A Penetration Metric for Deforming Tetrahedra using Object Norm

Jisu Kim[†] and Young J. Kim[†]

Abstract— In this paper, we propose a novel penetration metric, called *deformable penetration depth* PD_d, to define a measure of inter-penetration between two linearly deforming tetrahedra using the object norm [1]. First of all, we show that a distance metric for a tetrahedron deforming between two configurations can be found in closed form based on object norm. Then, we show that the PD_d between an intersecting pair of static and deforming tetrahedra can be found by solving a quadratic programming (QP) problem in terms of the distance metric with non-penetration constraints. We also show that the PD_d between two, intersected, deforming tetrahedra can be found by solving a similar QP problem under some assumption on penetrating directions, and it can be also accelerated by an order of magnitude using pre-calculated penetration direction. We have implemented our algorithm on a standard PC platform using an off-the-shelf QP optimizer, and experimentally show that both the static/deformable and deformable/deformable tetrahedra cases can be solvable in from a few to tens of milliseconds. Finally, we demonstrate that our penetration metric is three-times smaller (or tighter) than the classical, rigid penetration depth metric in our experiments.

I. INTRODUCTION

Handling soft objects is increasingly popular and becoming important in the robotics community. For instance, soft object manipulation is gaining broad attention recently due to robustness in manipulation and practical real-world demand [2]. In bio-inspired robotics, compliant actuation is a must to mimic that of biological entities. In all these areas, accurately simulating deformation for soft objects possibly with intensive contact is very crucial to ensure the robustness of robots and reduce their building cost.

The finite element method (FEM) is a general method to simulate deformable motion for soft objects with diverse material and structural properties, and it has been extensively studied for many decades in the area of computer-aided engineering (CAE), structural dynamics, computer animation and soft robotics [3], [4]. Typically, the FEM models a soft object with a network of many finite elements (FEs), such as tetrahedra. In order to accurately simulate contact dynamics using FEM, it is important to define a proper penetration measure between FEs and apply proper responsive forces to them.

However, unfortunately, there has been no rigorous formulation in the literature to define a penetration measure or metric for deformable FEs, in particular tetrahedra. In this paper, we propose a novel penetration metric, called *deformable penetration depth* PD_d, to define a measure of inter-penetration between two linearly deforming tetrahedra

[†]J. Kim and Y. J. Kim are with the Department of Computer Science and Engineering at Ewha Womans University in Korea `cwyh5526@ewhain.net,kimy@ewha.ac.kr`

using the object norm [1]. The main contribution of our paper can be summarized as follows:

- In order to define PD_d, we generalize the concept of object norm to a deformable case, and show that a distance metric for a tetrahedron deforming between two configurations can be found in closed form.
- Case 1: we show that the PD_d between an intersected pair of one static and one deforming tetrahedra can be found by solving a quadratic programming (QP) problem over all possible penetrating directions in terms of the distance metric with non-penetration constraints.
- Case 2: we show that the PD_d between an intersected pair of deforming tetrahedra can be found by solving a similar QP problem.
- Case 3: We approximate the case 2 using rigid penetration depth calculation based on the separating axis theorem, and demonstrate that this computation can be accelerated by an order of magnitude compared to the case 2 while the approximation error is kept to less than by 5%.
- We compare the metric results of our deformable penetration depth against those of rigid penetration depth and show that our results are three-times tighter than the rigid case.

The rest of this paper is organized as follows. In the next section, we briefly survey works relevant to penetration depth computation. In Sec. III, we show a closed-form solution of our penetration metric and subsequently apply it to obtain PD_d for static vs. deformable (Sec. IV) and deformable vs. deformable tetrahedral cases (Sec. V). In Sec. VI, we show that these solutions can be obtained more rapidly. In Sec. VII, we show various experimental results of our penetration depth algorithms and conclude the paper in Sec. VIII.

II. RELATED WORKS

A. Penetration Metric for Rigid Objects

Various penetration metrics for rigid objects have been suggested in the literature [5]. The (translational) penetration depth PD is the most well-known metric, defined by a minimal distance that separates two intersecting objects [6], [7]. Approximation methods have been proposed for convex models [8], [9], and for non-convex models, exact PD [10] and approximations [11], [12], [13], [14], and real-time algorithms [15] are suggested. Kim et al. [16] presented a hybrid algorithm that combines local optimization with machine learning.

The continuity of penetration metric has been considered; the Phong projection [17] and dynamic Minkowski sums [18] were used to compute continuous PD. Both positive and negative distances (i.e., PD), and the minimal distance value over a continuous time interval can be computed [19]. The generalized penetration depth PD_g is defined as a minimal rigid transformation that interpenetrating objects must undergo to resolve the penetration [20]. The works by [21], [22], [23] use penetration volume as a continuous penetration metric. Nirel and Lischinski [24] also presented a volume-based global collision resolution method. The pointwise penetration depth is defined as a distance between the deepest intersecting points of two objects [25].

B. Penetration Metric for Deformable Objects

Computing a penetration metric (or penetration depth) for deformable objects has been relatively less studied than for rigid objects. In particular, no attempt has been made to rigorously define such a metric, and our work tackles this problem.

