CSE 493V Final Project Report

3D Physics Interactions with Hand Tracking in VR

AIDAN GALL and SAM GAN, University of Washington

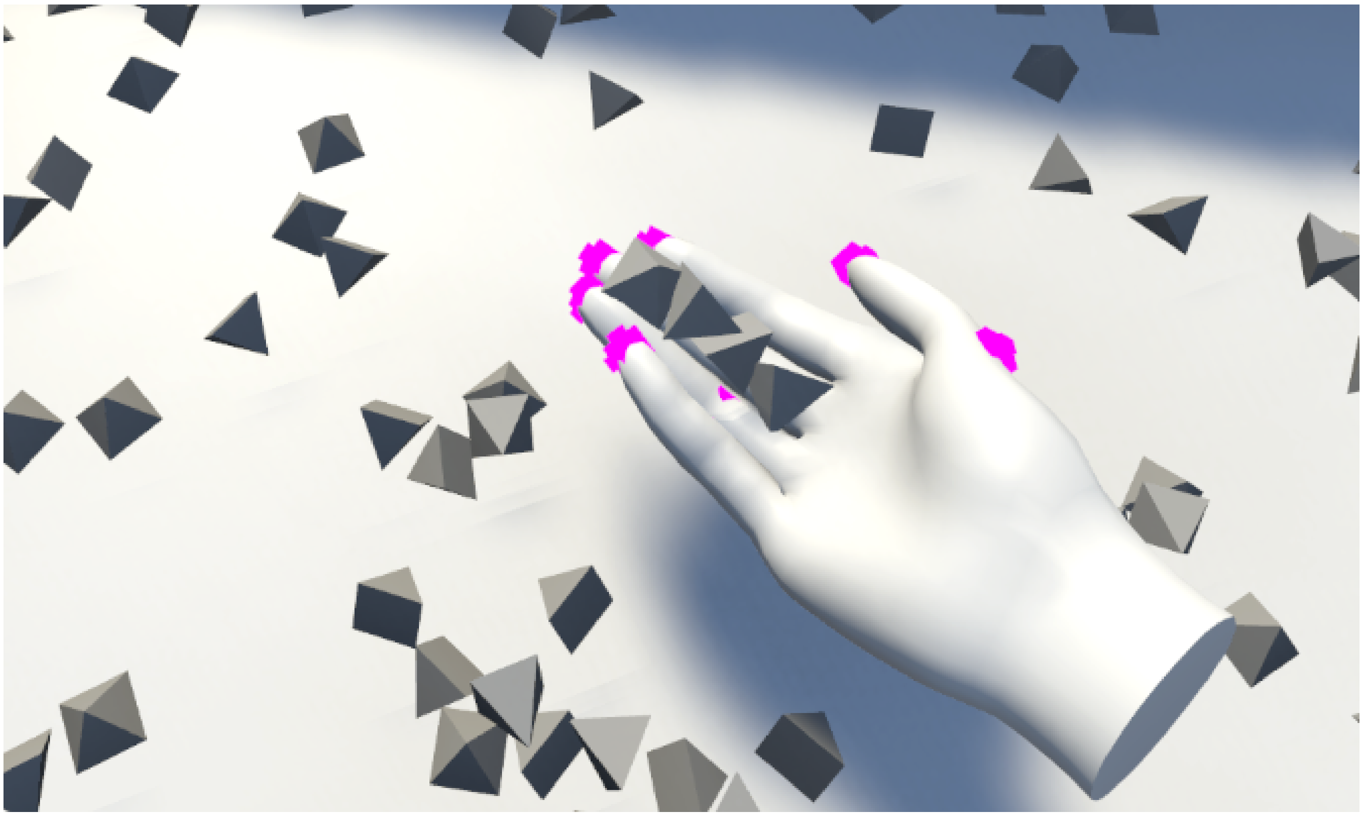

Fig. 1. Teaser

In this project, we explore ways to allow interaction between the virtual world and the real world without the use of controllers. While many applications of hand tracking with virtual interfaces exist, not many of them explore interacting with objects affected by physics in the virtual world. We decided to take hand tracking and use it to place the user into a physics-affected world in which they may interact with objects: pushing, grabbing, holding, etc. In perfecting VR/AR/XR, the ideal solution would involve as little hardware as possible, and by using hand tracking, we strive for a seamless connection between the virtual world and the real world, without the need for controllers.

