_j}, \frac{b_j^+ - o_j}{r_j}\right), \qquad t_j^+ = \max\left(\frac{b_j^- - o_j}{r_j}, \frac{b_j^+ - o_j}{r_j}\right) \tag{22}$$

For $r_j = 0$, the interval is empty if o_j lies outside the slab and unbounded otherwise. The three-dimensional interval is

$$t_{\text{enter}} = \max_j t_j^-, \qquad t_{\text{exit}} = \min_j t_j^+ \tag{23}$$

The node is traversed iff

$$t_{\text{exit}} \geq \max(t_{\text{enter}}, 0) \quad \text{and} \quad t_{\text{enter}} \leq \bar{t} \tag{24}$$

The implementation precomputes the componentwise inverse of $\boldsymbol{r}$ per ray and uses the equivalent branch-minimized slab calculation. At a leaf, ray–triangle intersection supplies the barycentric coordinates and ray parameter; faces incident on the source vertex are rejected.

5.2 EE Sweeps

For a source edge with endpoints $\boldsymbol{p}_0, \boldsymbol{p}_1$ and direction $\boldsymbol{r}$, the swept set is the parallelogram

$$\boldsymbol{p}(u,t) = \boldsymbol{p}_0 + u(\boldsymbol{p}_1 - \boldsymbol{p}_0) + t\boldsymbol{r}, \qquad (u,t) \in [0,1] \times [0,\bar{t}] \tag{25}$$

Its axis-aligned box is the componentwise hull of

$$\{\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_0 + \bar{t}\boldsymbol{r}, \boldsymbol{p}_1 + \bar{t}\boldsymbol{r}\} \tag{26}$$

A node is first rejected when this hull and the node box do not overlap. Two inexpensive tests further tighten traversal.

First, let the node center and half extent be $\boldsymbol{c}$ and $\boldsymbol{a}$. The node projection interval along $\boldsymbol{r}$ has center $\boldsymbol{c}^\top \boldsymbol{r}$ and radius

$$\rho_r = \sum_{j=1}^{3} a_j |r_j| \tag{27}$$

It must overlap the sweep projection

$$\left[\min_{i\in\{0,1\}} \boldsymbol{p}_i^\top \boldsymbol{r}, \max_{i\in\{0,1\}} \boldsymbol{p}_i^\top \boldsymbol{r} + \bar{t}\|\boldsymbol{r}\|^2\right] \tag{28}$$

The batched implementation sets $\bar{t}$ from the largest target-edge projection, avoiding traversal beyond all admissible targets.

Second, when $\boldsymbol{p}_1 - \boldsymbol{p}_0$ and $\boldsymbol{r}$ are not parallel, the sweep lies in the plane

$$\boldsymbol{n} = \frac{(\boldsymbol{p}_1 - \boldsymbol{p}_0) \times \boldsymbol{r}}{\|(\boldsymbol{p}_1 - \boldsymbol{p}_0) \times \boldsymbol{r}\|}, \qquad d = \boldsymbol{n}^\top \boldsymbol{p}_0 \tag{29}$$

The node can meet this plane only if

$$|\boldsymbol{n}^\top \boldsymbol{c} - d| \leq \sum_{j=1}^{3} a_j |n_j| + \varepsilon_{\text{box}} \tag{30}$$

The plane test is skipped for a degenerate sweep. At a leaf, the edge–edge hit is the solution of

$$t\boldsymbol{r} + u(\boldsymbol{p}_1 - \boldsymbol{p}_0) - v(\boldsymbol{q}_1 - \boldsymbol{q}_0) = \boldsymbol{q}_0 - \boldsymbol{p}_0 \tag{31}$$

The solution must satisfy $t \in [0, \bar{t}]$ and $u, v \in [0,1]$. Topologically incident source and target edges are excluded.

6 Batched GPU Ray Casting and Cluster Culling

6.1 Task Packing and Batched Queries

A GPU task $t = (c, d, k_{\text{s}}, k_{\text{d}})$ consists of a contour, one proposed direction, and the active source/destination candidate radii. For each task, request preparation emits

$$Q_t^{\text{VF}} = \{(v_{\text{s}}, t) : v_{\text{s}} \in \mathcal{V}_{\text{s}}^t\} \tag{32}$$

$$Q_t^{\text{EE}} = \{(e_{\text{s}}, t, \bar{t}_e) : e_{\text{s}} \in \mathcal{E}_{\text{s}}^t\} \tag{33}$$

$$Q_t^{\text{FV}} = \{(v_{\text{d}}, t) : v_{\text{d}} \in \mathcal{V}_{\text{d}}^t\} \tag{34}$$

The target-face, target-edge, and source-face sets are stored as task-indexed contiguous ranges together with exact membership lists. Topology-dependent request buffers are cached across direction trials. Only direction-dependent quantities, including the EE projection bounds, are refreshed.

