ハプティクス研究委員会　第 33 回研究会 (2024 年 2 月)

実時間大変形シミュレーションのための局所体積FEM

Local Volume-Based FEM for Real-Time Large Deformation Simulation
with Position-Based Dynamics

王思宇 1)，徐云岫 2)，長谷川晶一 3)
Siyu WANG, Yunxiu XU, and Shoichi HASEGAWA

1) 東京工業大学 工学院（〒 226-8503 神奈川県横浜市緑区長津田町 4259, siw131@haselab.net）
2) 東京工業大学 工学院（〒 226-8503 神奈川県横浜市緑区長津田町 4259, yunxiu@haselab.net）
3) 東京工業大学 工学院（〒 226-8503 神奈川県横浜市緑区長津田町 4259, hase@haselab.net）

概要: VR でのリアルなインタラクションが重要となり、柔軟物の実時間でのリアルな変形シミュレーションが求められている。従来の Corotated FEM を改良した Local Volume Based Corotated FEM を提案し、リアルタイムでの柔軟変形シミュレーションの高速化を提案する。この方法は、モデルをグループに分割し、ローカル座標を用い、計算効率を向上させせます。また、Position Based Dynamics (PBD) と制約条件を統合し、隣接グループを統合します。

キーワード： 有限要素法、大変形、Position Based Dynamics

1. Introduction

In Virtual Reality (VR), achieving rapid and accurate interactions is paramount for an enhanced user experience. With the evolution of the Metaverse, the ultimate goal is to establish a virtual world that closely mimics reality, making the realism of interactions and physical effects especially crucial. Haptic feedback, a vital component of immersive VR experiences, hinges on the precise and quick deformation of virtual objects. In the real world, object deformations are categorized into rigid and soft body deformations. For rigid bodies, linear models are typically employed due to their simplicity and computational efficiency. However, most real-world objects undergo deformations that are too complex to be accurately described by rigid body transformations alone, necessitating the use of nonlinear models. Although more accurate, nonlinear models are computationally intensive, posing challenges to both computational speed and accuracy.

Finite Element Method (FEM) is a numerical technique widely used across engineering to solve various problems by subdividing objects into a finite number of elements. When an object is divided into an excessively large number of elements, the computational cost escalates, particularly for soft bodies where a physical-based FEM requires the construction of massive matrices at every timestep for equation solving. To address this, the corotated FEM approach was proposed, which involves rotating the object to an unrotated coordinate system, stretching or compressing it, and then rotating it back. This method allows the stiffness matrix K to remain unchanged at each step, thereby speeding up computations. While most research on accelerating corotated FEM focuses on extracting rotation matrices more quickly and stably, few have explored acceleration through matrix computation. This study proposes dividing the object's elements into groups, each calculated in its local coordinate system, effectively reducing the overall matrix size and accelerating the corotated FEM method.

After dividing the object into several groups, a method is needed to integrate adjacent groups to maintain the object's integrity. We introduce a position-based constraint that ensures common points between neighboring groups occupy the same location. This constraint is translated into a constraint force that allows for a more natural connection between groups by iterative calculations. Unlike other external forces, such as gravity, this constraint force is calculated separately, acting as a deformation correction. This method, commonly referred to as the Position Based Method, directly controls positions. It can accommodate arbitrary position constraints and model a wide range of physical phenomena in a visually plausible manner due to its robustness and efficiency.

In summary, our method combines the advantages of corotated FEM and Position Based Dynamics by calculating the body in groups and ultimately merging them with PBD. Preliminary results indicate a significant improvement in speed while maintaining simulation accuracy.

2. Related Work

In order to address non-linear issue, Muller et al. proposed Corotated FEM [1] [2] to linearize the problem and made stiffness matrix precomputed, overcame a major obstacle in Finite Element computing used for real-time simulation. As corotated FEM proposed and used, handling rotations is an issue affecting simulation speed and error. There are robust ways to handle rotations, such as [3] and [4]. There are also other ways to accelerate FEM computing. For example, some researchers use GPU [5][6] for parallel computing to realize fast computation.