Distance-fields-based representations are often used to compute penalty forces for deformable objects [26] and can be accelerated using GPUs [27], [28]. Heidelberger et al. [29] improved the consistency of penetration depth using a propagation scheme. There are few FEM-based methods that calculate penetration metric. [30] suggested material depth, an approximation of distance field, which is the distance between the interior and object-boundary points. The energy-based method [31] also used a signed distance in material space to compute the penetration depth. Layered depth images (LDIs) [32] are a data structure representing multiple layers of geometry rendered from a fixed viewpoint. Heidelberger et al. [33], [34] suggested a method to estimate penetration volume using pixel depths and normals from LDIs. The methods by [35] used a surface rasterization method based on LDIs [36], [22] to compute repulsive forces.

However, most of these existing works are heuristically defined and more seriously, they do not guarantee full separation of intersecting objects. In other words, all these metrics provide only a lower bound of penetration depth, not an upper bound, unlike our method.

III. METRIC FORMULATION

Given a tetrahedron $\mathcal{T} \in \mathbb{R}^3$ linearly deforming between rest $\mathbf{q_0} \in \mathbb{R}^{12}$ and deformed configurations $\mathbf{q_1} \in \mathbb{R}^{12}$, an arbitrary point $\mathbf{r} \in \mathcal{T}(\mathbf{q_0})$ and its counterpart $\mathbf{p} \in \mathcal{T}(\mathbf{q_1})$ can be represented using barycentric coordinate $\mathbf{b} = (b_1, b_2, b_3)$ which is constant during deformation [4] (also illustrated in Fig. 1):

$$\mathbf{r} = \mathbf{r}_0 + (\mathbf{r}_1 - \mathbf{r}_0)b_1 + (\mathbf{r}_2 - \mathbf{r}_0)b_2 + (\mathbf{r}_3 - \mathbf{r}_0)b_3$$
$$\mathbf{p} = \mathbf{p}_0 + (\mathbf{p}_1 - \mathbf{p}_0)b_1 + (\mathbf{p}_2 - \mathbf{p}_0)b_2 + (\mathbf{p}_3 - \mathbf{p}_0)b_3.$$

Then, the distance metric $\sigma(\mathbf{q_0}, \mathbf{q_1})$ using object norm [1], [37] can be formulated as:

$$\sigma(\mathbf{q_0}, \mathbf{q_1}) = \frac{1}{V} \int \|\mathbf{p} - \mathbf{r}\|^2 \, dV = 6 \int \|\mathbf{Db} + \mathbf{d}_0\|^2 \, d\mathbf{b}, \quad (1)$$

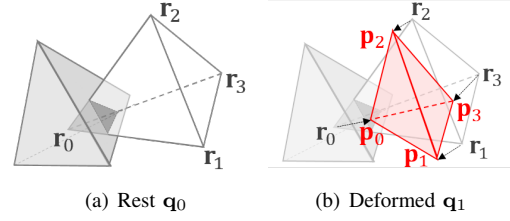
(a) Rest $\mathbf{q}_0$ (b) Deformed $\mathbf{q}_1$

Fig. 1. Linear deformation of a tetrahedron to resolve inter-penetration from the rest $\mathbf{q}_0$ to deformed configuration $\mathbf{q}_1$.

where $V = \frac{1}{6}$ is the volume of $\mathcal{T}$ in terms of the barycentric coordinate, $\mathbf{d}_i = (\mathbf{p}_i - \mathbf{r}_i)$, and $\mathbf{D} \in \mathbb{R}^{3 \times 3} = [\mathbf{d}_1 - \mathbf{d}_0 | \mathbf{d}_2 - \mathbf{d}_0 | \mathbf{d}_3 - \mathbf{d}_0]$. Eq. 1 induces a new tetrahedron $\mathcal{T}_d$, called *displacement tetrahedron*, with four vertices $\mathbf{d}_i = (\mathbf{p}_i - \mathbf{r}_i), i = 0, \cdots, 3$, and the object norm is a sum of squared distances from $\mathbf{d}_0$ for $\forall \mathbf{d} \in \mathcal{T}_d$.

Let $\mathbf{d}_i = (x_i, y_i, z_i)^T$. Then the closed-form solution of the proposed metric can be calculated as:

$$\sigma(\mathbf{q_0}, \mathbf{q_1}) = \frac{1}{10} \sum_{\forall i \geq j \in \{0, \ldots, 3\}} \mathbf{d}_i^T \mathbf{d}_j \quad (2)$$

Let $\mathbf{I}_d$ be the inertia tensor of the displacement tetrahedron $\mathcal{T}_d$. Then, the moments of inertia of $\mathbf{I}_d$, $\mathbf{I}_{xx}, \mathbf{I}_{yy}, \mathbf{I}_{zz}$, with *constant* mass density μ_d are:

$$\mathbf{I}_{xx} = \frac{1}{60} \mu_d |det(\mathbf{D})| \sum_{\forall i \geq j} \mathbf{d}_i^T [\mathbf{0} \ \mathbf{e}_2 \ \mathbf{e}_3] \mathbf{d}_j$$

$$\mathbf{I}_{yy} = \frac{1}{60} \mu_d |det(\mathbf{D})| \sum_{\forall i \geq j} \mathbf{d}_i^T [\mathbf{e}_1 \ \mathbf{0} \ \mathbf{e}_3] \mathbf{d}_j$$

$$\mathbf{I}_{zz} = \frac{1}{60} \mu_d |det(\mathbf{D})| \sum_{\forall i \geq j} \mathbf{d}_i^T [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{0}] \mathbf{d}_j,$$

where $\mathbf{e}_i$ is the standard basis for $\mathbb{R}^3$. Then, since $|det(\mathbf{D})| = 6V_d = 1$ where V_d is the volume of $\mathcal{T}_d$ (which is always $\frac{1}{6}$),

$$\sigma(\mathbf{q_0}, \mathbf{q_1}) = \frac{3}{\mu_d}(\mathbf{I}_{xx} + \mathbf{I}_{yy} + \mathbf{I}_{zz}) = \frac{3}{\mu_d} tr(\mathbf{I_d}) \quad (3)$$

