Received: 18 April 2023 | Revised: 4 May 2023 | Accepted: 4 May 2023

DOI: 10.1002/cav.2183

Inversion alleviation for stable elastic body simulation

JaeHyun Lee[1] | Seung-wook Kim[1] | Kiwon Um[2] | Min Hyung Kee[1] | JungHyun Han[1]

[1]Department of Computer Science, Korea University, Seongbuk-gu, Republic of Korea

[2]LTCI, Telecom Paris, IP Paris, Paris, France

Correspondence
JungHyun Han, Department of Computer Science, Korea University, Seongbuk-gu, Republic of Korea.
Email: jhan@korea.ac.kr

Funding information
Ministry of Science and ICT, South Korea, Grant/Award Number: IITP-2023-2020-0-01819; Information Technology Research Center, Grant/Award Numbers: IITP-2023-2020-0-01460, 2020-0-00861

Abstract

In general, it is not easy to simulate an elastic body that undergoes large deformations. Especially when its elements are inverted or tangled, that is, when its vertices penetrate its polygons, simulation often fails. In this paper, we propose a simple yet highly effective method for alleviating the inversion problems of elastic bodies. Our experiments made with typical optimization-based solvers demonstrate that the proposed method successfully stabilizes the solvers and produces visually plausible motions. We believe that our method can be widely adopted by a variety of state-of-the-art elastic-body simulators thanks to its simplicity.

KEYWORDS

elasticity, inversion, optimization, physically-based simulation

1 | INTRODUCTION

Simulation of elastic bodies is a fundamental tool for a variety of computer graphics applications such as animation films, video games, surgical simulations, and virtual try-on of clothing. Such applications urge us to develop a robust and reliable method for realistic yet stable and efficient simulation of elasticity. Among others, the nonlinear optimization approach, such as projective dynamics (PD)[1] and position-based dynamics (PBD)[2] methods, has received great attention thanks to its stability and effectiveness. PD and PBD solve for dynamics of elastic bodies via two steps: *prediction* and *projection*. In the prediction step, the solver integrates the physical state forward in time without considering elasticity properties. In the projection step, the state is projected onto the solution manifold, where the given elasticity properties are satisfied as much as possible. Here, the elasticity model varies depending on the target materials, for example, simple linear elastic or nonlinear hyperelastic materials, and the optimization methods can flexibly adopt the elasticity energy models for different elastic materials.

Unfortunately, the optimization methods often suffer from the *element inversion* problem in the prediction step, that is, a vertex in an element penetrates the opposite face, as depicted in Figure 1, particularly in large deformations. Consequently, the projection step may not work properly, leading to overall instability of the solver.

In order to tackle the inversion problem, we propose a simple yet highly effective alleviation method, which iteratively resolves the inverted elements after the prediction step. It enables the nonlinear optimization methods to produce stable motions of elastic bodies even in large deformations. Our method is implemented for three classical hyperelastic models, Neo-Hookean, St. Venant-Kirchoff, and co-rotational, with two optimization-based solvers, PD and PBD. The improved stability and robustness are demonstrated with many examples, where the standard versions of the two solvers easily fail to converge due to the inversion.

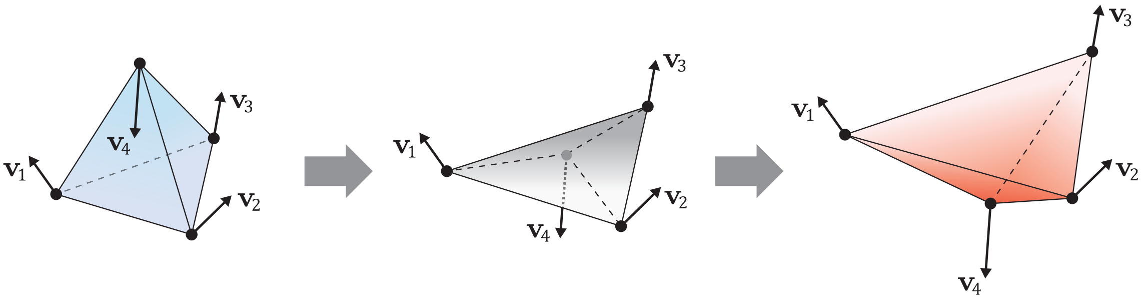


FIGURE 1 A tetrahedral element goes through inversion due to the velocity of the single mass point.

2 | RELATED WORK

Optimization-based solvers. Since the pioneering work of Baraff and Witkin,[3] there have been numerous efforts to improve optimization-based solvers for simulating deformable bodies. It is worth noting that such efforts focus on approximating the Hessian matrices in Newton's method. Particularly, Bouaziz et al.[1] proposed the PD, which approximates the Hessian with a constant matrix and then applies local projections. Wang[4] accelerated PD by applying Chebyshev semi-iterative approach,[5] which suits well for GPU implementation. Narain et al.[6] and Overby et al.[7] interpreted PD as a special form of alternating direction method of multipliers (also known as ADMM),[8] allowing PD to support hyperelastic materials such as Neo-Hookean and St. Venant-Kirchoff. Liu et al.[9] achieved the same goal by showing that PD can be classified as a special case of Limited-memory BFGS.[10]

On the other hand, some researchers extended the optimization problems into constrained ones subject to nonlinear constraints that enforce zero-potential energy, in order to simulate extremely stiff materials such as cloth. One of the most popular examples is the PBD,[2] which achieves fast approximations via dividing the original problem into smaller ones and then solving them sequentially. PBD has been steadily improved[11] and extended for fluids,[12] rigid bodies,[13] and unified simulations.[14] More recently, Macklin and Müller[15] successfully incorporated stable Neo-Hookean[16] constraint into PBD framework. However, such constraint cannot prevent element inversions from occurring during the prediction step in the PBD solver.

