A version of this paper will appear in ACM Multimedia Systems (MMSys) 2025.

RemoteVIO: Offloading Head Tracking in an End-to-End XR System

Qinjun Jiang
University of Illinois at Urbana-Champaign
Urbana, USA
qinjunj2@illinois.edu

Yihan Pang
University of Illinois at Urbana-Champaign
Urbana, USA
yihanp2@illinois.edu

William Sentosa
University of Illinois at Urbana-Champaign
Urbana, USA
sentosa2@illinois.edu

Steven Gao
University of Illinois at Urbana-Champaign
Urbana, USA
hongyig3@illinois.edu

Muhammad Huzaifa
University of Illinois at Urbana-Champaign
Urbana, USA
huzaifa2@illinois.edu

Jeffrey Zhang
University of Illinois at Urbana-Champaign
Urbana, USA
jfzhang2@illinois.edu

Javier Perez-Ramirez*
Ofinno
Reston, USA
jperez-ramirez@ofinno.com

Dibakar Das
Intel Corporation
Portland, USA
dibakar.das@intel.com

David Gonzalez-Aguirre
Intel Labs
Hillsboro, USA
david.i.gonzalez.aguirre@intel.com

Brighten Godfrey
University of Illinois at Urbana-Champaign
Urbana, USA
pbg@illinois.edu

Sarita Adve
University of Illinois at Urbana-Champaign
Urbana, USA
sadve@illinois.edu

Abstract

Power consumption, and the resulting limitation to computational load, is a first-order constraint in designing comfortable all-day-wear extended reality (XR) devices that can provide rich immersive experiences. This paper concerns reducing XR device power consumption by offloading head tracking, one of the top CPU and power consumers, to a remote server. We present RemoteVIO, the first open-source end-to-end XR system that offloads head tracking (visual inertial odometry or VIO) to a remote server. Our work distinguishes itself from past studies on computation offloading in XR by properly addressing two under-explored but critical aspects: 1) a comprehensive evaluation of *user experience in a complete end-to-end XR system* and 2) a quantification of the *net power savings on real hardware*.

Through an Institutional Review Board (IRB) approved study, we find that RemoteVIO provides a satisfactory user experience under typical network conditions, but often degrades for network round trip time above 200ms. We also demonstrate the first measured power savings from offloading head tracking on real hardware: compared with on-device tracking, RemoteVIO reduces CPU power by up to 52%, CPU+network power by up to 39%, and end-to-end full system power by up to 20%. Of equal importance, we examine the traditional approach of evaluating XR offloading techniques with datasets and quantitative metrics. Our results reveal that traditional head tracking metrics do not correlate with user experience, questioning the use of such metrics in XR systems research and underscoring the importance of using end-to-end systems that allow for user experience studies.

CCS Concepts

• **Computing methodologies** → **Extended Reality**; • **Human-centered computing** → *Ubiquitous and mobile computing*; • **Computer systems organization** → *Real-time system architecture, Power-efficient system design*.

Keywords

Extended Reality, Head Tracking, Remote-assisted XR, Power Efficient Systems

1 Introduction

Recent advances in virtual, augmented, and mixed reality (VR/AR/MR), collectively referred to as extended reality (XR), have the potential to transform most industries and human activities [5–7, 15, 23]. However, with the end of conventional CMOS scaling [25, 64, 72], power consumption has become a first order design constraint to

*The work was conducted while the author was affiliated with Intel Labs.

MMSys'25, March 31–April 03, 2025, Stellenbosch, South Africa

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

achieving the potential of XR. Power consumption directly affects an XR device's form factor, weight, and battery life, and the user's thermal discomfort through the battery and cooling support required. It also determines the richness of the immersive experience through the compute capacity afforded within the power constraint. Unfortunately, there is currently an orders-of-magnitude gap between the power, compute capacity, and quality of experience of current and desired XR devices [39]. Specifically, desired compute power consumption for comfortable all-day wear XR devices is a few hundred milliwatts [70] while today's commercial devices typically consume 10s of Watts [33, 67]. The battery lives of the recent Meta Quest Pro and Apple Vision Pro are reported to be around two hours [9, 73] with the latter requiring an external battery pack [9].