1 INTRODUCTION

Virtual and augmented reality (VR/AR) have made significant strides in immersion, but many experiences still rely on controllers, limiting their realism and accessibility. While modern VR systems have introduced hand tracking, its primary use has been in interface navigation rather than physics-based interactions. Our project aims to bridge this gap by enabling natural, physics-driven interactions between users' hands and virtual objects—allowing for grabbing, pushing, stacking, and other tactile interactions without the need for controllers.

Interacting with physics-based objects in VR without controllers presents several challenges. Hand tracking systems introduce inaccuracies due to imperfect tracking, and physics engines often struggle with clipping, object stability, and collision resolution when applied to dynamic hand movements. Despite these challenges, a well-executed solution could lead to more immersive VR experiences, benefiting applications from gaming to professional simulations and accessibility.

Our approach leverages the Meta Quest 2's hand tracking in Unity's physics engine to create a robust interaction system. We experiment with different collision models, friction settings, and constraint-based grabbing methods to refine the realism of interactions. Inspired by projects such as Hand Physics Lab, we seek to implement a physics-based approach rather than rely on artificial constraints or animations.

Authors' address: Aidan Gall, algall@cs.washington.edu; Sam Gan, samganyx@cs.washington.edu, University of Washington.

1.1 Contributions

- Contribution 1: Hardware and Development Set Up:
  Sam Gan
- Contribution 2: Algorithm Design:
  Aidan Gall (Primary) and Sam Gan
- Contribution 3: Implementation and Testing:
  Sam Gan (Primary) and Aidan Gall
- Contribution 4: Research Aidan Gall (Primary) and Sam Gan
- Contribution 5: Write Up:
  Aidan Gall (Primary) and Sam Gan

2 RELATED WORK

Partway while working on our project, we found a game on the Meta Library: **Hand Physics Lab** Their game uses hand tracking to interact with objects in order to solve puzzles, in which you can grab things, throw them, as well as other applications.

Unfortunately that does a lot of the stuff that we have planned, and was discouraging. We were unable to test it out as it costs money. Although it accomplishes what we wanted to do, since we did not know the implementation behind it, we thought that we would still explore our path that may work differently from theirs.

3 METHOD/IMPLEMENTATION

The base of our project was the Meta Quest 2 and Unity. Using this, we have a well-tested and robust virtual reality system with hand tracking capabilities, as well as a physics engine. Before we started our project, we knew we had several potential limitations:

1. Time:

With the limited time and manpower at our disposal, we may not be able to complete a fully realized project by the deadline. Furthermore, we are working with hardware and software we are not familair with.

2. Hardware:

Although the Meta Quest 2 has a good hand tracking system, it is not perfect, and can result in errors that may cause issues.

3. Software:

The Unity Engine may not be sufficient to do the type of physics we are hope are possible, and we may run into issues with clipping, low polling rate, etc.

3.1 Initial Prototype

After setting up a simple Unity environment playground with objects and physics, we made a proof of concept. Our first implementation was a simple rigid body collider on each of the hand's main parts (finger joints, palm, etc). While it was able to interact with objects in the virtual world, there were several issues: it was very difficult to pick up an object without clipping it, and objects would slide off the hands at the slightest angle. Furthermore, the fingers were difficult to work with as small objects would slip between the collision boxes.

3.2 Tweaking

Before we added any complexity to the hand logic, some tweaking was in order. After playing around with different models, (thickness, shape, length) we found a model that world pretty well. We made the decision to thicken the collision boxes to be slightly thicker than the fingers, which would allow the fingers to touch and prevent small objects from slipping through. We made the fingertips rounded for realism while the finger joints and palms were rectangular, which was easier on the physics.

3.3 Friction

One of the biggest issues we ran into was objects constantly slipping out of our hands. The default physics settings in Unity didn't provide enough friction, making it tough to grip anything properly.

To fix this, we played around with static and dynamic friction—static friction helped objects stay put when grabbed, while dynamic friction controlled how easily they slid. We also tweaked material properties and even added an artificial friction boost when fingers closed around an object.

Another problem was that some objects didn't have enough contact points with the hand, making them slip no matter what. We adjusted collision shapes and experimented with adaptive friction, where grip strength affected how "sticky" the hold was.

These tweaks made a huge difference—objects stayed in our hands more naturally. There were still some hiccups, like occasional tracking errors causing weird drops, but overall, friction adjustments made interactions feel way more realistic.

3.4 Clip Fix

To tackle clipping in a more natural way, we changed how the hand moves in VR. Instead of instantly snapping to the tracked position, we applied a physics-based approach where the hand moves using forces.