Handling inverted elements. While there have been significant improvements in optimization-based solvers, some researchers focused on handling inverted elements in three-dimension (i.e., tetrahedra), since inverting an element is physically impossible in the real world and such elements lead to erroneous results for the optimization-based solvers. The early works attempted to handle inverted elements with untangling meshes[17] but unfortunately such methods do not work for all cases. Therefore, Irving et al.[18] proposed a modified constitutive models in FEM, which enforces all the inverted elements to return to noninverted states. Furthermore, Schmedding et al.[19] proposed a method that always chooses the most appropriate direction to reinvert an inverted element based on a heuristic assumption.

3 | OPTIMIZATION-BASED SOLVERS

As an underlying solver for simulation of elastic bodies, we adopt the optimization-based approach. This section briefly presents how the elastic body simulation is formulated as an optimization problem.

Given a discrete representation of an elastic body with m vertices, its physical state at time t^n can be defined as a set of positions $\mathbf{x}^n (\in \mathbb{R}^{3m})$ and velocities $\mathbf{v}^n (\in \mathbb{R}^{3m})$. Integrating the state forward in time via the backward Euler scheme, we can evaluate the next state, $\mathbf{x}^{n+1}$ and $\mathbf{v}^{n+1}$, as follows:

$$\begin{aligned} \mathbf{x}^{n+1} &= \mathbf{x}^n + h\mathbf{v}^{n+1}, \\ \mathbf{v}^{n+1} &= \mathbf{v}^n + h\mathbf{M}^{-1}(\mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{int}}(\mathbf{x}^{n+1})), \end{aligned} \tag{1}$$

where h is the time step size, $\mathbf{M}$ is the mass matrix, $\mathbf{f}_{\text{ext}}$ represents the external forces such as gravity, and $\mathbf{f}_{\text{int}}(\mathbf{x})$ represents the internal forces.

For modeling the elasticity of an elastic body, the internal forces, $\mathbf{f}_{\text{int}}(\mathbf{x})$, can be derived from the elastic potential energy, W:

$$\mathbf{f}_{\text{int}}(\mathbf{x}) = -\nabla W(\mathbf{x}). \tag{2}$$

It is worth noting that W varies depending on the target material to model and defines the state's constraints. Without considering the internal forces in Equation (1), we can first evaluate the predicted state $\mathbf{y}$, that is, $\mathbf{y} = \mathbf{x}^n + h\mathbf{v}^n + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$. Then, taking the elasticity into account leads to the following optimization problem:

$$\underset{\mathbf{x}^{n+1}}{\arg\min} \underbrace{\frac{1}{2h^2}\|\mathbf{x}^{n+1} - \mathbf{y}\|^2_{\mathbf{M}} + W(\mathbf{x}^{n+1})}_{g(\mathbf{x}^{n+1})}. \tag{3}$$

Depending on the elastic potential energy function, W, adopted for the target material, the objective function $g(\mathbf{x}^{n+1})$ in Equation (3) can be solved differently. For example, Newton's method can be adopted as in the work of Baraff and Witkin[20] although this method requires solving a large system of equations at every time step. In contrast, PBD[2,11] is based on the Gauss-Seidel scheme that tackles the elasticity constraints individually in a number of iterations. On the other hand, PD[1] exploits an approximate Hessian matrix that remains constant over iterations and a quasi-Newton method that enables efficient simulations of general hyperelastic materials.[9]

4 | HYPERELASTIC ENERGY MODELS

This section briefly presents three hyperelastic energy models popularly used in the computer graphics field. We note that the elastic potential energy W in Section 3 is evaluated using integral of an energy density function Ψ, which can be defined differently depending on the model.

St. Venant Kirchhoff model (henceforth, StVK) represents its energy density function using a rotational invariant Green's strain, $\mathbf{E}$:

$$\Psi_{\text{StVK}} = \mu\mathbf{E} : \mathbf{E} + \frac{\lambda}{2}\text{tr}^2(\mathbf{E}), \tag{4}$$

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^{\text{T}}\mathbf{F} - \mathbf{I}), \tag{5}$$

where μ and λ are Lamé's first and second coefficients, respectively, $\mathbf{F}$ is the deformation gradient, and $\mathbf{I}$ is the identity matrix. Although StVK has been widely used thanks to its simplicity, it can easily suffer from failure at restoring inverted elements when an elastic body is significantly compressed.[21]

Being more restorative and stabler than StVK, co-rotational model (henceforth, CoRot) penalizes deviotric and scaling behavior. Its energy density function can be written as follows:

$$\Psi_{\text{CoRot}} = \frac{\mu}{2}\|\mathbf{F} - \mathbf{R}\|^2_{\text{F}} + \frac{\lambda}{2}\text{tr}^2(\mathbf{S} - \mathbf{I}), \tag{6}$$

$$\mathbf{F} = \mathbf{R}\mathbf{S}, \tag{7}$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{S}$ represents the nonrotational part that consists of scaling and shearing. Whereas CoRot enforces the inverted element to restore well, it does not prevent the inversion, which often makes the simulation implausible.

