Testing Approximations for Augmented and Virtual Reality

Samuel Grayson
grayson5@illinois.edu
University of Illinois at Urbana-Champaign

Navi Ning
xning5@illinois.edu
University of Illinois at Urbana-Champaign

ABSTRACT

There is a lot of commercial interest in the novel domain of augmented and virtual reality (AR/VR), but it has stringent realtime performance, power, and area constraints, which are difficult to simulatenously satisfy.

One way to bring this difficulty down is through the use of *approximation techniques*; In many cases, the output only has to fool a human, so the algorithms can be simplified or reduced. However, one does not know *a priori* which approximations will 'fool the human.'

This project seeks to satsify stringent constraints by automatically searching over the space of approximations.

CCS CONCEPTS

• **Software and its engineering** → *Functionality*.

KEYWORDS

approximation tuning

1 INTRODUCTION

- A virtual reality system presents the user with an opaque visual display that consumes their entire field-of-vision. Within this field, what the user sees is determined by estimating the pose (position and orientation) of the users head in the real-world and rendering a virtual world from that pose. This gives the user the illusion of being immersed in the virtual world.
- An augmented reality system uses a transparent display, so virtual elements can be overlayed on the physical world. THe virtual objects should move synchronously with the physical objects they overlay, since they are rendered from the user's head-pose.

AR/VR has important applications in remote medicine, education, and job training. There are some commercially available AR and VR systems, such as HTC Vive Pro (example of VR) and Microsoft HoloLense 2 (example of AR). However, the quality-of-experience in these systems can stand to be improved. In order to fully immerse the user, more resolution and a faster latency are needed. Furthermore, AR/VR systems need to be mobile/tetherless to support the full breadth of AR/VR applications, and tetherless systems imply a strict power constraint on top of the existing performance constraints. There are multiple orders of magnitude between state-of-the-art AR/VR systems and ideal futuristic systems in performance, power, and area.

XR systems typically use an IMU sensor and a pair of stereo cameras to capture the environment. The system detects visual features in the physical environment, builds up a map of them over time, and uses this map to localize itself (simultaneous localization and mapping or SLAM). Prior work shows that XR systems without audio spend between 10 and 30% in the SLAM computation, depending on the application, platform, physical environment, and user[2]. Existing SLAM algorithms *already have* approximation knobs. In fact, they have 'too many;' nobody knows which ones to turn. Therefore, automatically searching for feasible SLAM approximations could greatly improve system performance. We wanted to build out a system *in practice* not just in theory. Therefore, we built off of Illinois eXtended Reality System (ILLIXR): an open-source runtime for AR/VR applications cite ILLIXR.

This project seeks to answer or partially answer three questions:

R.Q. 1 How can we test SLAM approximations in ILIXR?

R.Q. 2 How can we test SLAM approximations in ILLIXR *automatically*?

R.Q. 3 How can we test SLAM approximations in ILLIXR *automatically* and *quickly*?

- Due to time constriants, We only provide a partial implementation of this research question.

2 IMPLEMENTATION

Our implementation is located at .

2.1 R.Q. 1 Testing Approximations

2.1.1 Approximation Knobs. ILLIXR uses state-of-the-art components such as OpenVINS[1] for SLAM. We located approximation knobs in OpenVINS and consulted domain experts on which ones should be modified and tuned. These are given in href, see comment in LaTeX source.

We can also enable/disable asynchronous reprojection (elsewhere in ILLIXR; outside of OpenVINS).

The exact set of knobs is not so significant, as long as I have some useful ones, some useless ones, and they are orthogonal.

2.1.2 Timing Infrastructure. In order to test the approximations, I need to know how much time they are saving.

ACKNOWLEDGMENTS

We would like to acknowledge the ILLIXR group at the University of Illinois at Urbana-Champaign for the ILLIXR research platform itself and timely support. Specifically, we would like to thank Boyuan Tian and Yihan Pang for helping with the trajectory alignment and Muhammed Huzaifa for the coordinate transformation.

REFERENCES

[1] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. 2020. OpenVINS: A Research Platform for Visual-Inertial Estimation. In *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France. https://github.com/rpng/open_vins

[2] Muhammad Huzaifa, Rishi Desai, Xutao Jiang, Joseph Ravichandran, Finn Sinclair, and Sarita V. Adve. 2020. Exploring Extended Reality with ILLIXR: A New Playground for Architecture Research. arXiv:cs.DC/2004.04643