



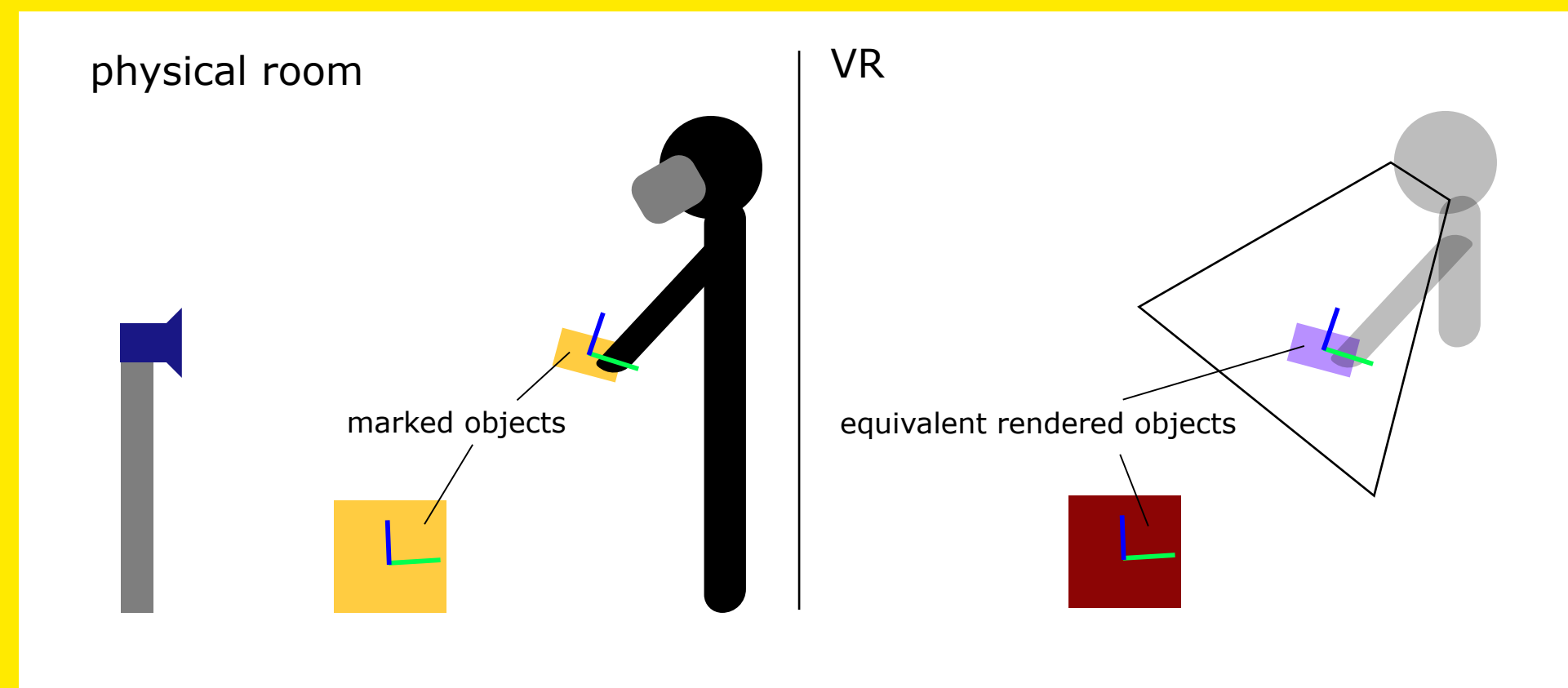
CSE 493V Rendering Physical Objects in VR

VR SYSTEMS

Jason Langley
University of Washington

PROBLEM

Augmented reality techniques, computer vision tools, and an instrumented environment can provide tactile feedback in VR and allow users to physically interact with a simulated scene.



RELATED WORK / MOTIVATION

A 2016 MIT Media Lab experiment first digitized the entire physical space using Google Tango. They additionally used Microsoft Kinects and large marker tags for user/object pose tracking.

A 2016 presentation from motion capture research group Artanim used a Vicon mocap system and had users wear a backpack covered in reflective markers for sending/receiving positional data wirelessly.