Solving large-dimension matrix equations is also affecting efficiency in FEM computing. Thus, in order to get low dimensional DOFs to reduce the problem size, model reduction has been used to reduce the complexity of a model while retaining its essential features. Barbic and James [7] utilized the modal derivative technique to determine an invariant subspace U when simulating nonlinear elasticity. In[8], a rapid method for computing linear subspaces was introduced, using Krylov iteration to estimate the subspace spanned by the eigenvectors.

Position Based Dynamics (PBD) [9] [10] has been popular in recent years since its robustness and easy implementation. Its high performance allows its potential applications in computer games and other interactive environments. Multiple studies used PBD to achieve robust and rapid results. Rigid body simulation in real-time is realized using PBD [11][12], yielding fast and stable simulations and enabling real-time interactions in games and VR environments. In [13], authors combined a continuum-based formulation with a position-based solver and achieved complex physics effects like anisotropy or elastoplasticity. Later, based on PBD, a more accurate and efficient method is proposed compared to the PBD, named XPBD [14]. XPBD extends PBD constraints to have a direct correspondence to well-defined elastic and dissipation energy potentials and introduces the concept of a total Lagrange multiplier to PBD, allowing for time step and iteration count independent constraint solving.

3. Implementation

3.1 Dynamic Equation

We first introduce how to develop a dynamic equation for the full body and then decompose it to a group scale. The general. Newton's second law describes dynamic equilibrium equation for the body, and it's written as:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{F}(\mathbf{x} - \mathbf{x}_0) = \mathbf{f} \tag{1}$$

where $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ are first and second order differentiation of $\mathbf{x}$. $\mathbf{M}$ and $\mathbf{C}$ are mass matrix and damping matrix.

As Euler implicit method allows for larger time steps without the risk of instability, we use this method to discretize Equation (2) and (3).

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \Delta t \mathbf{v}^{i+1} \tag{2}$$

$$\mathbf{M}\mathbf{v}^{i+1} = \mathbf{M}\mathbf{v}^i + \Delta t(-\mathbf{C}\mathbf{v}^{i+1} - \mathbf{F}(\mathbf{x}^{i+1} - \mathbf{x}^0) + \mathbf{f}^{i+1}) \tag{3}$$

In linear elasticity, we approximate that

$$\mathbf{F}(\mathbf{x} - \mathbf{x}_0) = \mathbf{K}(\mathbf{x} - \mathbf{x}_0) \tag{4}$$

where $\mathbf{K}$ is the stiffness matrix of the system. To make the stiffness matrix precomputed, we adopted corotated method [1] to simulate non-linear deformation. In this study, as we use local coordinates for each group rather than global coordinates, Equation (4) is written as:

$$\mathbf{K}(\mathbf{x} - \mathbf{x}_0) = \mathbf{R}\mathbf{K}(\mathbf{R}^{-1}(\mathbf{x} - \mathbf{x_{cm}}) - (\mathbf{x_0} - \mathbf{x_{0,cm}})) \tag{5}$$

where $\mathbf{x_{0,cm}}$ is the initial center of mass and $\mathbf{x_{cm}}$ is center of mass in run-time calculations.Use $\boldsymbol{C'} = \Delta t \boldsymbol{C}$ to simplify. We also use predicted position $\bar{\mathbf{x}}^{i+1}$ to represent positions adding external force (Equation (6)), and our iterative position constraints will correct it.

$$(\mathbf{M} + \mathbf{C}')\bar{\mathbf{x}}^{i+1} = (\mathbf{M} + \mathbf{C}')\mathbf{x}^i + \Delta t \mathbf{M}\mathbf{v}^i + \Delta t^2 \mathbf{f}_{ext}^{i+1} \tag{6}$$

Finally, we can give the dynamic equation applying corotated FEM for each group.