Definition of Deformable PD We use Eq. 2 or 3 to define the deformable penetration depth PD_d as follows. Given an intersecting pair of linearly deforming tetrahedra, $\mathcal{T}_1, \mathcal{T}_2$, with the corresponding, initial configurations $\mathbf{q}_0^{\mathcal{T}_1}, \mathbf{q}_0^{\mathcal{T}_2}$, their PD_d is:

$$\mathrm{PD}_d(\mathcal{T}_1(\mathbf{q}_0^{\mathcal{T}_1}), \mathcal{T}_2(\mathbf{q}_0^{\mathcal{T}_2}))$$
$$= \min_{(\mathbf{q}_1^{\mathcal{T}_1}, \mathbf{q}_1^{\mathcal{T}_2}) \in \mathcal{C}} \sqrt{\sigma(\mathbf{q}_0^{\mathcal{T}_1}, \mathbf{q}_1^{\mathcal{T}_1}) + \sigma(\mathbf{q}_0^{\mathcal{T}_2}, \mathbf{q}_1^{\mathcal{T}_2})} \quad (4)$$

where $\mathcal{C} \subset \mathbb{R}^{24}$ is the contact (configuration) space imposed by $\mathcal{T}_1, \mathcal{T}_2$. In Sec. IV, we present our PD_d computation algorithm when $\mathcal{T}_1$ is static; i.e., $\mathbf{q}_0^{\mathcal{T}_1} = \mathbf{q}_1^{\mathcal{T}_1}$, and relax this restriction in Sec. V by allowing both $\mathcal{T}_1, \mathcal{T}_2$ deformable.

Generalization of Rigid PD Note that the proposed PD_d is a generalization of the classical rigid (or translational) penetration depth [6], [7]. Since the object norm in Eq. 3 can be interpreted as an average displacement of all the points inside a deformable object during deformation, when

the object purely translates by $\mathbf{d}$, the object norm will be equivalent to the squared norm of the translation vector $\mathbf{d}$ from Eq. 2: i.e.,

$$\sigma = \frac{1}{10} \sum \mathbf{d}_i^T \mathbf{d}_i = \mathbf{d}_i^T \mathbf{d}_i = \|\mathbf{d}\|^2 \qquad (5)$$

Thus, according to Eq. 4, $\mathrm{PD}_d = \mathrm{PD}$ when $\mathbf{q} \in \mathrm{SE}(3)$.

IV. STATIC VS DEFORMING TETRAHEDRA

In this section, we solve the optimization problem of Eq. 4 when $\mathcal{T}_1$ is static but $\mathcal{T}_2$ is still deformable. Since the contact space $\mathcal{C}$ in Eq. 4 can be realized by face-vertex (FV), vertex-face (VF) or edge-edge (EE) contacts (Fig. 2), the normal vector of the contact plane is used as a direction to separate $\mathcal{T}_1, \mathcal{T}_2$, which can also minimize Eq. 4. According to the separating axis theorem [38], the total number of possible separating directions that we need to consider is 44 since there are 8 VF (or FV) and 36 EE contact pairs between two tetrahedra.

In general, when both $\mathcal{T}_1, \mathcal{T}_2$ are deforming, even for linear deformation, the separating directions can be non-linear, which makes Eq. 4 hard to solve. In our work, to make the problem tractable, we assume that the separating directions are always obtained from the rest configurations $\mathbf{q}_0^{\mathcal{T}_1}, \mathbf{q}_0^{\mathcal{T}_2}$ of $\mathcal{T}_1, \mathcal{T}_2$, thus constant. Note that this is a reasonable assumption unless objects deform severely, which is reasonable for most practical robotic applications. Let us call a vertex in $\mathcal{T}_1$ or $\mathcal{T}_2$ *constrained* if they are involved in calculating the separating direction (i.e., the contact normal); otherwise, call it *free*. For example, if a face normal of $\mathcal{T}_1$ (e.g., $\mathbf{n}_{\mathrm{FV}}$ in Fig. 2(a)) is selected as a separating direction, the vertices incident to the face (e.g., $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2$, in Fig. 2(a)) are constrained. After optimization, free vertices may or may not be on the contact plane, but constrained vertices are always on the plane.