For an EE source edge e and direction $\boldsymbol{r}_t$, the target projections give the ray-parameter bound

$$\bar{t}_e = \frac{\max\limits_{e' \in \mathcal{E}_{\text{d}}^t} \max\limits_{\boldsymbol{x} \in e'} \boldsymbol{x}^\top \boldsymbol{r}_t - \min\limits_{\boldsymbol{x} \in e} \boldsymbol{x}^\top \boldsymbol{r}_t}{\|\boldsymbol{r}_t\|^2} + \varepsilon_t \tag{35}$$

VF and EE requests sweep along $\boldsymbol{r}_t$; FV requests equivalently cast destination vertices along $-\boldsymbol{r}_t$ against the source-face BVH. All requests from all active tasks are processed in three large batches.

Each request first counts its accepted leaf intersections. An exclusive prefix sum allocates a compact output interval, and a second pass writes the hits and their geometric attributes. The output therefore has size proportional to the number of actual hits rather than to the Cartesian product of source and target primitives. Per-task output slices are grown and the batch is replayed if a rare capacity overflow occurs.

6.2 Connected Components of the Hit Graph

For a task t, let $\mathcal{H}_t$ be its raw hits. We define the undirected product-complex hit graph

$$\begin{aligned}
\mathcal{G}_t &= (\mathcal{H}_t, \mathcal{A}_t), \\
(i,j) \in \mathcal{A}_t \iff &\Big(\text{adj}_{\text{prod}}(h_i, h_j) = 1 \\
&\quad \vee \big(\mathbb{I}_{\text{solid}}(\mathcal{B}) \wedge \text{adj}_{\text{solid}}(h_i, h_j) = 1\big)\Big) \\
&\quad \wedge\ \mathcal{B}_t(h_i, h_j) = 1,
\end{aligned} \tag{36}$$

where adj_{prod} is the dual-sided product-cell adjacency predicate (Sec. 6.3), $\text{adj}_{\text{solid}}$ is the solid interior adjacency predicate for watertight solid bodies (Sec. 6.4), and $\mathcal{B}_t$ enforces contour-boundary containment guards.

To eliminate quadratic $O(|\mathcal{H}_t|^2)$ pairwise comparisons across large hit buffers, the GPU constructs task-aware inverted lists from mesh primitives to incident hits. Key counts are accumulated per pair (t, p), an exclusive prefix sum allocates a compressed-sparse-row (CSR) buffer, and hit indices are scattered into the corresponding rows. Each hit thread traverses only CSR rows matching its incident primitives, evaluating adj_{prod} and boundary guards on-the-fly.

Connected components are extracted via parallel lock-free union–find with path compression. Initially, each hit defines a singleton set with parent $P_i = i$. For every valid edge $(i,j) \in \mathcal{A}_t$, the thread

Algorithm 4: Batched evaluation of contour directions

Input: Contour tasks $\mathcal{T}$, candidate regions, proposed directions $\{\boldsymbol{r}_t\}$, face and edge BVHs
Output: One selected cluster and score for each valid task

1. **Parallel request preparation:** pack Eqs. (34) and (35); cache task membership ranges;
2. **Batched ray casting:** evaluate all VF requests along $\boldsymbol{r}_t$, all EE sweeps along $\boldsymbol{r}_t$, and all FV requests along $-\boldsymbol{r}_t$;
3. Count, prefix-sum, and compact accepted leaf hits into task-contiguous arrays;
4. **Batched cluster culling and selection:** construct each hit graph by Eq. (36), then invoke Algorithm 5 to obtain one eligible component per task;
5. Evaluate the direction objective from the selected component;
6. **if** *the task enters direction refinement* **then**
7. compute a tangent-space trial direction on $\mathbb{S}^2$;
8. append the trial to a subsequent batch and repeat ray casting and cluster culling;
9. Return the best accepted direction, its rebuilt hit set, and the associated response support;

resolves the current representative roots r_i, r_j and atomically hooks the higher root index to the lower root:

$$P_{\max(r_i,r_j)} \leftarrow \min\left(P_{\max(r_i,r_j)}, \min(r_i, r_j)\right). \tag{37}$$

Because roots only decrease monotonically, this atomic minimization is cycle-free and requires no global locks. Path compression ($P_i \leftarrow P_{P_i}$) executes in parallel after each hooking pass. Termination is governed by a sticky device-side convergence flag Δ, eliminating host-device synchronization latency during iteration. The resulting connected components are

$$\Gamma_r = \{i \in \mathcal{H}_t : \mathrm{root}(i) = r\}. \tag{38}$$

To reject disconnected ray evidence, a cluster is eligible only if it contains at least one contour-anchored hit, i.e. a hit whose source primitives touch $\mathcal{N}_1(\mathcal{A}_0)$ and whose destination primitives touch $\mathcal{N}_1(\mathcal{A}_1)$ simultaneously. For an eligible cluster Γ, let $\mathcal{A}(\Gamma)$ be the distinct nearest contour-seed anchors covered by its source and destination features, and let

$$K(\Gamma) = \{h \in \Gamma : d_{\mathrm{hop}}(h) \le d_{\max}\} \tag{39}$$

The set $K(\Gamma)$ contains the hits retained by the active locality bound $d_{\max}$, where $d_{\mathrm{hop}}(h) = \max_{v \in \mathrm{supp}(h)} \ell_s(v)$ measures the topological hop distance from hit h's incident vertices to the contour seed boundary. The default selection is the deterministic lexicographic rule