Neo-Hookean model (henceforth, NH) has gained its popularity in the computer graphics field, thanks to its superior volume-preserving feature,[16,22] and many variants of NH have been proposed. Among others, the most common variant utilizes a logarithmic term.[23] This model consists of two parts: a one-dimensional strain penalty term denoted as Ψ_{1D} and a three-dimensional volumetric penalty term denoted as Ψ_{3D}:

$$\Psi_{\text{NH}} = \Psi_{\text{1D}} + \Psi_{\text{3D}}, \tag{8}$$

$$\Psi_{\mathrm{1D}} = \frac{\mu}{2}(I_C - 3), \tag{9}$$

$$\Psi_{\mathrm{3D}} = -\mu \log J + \frac{\lambda}{2}(\log J)^2, \tag{10}$$

where I_C is the first Cauchy-Green invariant and $J = \det(\mathbf{F})$. Note that using NH can instantly fail when $J < 0$ because the logarithm function is only valid for positive values. In order to prevent this problem, the clamped Neo-Hookean model (henceforth, CNH)[9] replaces Ψ_{3D} with a quadratic function when $J < 0.5$. On the other hand, the stable Neo-Hookoean model (henceforth, SNH)[16,22] replaces the logarithmic term in Ψ_{3D} with a rest-stable quadratic function. We will use these two variants of NH in our experiments.

5 | INVERSION ALLEVIATION

The element inversion may happen in the prediction step of the underlying optimization-based solver, particularly when large deformations, collisions, and interactive manipulations are coming into play. Figure 1 shows an example. Such an inverted element represents a physically incorrect state, causing instabilities of the solver or yielding visually implausible results.

Our target hyperelastic material models are vulnerable to this inversion problem. Specifically, StVK in Equation (4) is unable to distinguish between inverted and noninverted elements and thus cannot restore to its uninverted state (See the StVK cube in Figure 2). Although CoRot can detect such inversions from Equation (6), it can fail to effectively resolve the problem when a large amount of inversion occurs, and consequently result in visually implausible simulations, as shown in Figure 4. In NH, the inversion often makes the instability problem even worse. In Equation (10), the determinant of deformation gradient, J, can be considered as a compression estimator, that is, if $J < 1$, it means volume compression. When an element is inverted, J becomes negative, making it impossible to evaluate Equation (10) and thus resulting in simulation failure. Even though CNH and SNH removed the singularities successfully and enhanced robustness to element inversion compared to NH, the element inversion still remains, as demonstrated in Figure 3.

We propose a simple yet highly effective method, which iteratively resolves the inverted elements after the prediction step. For the simplicity of exposition, we first tackle a single element inversion and then move to multiple inverted elements.

5.1 | Single inverted element

Consider a tetrahedral element that consists of four vertices. It is represented using the mass m_i, current position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, and predicted position $\mathbf{y}_i$, where i denotes the index of each vertex (i.e., $i \in \{1, 2, 3, 4\}$).

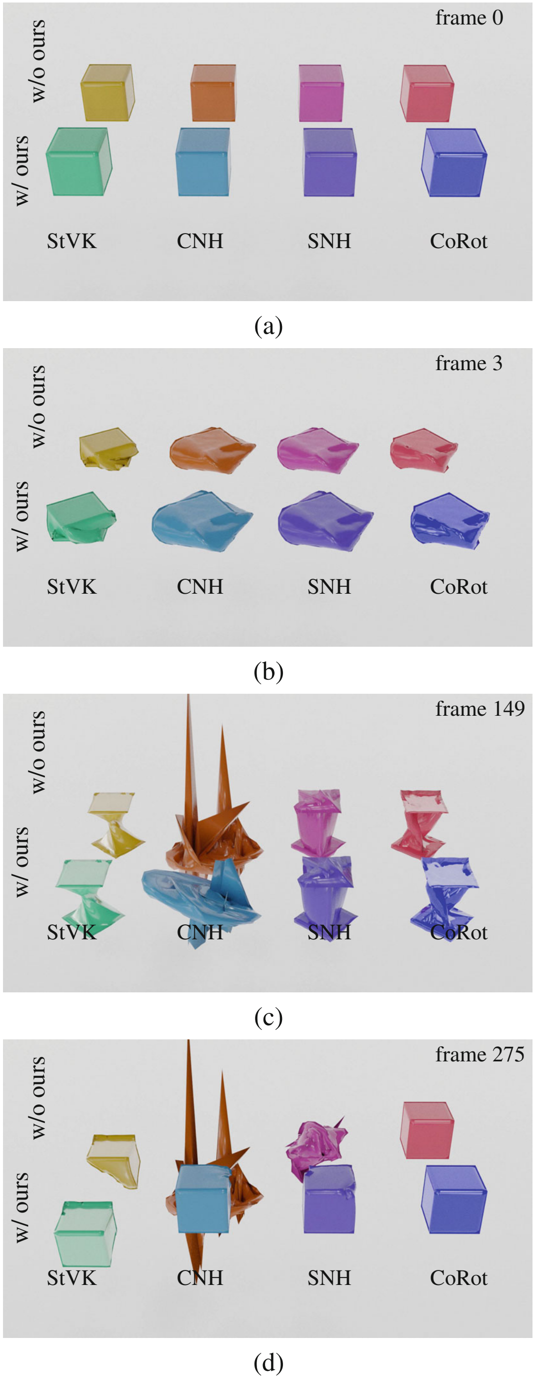
Note that an element inversion can be readily evaluated using the determinant of its deformation gradient; the negative determinant means inversion of the element. Once we find an inverted element at the predicted state, we resolve the inversion by reinverting $\mathbf{y}_i$ to $\mathbf{x}_i$ and predicting $\mathbf{y}_i$ again without the loss of momentum. To this end, we first compute the predicted velocity $\tilde{\mathbf{v}}_i$ as follows:

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{y}_i - \mathbf{x}_i}{h} = \mathbf{v}_i + h\mathbf{M}^{-1}\mathbf{f}_{\mathrm{ext}}. \tag{11}$$

Assuming perfectly inelastic collision of vertices for the inverted element during the inversion resolution, the post-resolution motion of the element can yield a rigid motion. Accordingly, we can evaluate the new velocity $\tilde{\mathbf{v}}^*$ of the element as follows:

$$\tilde{\mathbf{v}}^* = \frac{\sum_i m_i \tilde{\mathbf{v}}_i}{\sum_i m_i}. \tag{12}$$

The new prediction, then, can be evaluated as $\mathbf{y}_i = \mathbf{x}_i + h\tilde{\mathbf{v}}^*$.



