Physics-Based Animation of Characters in Fluids

Member: Wang Sinan, Guo Zebin
Supervisor: Prof. Komura, Taku

Project Background

In recent years, the concept of Metaverse has gained popularity, and character animation in computer graphics has emerged as a popular and evolving field. Kinematic-based and physics-based animation are two distinct directions that have gained attention. While significant progress has been made in fluid simulation, the complexity and real-world significance of this area call for further exploration. Notably, the work of Stream Function and DayDreamer has demonstrated the potential of incorporating the stream function into vorticity simulation and developing a fully differentiable model, respectively. However, there is a lack of notable integration between these techniques, leaving ample room for future influential work. Therefore, our proposed final year project is centered around "physical-based character animation in fluids." Our main objective is to address the challenges associated with vorticity simulation in fluids by utilizing a differentiable world model trained through supervised learning. By combining vorticity simulation and a differentiable physical engine, we aim to take a significant step forward in fluid simulation and establish a new approach to building physical solvers. Our approach involves developing a discrete physical solver and making it differentiable using supervised learning. The differentiable world model will enable us to train characters for more realistic animation purposes. The project aims to deliver a well-developed differentiable solver capable of simulating various tasks and a few well-trained characters by the final phase, with the complete differentiable world model expected to be available in February and the final product delivered in May. The proposed project is highly innovative and has great potential to advance the current state of fluid simulation and character animation. Currently, there are open-source codes available online for world models, such as DayDreamer, which we plan to utilize. Additionally, we have already built a vorticity-based solver that can handle one-way solid-fluid coupling. The motivation behind this project stems from the observation that while many people are involved in robotics, their robots typically only interact with solids. However, fluids play a crucial role in the world, be it water or air. We aspire to create virtual robotics that can effectively interact with these fluids, recognizing their importance and potential applications. The proposed project aligns well with the current trend towards creating more realistic virtual environments, which require advanced fluid simulation techniques and character animation. By developing a differentiable world model and physical solver, we aim to contribute to this growing area of research and establish a new approach to building physical solvers.

Project Objective

We aim to achieve highly realistic visual effects of characters moving within fluids. Additionally, if feasible, we also aspire to accurately calculate the forces and torques involved, enabling real-world robots to navigate and interact with fluids effectively.

Project Methodology

Overview of the experiment setup

We will first construct a simulated environment (a vorticity-based physical solver), then ask the agent to do operations in the virtual space to generate sufficient datasets for future training. Then, we implement a world model created by model-based reinforcement learning and then trained by supervised learning which can create differentiable physical engines, handling and representing solid fluid coupling very well. After that, by using this world model, we can train our agent in a differentiable manner using some gradient-based optimization methods. Hence, we will be able to achieve our goal.

Hardware

Our robot is estimated to be trained in Linux Environment, deployed on a Linux computer locally, enabling better illustrations of graphics. The programming language should be mainly python, involving popular APIs like Pytorch, TensorFlow and Taichi Lang to accelerate the speed of computation given the slow computing speed of native python.

Physical-based animation

Physical-based animation is a computational technique that simulates the movement and behavior of objects and characters in animation using principles of physics. It involves modeling the physical properties of objects, such as mass, elasticity, and friction, and applying equations that describe their motion. By considering forces like gravity, collisions, and constraints, the animation can accurately depict realistic interactions and dynamics. Physical-based animation algorithms often use numerical methods to solve these equations and update the positions and velocities of objects over time. This approach creates animations that are visually convincing, as they reflect the laws of physics and mimic real-world behavior.

Physical Solver

Our physical solver of fluids will be based on vorticity. The initial momentum equation is the famous Navier–Stokes equations, i.e.,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 u$$

The details of this equation can be found at [1]. However, for a vorticity-based solver, the equation transforms into another form, based on the fact that