A. Separating Direction Calculation

Let $\{\mathbf{s}_i\} \subset \mathcal{T}_1, \{\mathbf{r}_i\} \subset \mathcal{T}_2(\mathbf{q}_0^{\mathcal{T}_2}), \{\mathbf{p}_i\} \subset \mathcal{T}_2(\mathbf{q}_1^{\mathcal{T}_2}), i = 0 \sim 3$ be the four vertices of $\mathcal{T}_1, \mathcal{T}_2(\mathbf{q}_0^{\mathcal{T}_2}), \mathcal{T}_2(\mathbf{q}_1^{\mathcal{T}_2})$, respectively. We first calculate the constant normal vector of a separating plane using the rest configurations. Let $\mathbf{n}$ be such a normal vector for each contact state. Then the following is the result of normal vector calculation:

$$\begin{aligned}
\mathbf{n}_{\mathrm{FV}} &= (\mathbf{s}_1 - \mathbf{s}_0) \times (\mathbf{s}_2 - \mathbf{s}_0) \\
\mathbf{n}_{\mathrm{VF}} &= (\mathbf{r}_1 - \mathbf{r}_0) \times (\mathbf{r}_2 - \mathbf{r}_0) \qquad (6) \\
\mathbf{n}_{\mathrm{EE}} &= (\mathbf{r}_1 - \mathbf{r}_0) \times (\mathbf{s}_1 - \mathbf{s}_0)
\end{aligned}$$

Furthermore, to make the separation direction consistent, the direction is decided based on the following rules:

1) $\mathbf{n}_{\mathrm{FV}}$ should be outward from $\mathcal{T}_1$.
2) $\mathbf{n}_{\mathrm{VF}}$ should be inward to $\mathcal{T}_2$.
3) $\mathbf{n}_{\mathrm{EE}}$ should point away from the non-contacting vertices of $\mathcal{T}_1$ (e.g., $\mathbf{s}_2, \mathbf{s}_3$ in Fig. 2(c)).

Note that the third case for $\mathbf{n}_{\mathrm{EE}}$ works only when the non-contacting vertices of $\mathcal{T}_1$ lie in the same half space (Fig. 3(a)).

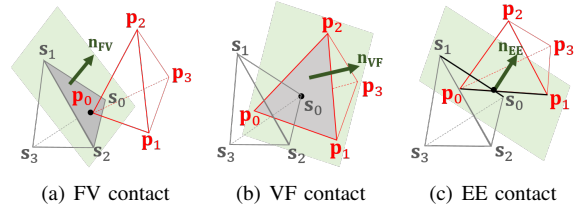(a) FV contact   (b) VF contact   (c) EE contact

Fig. 2. Three contact cases (FV, VF, EE) after penetration resolution. $\{\mathbf{s}_i\} \subset \mathcal{T}_1$ is static and $\{\mathbf{p}_i\} \subset \mathcal{T}_2$ is a deforming tetrahedron

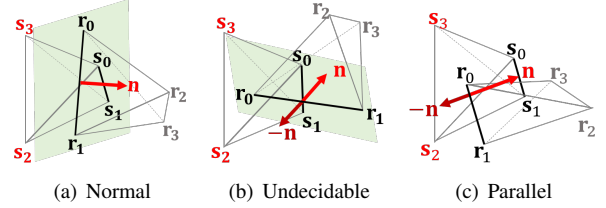(a) Normal   (b) Undecidable   (c) Parallel

Fig. 3. Various cases to deal with the separating direction $\mathbf{n}$ for an EE contact.

If this is not the case (Fig. 3(b)), both directions are tested to minimize Eq. 4. There also exist degenerate cases for $\mathbf{n}_{\mathrm{EE}}$ when two contacting EE pairs are parallel to each other. In this case, the normal is calculated from the shortest distance vector between the two EE pairs (Fig. 3(c)). When the two EE pairs are co-linear, the separating direction can be any of the vectors perpendicular to the edge. Thus, we can use the normal directions of the faces incident to the edges as a candidate separating direction, which is redundant as it can be covered by VF or FV cases.

B. Non-penetration Constraints

Eq. 4 is essentially a constrained optimization problem where the constraints are non-penetration constraints. Due to the constant assumption on the separating directions employed in Sec. IV-A, now we can linearize the non-penetration constraints and thus set up a solvable QP problem afterwards. Specifically, we write the non-penetration constraints for each contact case as follows:

1) FV case (Fig. 2(a)): since all the vertices $\mathbf{s}_i$ of the static tetrahedron $\mathcal{T}_1$ lie in the same half space, we simply impose non-penetration constraints for the vertices $\mathbf{p}_i$ of the deforming tetrahedron $\mathcal{T}_2$ as follows, where $\mathbf{s}_0$ is the vertex defining the FV contact:

$$\mathbf{n}_{\mathrm{FV}} \cdot (\mathbf{p}_i - \mathbf{s}_0) \geq 0 \qquad (7)$$

2) VF case (Fig. 2(b)): with constrained vertices ($\mathbf{p}_k, k = 0 \sim 2$) of $\mathcal{T}_2$ and free vertices ($\mathbf{s}_j, j = 0 \sim 3$) of $\mathcal{T}_1$, the free vertex $\mathbf{p}_3$ and the constrained vertex $\mathbf{p}_0$ on the plane, we formulate seven non-penetration constraints:

$$\begin{aligned}
\mathbf{n}_{\mathrm{VF}} \cdot (\mathbf{p}_k - \mathbf{p}_0) &= 0, \\
\mathbf{n}_{\mathrm{VF}} \cdot (\mathbf{s}_j - \mathbf{p}_0) &\leq 0, \qquad (8) \\
\mathbf{n}_{\mathrm{VF}} \cdot (\mathbf{p}_3 - \mathbf{p}_0) &\geq 0,
\end{aligned}$$

where $k \in \{0, \ldots, 2\}, j \in \{0, \ldots, 3\}$.