$$\begin{aligned}(\mathbf{M}_j + \mathbf{C}_j^{'})\mathbf{x}_j{}^{i+1} = (\mathbf{M}_j + \mathbf{C}_j^{'})\bar{\mathbf{x}}_j^{i+1} - \Delta t^2 \mathbf{R}_j{}^{i+1}\mathbf{K}_j \\ (\mathbf{R}_j{}^{i+1^T}(\mathbf{x}_j{}^{i+1} - \mathbf{x}_j{}_{cm}^{i+1}) - (\mathbf{x}_j{}^0 - \mathbf{x}_j{}_{cm}^0)) \\ +\Delta t^2 \mathbf{f_{c_j}^{i+1}}\end{aligned} \tag{7}$$

Here, $\mathbf{f_{c_j}^{i+1}}$ means constraint force between groups, and $\mathbf{j}$ represents group number. To implement this equation, we need to extract rotation from positions of vertices and take reference from shape matching method [4]. Then, we exert one rotation for all vertices in that group.

In order to keep the integrity of the model as we divide it into several groups, constraints that keep the nodes shared by the adjacent groups in the same position are added.

$$\mathbf{x}_{G1}^{i+1}(common) = \mathbf{x}_{G2}^{i+1}(common) \tag{8}$$

In Equation (7), we use $\mathbf{M}_{j,cm}\mathbf{x}_{j,cm}^{i+1}$ to calculate group center of mass, where $\mathbf{M}_{j,cm}$ is the mass distribution matrix of each group. Although this equation may be solved as it is, the following substitutions are made to reduce the error. We solve $\Delta\mathbf{x^{i+1}} = \mathbf{x^{i+1}} - \mathbf{\bar{x}^{i+1}}$ rather than $\mathbf{x^{i+1}}$. Then we formed a matrix equation $\mathbf{A}_j^{i+1}\Delta\mathbf{x}_j^{i+1} = \mathbf{b}_j^{i+1}$. In both $\mathbf{A}_j^{i+1}$ and $\mathbf{b}_j^{i+1}$, some variables need to be calculated at every time step. In order to accelerate calculation, we use a property of orthogonal matrix $\mathbf{R}_j^{i+1}$ ($\mathbf{R}^{-1} = \mathbf{R}^{T}$)). In addition, we use Rayleigh Damping $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$ to formulate damping matrix $\mathbf{C}$ and set $\beta = 0$ to to move the time variant rotation matrix $\mathbf{R_j^{i+1}}$. Then, the equation to be solved becomes

$$\begin{aligned}
&\left(\mathbf{I} + \left(\mathbf{M}_j + \mathbf{C}'_j\right)^{-1}\Delta t^2\mathbf{K}_j - \left(\mathbf{M}_j + \mathbf{C}'_j\right)^{-1}\Delta t^2\mathbf{K}_j\mathbf{M}_{j,cm}\right)\Delta\mathbf{y}_j^{i+1} \\
= \; & \Delta t^2\left(\mathbf{M}_j + \mathbf{C}'_j\right)^{-1}\mathbf{K}_j\left(\mathbf{x}_j^0 - \mathbf{x}_{j,cm}^0\right) \\
&- \left(\mathbf{M}_j + \mathbf{C}'_j\right)^{-1}\Delta t^2\mathbf{K}_j\mathbf{R}_j^{i+1\,T}\bar{\mathbf{x}}_j^{i+1} \\
&+ \left(\mathbf{M}_j + \mathbf{C}'_j\right)^{-1}\Delta t^2\mathbf{K}_j\mathbf{R}_j^{i+1^T}\bar{\mathbf{x}}_j^{i+1}\mathbf{M}_{j,cm} \\
&+ \Delta\mathrm{t}^2\mathbf{R}_j^{i+1\,-1}\,\mathbf{f}_\mathrm{c}(k,j)
\end{aligned} \tag{9}$$

$$\Delta\mathbf{x_j^{i+1}} = \mathbf{R_j^{i+1}}\Delta\mathbf{y_j^{i+1}} \tag{10}$$

This is the final dynamic equation to solve in our simulation.