$$\omega = \nabla \times \mathbf{u}$$

The transformed equation is called vorticity transport equation

$$\frac{Dw}{Dt} = (\mathbf{w} \cdot \nabla)\mathbf{u} + \nu \nabla^2 w$$

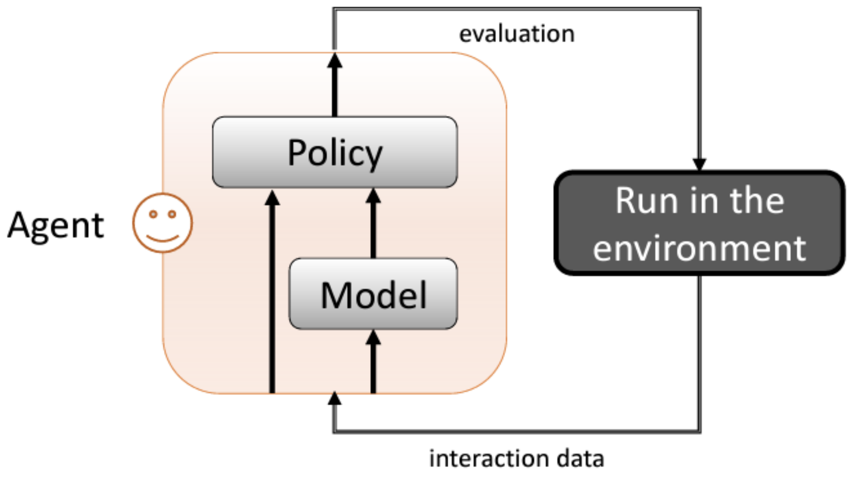
Supervised Learning

Supervised learning is a machine learning paradigm where an algorithm learns to make predictions or decisions by training on a labeled dataset. In this approach, input data is paired with corresponding output labels, forming a training set. The algorithm iteratively adjusts its model parameters to minimize the discrepancy between its predictions and the true labels, typically using techniques like gradient descent. Once trained, the model can generalize to make accurate predictions on new, unseen data. Supervised learning is widely used in tasks such as classification (assigning labels to inputs) and regression (predicting numeric values), making it a fundamental method in data-driven problem solving.

Model-based reinforcement learning

Model-based reinforcement learning is an approach that combines the use of a learned model of the environment with traditional reinforcement learning techniques. In this framework, the agent learns a model that approximates the dynamics of the environment, including state transitions and rewards. This learned model is then used to plan and make decisions, allowing the agent to explore and learn in a more efficient manner. One of the key advantages of model-based reinforcement learning is the ability to simulate and explore potential future scenarios without actually interacting with the environment. By using the learned model, the agent can simulate different actions and their consequences, enabling it to plan ahead and make more informed decisions. This planning ability can be especially useful in complex and uncertain environments, where the agent can use the model to anticipate potential outcomes and choose actions that maximize long-term rewards. Another benefit of model-based reinforcement learning is the potential for faster learning and improved sample efficiency. Since the agent can generate its own training data through simulations, it can learn from a larger amount of data in a shorter period of time. This can be particularly advantageous in situations where real-world interactions are time-consuming, expensive, or risky. By learning from simulated experiences, the agent can iteratively improve its performance and generalize its knowledge to real-world scenarios. However, model-based reinforcement learning also comes with its own challenges.

One major issue is the accuracy and reliability of the learned model. If the model does not accurately capture the true dynamics of the environment, the agent's planning and decision-making may be suboptimal or even incorrect. Additionally, learning an accurate model can be computationally expensive, especially in complex environments with high-dimensional state and action spaces. Despite these challenges, model-based reinforcement learning holds great promise for improving the efficiency and effectiveness of reinforcement learning algorithms. By leveraging learned models, agents can plan and make decisions in a more informed manner, leading to faster learning and better overall performance. With ongoing research and advancements in this field, model-based reinforcement learning has the potential to revolutionize the way agents learn and adapt in complex and dynamic environments.



Differentiable World Model

A Differentiable World Model refers to a computational model that approximates the dynamics of an environment and is designed to be differentiable end-to-end. It enables gradient-based optimization, allowing for efficient and effective training using techniques such as backpropagation. In the context of reinforcement learning, a differentiable world model is used to capture the complex relationships between states, actions, and rewards. It provides a differentiable mapping from the current state and action to the predicted next state and reward. This enables agents to simulate and explore potential future scenarios without direct interaction with the environment. The differentiability of the world model is crucial as it allows for seamless integration with gradient-based optimization algorithms. The model parameters can be updated using gradients computed through backpropagation, enabling efficient learning and adaptation from data. To train a differentiable world model, a combination of supervised and reinforcement learning techniques is typically employed. Initially, the model is trained in a supervised manner using pairs of states and corresponding next states obtained from real or simulated interactions. This supervised training helps the model learn the underlying dynamics of the environment. Once the model is trained, it is incorporated into reinforcement learning algorithms. The agent uses the differentiable world model to simulate potential future states and rewards, allowing for efficient planning and decision-making. This approach can significantly improve sample efficiency, as the agent can learn from simulated experiences instead of relying solely on real-world interactions. The differentiable world model has various applications in model-based reinforcement learning. By leveraging the model's differentiability and optimization capabilities, agents can learn and plan more efficiently, enabling them to tackle complex and uncertain environments. In summary, a differentiable world model is a computational model that approximates the dynamics of an environment and is designed to be differentiable end-to-end. It enables efficient gradient-based optimization, allowing for effective training and integration with reinforcement learning algorithms. By simulating potential future scenarios, the differentiable world model improves sample efficiency and enhances the agent's decision-making capabilities in complex environments.

Project Schedule and Milestones

We plan to first implement a two-way vorticity-based fluid solver based on our current solver before December, then we will try to implement the world model by Feburary,2024 and use the following time to train the agents.

Phase	Description	Timeline
Phase 1: Fluid Solver	Implement a two-way vorticity-based fluid solver based on the current solver	Before December 2023
Phase 2: World Model	Implement the differentiable world model using the fluid solver	By February 2024
Phase 3: Training	Train agents using the differentiable world model	February 2024 onwards

Phase 4: Evaluation	Evaluate the performance of trained agents and fine-tune the model if necessary	March-April 2024
Phase 5: Finalization	Finalize the differentiable world model and deliver the project	May 2024

References

1. Bridson, R. (2015). Fluid Simulation for Computer Graphics (2nd ed.). A K Peters/CRC Press. https://doi.org/10.1201/9781315266008