(a)

(b)

(c)

(d)

FIGURE 2 Being twisted, the cubes are pressed downward so that their top faces cross over the bottom ones: (a) The initial states. (b) The states under pressure. (c) The states where the cubes' top faces are below the bottom ones. (d) The released states.

5.2 | Multiple inverted elements

We extend the idea for resolving a single element inversion to multiple inverted elements. Because of the connectivity of tetrahedra in a mesh, however, resolving the individual inverted elements does not guarantee the inversion-free prediction. One can easily imagine that the newly predicted position of a resolved element can invert its neighbor elements. Our solution to this problem is to repeat "traversing all elements of the given mesh and reinverting the inverted elements one by one." This iteration is made the number of pre-specified times. Such an iterative method is not guaranteed to yield

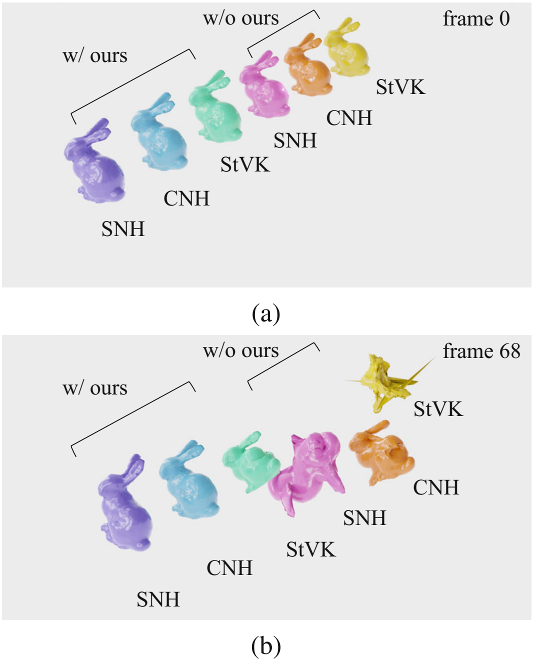


FIGURE 3 Bunnies moving back and forth: (a) The initial states. (b) Without the aid of our method, all hyperelastic models go through visually implausible motions. In contrast, our method enables stable simulations.

a perfectly inversion-free state. In all of our experiments with challenging large deformation scenarios, however, all the inversions are resolved within five iterations, except a few failure cases presented in Section 6.

6 | RESULTS

This section demonstrates the simulation stability improved by our method for two optimization-based solvers, PD and PBD. The aforementioned classical hyperelastic energy models are used for the experiments, and it is shown that our method can handle a wide range of material stiffness stably and robustly.

All our experiments are made with a 3.90 GHz AMD Ryzen 7 3800XT 8-Core processor and 16GB memory. In order to focus on the inversion problem, only the tetrahedral meshes are used in our experiments. The simulation results are visualized offline using Blender.[24] Table 1 summarizes the statistics and parameters in the experiments (In all experiments, the underlying solver is PD except the example presented in Figure 6b for which PBD is used.).

Cube: This experiment uses a simple elastic cube yet with a large deformation scenario, which includes compression and twisting. Figure 2 compares four hyperelastic models (i.e., StVK, CNH, SNH, and CoRot from left to right) with and without our method involved. It is demonstrated that the simulation results of the solver are significantly enhanced with our method. Without it, the simulations lead to either instabilities or visually implausible motions.

Bunny: This experiment is made with a higher-resolution mesh: The tetrahedron count of the bunny is twice that of the previous cube. The bunnies experience strong wagging motions horizontally such that the elements can be easily inverted. As clearly demonstrated in Figure 3, our method improves the simulation stability for the different energy models, SNH, CNH, and StVK.

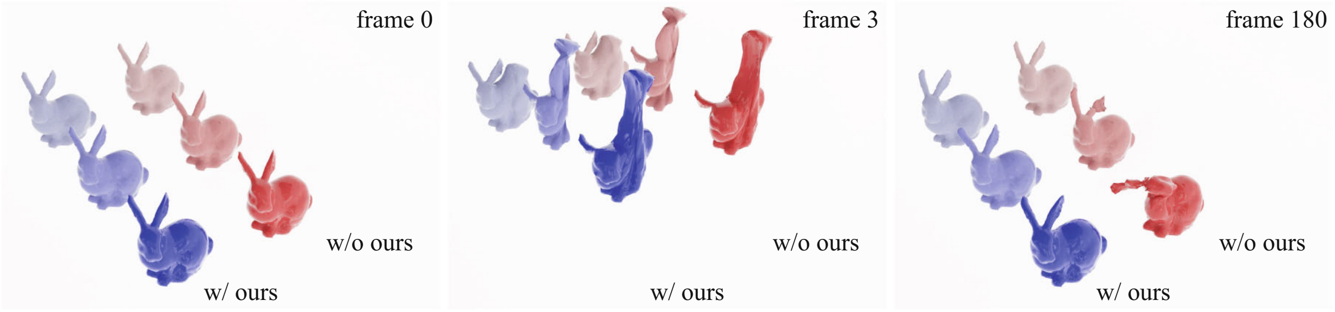
With CoRot, Figure 4 presents the results of simulation made with vertical wagging motions. The darker color implies the faster wagging speed. Despite the extremely fast wagging speed which yields severe element inversions, our method successfully resolves the problem.