3.2 Iterative Method

In the previous section, we derived a dynamic equation to be solved. However, it is difficult to solve the equation and satisfy the constraints directly. Thus, we came up with an iterative method and transformed the position constraints into constraint forces between groups. We use the difference between the average position of the common points of two adjacent groups and the current position to calculate the constraint force. At first, we set $\mathbf{f_c(j, k+1)} = 0$ and solve for the $\Delta\mathbf{x}_j^{i+1,k+1}$. Then, use it to calculate a new constraint force with Equation (12). After iterations, we correct the current position and velocity and then enter the next time step. Here $\mathbf{b_j^{i+1}}'$ represents other terms excluded for bind force. Here we introduce k, bind force coefficient, which determines the value of bind force.

$$\mathbf{A}_j^{i+1}\Delta\mathbf{x}_j^{i+1,n+1} = \mathbf{b}_j^{i+1\prime} + \mathbf{f_c(j, n)},\; j = 1, ..., m \tag{11}$$

$$\begin{aligned}
&\mathbf{f_c(j, n+1)} = \mathbf{f_c(j, n)} \\
&\quad + k(\bar{\mathbf{x}}_j^{i+1} + \Delta\mathbf{x}_j^{i+1,n+1} - \bar{\mathbf{x}}_l^{i+1} - \Delta\mathbf{x}_l^{i+1,n+1}) \\
&\quad (j = 1, ..., m)
\end{aligned} \tag{12}$$

4. Results

We evaluated our method concerning speed, accuracy, and stability. For speed, we used a cantilever beam divided into different numbers of elements and then compared the average calculation time for each timestep.

According to Figure 1, for the corotated FEM (1 group), the computational time exhibits a noticeable increase as the number of tetras grows. In contrast, dividing the computation into multiple groups (2, 3, 4, and 8 groups) consistently reduces the computational time across all tetras counts. This shows that the proposed method is effective in reducing matrix dimensions and accelerating calculation.

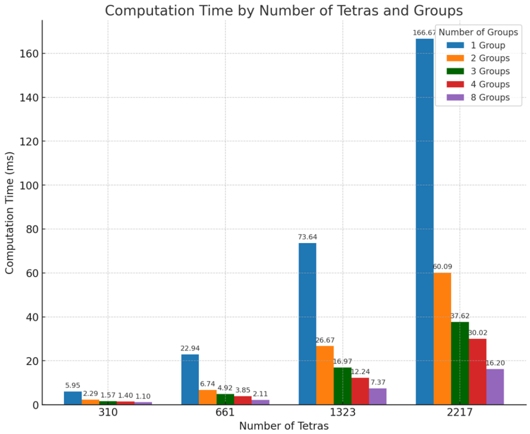

図 1: Calculation Speed by Different Number of Tetras and Groups

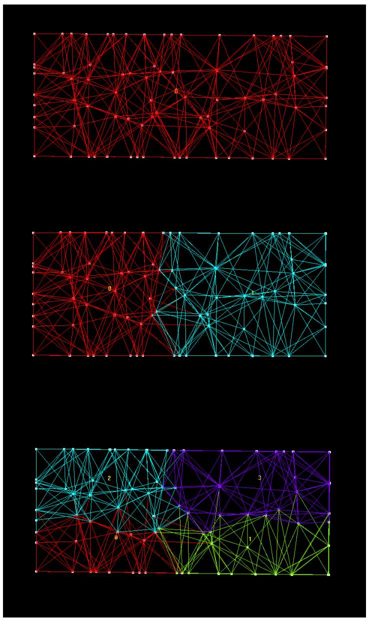

図 2: Comparison Accuracy Between Corotated FEM (Top) and Our Method (Middle, 2 Groups and Bottom, 4 Groups)

As for accuracy, we compared the beam divided into 4 groups and 2 groups, as well as the corotated FEM. We set the model and initial condition the same, with each vertex added 5 times its gravity. After the beam becomes balanced, we compared the final results of each beam. As shown in Figure 2, we can observe that there is no significant difference between corotated FEM and our method.