3) EE case (Fig. 2(c)): with constrained vertices $(\mathbf{s}_0, \mathbf{s}_1, \mathbf{p}_0, \mathbf{p}_1)$, the free vertices $(\mathbf{s}_2, \mathbf{s}_3)$ should lie in

the same half space, and free vertices $(\mathbf{p}_2, \mathbf{p}_3)$ should lie in the other half space. Without loss of generality, assuming that $\mathbf{p}_0$ is on the separating plane, we can formulate seven constraints.

$$
\begin{aligned}
\mathbf{n}_{EE} \cdot (\mathbf{s}_k - \mathbf{p}_0) &= 0 \\
\mathbf{n}_{EE} \cdot (\mathbf{p}_1 - \mathbf{p}_0) &= 0 \\
\mathbf{n}_{EE} \cdot (\mathbf{s}_j - \mathbf{p}_0) &\leq 0 \\
\mathbf{n}_{EE} \cdot (\mathbf{p}_j - \mathbf{p}_0) &\geq 0
\end{aligned}
\tag{9}
$$

where $k \in \{0, 1\}, j \in \{2, 3\}$.

C. Constrained Optimization

Since now we set up both an objective function (Eq. 3) and non-penetration constraints for each contact case (Eqs. 7,8,9), we solve the constrained optimization in Eq. 4. Since the objective function is quadratic and the constraints are linear, our problem is a QP problem. This problem is solvable as the objective function is semi-positive definite and the constraint space is convex. We use an off-the-shelf QP solver like Gurobi Optimizer [39] to effectively solve this problem.

V. DEFORMING VS DEFORMING TETRAHEDRA

In this section, we extend the optimization problem of Eq. 4 to the case where both $\mathcal{T}_1$ and $\mathcal{T}_2$ are deforming.

A. Non-penetration Constraints

The non-penetration constraints used for this section requires a new variable $\mathbf{p}_s \in \mathbb{R}^3$, which is a point on the separating plane to decide the position of the plane. For the static and deforming tetrahedron case, the position of a separating plane is automatically decided once we determine the separating direction since there will be at least one vertex of the static tetrahedron involved in contact. But now, not only the configurations of both tetrahedra but also the position of the separating plane should be optimized at the same time because there is no static vertex to decide the position. By introducing $\mathbf{p}_s$, it becomes easier to write the constraints since we do not have to consider which vertex to select as a point on the plane. The only difference between each contact case is the separating direction and a set of constrained vertices, $\mathcal{C}_s$ for $\mathcal{T}_1$ and $\mathcal{C}_p$ for $\mathcal{T}_2$; for instance, $\mathcal{C}_s = \{\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2\}, \mathcal{C}_p = \emptyset$ in Fig. 4(a) and $\mathcal{C}_s = \{\mathbf{s}_0, \mathbf{s}_1\}, \mathcal{C}_p = \{\mathbf{p}_0, \mathbf{p}_1\}$ in Fig. 4(b). Once they are decided, we can write the non-penetration constraints in a more general form than those presented in Sec. IV. With a chosen separating direction $\mathbf{n}$ for each contact case, the corresponding constrained vertices $\mathbf{s}_c \in \mathcal{T}_1$, $\mathbf{p}_c \in \mathcal{T}_2$ and the free vertices $\mathbf{s}_f \in \mathcal{T}_1$, $\mathbf{p}_f \in \mathcal{T}_2$ of each deforming tetrahedron $\mathcal{T}_1, \mathcal{T}_2$, the eight non-penetration constraints can be formulated as:

$$
\begin{aligned}
\mathbf{n} \cdot (\mathbf{s}_c - \mathbf{p}_s) &= 0, \ \forall \mathbf{s}_c \in \mathcal{C}_s, \\
\mathbf{n} \cdot (\mathbf{p}_c - \mathbf{p}_s) &= 0, \ \forall \mathbf{p}_c \in \mathcal{C}_p, \\
\mathbf{n} \cdot (\mathbf{s}_f - \mathbf{p}_s) &\leq 0, \ \forall \mathbf{s}_f \in \mathcal{F}_s, \\
\mathbf{n} \cdot (\mathbf{p}_f - \mathbf{p}_s) &\geq 0, \ \forall \mathbf{p}_f \in \mathcal{F}_p,
\end{aligned}
\tag{10}
$$

where $\mathcal{F}_s, \mathcal{F}_p$ are the set of free vertices for $\mathcal{T}_1, \mathcal{T}_2$, respectively.