YOUR APPROACH / SOLUTION

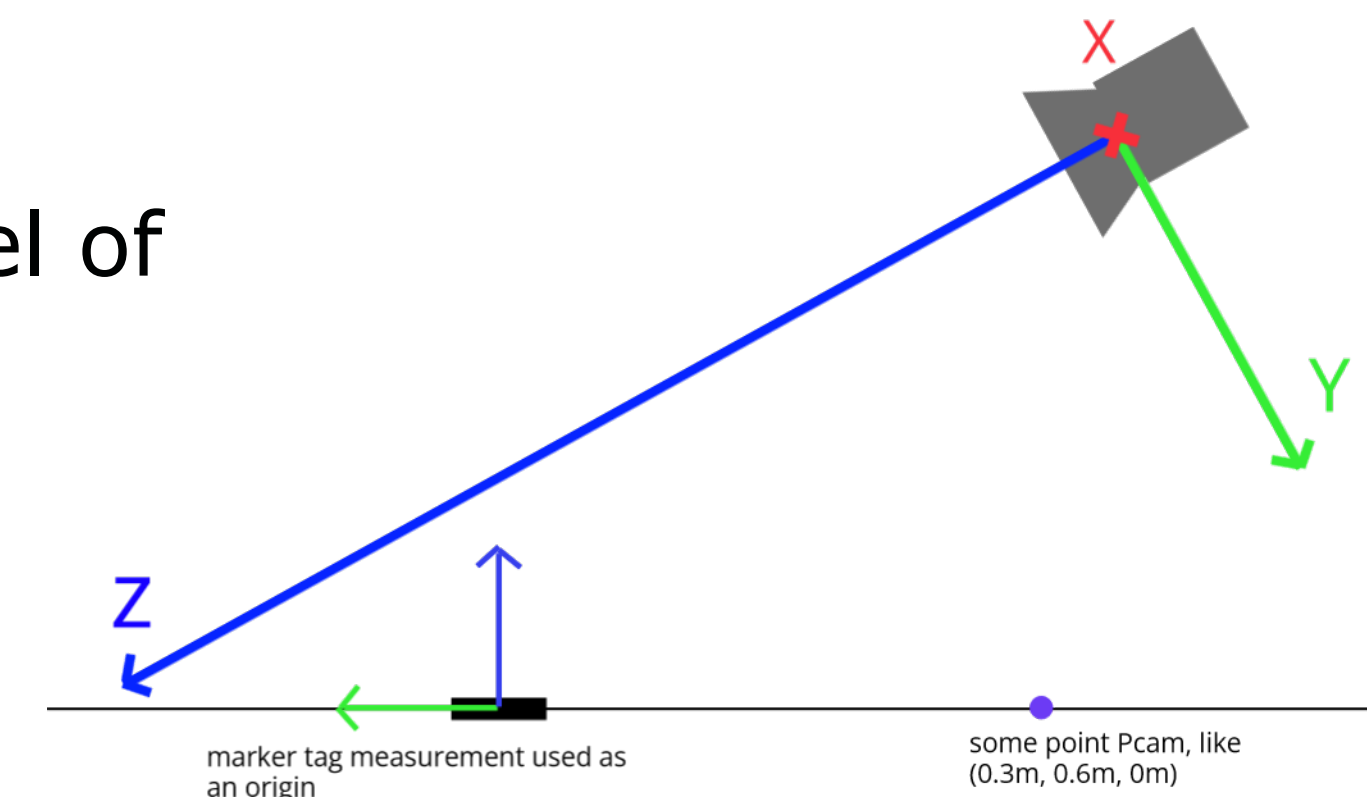
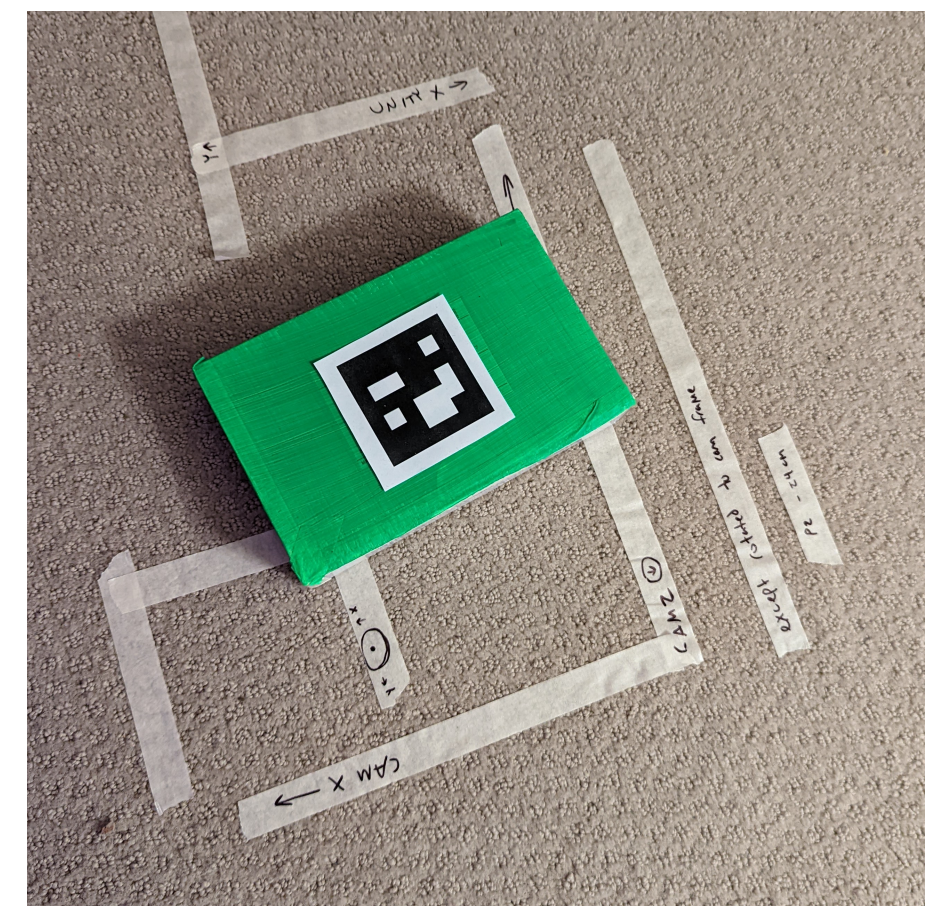
ArUco markers tracked by OpenCV, viewed through a webcam, for estimating pose for objects in the environment.