As for stability, we added 200 times the cube's gravity and observed its deformation. Upon application of the vertical force, the object undergoes significant distortion, elongating, and bending. As the force is released, the

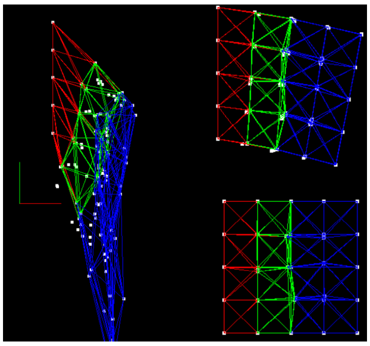

図 3: **After adding a huge force, the cube can recover to its initial state.**

object begins to return to its original form. This process shows that the cube can recover to a stable state under huge force.

5. Conclusion

In this study, we proposed a method to accelerate the FEM method for soft body simulation by dividing the model into groups and using PBD to control its shape integrity. The preliminary results show that our method can significantly reduce computation cost compared with the corotated FEM. In the meantime, our method shows no significant discrepancies were observed between the corotated FEM and the proposed method, suggesting that our method does not sacrifice accuracy for speed. In addition, the object could withstand and recover from substantial forces using our method, indicating that the method is robust and maintains stability under extreme conditions.

6. Limitation

One limitation of our method is that the constraint forces between groups introduce instability. If the bind force coefficient is set too large, the simulation will crash. We need to fine-tune the coefficient to realize a stable result. In the future, we will aim at building more robust constraints to realize stable simulation and apply it to haptic rendering algorithm.

参考文献

[1] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 49–54, 2002.

[2] Matthias Müller and Markus H Gross. Interactive virtual materials. In *Graphics interface*, Vol. 2004, pp. 239–246, 2004.

[3] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM transactions on graphics (TOG)*, Vol. 24, No. 3, pp. 471–478, 2005.

[4] Tassilo Kugelstadt, Dan Koschier, and Jan Bender. Fast corotated fem using operator splitting. In *Computer Graphics Forum*, Vol. 37, pp. 149–160. Wiley Online Library, 2018.

[5] Peter Huthwaite. Accelerated finite element elastodynamic simulations using the gpu. *Journal of Computational Physics*, Vol. 257, pp. 687–707, 2014.

[6] Serban Georgescu, Peter Chow, and Hiroshi Okuda. Gpu acceleration for fem-based structural analysis. *Archives of Computational Methods in Engineering*, Vol. 20, pp. 111–121, 2013.

[7] Jernej Barbič and Doug L James. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM transactions on graphics (TOG)*, Vol. 24, No. 3, pp. 982–990, 2005.

[8] Yin Yang, Dingzeyu Li, Weiwei Xu, Yuan Tian, and Changxi Zheng. Expediting precomputation for reduced deformable simulation. *ACM Trans. Graph*, Vol. 34, No. 6, 2015.

[9] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. In *Computer graphics forum*, Vol. 25, pp. 809–836. Wiley Online Library, 2006.

[10] Jan Bender, Matthias Müller, and Miles Macklin. A survey on position based dynamics, 2017. *Proceedings of the European Association for Computer Graphics: Tutorials*, pp. 1–31, 2017.

[11] Crispin Deul, Patrick Charrier, and Jan Bender. Position-based rigid-body dynamics. *Computer Animation and Virtual Worlds*, Vol. 27, No. 2, pp. 103–112, 2016.

[12] Jan Bender, Kenny Erleben, and Jeff Trinkle. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, Vol. 33, pp. 246–270. Wiley Online Library, 2014.

[13] Jan Bender, Dan Koschier, Patrick Charrier, and Daniel Weber. Position-based simulation of continuous materials. *Computers & Graphics*, Vol. 44, pp. 1–10, 2014.

[14] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pp. 49–54, 2016.