Here's how it works:

Force-Based Movement – The hand doesn't teleport instantly. Instead, it tries to move toward the tracked position using a force, like a spring pulling it into place. This makes interactions feel more physical and prevents fingers from phasing through objects. Collision-Responsive Hands – If the hand collides with an object, it stops moving and reacts naturally, rather than pushing through. This prevents fingers from clipping inside objects while still allowing natural movement. Failsafe Teleportation – If the hand gets too far from the tracked position (due to lag or occlusion), it snaps back to the correct place. This prevents drift while still allowing for realistic interactions. This approach made a big difference. Hands now feel like they exist in the world, rather than just being ghost-like visual elements. There's still some tuning needed for edge cases, but overall, this method drastically reduced clipping while keeping interactions smooth and immersive.

4 EVALUATION OF RESULTS AND BENEFITS AND LIMITATIONS

Looking back on this project, we're happy with the progress we made in creating a more natural and interactive hand-tracking system for VR. We were able to make meaningful interactions like pushing, holding, and grabbing objects feel intuitive and grounded. A key goal was to minimize reliance on artificial "helper" techniques—things like automatically snapping the hands into place or making objects auto-grab when a gesture was detected. Instead, we wanted the tracked hands to behave in a way that felt more natural, like they truly existed in the virtual world, and that's something we were able to achieve with the system we built.

The force-based movement we implemented worked well, giving the hands a smooth transition between positions rather than snapping instantly into place. We also made tweaks to the hand collision models and the friction settings, which helped make grabbing and holding objects feel more stable and realistic.

There were a few limitations with our implementation. Since we relied purely on the physics engine to make the virtual hands as close to the real hands as possible, there were difficulties in manipulating objects on small magnitudes or at high speeds. Whereas other hand tracking interactions involve gestures that would automatically apply an action, relying solely on the physics reduced dexterity significantly.

That said, there's always room for improvement. Looking ahead, we can think of a few ways to make this experience even better through further research or as the world advances further on hand tracking and physics simulation technology.

5 FUTURE IMPROVEMENTS AND TECHNOLOGIES TO FOLLOW:

Better Finger Tracking: While the hands tracked fairly well, the finger tracking could definitely be more precise, especially when it comes to things like picking up small objects or performing finer actions. As hand tracking technology advances, our interpretation of its interaction with virtual physics may improve as well.

Smoother Occlusion Handling: When the hands are out of view (like if the user's hands are out of the camera's line of sight), the system could be better at predicting where the hands should be and smoothing out any movement so the user doesn't experience noticeable drifting.

Haptic Feedback: Adding haptic feedback (like vibrations when touching or grabbing an object) would give users a more physical sense of interaction, making the virtual world feel even more immersive.

Improved Object Physics: We could fine-tune the physics system to handle different types of objects more realistically—things like heavier items, objects with irregular shapes, or fragile objects could all behave differently in the hands, making the interactions feel more varied and lifelike.

Alternative Approaches We Didn't Use Predefined Grabbing Gestures: We could have gone with a system where certain gestures automatically trigger grabbing or interacting with objects (like pinching to grab something), but we wanted to avoid making things feel too robotic. The goal was to let the hand movements be more freeform and physical, and predefined gestures might have limited that flexibility.

Teleporting Hands: Another option was to have the hands snap directly to the tracked position when they drifted, but we avoided this because it would have taken away from the physicality of the interactions. Instead, we chose a more gradual, force-based movement, which feels more natural even if it means occasional slight drift.

AI-Driven Object Behavior: We didn't explore the possibility of making objects react intelligently to hand movements (for example, objects that resist being grabbed or interact differently based on the user's actions). While that could have added another layer of complexity and immersion, it wasn't central to our goal, which was to make the hand interactions feel as realistic as possible.

6 CONCLUSION

Overall, we're pleased with where we ended up. The system feels intuitive, and interactions are much smoother and more natural than when we started. Of course, there's always room to make it even better, but we've built a solid foundation, and we're excited to hopefully continue refining it in the future. Although there are many flaws with direct physics with hand tracking, we believe that many of them can be worked out and in the future, our hands in the readl world will be able to interact with virtual objects just as easily as solid objects.

ACKNOWLEDGMENTS

Ackowlegments to Dougas Lanman, John Akers, Shaan Chattrath, and Evan Zhao as staff of the 493V Class for providing support, equipment, and resources.