A Virtual Reality Platform for Dynamic Human-Scene Interaction

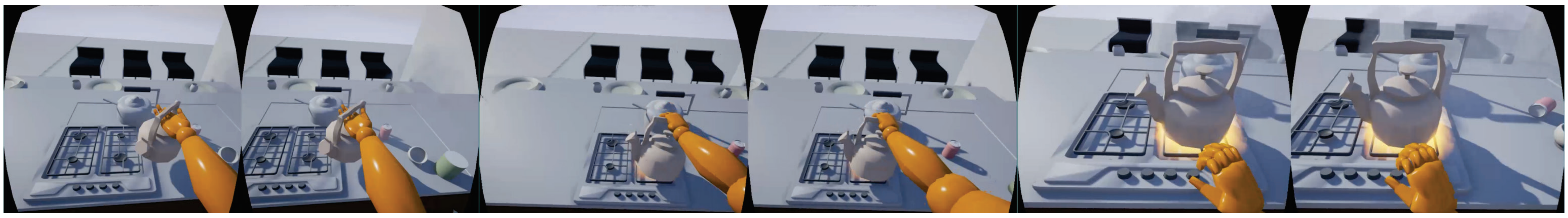
Jenny Lin*[1] Xingwen Guo*[12] Jingyu Shao*[1]
Chenfanfu Jiang[3] Yixin Zhu[1] Song-Chun Zhu[1]

* Equal contributors

1. UCLA Center For Vision, Cognition, Learning and Autonomy
2. University of Hong Kong Electrical and Electronic Engineering Department
3. UCLA Computer Graphics & Vision Laboratory

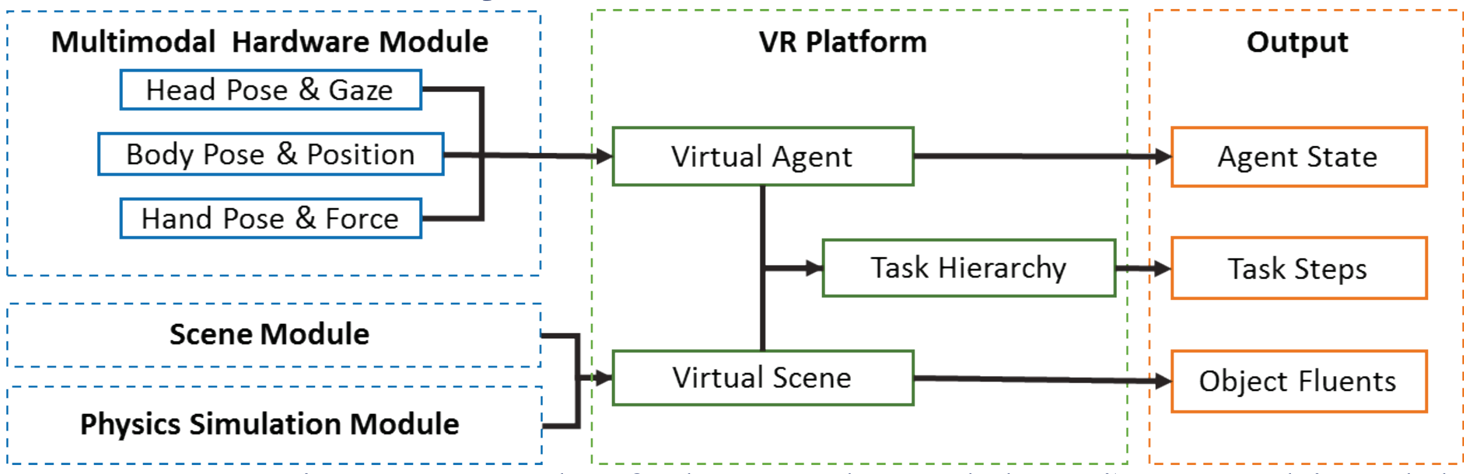
SIGGRAPH ASIA 2016 MACAO

Introduction



We propose a system in which Virtual Reality and human / finger pose tracking is integrated to allow agents to interact with virtual environments in real time. Segmented object and scene data is used to construct a scene within Unreal Engine 4, a physics-based game engine. We then use the Oculus Rift head-set with a Kinect sensor, Leap Motion controller and a dance pad to navigate and manipulate objects inside synthetic scenes in real time.