$$\Gamma^* = \underset{\Gamma \text{ eligible}}{\arg\overset{\mathrm{lex}}{\max}} \left(|\mathcal{A}(\Gamma)|, |K(\Gamma)|, -\kappa(\Gamma), -r(\Gamma)\right) \tag{40}$$

where the lexicographical $\arg\max^{\mathrm{lex}}$ maximizes the four-tuple in left-to-right priority order: candidate clusters are ranked primarily by the number of covered seed anchors $|\mathcal{A}(\Gamma)|$, breaking ties sequentially by the retained local hit count $|K(\Gamma)|$, the negative canonical feature key $-\kappa(\Gamma)$, and the component root $-r(\Gamma)$. Coverage therefore takes precedence over raw multiplicity, while the final two tie-breaking keys make the selection strictly deterministic and independent of parallel thread execution order.

6.3 Why Product-Complex Adjacency

In discrete uncut triangle meshes, intersection curves cut transversely through triangle interiors without modifying the underlying mesh 1-skeleton $G = (\mathcal{V}, \mathcal{E})$. Standard single-mesh graph traversal allows 1-ring vertex paths to circumvent the uncut intersection line, spuriously merging logically separate surface regions into a single connected component. We resolve this ambiguity without dynamic remeshing by defining hit adjacency on the discrete Cartesian product complex $\mathcal{K}_{\mathrm{s}} \times \mathcal{K}_{\mathrm{d}}$.

Product-cell structure. The three hit types VF, EE, and FV reside in the 2-dimensional product cells

$$\mathcal{V}_{\mathrm{s}} \times \mathcal{F}_{\mathrm{d}}, \quad \mathcal{E}_{\mathrm{s}} \times \mathcal{E}_{\mathrm{d}}, \quad \mathcal{F}_{\mathrm{s}} \times \mathcal{V}_{\mathrm{d}},$$

whose dimensions sum to (0+2), (1+1), and (2+0), respectively. In the Cartesian product of two simplicial complexes, two k-cells $(\sigma_1 \times \tau_1)$ and $(\sigma_2 \times \tau_2)$ share a codimension-one face (a $(k-1)$-cell of the product) if and only if exactly one factor steps across a facet while the other factor stays fixed [Hatcher 2002]. The adjacency predicates in the main paper Table 1 instantiate precisely this codimension-one facet adjacency for the three 2-cell types: each rule either advances the source primitive by one facet step while keeping the destination primitive identical, or vice versa.

Virtual-remesh interpretation. If the intersection contour were explicitly cut into the mesh (by splitting triangles along intersection curves and inserting new edges), a standard BFS on the refined 1-skeleton would naturally separate the two sides of the intersection line, because the new edges would form a topological barrier. Product adjacency achieves the same separating effect on the *unmodified* mesh: a graph path in $G_{\boldsymbol{r}}$ that attempts to cross from the "interior" side of the contour to the "exterior" side must, at some step, pair a source primitive on one side with a destination primitive on the other. Since the destination-side faces that are actually struck by rays are not adjacent across the uncut intersection line (they lie on opposite sides of the transverse cut in the face interior), the product-adjacency predicate blocks the crossing. The construction thus provides a *virtual remesh*—the topological separation that dynamic remeshing would provide, without any mesh modification.

6.4 Solid Interior Hit Adjacency for Watertight Rigid Bodies

For thin-shell cloth meshes, the simulation domain is an embedded 2-manifold Σ whose material connectivity is governed by surface topology. Consequently, product-complex adjacency $\mathrm{adj}_{\mathrm{prod}}$ (Sec. 6.3) faithfully propagates cluster connectivity along Σ. However, for solid objects (such as rigid, affine, or tetrahedral bodies with volumetric interior $\Omega \subset \mathbb{R}^3$ bounded by $\partial\Omega = \Sigma$), severe penetrations frequently pierce entirely through the solid cross section. In such configurations, a separation ray cast from a penetrating source primitive enters the solid body at an entrance boundary element and exits on the opposite side at an exit boundary element. Under pure surface product-complex adjacency, these entrance and exit hits can only be linked via boundary paths lying on $\partial\Omega$. For thick or complex geometry, connecting them requires circumnavigating the outer surface over a large number of topological hops, often exceeding the local candidate radius $d_{\max}$ or failing altogether when intermediate non-penetrating surface triangles register no ray hits. As a result, the exit hits become isolated from the contour seed anchors $\mathcal{N}_1(\mathcal{A}_0), \mathcal{N}_1(\mathcal{A}_1)$ and are discarded by the cluster selection rule (Eq. (40)). Deprived of bilateral exit responses, the solver produces asymmetric untangling forces that stall convergence.