The non-penetration constraints for the static/deforming tetrahedron in Sec. IV can be viewed as a special case of this form where $\mathbf{s}_c$ and $\mathbf{s}_f$ are fixed and $\mathbf{p}_s$ is chosen among constrained vertices. For example, in FV case, since $\mathcal{C}_p = \emptyset$, the second constraint in Eq. 10 can be ignored. Moreover, as $\mathbf{p}_s$ is chosen from $\mathcal{C}_s$, the first and third constraints in Eq. 10 are automatically satisfied and the rest of four constraints remain just like in Eq. 7. The VF case (Eq. 8) and EE case (Eq. 9) have seven constraints because the chosen constrained vertex removes one of the equality constraints; for example, since $\mathbf{p}_s = \mathbf{p}_0 \in \mathcal{C}_p$, $\mathbf{n} \cdot (\mathbf{p}_0 - \mathbf{p}_0) = 0$.
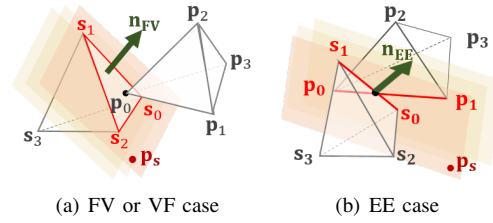

| (a) FV or VF case | (b) EE case |

Fig. 4. A new variable $\mathbf{p}_s$ constrains the location of the separating plane. Constrained vertices (red) should lie on the separating plane and free vertices (black) should lie inside either side of the plane.

B. Constrained Optimization

Similarly to the previous section, PD_d can be calculated by solving the QP problem of Eq. 4. Note that the non-penetration constraints in Eq. 10 is still linear since the separating directions are constant.

VI. ACCELERATION TECHNIQUE

In order to compute the PD_d, we have to solve a QP problem with constraints, each of which is associated with a separation direction, typically the entire solution taking tens of milliseconds using an off-the-shelf QP solver. However, if we can pre-select a set of candidate directions, possibly containing the optimal separating direction, we can significantly reduce the overall computation time. Under the hypothesis that the deformation of soft objects is not severe, we assume that the penetration (equivalently separating) direction for deformable objects is close to the penetration direction when the objects behave rigidly. Moreover, in our experiment, we have found that, on average, 52.33% of the result obtained by rigid penetration depth coincides with the optimal direction after running full optimization on PD_d.

To leverage this observation and accelerate PD_d computation, we calculate the rigid penetration depth PD based on the separating axis theorem (SAT) [38] and feed it to the optimization problem in Eq. 4; this is equivalent to using a single contact constraint in Eq. 4. There are various methods to find rigid penetration depth such as using GJK algorithm or Minkowski sums [6], [7]. However, we choose to use a SAT-based algorithm since our PD_d algorithm can be considered as a general case of PD and also can be implemented similarly to [40]. Specifically, in order to compute

PD, we use the overlap length of tetrahedra projected to each separating direction and take the projected, minimum overlap length as PD. A little caution has to be taken in our case, however, since tetrahedra are not symmetric. Thus, we need to project all tetrahedral vertices to 44 separating direction to calculate the minimum overlapping length.

Let $L_{\mathbf{n}}(\mathcal{T}) = \max(\{|(\mathbf{x}_i - \mathbf{x}_j) \cdot \mathbf{n}| \mid \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{T}\})$ for a tetrahedron $\mathcal{T}$, vertices $\mathbf{x}_i, \mathbf{x}_j$ in $\mathcal{T}$, and $\mathbf{n}$ is a given separating direction. The PD can be calculated as:

$$\mathrm{PD}(\mathcal{T}_1, \mathcal{T}_2) = \min(\{(L_{\mathbf{n}}(\mathcal{T}_1) + L_{\mathbf{n}}(\mathcal{T}_2) - L_{\mathbf{n}}(\mathcal{T}_1 \cup \mathcal{T}_2) \mid \forall \mathbf{n} \in N\}), \quad (11)$$

where $N = \{\mathbf{n}_{\mathrm{FV}}, \mathbf{n}_{\mathrm{VF}}, \mathbf{n}_{\mathrm{EE}}\}$ is a set of possible separating directions between $\mathcal{T}_1, \mathcal{T}_2$. Interestingly, instead of taking the "minimum direction" in Eq. 11, even if we take only the first six directions in ascending order of minimum distances, more than 90% of PD_d direction from Sec. V can be still found. This observation opens up a new possibility of reducing the possible error from this approximation while spending a little more time on search.

VII. RESULTS AND DISCUSSION

Now we show implementation results of our penetration depth algorithms for static/deformable, deformable/deformable, and deformable/deformable with acceleration. For our experiments, the target tetrahedra are randomly generated, which are bounded by a cube with a side length of 10. We tested 10^4 intersecting pairs of tetrahedra with different volume sizes ranging from 0.1 to 130, and compute their PD_d's. We implemented our algorithms using C++ on a Windows 10 PC with an AMD Rizen7 1700x 3.6GHz CPU, and 32GB memory. We used the Gurobi Optimizer to solve the QP problems.

A. Performance

As illustrated in Fig. 5, every result using PD_d resolves penetration. The average performance results of each case are shown in Table. I. The running time for the static/deformable case is slightly faster than the deformable/deformable case since a fewer number of variables are required to optimize. The accelerated deformable/deformable case can be calculated in 1.07 milliseconds on average.