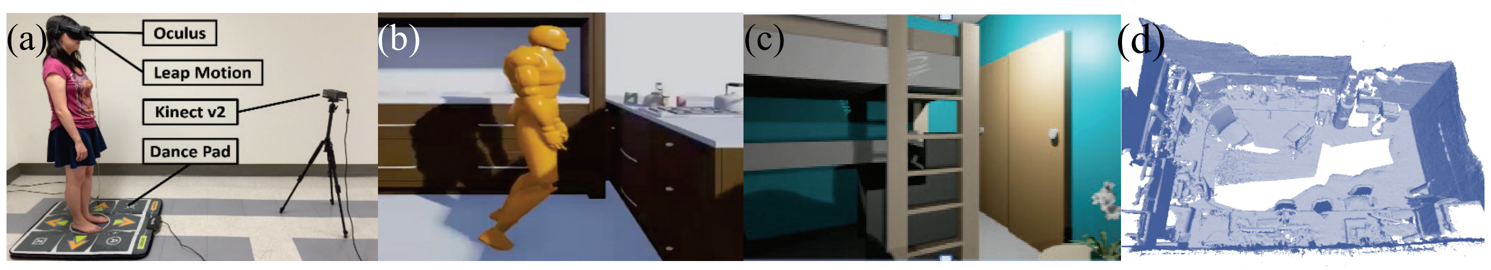
System Architecture



Our system is composed of three sub-modules: i) a Multimodal Hardware, which collects data about the real world agent (*e.g.* head pose, body pose, hand pose), ii) a Scene Module, which provides 3D data about the virtual scene, and iii) a Physics Simulation Module, which simulates the interactions between virtual agent and scenes. These sub-modules are consolidated in our VR platform, allowing the real world agent to interact with the virtual environment.

Multimodal Hardware

Our hardware setup (a) is used to map the agent's skeleton information to a virtual agent (b). An Oculus Rift is used to display the virtual scene to the agent and track head orientation. Mounted on the Oculus Rift is a Leap Motion tracker, which is used for detailed hand pose. The Kinect is used for overall skeleton pose. A dance pad receives locomotion input.