Figure 5 presents the numbers of inverted elements over frames, for the fastest wagging speed in Figure 4. Observe that our method successfully and very rapidly reduces the number of inverted elements.

CASA2023: The scene shown in Figure 6 contains multiple instances of “CASA2023” alphabets modeled in CNH. The bluish alphabets are simulated with our method whereas the reddish ones are not.

TABLE 1 Statistics of the experiments (times in milliseconds).

	# elements	Time step size	Material type	Solver type	Solver time	Solver # iter.	Our overhead (%)	Our # iter.
Cube (Figure 2)	2911	33	CoRot	PD	12.9	10	6.5	3
Bunny (Figure 3)	5995	33	SNH	PD	27.8	10	40.7	30
Bunny (Figure 4)	5995	33	CoRot	PD	12.5	5	60.5	30
CASA2023 (Figure 6a)	6147	6.6	CNH	PD	20.3	10	14.3	5
CASA2023 (Figure 6b)	6147	3.3	CNH	PBD	16.8	20	15.1	5
CASA2023 iter. 0 (Figure 7)	6147	11.1	CNH	PD	13.9	10	0	0
CASA2023 iter. 2 (Figure 7)	6147	11.1	CNH	PD	35.5	10	3.5	2
CASA2023 iter. 3 (Figure 7)	6147	11.1	CNH	PD	31.4	10	5.4	3
Hippo (Figure 8)	8406	33	CoRot	PD	10.4	5	71.2	30
Dino iter. 0 (Figure 10)	3486	6.6	CoRot	PD	22.5	10	0	0
Dino iter. 1 (Figure 10)	3486	6.6	CoRot	PD	18.3	10	2.1	1
Dino iter. 5 (Figure 10)	3486	6.6	CoRot	PD	14.4	10	11.1	5
Dino iter. 15 (Figure 10)	3486	6.6	CoRot	PD	14.4	10	26.9	15
Dino iter. 50 (Figure 10)	3486	6.6	CoRot	PD	14	10	56.5	50
Tire (Figure 11)	3486	33	SNH	PD	42.9	20	28.5	30



(a) (b) (c)

FIGURE 4 Bunnies moving up and down: (a) The initial states. (b) Bunnies under vertical dragging forces. (c) The states after dragging are finished.

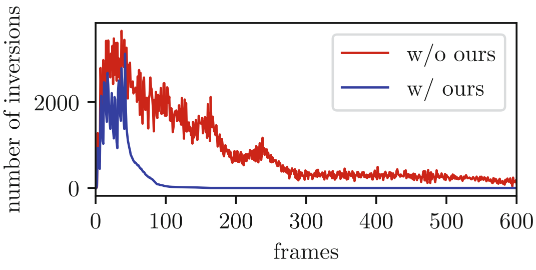


FIGURE 5 These graphs illustrate how the number of inversions changes over frames.

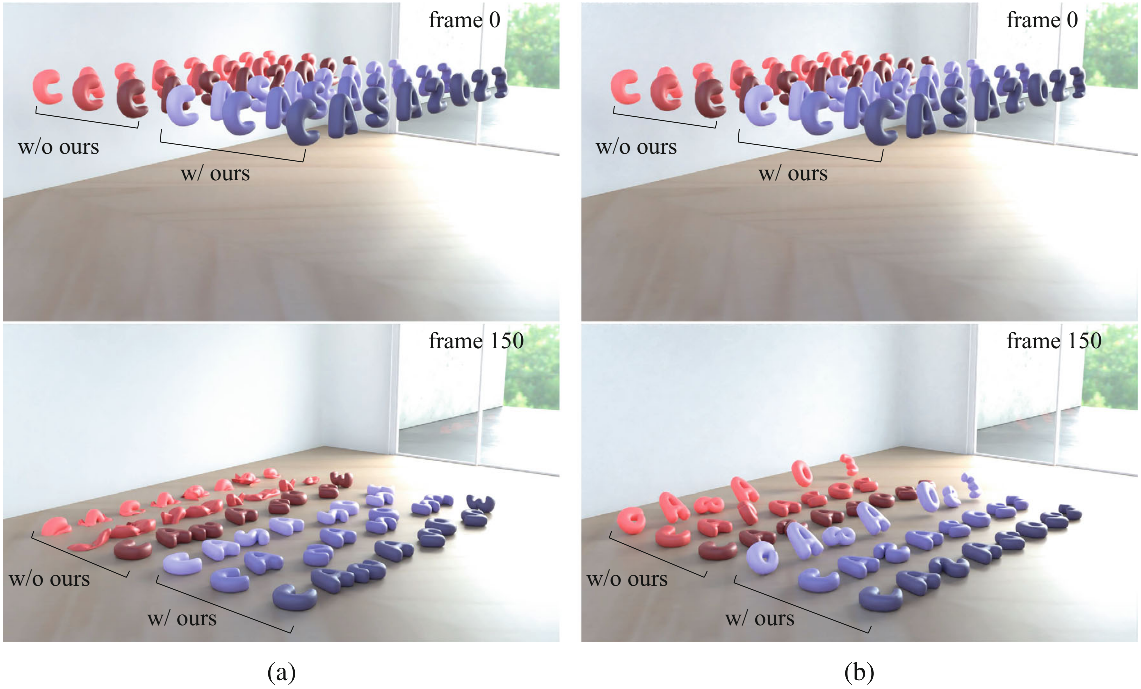


FIGURE 6 Elastic bodies freely fall onto the ground: (a) Simulation with PD. (b) Simulation with PBD.