TABLE I
PERFORMANCE STATISTICS

Criterion	STAT/DEF	DEF/DEF	DEF/DEF(ACCEL)
Performance	32.26ms	46.29 ms	1.07 ms
PD_d/PD	43.59 %	27.94 %	29.39%
(Standard Deviation)	(13.17%)	(3.15%)	(3.47%)

B. Discussion

To the best of our knowledge, there are no penetration depth algorithms for deformable models available to guarantee full separation as a result (i.e. penetration metric with a tight upper bound). All the known penetration depth algorithms for deformable models provide only a lower bound, which does not guarantee full penetration resolution. Thus, it is

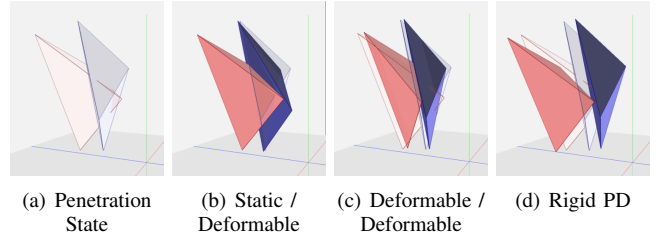(a) Penetration State   (b) Static / Deformable   (c) Deformable / Deformable   (d) Rigid PD

Fig. 5. Implementation results of deformable penetration depth and rigid penetration depth. (a) shows the initial penetrated state of two tetrahedra. In (b)∼(d), the solid colored tetrahedra are the results with $\mathrm{PD}_d = 0.5692, 0.3469, \mathrm{PD} = 1.496$, respectively.
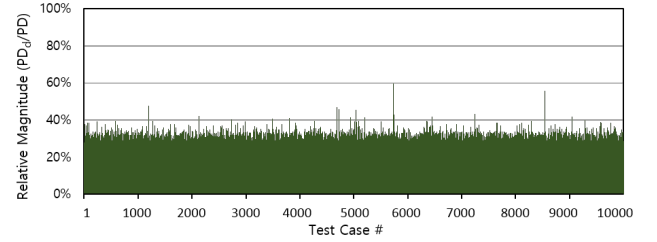

Fig. 6. Relative magnitude of PD_d over rigid penetration depth PD in 10^4 times of penetration resolution tests of randomly intersecting tetrahedron pairs. The average magnitude is 27.94% with the standard deviation of 3.15%.

unfair to compare our algorithms against existing algorithms for deformables and instead we compare our algorithms with rigid PD, which can be considered as an upper bound for deformable penetration depth. In this case, in order to show the tightness of the metric upper bounds, we calculate the relative magnitude of PD_d with respect to rigid penetration depth PD. Fig. 6 shows that the magnitude of PD_d is less than $\frac{1}{3}$ of that of the rigid case most times. This means that PD_d provides a much tighter deformation metric than using the rigid penetration depth.

VIII. CONCLUSION

We have formulated a new penetration metric based on object norm for a pair of intersecting tetrahedra undergoing linear deformation. The new metric, called deformable PD PD_d, optimizes an average displacement of all the points inside the tetrahedra under linear deformation to separate them. The deformable PD can be computed by solving a QP problem based on the distance metric with non-penetration constraints.

We have implemented three cases of computing PD_d, rigid vs. deformable, deformable vs. deformable with and without acceleration. Our experimental results show that we can compute PD_d in a fraction of milliseconds for intersecting, deformable tetrahedra.

There are a few limitations to our algorithm. To derive a tractable optimization problem for PD_d, we have assumed that the separation direction can be obtained from the rest pose of deforming tetrahedra. Even though it is straightforwardly extendable to a set of tetrahedra by applying our metric to each element in the set, it might be interesting to pursue a technique to accelerate this computation, to be more useful for FEM-type simulation. One plausible direction would be to combine the iterative contact space projection

technique [15] with our deformable metric. Our metric does not guarantee volume-preservation during deformation, which is another interesting future work.

REFERENCES

[1] K. Kazerounian and J. Rastegar, "Object norms: A class of coordinate and metric independent norms for displacement," in *in Flexible Mechanism, Dynamics and Analysis: ASME Design Technical Conference*, vol. 47, no. 1-2, 1992, pp. 271–275.

[2] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[3] S. S. Rao, *The finite element method in engineering*. Butterworth-heinemann, 2017.

[4] E. Sifakis and J. Barbic, "Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction," in *Acm siggraph 2012 courses*. ACM, 2012, p. 20.

[5] M. C. Lin, D. Manocha, and Y. J. Kim, *Handbook of Discrete and Computational Geometry*. CRC Press, 2017, ch. Collision and Proximity Queries.

[6] S. Cameron and R. Culley, "Determining the minimum translational distance between two convex polyhedra," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 591–596.

[7] D. Dobkin, J. Hershberger, D. Kirkpatrick, and S. Suri, "Computing the intersection-depth of polyhedra," *Algorithmica*, vol. 9, no. 6, pp. 518–533, 1993.

[8] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3d game objects," in *Game developers conference*, vol. 170, 2001.

[9] Y. J. Kim, M. C. Lin, and D. Manocha, "Incremental penetration depth estimation between convex polytopes using dual-space expansion," *IEEE transactions on visualization and computer graphics*, vol. 10, no. 2, pp. 152–163, 2004.

[10] P. Hachenberger, "Exact minkowksi sums of polyhedra and exact and efficient decomposition of polyhedra into convex pieces," *Algorithmica*, vol. 55, no. 2, pp. 329–345, 2009.

[11] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, "Fast penetration depth computation for physically-based animation," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 23–31.