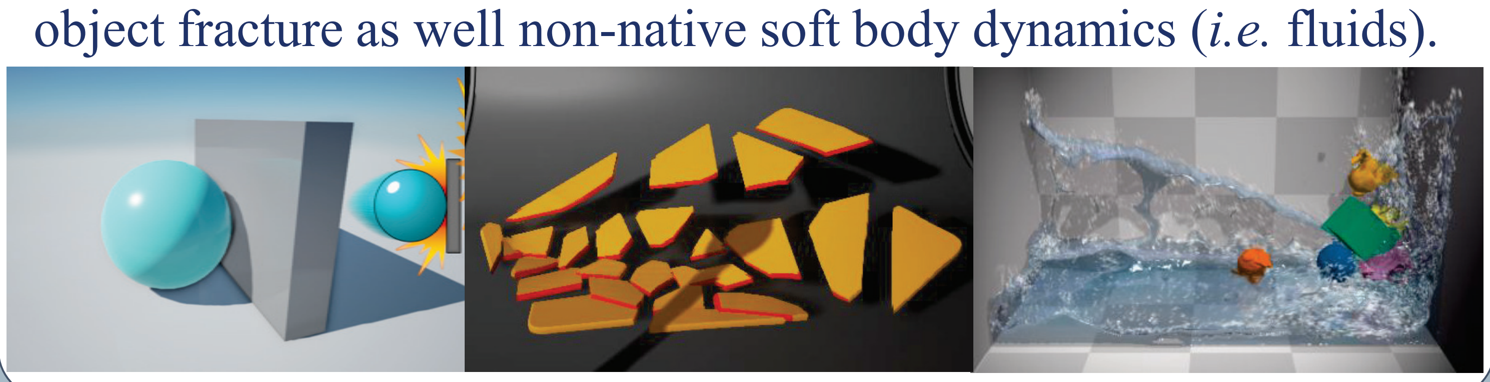


Scene Module

Scenes are constructed in the 3D modeling software SketchUp using assets downloaded from 3D Warehouse (c). We also use scanned scenes (d) from real environments and apply additional segmentation and scaling to the mesh to obtain movable sub-parts and maintain consistent scale.