Virtual volumetric discretization. Physically, the bounded interior $\Omega = \mathrm{int}(\mathcal{B})$ is filled with solid material rather than empty space. If the interior were explicitly discretized by a conforming tetrahedral mesh $\mathcal{T}$ with boundary $\partial\mathcal{T} = \partial\Omega$, a separation ray traversing Ω would directly step through a sequence of facet-adjacent tetrahedra connecting the entrance face to the exit face across shared 2D triangular faces. Parallel to how product-complex adjacency provides a *virtual surface remesh* across the uncut intersection line on Σ, we introduce *solid interior hit adjacency* $\mathrm{adj}_{\mathrm{solid}}$ as a *virtual volumetric discretization* across Ω, directly bridging entrance and exit hits along interior ray chords without constructing or maintaining runtime tetrahedral meshes.

Solid adjacency predicate. Let $\mathcal{B}$ denote a watertight solid mesh with outward unit surface normals $\boldsymbol{n}(\cdot)$, and let $\mathbb{I}_{\mathrm{solid}}(\mathcal{B}) \in \{0,1\}$ indicate that $\mathcal{B}$ represents a solid continuum (e.g., an affine rigid body or tetrahedral solid). For a contour task t with separation direction $\boldsymbol{r} \in \mathbb{S}^2$, two hits $h_i, h_j \in \mathcal{H}_t$ of matching collision type satisfy the solid interior adjacency predicate:

$$\mathrm{adj}_{\mathrm{solid}}(h_i, h_j) = 1 \tag{41}$$

if and only if they satisfy four conditions:

(1) **Shared sweeping primitive:** Both hits share the same originating feature. For VF hits $h_i = (v, f_i)$ and $h_j = (v, f_j)$, they share source vertex $v \in \mathcal{V}_{\mathrm{s}}$; for FV hits $h_i = (f_i, v)$ and $h_j = (f_j, v)$, they share destination vertex $v \in \mathcal{V}_{\mathrm{d}}$; for EE hits $h_i = (e, e_i)$ and $h_j = (e, e_j)$, they share source edge $e \in \mathcal{E}_{\mathrm{s}}$.

(2) **Identical solid body component:** The struck target (or source) features belong to the same watertight solid mesh $\mathcal{B}$, with $\mathbb{I}_{\mathrm{solid}}(\mathcal{B}) = 1$.

(3) **Opposing piercing orientation:** The ray chord pierces into the interior at h_i and out of the interior at h_j. Using outward boundary normals and angular tolerance $\varepsilon_n > 0$:

$$\text{VF:} \quad \boldsymbol{n}(f_i) \cdot \boldsymbol{r} < -\varepsilon_n \quad \text{and} \quad \boldsymbol{n}(f_j) \cdot \boldsymbol{r} > \varepsilon_n, \tag{42}$$

$$\text{FV:} \quad \boldsymbol{n}(f_i) \cdot \boldsymbol{r} > \varepsilon_n \quad \text{and} \quad \boldsymbol{n}(f_j) \cdot \boldsymbol{r} < -\varepsilon_n, \tag{43}$$

$$\text{EE:} \quad \boldsymbol{n}(e_i) \cdot \boldsymbol{r} < -\varepsilon_n \quad \text{and} \quad \boldsymbol{n}(e_j) \cdot \boldsymbol{r} > \varepsilon_n. \tag{44}$$

For FV queries, because the ray is cast from destination vertex v along $-\boldsymbol{r}$ against the source body, the entrance condition $\boldsymbol{n}(f_i) \cdot (-\boldsymbol{r}) < -\varepsilon_n$ and exit condition $\boldsymbol{n}(f_j) \cdot (-\boldsymbol{r}) > \varepsilon_n$ invert the sign of the dot products with respect to separation vector $\boldsymbol{r}$.

(4) **Consecutive ray chord:** The hits are ordered along the ray parameter with $t(h_j) - t(h_i) > \varepsilon_d$, and h_j is the immediate subsequent exit hit with no intermediate boundary intersections along the segment between h_i and h_j.

GPU batched atomic hooking. Solid interior adjacency integrates into the GPU clustering pipeline (Algorithm 5) without requiring additional global synchronization. Following CSR inverted-list assembly, a specialized parallel kernel inspects all hits. For each entrance hit h_i satisfying $\boldsymbol{n} \cdot \boldsymbol{r} < -\varepsilon_n$ (or $\boldsymbol{n} \cdot \boldsymbol{r} > \varepsilon_n$ for FV), the thread iterates over hits sharing its sweeping primitive within the task's slice, identifies the immediate exit counterpart $h_j = \arg\min_{k:t(h_k)-t(h_i)>\varepsilon_d}(t(h_k)-t(h_i))$ matching the orientation predicates, and executes atomic root minimization:

$$P_{\max(r_i,r_j)} \leftarrow \min\left(P_{\max(r_i,r_j)}, \min(r_i, r_j)\right). \tag{45}$$

Because root values decrease monotonically, this atomic hooking operates concurrently with path compression and preserves lock-free cycle-free convergence.

Bilateral coupling and rigid momentum balance. Connecting entrance and exit hits across the interior unifies both penetration interfaces into a single connected component that preserves bilateral anchor coverage $|\mathcal{A}(\Gamma^*)|$. When combined with the translational projection in Sec. 7.2, the opposing normal forces at the entrance and exit surfaces cancel spurious internal moments and deliver a net translational thrust along $\boldsymbol{r}$, resolving deep penetrations in 3–8 Newton iterations on watertight geometries where pure surface traversal stalls indefinitely.

