HOOD-report

Yuchen Liu

November 25, 2023
v0.1.1.0

Contents

Abstract

A report after reading HOOD [1]. Including the analysis of its method and possible improvements.

1 Innovations

- Using graph neural networks. Learning local states. Based on MeshGraphNets [2].
- Build a message-passing scheme over a hierarchical graph that interleaves propagation steps at different levels of resolution.
- Using physics-based loss function which is an incremental potential for implicit time stepping.
- Using a simple and efficient graph coarsening strategy which allowed the network implicitly learn transition etween graph levels.

2 Garment Preprocessing

2.1 Build up basic graph and extensions

In computer graphics, a common method to model garments is to build a garment mesh of vertices and edges. Apparently, this structure is easy to translate into a graph consisting of the same vertices and edges. Moreover, the graph augments *body edges* : the edge between vertices and nearest body node.

Feature vectors:

- To nodes, v_i includes velocity $\vec{v}$, normal vector $\vec{n}$, mass $\vec{m}$, node type, and local material parameters($\mu_{Lame}, \lambda_{Lame}$ and $k_{bending}$).
- To edges, e_{ij} includes the relative position between their two nodes.

2.2 Hierarchical graph construction

In MeshGraphNets [2], the simulation was implemented by the message passing through the graph. But in cloth simulation, the result is sensitive to the number of message passing steps, which makes it hard to manually set the number of steps. To address this problem, a message-passing scheme over a hierarchical graph that interleaves propagation steps at different levels of resolution was applied.

The hierarchical graph is a combination of several levels of coarsened garment graphs which were built by recursively applying the algorithm below:

input : fine graph $G_f(V_f, E_f)$
output: coarse graph $G_c(V_c, E_c)$ $\triangleright$ *$G_c \subseteq G_f, V_c \subseteq V_f, E_c \subseteq E_f$*
$v_{center} \leftarrow$ center of G_f $\triangleright$ *v_{center}'s eccentricity is equal to the graph's radius.*
for v_i in V_f **do**
    $d_i \leftarrow distance(v_{center}, v_i)$
$V_c \leftarrow \{v_i \in V_f | d_i \mod 2 = 0\}$
$E_c \leftarrow \{\}$
$V^{interm} \leftarrow \{v_i \in V_f | d_i \mod 2 = 1\}$
for $v_i \in V^{interm}$ **do**
    $V^{from} \leftarrow \{v_j \in V_f | e_{ij} \in E_f \ \& \ d_j = d_i - 1\}$ $\triangleright$ *$V^{from} \subseteq V_c$*
    $V^{to} \leftarrow \{v_j \in V_f | e_{ij} \in E_f \ \& \ d_j = d_i + 1\}$ $\triangleright$ *$V^{to} \subseteq V_c$*
    for v_j in V^{from} **do**
        for v_k in V^{to} **do**
            $push(e_{jk}, E_c)$ $\triangleright$ *e_{jk} connect nodes in V_c, thus $e_{jk} \in E_c$*

Through this algorithm, we build coarsened graph G_c from the input graph G_f, and this coarsened level is the cornerstone of Section 3.2.

3 Message Passing

In the GNN for simulation series, the GNN is built up by MLPs which learn the relationship between edges and nodes. And the model predict next time step through N message-passing steps.

The full model of HOOD consists of:

- Several encoder MLPs: 1 for nodes, 1 for body edges, 1 for each level's edges.
- N message-passing steps: 1 for nodes, 1 for body edges, 1 for each level's edges(levels processed in current step).
- Decoder MLP: only 1, predict the acceleration, preparing for next time steps.

3.1 Basic message passing

In each single message-passing step, edge feature are first updated as:

$$e_{ij} \leftarrow f_{v \to e}(e_{ij}, v_i, v_j),$$

And then, node feature are updated as:

$$v_i \leftarrow f_{e \to v}(v_i, \sum_j e_{ij}^{body}, \sum_j e_{ij}),$$

$f_{v \to e}$ and $f_{e \to v}$ are both multi-layer perceptions. All nodes share the same MLP, and each set of edges share the separate MLPs.

The encoder convert the input feature vectors into latent vectors with h dimensions, in HOOD $h = 128$. In each message-passing steps, the latent vector of each node and edge of the graph is updated. And then, the decoder convert latent vectors to scalar accelerations.

3.2 Multi-level message passing

In Section 2.2, a hierarchical graph was built up, where nodes are shared across all levels and each level takes different edges.

For each level l, e_{ij}^l is the feature vector of its edges. In each step, we update edges of all levels first:

$$e_{ij}^l \leftarrow f_{v\to e}^l(e_{ij}^l, v_i, v_j)$$

Then update nodes' feature vectors(L is the number of levels processed in this step):

$$v_i \leftarrow f_{e\to v}(v_i, \sum_j e_{ij}^{body}, \sum_j e_{ij}^1, \dots, \sum_j e_{ij}^L)$$

These Multi-level message passing steps transfer information between levels efficiently, since it can simultaneously handle any number of levels. It plays the role of explicit averaging or interpolation in other methods. In HOOD, there's a three-level hierarchy and each message-passing step operates two adjacent level at a time.