The alphabets fall down under the gravity, and Figure 6a shows the results of simulation made with PD. The alphabets have different stiffness parameters, that is, Lamé's coefficients. The darker the color is, the stiffer the material is. From the stiffest to the softest materials, we assign 357, 337, and 208 to μ as well as 16442, 1428, and 138 to λ. All three materials are simulated stably with our method, whereas two softer materials experience simulation failure without our method.

Figure 6b shows the same setup as Figure 6a, where the darker the color is, the stiffer the material is. However, the underlying solver is PBD, and the solver iterations are made to vary depending on the material stiffness, that is, 40, 20, and 5 iterations are made for three materials respectively, from the stiffest to the softest. Regardless of whether our method is involved or not, the stiffest material is not bothered by the inversion problem whereas the softest one experiences simulation failure. In contrast, our method helps simulate stably the material of medium stiffness, which cannot be simulated using 20 solver iterations without our method.

Figure 7 compares the results of PD simulation made with different iteration counts. (Zero iteration implies that our method is not involved.) In this example, two iterations are sufficient for generating visually plausible motions, and additional iterations do not yield noticeably different results.

Hippo: In this experiment, the hippo is modeled in CoRot and its vertices are initialized with randomly selected velocities such that excessive inversions may happen from the beginning. Figure 8 compares the results of simulation made without our method (red) and with our method (blue). Observe that the red hippo experiences noisy motions particularly at its legs due to unresolved inversions. In contrast, the elastic motions of the blue hippo are kept natural over time as our method successfully resolves the inversions.

The graph of Figure 9 depicts the number of inverted elements that remain after each iteration of our method at the initial frame. This graph shows that most of inversions are resolved in five iterations.

Dino: In this experiment, we demonstrate the robustness of our method in an excessively stressed scenario. A dino (modeled in CoRot) rushes toward a wall with an extremely fast speed and collides with the wall, which results in severe element inversions. Figure 10 compares the results of simulation made with different numbers of iterations, that is, 0, 1, 5, 15, and 50 from left to right, where 0 implies that our method is not used. In this extreme example, it is observed that 15 iterations are sufficient for stable simulations.

Tire: This example shows the simulation result of an elastic body, a tire modeled in SNH. It keeps interacting with the ground in bouncing and rolling motions. As shown in Figure 11, our method neatly produces the desired dynamics.

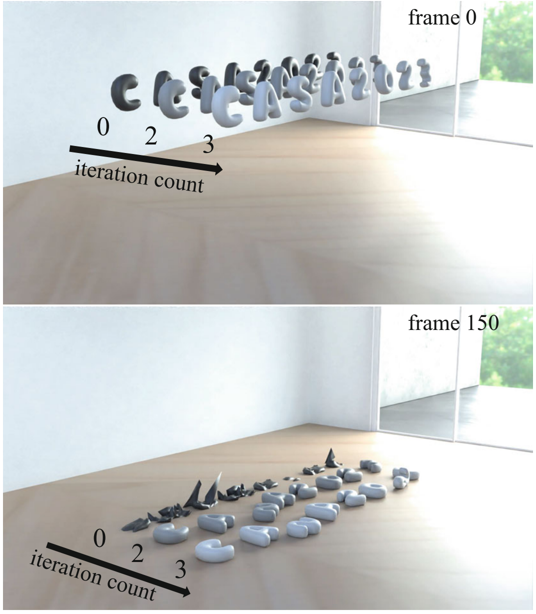


FIGURE 7 Simulation with varying number of iterations in our method.

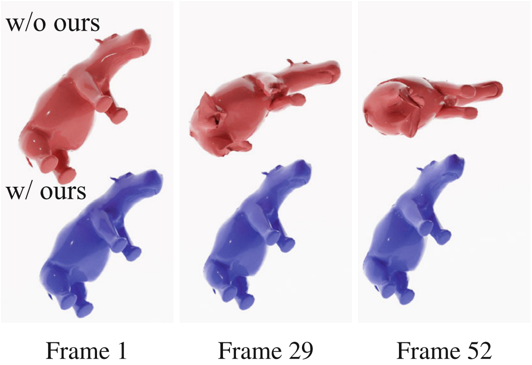


FIGURE 8 The vertices of elastic hippos are initialized with random velocities. Without our method, the hippo shows visually awkward motions.

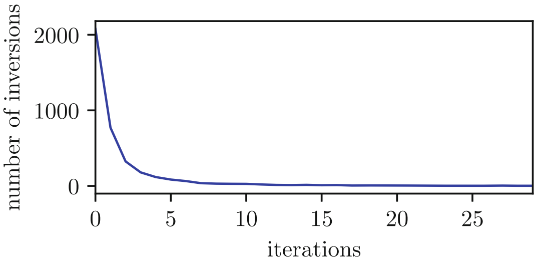


FIGURE 9 The graph illustrates the number of inversions over iterations at frame 0.