6.5 Coupling to Direction Optimization

All finite initial candidates are evaluated through Algorithm 4. For an accepted direction $\boldsymbol{r} \in \mathbb{S}^2$, its selected cluster defines the active geometric objective $\Phi(\boldsymbol{r})$. A local trial is computed in the tangent plane using

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{r}\boldsymbol{r}^{\top}, \qquad (\boldsymbol{P}\nabla^2\Phi\boldsymbol{P} + \lambda\boldsymbol{P})\boldsymbol{\delta} = -\boldsymbol{P}\nabla\Phi, \qquad \boldsymbol{r}^+ = \frac{\boldsymbol{r} + \boldsymbol{\delta}}{\|\boldsymbol{r} + \boldsymbol{\delta}\|} \tag{46}$$

$\boldsymbol{r}^+$ is not scored using a frozen hit set. It is submitted as a new task, after which VF/EE/FV ray casting, cluster culling, and cluster selection are repeated. The optimization therefore couples the spherical update to the discrete geometric support and accepts only objectives evaluated on a freshly reconstructed cluster.

7 Constitutive Models, Affine Coordinates, and Bending Formulation

The notation follows the main paper: a contour has two *logical sides*, source and destination, even for self-intersection within a single mesh. Thus, the subscripts s and d identify geometric roles rather than necessarily distinct mesh objects.

7.1 Affine-Body Coordinates, Inertial Potentials, and Orthogonality

Let $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{F}$ denote the mesh vertices, edges, and triangles. Soft vertices use world-space nodal positions $\boldsymbol{x}_i \in \mathbb{R}^3$ as degrees of freedom. Following affine body dynamics [Lan et al. 2022], an affine body b instead uses twelve scalar generalized degrees of freedom,

$$\boldsymbol{q}_b = (\boldsymbol{p}_b, \boldsymbol{a}_{b1}, \boldsymbol{a}_{b2}, \boldsymbol{a}_{b3}) \in \mathbb{R}^{12}, \qquad \boldsymbol{A}_b = [\boldsymbol{a}_{b1}\ \boldsymbol{a}_{b2}\ \boldsymbol{a}_{b3}] \in \mathbb{R}^{3\times 3} \tag{47}$$

where $\boldsymbol{p}_b \in \mathbb{R}^3$ is the center-of-mass translation and $\boldsymbol{A}_b \in \mathbb{R}^{3\times 3}$ represents linear affine deformation and orientation. These affine coordinates map a material-space vertex $\bar{\boldsymbol{x}}_i$ to its current spatial position via:

$$\boldsymbol{x}_i(\boldsymbol{q}_b) = \boldsymbol{p}_b + \boldsymbol{A}_b \bar{\boldsymbol{x}}_i \tag{48}$$

The corresponding constant vertex Jacobian mapping generalized displacements to nodal positions is

$$\boldsymbol{J}_i = \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{q}_b} = \begin{bmatrix} \boldsymbol{I}_3 & \bar{x}_{i1}\boldsymbol{I}_3 & \bar{x}_{i2}\boldsymbol{I}_3 & \bar{x}_{i3}\boldsymbol{I}_3 \end{bmatrix} \in \mathbb{R}^{3\times 12} \tag{49}$$

Consequently, all positional energies, including contact and untangling responses, share the same local derivatives; only the linear transfer map from vertex positions to global degrees of freedom changes.

For each affine body b, the generalized inertial potential over time step h is formulated as:

$$E_{\text{in}}^{\text{aff}}(\boldsymbol{q}_b) = \frac{1}{2h^2}(\boldsymbol{q}_b - \widetilde{\boldsymbol{q}}_b)^\top \boldsymbol{M}_b (\boldsymbol{q}_b - \widetilde{\boldsymbol{q}}_b) \tag{50}$$

where $\widetilde{\boldsymbol{q}}_b$ is the unconstrained inertial prediction, and the generalized mass matrix $\boldsymbol{M}_b$ integrates body mass m_b and material-space second moment of inertia:

$$\boldsymbol{M}_b = \operatorname{diag}(m_b \boldsymbol{I}_3, \boldsymbol{I}_{\text{mat}} \otimes \boldsymbol{I}_3) \in \mathbb{R}^{12\times 12}, \qquad \boldsymbol{I}_{\text{mat}} = \int_{\Omega_b} \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\top \rho \, \mathrm{d}V \tag{51}$$