METHOD / PIPELINE / ALGORITHM / PROCESS

To move quantities from the webcam's 'camera space', we first record the OpenCV **rvec** and **tvec** for a marker tag placed on the floor to act as the world origin. We use this recorded P_C and R (the rotation from camera axes into the origin marker pose) to solve for world coordinates of other detected markers, using their **tvec**. $R^{-1}(P_C - tvec) = P_W$

For the rotation of the object, we apply the original R we recorded from the origin to a detected marker's orientation, then use the 'front' and 'up' vectors as parameters to Unity's LookRotation function.

Data is sent to Unity via UDP and used to set the translation and rotation on a box model of the same size as the physical object.



RESULTS

Not great! I didn't have time to do much other than a very limited proof of concept.

- There's only one marker tag, so if the user rotates the tag away from the sole camera, the object tracking is lost. It seems reasonably possible to incorporate multiple tags.
- The granularity of rotation and position OpenCV can detect aren't high enough to be used without processing, and it's also very noisy. The box jitters around a lot at a pretty low framerate. Some of this could likely be smoothed out by throwing out unrealistic measurements and doing some simple low-pass filtering. Doing interpolation between data would also help.
- There's only one camera, so occlusions and marker angle are a problem. Additionally, one camera is not enough to, say, determine the pose of a flat-colored spherical object. I fiddled with color-based contour tracking at the last minute to see if I could render green balls; it seems possible but would be easier with two cameras to triangulate with.
- An ArUco tag doesn't have to be moving very fast to blur and be unreadable.

REFERENCES

- Sylvain Chagué and Caecilia Charbonnier. 2016. Real Virtuality: A Multi-User Immersive Platform Connecting Real and Virtual Worlds. In Proceedings of the 2016 Virtual Reality International Conference (Laval, France) (VRIC '16). Association for Computing Machinery, New York, NY, USA, Article 4, 3 pages. <https://doi.org/10.1145/2927929.2927945>
- Misha Sra and Chris Schmandt. 2015. MetaSpace II: Object and full-body tracking for interaction and navigation in social VR. CoRR abs/1512.02922 (2015). arXiv:1512.02922 <http://arxiv.org/abs/1512.02922>