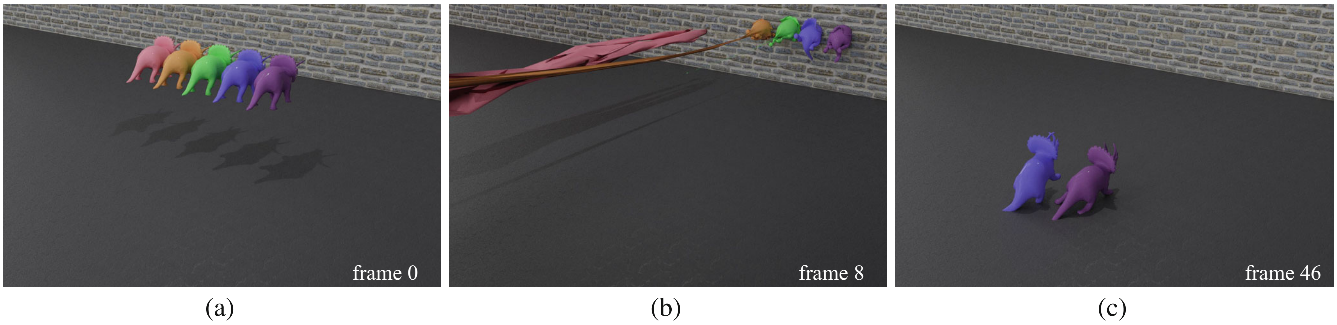


FIGURE 10 Dinosaur-shaped elastic bodies rush onto the wall: (a) The initial states. (b) The states, where the dinosaurs are colliding with the wall. (c) The states, where the dinosaurs are separated from the wall.

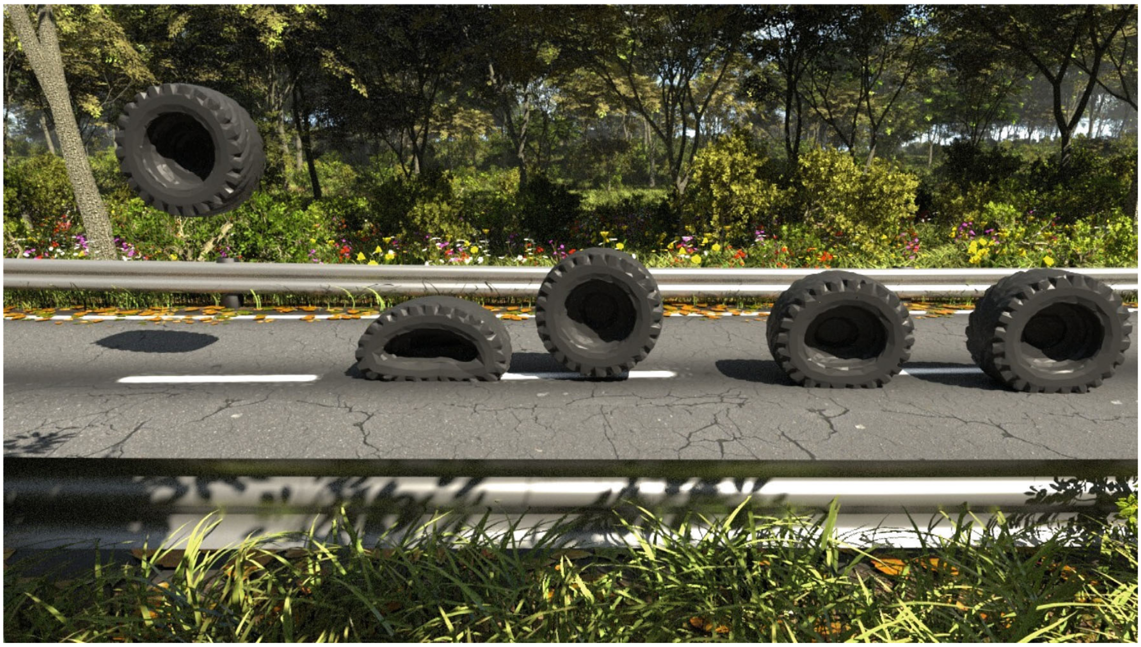


FIGURE 11 A tire is rolling on the road.

7 | DISCUSSION AND LIMITATIONS

Although our iterative method successfully resolves inversions within a small number of iterations, thus efficiently stabilizing the underlying solver, we note that it does not guarantee the inversion-free state particularly when multiple inversions happen due to a complicated deformation. Moreover, the iteration count of our resolving method, which directly affects the performance of our method, is heuristically chosen. Our experiments show that two or three iterations are sufficient for simple deformation scenarios such as **Cube** and **CASA2023** whereas more iterations are required for more complex deformation scenarios such as **Bunny**, **Hippo**, **Dino**, and **Tire**. Finding an optimal or sub-optimal iteration count, which may depend on the deformation scenario, remains as our future work.

8 | CONCLUSION

In this paper, we proposed an effective algorithm for alleviating the inversion problem that often hampers stable simulations of elastic bodies. Our experiments demonstrated that the proposed method successfully stabilized two typical optimization-based solvers, PD and PBD. Our experiments also considered not only different hyperelastic models but also different material stiffness parameters, and the results showed that our method successfully turned many unstable simulations into stable and visually plausible ones.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science and ICT, Korea, under the ICT Creative Consilience Program (IITP-2023-2020-0-01819), ITRC (Information Technology Research Center) Support Program (IITP-2023-2020-0-01460) and the grant No. 2020-0-00861.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