To penalize non-rigid shearing and volume distortion, an affine body is regularized by an orthogonality strain energy:

$$E_{\text{ortho}}(\boldsymbol{A}_b) = \kappa_{\text{ortho}} \|\boldsymbol{A}_b^\top \boldsymbol{A}_b - \boldsymbol{I}_3\|_F^2 = \kappa_{\text{ortho}} \sum_{i=0}^{2}\sum_{j=0}^{2} (\boldsymbol{a}_{bi}^\top \boldsymbol{a}_{bj} - \delta_{ij})^2 \tag{52}$$

where $\kappa_{\text{ortho}} \geq 10^6$ is the shape-rigidity stiffness. The matrix gradient is $\nabla_{\boldsymbol{A}_b} E_{\text{ortho}} = 4\kappa_{\text{ortho}} \boldsymbol{A}_b(\boldsymbol{A}_b^\top \boldsymbol{A}_b - \boldsymbol{I}_3)$, yielding column-wise gradients:

$$\frac{\partial E_{\text{ortho}}}{\partial \boldsymbol{a}_{bi}} = 4\kappa_{\text{ortho}} \left[-\boldsymbol{a}_{bi} + \sum_{k=0}^{2} (\boldsymbol{a}_{bi}^\top \boldsymbol{a}_{bk}) \boldsymbol{a}_{bk} \right] \tag{53}$$

The corresponding 3×3 block Hessian components with respect to column vectors $\boldsymbol{a}_{bi}, \boldsymbol{a}_{bj}$ are:

$$\boldsymbol{H}_{ij} = 4\kappa_{\text{ortho}} \left[\boldsymbol{a}_{bj} \boldsymbol{a}_{bi}^\top + (\boldsymbol{a}_{bi}^\top \boldsymbol{a}_{bj}) \boldsymbol{I}_3 \right], \quad (i \neq j) \tag{54}$$

$$\boldsymbol{H}_{ii} = 4\kappa_{\text{ortho}} \left[\boldsymbol{a}_{bi} \boldsymbol{a}_{bi}^\top + (\|\boldsymbol{a}_{bi}\|^2 - 1) \boldsymbol{I}_3 + \sum_{k=0}^{2} \boldsymbol{a}_{bk} \boldsymbol{a}_{bk}^\top \right] \tag{55}$$

7.2 Translational Projection for Rigid-Body Interactions

Affine coordinates naturally encompass shear and non-uniform scaling modes. When resolving contact or untangling penetrations between two rigid bodies (or bodies constrained with high shape-stiffness regularizers $\kappa_{\text{ortho}} \geq 10^6$ to behave rigidly), allowing unconstrained linear affine degrees of freedom $\boldsymbol{A}_b$ to absorb large localized separation impulses can excite unphysical shearing artifacts or artificial mode coupling along the contact normal. To maintain physical rigidity and numerical robustness in rigid–rigid interactions, we project the contact Jacobian strictly onto the translational subspace:

$$\boldsymbol{J}_i^{\text{trans}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{56}$$

This restricts the separating response stencils to act purely as net translational impulses on the rigid body's center of mass, preventing high-frequency spurious shear deformation while deformable cloth meshes continue to utilize the full vertex-level nodal resolution. Together with the solid interior hit adjacency formulated in Sec. 6.4, this treatment ensures both topological completeness (retaining opposing entrance–exit pairs across the solid interior) and kinetic stability (concentrating response impulses purely along rigid degrees of freedom).

7.3 Coupled Affine-Deformable Incremental Potential

In the main paper, Eq. (1) is stated in terms of unreduced nodal positions $\boldsymbol{x} \in \mathbb{R}^{3N}$ for pedagogical clarity. When unreduced deformable thin shells (with nodal coordinates $\boldsymbol{x}_s \in \mathbb{R}^{3n_s}$) are coupled with α affine bodies (with generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{12\alpha}$), following the unified affine-deformable coupled IP framework of Chen et al. [2022] and Huang et al. [2025], the unified coupled incremental potential is defined as:

$$\begin{aligned} E(\boldsymbol{q}, \boldsymbol{x}_s) = {} & \frac{1}{2h^2}(\boldsymbol{x}_s - \widetilde{\boldsymbol{x}}_s)^\top \boldsymbol{M}_s (\boldsymbol{x}_s - \widetilde{\boldsymbol{x}}_s) + \Psi_s(\boldsymbol{x}_s) \\ & + \frac{1}{2h^2}(\boldsymbol{q} - \widetilde{\boldsymbol{q}})^\top \boldsymbol{M}_r (\boldsymbol{q} - \widetilde{\boldsymbol{q}}) + \Psi_r(\boldsymbol{q}) \\ & + \Psi_{\text{contact}}\left(\begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{x}(\boldsymbol{q}) \end{bmatrix} \right) + \sum_{C} \sum_{h \in \mathcal{S}_C^\star} E_h \left(\begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{x}(\boldsymbol{q}) \end{bmatrix} \right) \end{aligned} \tag{57}$$

where $\Psi_s(\boldsymbol{x}_s)$ collects deformable stretching and bending energies, $\Psi_r(\boldsymbol{q}) = \sum_{b=1}^{\alpha} E_{\text{ortho}}(\boldsymbol{A}_b)$ collects affine shape regularizations, Ψ_{contact} denotes proximity/barrier energies, and E_h are the untangling response stencils. Per time step, the coupled state is resolved by minimizing this joint objective:

$$(\boldsymbol{q}, \boldsymbol{x}_s)^{t+h} = \operatorname*{arg\,min}_{\boldsymbol{x}_s \in \mathbb{R}^{3n_s},\ \boldsymbol{q} \in \mathbb{R}^{12\alpha}} E(\boldsymbol{q}, \boldsymbol{x}_s) \tag{58}$$

matching the affine-deformable formulation in Huang et al. [2025].