[12] S. Redon and M. C. Lin, "A fast method for local penetration depth computation," *Journal of graphics tools*, vol. 11, no. 2, pp. 37–50, 2006.

[13] J.-M. Lien, "Covering minkowski sum boundary using points with applications," *Computer Aided Geometric Design*, vol. 25, no. 8, pp. 652–666, November 2008.

[14] J. M. Lien, "A simple method for computing minkowski sum boundary in 3d using collision detection," in *Algorithmic Foundation of Robotics VIII*, ser. Springer Tracts in Advanced Robotics, G. Chirikjian, H. Choset, M. Morales, and T. Murphey, Eds. Springer Berlin / Heidelberg, 2009, vol. 57, pp. 401–415.

[15] C. Je, M. Tang, Y. Lee, M. Lee, and Y. Kim, "PolyDepth: Real-time penetration depth computation using iterative contact-space projection," *ACM Trans. Graph.*, vol. 31, January 2012.

[16] Y. Kim, D. Manocha, and Y. J. Kim, "Hybrid penetration depth computation using local projection and machine learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4804–4809.

[17] Y. Lee and Y. J. Kim, "Phongpd: Gradient-continuous penetration metric for polygonal models using phong projection," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 57–62.

[18] Y. Lee, E. Behar, J.-M. Lien, and Y. J. Kim, "Continuous penetration depth computation for rigid models using dynamic minkowski sums," *Computer-Aided Design*, vol. 78, pp. 14–25, 2016.

[19] Y. Lee, A. Kheddar, and Y. J. Kim, "Continuous signed distance computation for polygonal robots in 3d," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[20] L. Zhang, Y. J. Kim, G. Varadhan, and D. Manocha, "Generalized penetration depth computation," *Computer-Aided Design*, vol. 39, no. 8, pp. 625–638, 2007.

[21] R. Weller and G. Zachmann, "Inner sphere trees for proximity and penetration queries," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[22] J. Allard, F. Faure, H. Courtecuisse, F. Falipou, C. Duriez, and P. G. Kry, "Volume contact constraints at arbitrary resolution," *ACM Trans. Graph.*, vol. 29, pp. 82:1–82:10, July 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778819

[23] B. Wang, F. Faure, and D. K. Pai, "Adaptive image-based intersection volume," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 97, 2012.

[24] D. Nirel and D. Lischinski, "Fast penetration volume for rigid bodies," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 239–250.

[25] M. Tang, M. Lee, and Y. J. Kim, "Interactive hausdorff distance computation for general polygonal models," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 74.

[26] S. Fisher and M. C. Lin, "Fast penetration depth estimation for elastic bodies using deformed distance fields," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 330–336.

[27] K. Hoff, A. Zaferakis, M. Lin, and D. Manocha, "Fast 3d geometric proximity queries between rigid and deformable models using graphics hardware acceleration," *UNC-CH Technical Report TR02-004*, 2002.

[28] A. Sud, N. Govindaraju, R. Gayle, I. Kabul, and D. Manocha, "Fast proximity computation among deformable models using discrete Voronoi diagrams," in *ACM SIGGRAPH*. ACM New York, NY, USA, 2006, pp. 1144–1153.

[29] B. Heidelberger, M. Teschner, R. Keiser, M. Müller, and M. H. Gross, "Consistent penetration depth estimation for deformable collision response," in *Vision, modeling and visualization 2004 : Proceedings, November 16-18, 2004, Stanford, USA*, B. Girod, M. A. Magnor, and H.-P. Seidel, Eds. AKA, 2004, pp. 339 – 346.

[30] G. Hirota, S. Fisher, A. State, C. Lee, and H. Fuchs, "An implicit finite element method for elastic solids in contact," in *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No. 01TH8596)*. IEEE, 2001, pp. 136–254.

[31] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw, "Robust quasistatic finite elements and flesh simulation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2005, pp. 181–190.

[32] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, "Layered depth images," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 231–242. [Online]. Available: http://doi.acm.org/10.1145/280814.280882

[33] B. Heidelberger, M. Teschner, and M. H. Gross, "Real-time volumetric intersections of deforming objects." in *VMV*, vol. 3, 2003, pp. 461–468.

[34] B. Heidelberger, M. Teschner, and M. Gross, "Detection of collisions and self-collisions using image-space techniques," *Journal of WSCG*, vol. 12, no. 3, pp. 145–152, 2004.

[35] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Soft tissue biomechanical modeling for computer assisted surgery*. Springer, 2012, pp. 283–321.

[36] F. Faure, J. Allard, F. Falipou, and S. Barbier, "Image-based collision detection and response between arbitrary volumetric objects," in *ACM Siggraph/Eurographics Symposium on Computer Animation*. Eurographics Association, 2008, pp. 155–162.

[37] L. Zhang, Y. Kim, and D. Manocha, "A fast and practical algorithm for generalized penetration depth computation," in *Proceedings of Robotics: Science and Systems*. Atlanta, USA, 2007.

[38] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: A hierarchical structure for rapid interference detection," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. Citeseer, 1996, pp. 171–180.

[39] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com

[40] Y. Lee and Y. J. Kim, "Simple and parallel proximity algorithms for general polygonal models," *Computer Animation and Virtual Worlds*, vol. 21, no. 3-4, pp. 365–374, 2010.