JungHyun Han https://orcid.org/0000-0001-6438-2974

REFERENCES

1. Bouaziz S, Martin S, Liu T, Kavan L, Pauly M. Projective dynamics: fusing constraint projections for fast simulation. ACM Trans Graph. 2014;33(4):1–11.
2. Müller M, Heidelberger B, Hennix M, Ratcliff J. Position based dynamics. J Vis Commun Image Represent. 2007;18(2):109–18.
3. Baraff D. Fast contact force computation for nonpenetrating rigid bodies. Proceedings of the 21st annual conference on computer graphics and interactive techniques. New York: Association for Computing Machinery; 1994. p. 23–34. https://dl.acm.org/doi/10.1145/192161.192168
4. Wang H. A Chebyshev semi-iterative approach for accelerating projective and position-based dynamics. ACM Trans Graph. 2015;34(6):1–9.
5. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. Numer Math. 1961;3(1):157–68.
6. Narain R, Overby M, Brown GE. ADMM ⊇ projective dynamics: fast simulation of general constitutive models. Symposium on computer animation. Goslar, Germany: Eurographics Association; 2016. p. 21–8. https://dl.acm.org/doi/10.5555/2982818.2982822
7. Overby M, Brown GE, Li J, Narain R. ADMM ⊇ projective dynamics: fast simulation of hyperelastic models with dynamic constraints. IEEE Trans Vis Comput Graph. 2017;23(10):2222–34.
8. Boyd S, Parikh N, Chu E. Distributed optimization and statistical learning via the alternating direction method of multipliers. Hanover, MA: Now Publishers Inc; 2011. https://dl.acm.org/doi/10.1561/2200000016
9. Liu T, Bouaziz S, Kavan L. Quasi-newton methods for real-time simulation of hyperelastic materials. ACN Trans Graph. 2017;36(3):1–16.
10. Nocedal J, Wright S. Numerical optimization. New York: Springer Science & Business Media; 2006. https://doi.org/10.1007/978-0-387-40065-5
11. Macklin M, Müller M, Chentanez N. XPBD: position-based simulation of compliant constrained dynamics. Proceedings of the 9th international conference on motion in games. New York: Association for Computing Machinery; 2016. p. 49–54. https://doi.org/10.1145/2994258.2994272
12. Macklin M, Müller M. Position based fluids. ACM Trans Graph. 2013;32(4):1–12.
13. Müller M, Macklin M, Chentanez N, Jeschke S, Kim T-Y. Detailed rigid body simulation with extended position based dynamics. Computer graphics forum. Volume 39. Wiley Online Library; 2020. p. 101–12. https://doi.org/10.1111/cgf.14105
14. Macklin M, Müller M, Chentanez N, Kim T-Y. Unified particle physics for real-time applications. ACM Trans Graph. 2014;33(4):1–12.
15. Macklin M, Muller M. A constraint-based formulation of stable neo-Hookean materials. Proceedings of the 14th ACM SIGGRAPH conference on motion, interaction and games; 2021. p. 1–7. https://doi.org/10.1145/3487983.3488289
16. Smith B, De Goes F, Kim T. Stable neo-Hookean flesh simulation. ACM Trans Graph. 2018;37(2):1–15.
17. Escobar JM, Rodrıguez E, Montenegro R, Montero G, González-Yuste JM. Simultaneous untangling and smoothing of tetrahedral meshes. Comput Methods Appl Mech Eng. 2003;192(25):2775–87.
18. Irving G, Teran J, Fedkiw R. Invertible finite elements for robust simulation of large deformation. Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on computer animation. Goslar, Germany: Eurographics Association; 2004. p. 131–40. https://doi.org/10.1145/1028523.1028541
19. Schmedding R, Teschner M. Inversion handling for stable deformable modeling. Vis Comput. 2008;24(7-9):625–33.
20. Baraff D, Witkin A. Large steps in cloth simulation. Proceedings of the 25th annual conference on computer graphics and interactive techniques. New York: Association for Computing Machinery; 1998. p. 43–54. https://doi.org/10.1145/280814.280821
21. Sifakis E, Barbic J. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. ACM siggraph 2012 courses. New York: Association for Computing Machinery; 2012. p. 1–50. https://doi.org/10.1145/2343483.2343501
22. Kim T, Eberle D. Dynamic deformables: implementation and production practicalities. ACM SIGGRAPH 2020 courses. New York: Association for Computing Machinery; 2020. p. 1–182. https://doi.org/10.1145/3532720.3535628
23. Bonet J, Wood RD. Nonlinear continuum mechanics for finite element analysis. Cambridge: Cambridge University Press; 2008. https://doi.org/10.1017/CBO9780511755446

24. Blender Online Community. Blender–a 3D modelling and rendering package. Stichting Blender Foundation, Amsterdam: Blender Foundation; 2018.

AUTHOR BIOGRAPHIES

JaeHyun Lee is a Master's student at Korea University, where he obtained his BS degree in Mechanical Engineering and Computer Science in 2021. His research interests focus on physics-based simulations of natural phenomena, especially in deformable bodies.

Seung-wook Kim is a postdoctoral researcher working at Korea University where he obtained his Ph. D degree in Computer Science at 2021. His research interests focus on physics animation in computer graphics.

Kiwon Um is an assistant professor of the Information Processing and Communications Laboratory (LTCI) in Telecom Paris, Institut Polytechnique de Paris. Before joining Telecom Paris, he worked as a postdoctoral researcher at the Technical University of Munich. He received his Ph.D., M.Sc., and B.Eng. degrees at Korea University. His research interests include physics-based simulations of natural phenomena and deep learning technologies for such simulations.

Min Hyung Kee is a PhD at Korea University where he obtained his Bachelor's degree in computer science in 2018. His research interests focus on physics-based simulation, large-scale simulation and differentiable-physics simulation.

JungHyun Han is a professor in the Department of Computer Science and Engineering at Korea University, where he directs the Interactive 3D Media Laboratory and VR/AR Research Center. Prior to joining Korea University, he worked at the School of Information and Communications Engineering of Sungkyunkwan University, in Korea, and at the Manufacturing Systems Integration Division of the US Department of Commerce National Institute of Standards and Technology. He received a BS degree in Computer Engineering at Seoul National University, an MS degree in Computer Science at the University of Cincinnati, and a PhD degree in Computer Science at the University of Southern California.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Lee JH, Kim S, Um K, Kee MH, Han JH. Inversion alleviation for stable elastic body simulation. Comput Anim Virtual Worlds. 2023;34(3):e2183. https://doi.org/10.1002/cav.2183