7.4 Dihedral Bending Formulation and Gauss–Newton Hessian

In the ablation study of the main paper, we investigate the effect of tuning the bending stiffness κ_{bend} (from $\kappa_b = 0.05$ to $\kappa_b = 1.0$) on regularizing localized pinching and preserving surface/volume fidelity during untangling. Here, we present the exact mathematical formulation and Gauss–Newton Hessian used in our solver.

Hinge Geometry and Signed Dihedral Angle. A discrete bending element is defined on a pair of adjacent triangles sharing an interior mesh edge $e_0 = (\boldsymbol{x}_0, \boldsymbol{x}_1)$, with opposite wing vertices $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$. Let the directed edge vectors be:

$$\begin{aligned} &\boldsymbol{e}_0 = \boldsymbol{x}_1 - \boldsymbol{x}_0, \quad \boldsymbol{e}_1 = \boldsymbol{x}_2 - \boldsymbol{x}_0, \quad \boldsymbol{e}_2 = \boldsymbol{x}_3 - \boldsymbol{x}_0, \\ &\boldsymbol{e}_3 = \boldsymbol{x}_2 - \boldsymbol{x}_1, \quad \boldsymbol{e}_4 = \boldsymbol{x}_3 - \boldsymbol{x}_1 \end{aligned} \tag{59}$$

The unnormalized outward face normals of the two sharing triangles are $\boldsymbol{n}_1 = \boldsymbol{e}_0 \times \boldsymbol{e}_1$ and $\boldsymbol{n}_2 = \boldsymbol{e}_2 \times \boldsymbol{e}_0$. The signed dihedral angle $\theta(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \in (-\pi, \pi]$ is defined as:

$$\theta = s \arccos\left(\frac{\boldsymbol{n}_1 \cdot \boldsymbol{n}_2}{\|\boldsymbol{n}_1\|\|\boldsymbol{n}_2\|}\right), \qquad s = \operatorname{sign}\left((\boldsymbol{n}_2 \times \boldsymbol{n}_1) \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2)\right) \tag{60}$$

Analytical Angle Gradients. Following discrete shell kinematics [Bridson et al. 2002; Grinspun et al. 2003], the spatial gradient of the signed dihedral angle with respect to each of the four hinge vertices is:

$$\frac{\partial \theta}{\partial \boldsymbol{x}_2} = -\frac{\|\boldsymbol{e}_0\|}{\|\boldsymbol{n}_1\|^2}\boldsymbol{n}_1 \tag{61}$$

$$\frac{\partial \theta}{\partial \boldsymbol{x}_3} = -\frac{\|\boldsymbol{e}_0\|}{\|\boldsymbol{n}_2\|^2}\boldsymbol{n}_2 \tag{62}$$

$$\frac{\partial \theta}{\partial \boldsymbol{x}_0} = \frac{\boldsymbol{e}_0 \cdot \boldsymbol{e}_1}{\|\boldsymbol{e}_0\|\|\boldsymbol{n}_1\|^2}\boldsymbol{n}_1 + \frac{\boldsymbol{e}_0 \cdot \boldsymbol{e}_2}{\|\boldsymbol{e}_0\|\|\boldsymbol{n}_2\|^2}\boldsymbol{n}_2 \tag{63}$$

$$\frac{\partial \theta}{\partial \boldsymbol{x}_1} = -\frac{\boldsymbol{e}_0 \cdot \boldsymbol{e}_3}{\|\boldsymbol{e}_0\|\|\boldsymbol{n}_1\|^2}\boldsymbol{n}_1 - \frac{\boldsymbol{e}_0 \cdot \boldsymbol{e}_4}{\|\boldsymbol{e}_0\|\|\boldsymbol{n}_2\|^2}\boldsymbol{n}_2 \tag{64}$$

Notice that translational invariance is preserved: $\sum_{i=0}^{3} \frac{\partial \theta}{\partial \boldsymbol{x}_i} = \boldsymbol{0}$.

Bending Energy and Gauss–Newton Hessian. Let θ_{rest} be the rest-state dihedral angle, $\bar{A}_{\text{edge}} = \frac{1}{3}(A_{f1} + A_{f2})$ the rest dual edge area, and κ_{bend} the material bending stiffness. The discrete bending potential is:

$$E_{\text{bend}}(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \frac{1}{2}\kappa_{\text{bend}}\bar{A}_{\text{edge}}(\theta - \theta_{\text{rest}})^2 \tag{65}$$

Its nodal gradient vector is:

$$\nabla_{\boldsymbol{x}_i} E_{\text{bend}} = \kappa_{\text{bend}}\bar{A}_{\text{edge}}(\theta - \theta_{\text{rest}})\frac{\partial \theta}{\partial \boldsymbol{x}_i}, \quad i \in \{0, 1, 2, 3\} \tag{66}$$