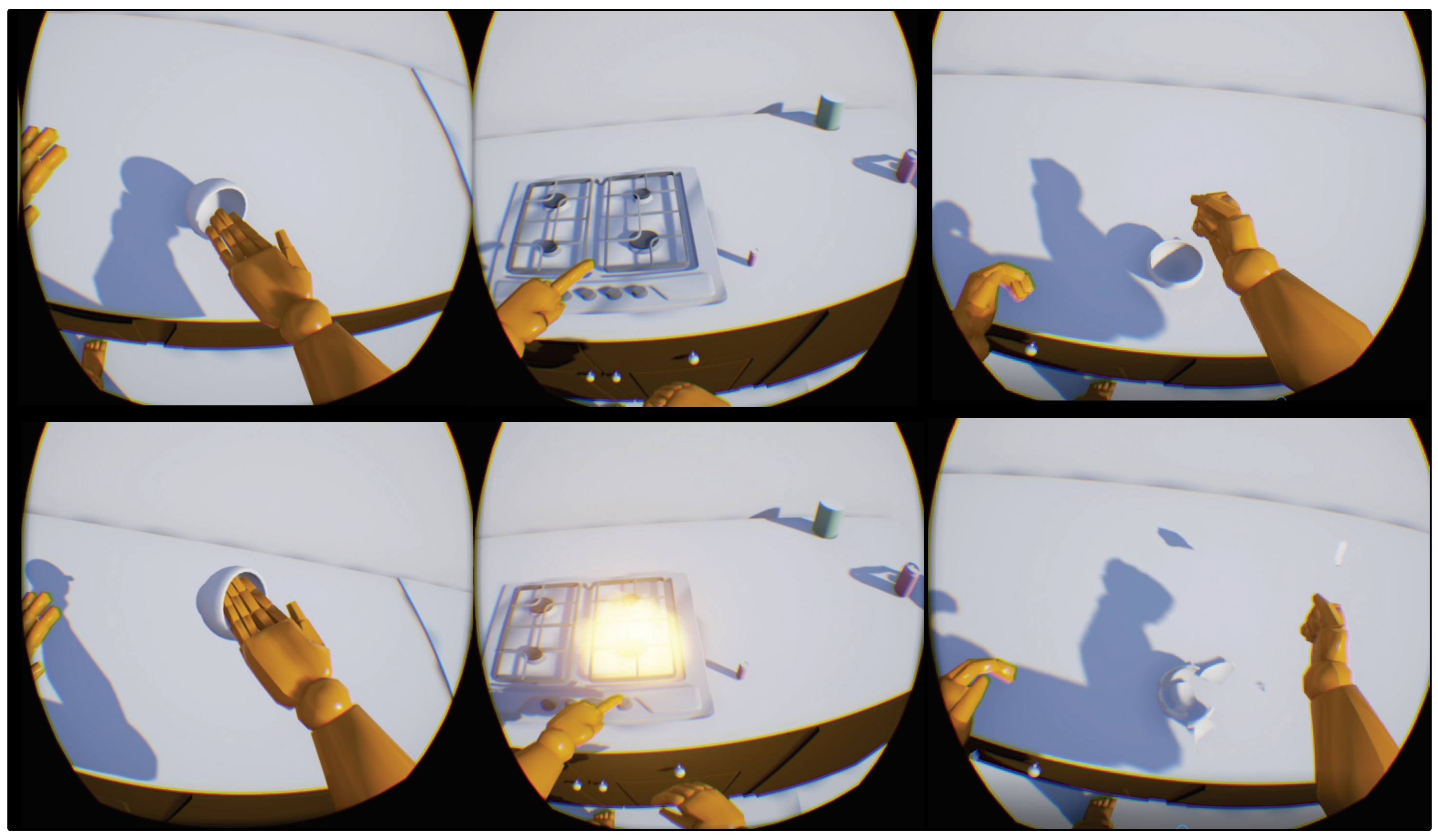
Physics Simulation Module

We chose Unreal Engine 4 (UE4) for our physics-based engine. UE4 natively supports the simulation of rigid body dynamics and object fracture as well non-native soft body dynamics (*i.e.* fluids).



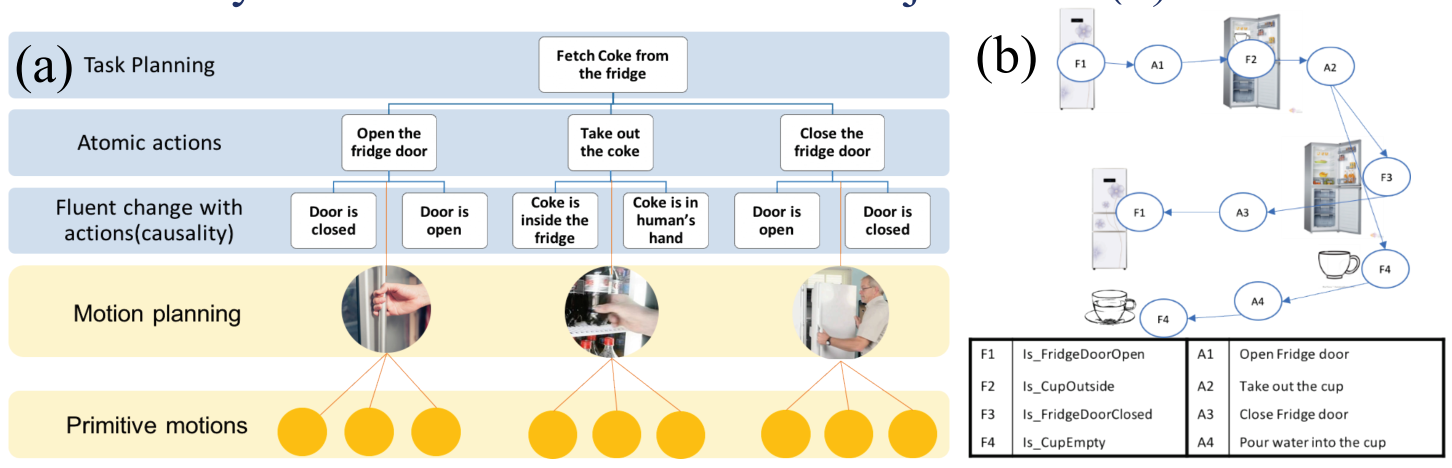
Simulating Virtual Interactions

In simulating interactions between the agent and the virtual scene, we use a combination of two methods: physical simulation and hand pose detection. Physical simulation is closer to reality, but has difficulty simulating delicate grasping. Hand pose detection is more reliable for actions that are difficult to physically simulate, but can be less realistic.



Task Hierarchy

We consider a high-level task as one which can be broken into sub-tasks recursively until reaching the level of atomic actions (a). These actions are then mapped to the actions of the agent, allowing actions to be further broken into primitive motions. To define these atomic actions, we can take advantage of the concept of causality between atomic action and object state (b).



Output

Our system can provide ground truth on full body agent trajectory as well as object state change over time. This means individual task steps and their primitive motions can be determined. This information can be used for training and validating task planning.