4 Physical Supervision(To Be completed)

4.1 Physical garment model

Actually I cannot understand the Physical formulas now(2023.11.18), so I simply put the $\mathcal{L}_{total}$ here:

$$\begin{aligned}\mathcal{L}_{total} =& \mathcal{L}_{stretching}(X^{t+1}) + \mathcal{L}_{bending}(X^{t+1}) + \mathcal{L}_{gravity}(X^{t+1}) + \\ & \mathcal{L}_{friction}(X^t, X^{t+1}) + \mathcal{L}_{collision}(X^t, X^{t+1}) + \\ & \mathcal{L}_{inertia}(X^{t-1}, X^t, X^{t+1})\end{aligned}$$

where X^{t-1}, X^t and X^{t+1} are positions of nodes at those time steps.

4.2 Novel terms in the loss function

HOOD takes SNUG [3]'s physically based loss function, and makes some change on it.

- Collision term: use body mesh, not vertex.
- Stretching term: using relaxed 3D resting pose triangles projected into 2D instead of 2D triangles from UV space.
- Friction term (novel): rough but useful. But I can't understand the formulas now(2023.11.18).

5 Conclusion

5.1 Enhancement

- Computationally cheap, compared to physics-based approaches.
- Train once, simulate every dynamic cloth, compared to other learning-based approaches.
- Be able to handle changes in topology and dynamic material parameters.
- Truly unsupervised, without the need of ground-truth data.
- State-of-the-art.

5.2 Problems remain

- Cannot simulate high speed animation, especially when body motions exceed the velocity seen at training time.
- Weak on solving garment self-collision.
- (Based on my observation) Low accuracy on prediction of inertia & creases in motion.
- Batch size was locked to 1, which means low reasoning speed.

6 Possible Improvements

After the analysis of HOOD, we could find several parts of it can be improved. Also, I got some new ideas after glancing at papers in related fields.

6.1 Fragmentary thoughts

- More precise $\mathcal{L}_{friction}$.
- Maybe a neural solver for Codium-IPC [4] to enhance the collision detection?, furthermore, how about introduce PINN into cloth simulation to predict that Physical information.
- Using STAR [5] instead of SMPL [6].
- Better ways to evaluation cloth simulation frameworks to replace the volunteer perception evaluation.
- Decompose high-speed motion into several parts and process them separately before combining them.
- Dynamic nodes and hierarchies. Maybe an extra network for prediction of density of folds? (From @yuki_hiroshi)

6.2 Model structure

1. It's easy to find out that different vertices of cloth has different contribution to the visual experience. Thus, we can introduce attention to the GNN to make it a GAT like what SwinGar [7] done.

2. HOOD learns something that fit physics law closely. Then we may break cloth into parts, and treat each parts as a whole to apply a learning based simulator and build a graph of all parts.

3. Not to update all nodes each time, such as only update nodes in levels ≥ 2 and use interpolation to handle other nodes, then update all nodes in next time step. This way we can build more complex networks.

6.3 "Serialization"

Noted that, the deformation is a continuous process. And we made a dicretisation of time to break the process into points in timeline.

So it's naturally to handle it as sequences. This way, let's introduce techniques for processing continuous text sequences in Natural Language Processing.

Similar thoughts have already appeared in papers, for example, Neural Cloth Simulation [8] used body motions sequences to indicate the state of cloth.

6.4 Hypergraph

This section is an expanded idea of the second element listed in Section6.2.

Hypergraph consist of hyperedges and nodes. Thus, HGNN(hypergraph neural networks) can learn the relationship between a group of nodes. To each small part of simulation object, we can approximately treat it as a hypergraph to accelerate the message-passing steps through nodes inside it.

Hypergraph has not been introduced to simulations, it may have the potential.

References

[1] A. Grigorev, M. J. Black, and O. Hilliges, "Hood: Hierarchical graphs for generalized modelling of clothing dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 965–16 974.

[2] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia, "Learning mesh-based simulation with graph networks," in *International Conference on Learning Representations*, 2021.

[3] I. Santesteban, M. A. Otaduy, and D. Casas, "Snug: Self-supervised neural dynamic garments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8140–8150.

[4] M. Li, D. M. Kaufman, and C. Jiang, "Codimensional incremental potential contact," *arXiv preprint arXiv:2012.04457*, 2020.

[5] A. A. A. Osman, T. Bolkart, and M. J. Black, "STAR: A sparse trained articulated human body regressor," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 598–613. [Online]. Available: https://star.is.tue.mpg.de

[6] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[7] T. Li, R. Shi, Q. Zhu, and T. Kanai, "Swingar: Spectrum-inspired neural dynamic deformation for free-swinging garments," *arXiv preprint arXiv:2308.02827*, 2023.

[8] H. Bertiche, M. Madadi, and S. Escalera, "Neural cloth simulation," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–14, 2022.