The exact second derivative contains geometric curvature terms $\nabla^2\theta$ that can become indefinite under large bending deformations. To maintain unconditional positive semi-definiteness for robust Newton convergence, we adopt the Gauss–Newton approximation:

$$\nabla^2_{\boldsymbol{x}_i \boldsymbol{x}_j} E_{\text{bend}} \approx \kappa_{\text{bend}}\bar{A}_{\text{edge}}\frac{\partial \theta}{\partial \boldsymbol{x}_i}\left(\frac{\partial \theta}{\partial \boldsymbol{x}_j}\right)^{\mathsf{T}} \in \mathbb{R}^{3\times 3} \tag{67}$$

In our GPU implementation, the 4×4 block matrix $\boldsymbol{H}_{\text{bend}} \in \mathbb{R}^{12\times 12}$ is assembled directly from the outer products of the gradient vectors, providing guaranteed positive curvature along the bending mode without requiring expensive per-element eigenvalue clamping.

References

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. ACM, New York, NY, USA, 594–603.

Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. 2022. A unified newton barrier method for multibody dynamics. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, New York, NY, USA, 62–67.

Allen Hatcher. 2002. *Algebraic Topology*. Cambridge University Press, Cambridge, UK.

Kemeng Huang, Xinyu Lu, Huancheng Lin, Taku Komura, and Minchen Li. 2025. StiffGIPC: Advancing GPU IPC for stiff affine-deformable simulation. *ACM Transactions on Graphics* 44, 3 (2025), 1–20.

Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022. Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials. arXiv preprint arXiv:2201.10022.

Algorithm 5: GPU Batched Hit Graph Cluster Culling

Input: Compacted hit buffer $\mathcal{H}$, task identifiers $\{t(h)\}$, candidate mesh primitives, contour seed anchors $\{\mathcal{A}_0, \mathcal{A}_1\}$

Output: Selected per-task response hit masks $\{\mathcal{M}_t\}$ and component statistics

`// 1. Task-Aware Keyed-CSR Inverted Index`

foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
  foreach *incident primitive* $p \in \mathrm{supp}(h_i)$ **do**
    Atomically increment histogram counter for key $(t(h_i), p)$;
  end
end
Execute parallel exclusive prefix-sum over primitive key counts;
foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
  Scatter hit index i into the inverted list CSR rows of its incident primitives;
end

`// 2. Parallel Atomic Min-Hooking and Path Compression`

Initialize disjoint-set parent array: $P_i \leftarrow i$, $\forall i \in \{1, \dots, |\mathcal{H}|\}$;
repeat
  $\Delta \leftarrow 0$;
  foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
    Traverse candidate neighbors $\{h_j\}$ via incident CSR rows;
    if $\big(\mathrm{adj}_{\mathrm{prod}}(h_i, h_j) = 1 \ \vee\ (\mathbb{I}_{\mathrm{solid}} \wedge \mathrm{adj}_{\mathrm{solid}}(h_i, h_j) = 1)\big)$ ***and*** $\mathcal{B}_{t(h_i)}(h_i, h_j) = 1$ **then**
      $r_i \leftarrow \mathrm{Find}(i)$; $\quad r_j \leftarrow \mathrm{Find}(j)$;
      if $r_i \neq r_j$ **then**
        $r_{\max} \leftarrow \max(r_i, r_j)$; $\quad r_{\min} \leftarrow \min(r_i, r_j)$;
        $\mathrm{old} \leftarrow \mathrm{AtomicMin}(P_{r_{\max}}, r_{\min})$;
        if $\mathrm{old} > r_{\min}$ **then** $\Delta \leftarrow 1$;
      end
    end
  end
  foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
    $P_i \leftarrow P_{P_i}$; `// Parallel one-pass path compression`
  end
until *sticky device convergence flag* $\Delta = 0$;

`// 3. Parallel Root Statistics and Anchor Coverage`

Reset root statistics buffers for all candidate roots $r \in \{1, \dots, |\mathcal{H}|\}$;
foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
  $r \leftarrow \mathrm{Find}(i)$;
  Atomically accumulate local hit count $|K(\Gamma_r)|$ and canonical tie-breaker key $\kappa(\Gamma_r)$;
  Query contour-seed distance; atomically insert touched anchors into root hash table to aggregate coverage $|\mathcal{A}(\Gamma_r)|$;
end

`// 4. Lexicographical Multi-Pass Winner Selection`

foreach *active task* t **do**
  Filter eligible roots containing bilateral contour-anchored hits touching both $\mathcal{N}_1(\mathcal{A}_0)$ and $\mathcal{N}_1(\mathcal{A}_1)$;
  Execute 5-pass parallel reduction over candidate roots via Eq. (40) to select optimal component Γ_t^*;
end

`// 5. Output Response Hit Masking`

foreach *hit* $h_i \in \mathcal{H}$ ***in parallel*** **do**
  $t \leftarrow t(h_i)$; $\quad r \leftarrow \mathrm{Find}(i)$;
  $\mathcal{M}_t[i] \leftarrow \big(r == \mathrm{Root}(\Gamma_t^*) \ \wedge\ d_{\mathrm{hop}}(h_i) \leq d_{\max}\big) ? 1 : 0$;
end