A tempting approach to reduce power consumption in an XR device is hardware acceleration. However, as there are many components in an XR system (e.g., head/hand/eye tracking, scene reconstruction, warping, spatial audio encoding and decoding, etc.) and the algorithms in these components continue to evolve in significant ways, developing a new hardware accelerator for each algorithmic advancement in every component is unsustainable. An alternative solution is to offload intensive computations to edge or cloud servers. Offloading does not suffer from the drawbacks of hardware acceleration as it is hardware independent and can readily adapt to changing algorithms. Since most XR use cases today have access to wireless networks through WiFi or cellular mobile hotspots, offloading becomes an appealing strategy for enabling a low-power, mobile XR experience.

This paper explores reducing XR device power consumption by offloading head tracking computations to remote servers. Head tracking is one of the most CPU-intensive tasks (e.g., it has been shown to use 35% to 55% of all CPU cycles in the XR system [37–39]), making it an ideal target for offloading to reduce on-device power usage. However, to ensure the effectiveness of such an offloaded XR system, two critical aspects must be thoroughly evaluated.

The first critical aspect is **the end-to-end user experience**. A typical XR system generates and displays virtual content by interpreting the user's environment, actions, and movements using sensor inputs such as cameras and inertial measurement units (IMUs). We term the entire sensor to display process as the *end-to-end or E2E system*, and the user's experience of the displayed content as the *end-to-end or E2E user experience*. The E2E XR experience is intimate to users, especially in fully immersive VR, where users wear head-mounted displays (HMDs) and are immersed in a virtual world with a wide field of view displayed directly in front of their eyes. In this context, any performance degradation within the E2E system has the potential to result in a poor user experience or even physical discomfort. Therefore, *ensuring an acceptable E2E user experience is the primary requirement for any useful XR system design.*

Offloading head tracking presents large challenges to the E2E user experience. Head tracking estimates the direction and position of the user's head, collectively referred to as the pose. Accurate and timely pose delivery is crucial for generating images and sounds that match the user's current viewpoint, directly impacting the user's quality of experience. Delays in the pose delivery due to high or unpredictable network latencies could potentially make the E2E user experience unacceptable.

Unfortunately, existing work on offloading head tracking in XR [12, 22, 34, 44] does not assess the E2E user experience. Most work [12, 22, 44] does not build the E2E system, which eliminates the possibility of evaluating E2E user experience. Instead, these works build partial systems consisting of only components relevant to the tasks to be offloaded, and calculate metrics such as Absolute Trajectory Error (ATE) to gauge the pose accuracy against the ground truth. While suitable for evaluating algorithmic improvements, the efficacy of this approach for system-level XR research where user experience is paramount remains unclear. FleXR [34] develops and implements a more complete system with a flexible offloading architecture. However, their evaluations for head tracking in VR rely on datasets from the robotics domain and metrics like pipeline latency ($\sim$80-350ms) and throughput ($<$ 30 FPS), which do not adequately reflect the E2E user experience.

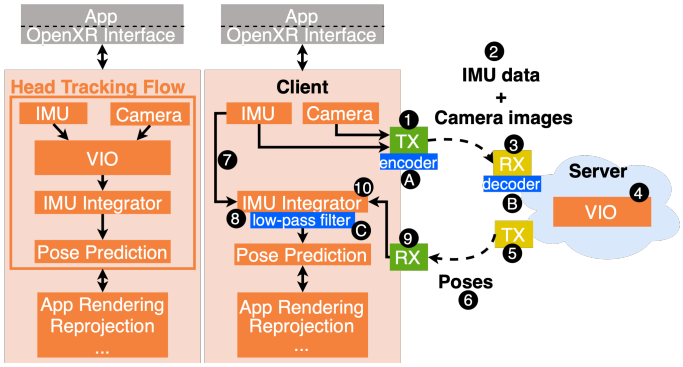
Our work is the first to **build an E2E immersive VR system with offloaded head tracking**, and the first to **evaluate whether offloading head tracking can preserve a satisfactory E2E user experience**.

The second critical aspect to ensure the effectiveness of offloaded head tracking is **the net power savings**. Offloading in XR does not necessarily bring power benefits [27] because of the overhead introduced by offloading, such as wireless communication, data serialization, and de/compression. The net power effects from offloading are difficult to evaluate without running an E2E system on real hardware. Unfortunately, existing works either do not discuss power consumption [12, 44], or use power models from the literature or observed CPU/GPU load without measuring the overhead from data transmission and codecs [22]. FleXR [34] reports the standalone energy consumption of Ethernet with synthetic traffic but not for the end-to-end system. Our work is the first to **evaluate whether offloading head tracking can provide power savings on real hardware with real wireless networks.**

Overall, our work makes the following contributions.

- We design, implement, and evaluate RemoteVIO, the first E2E XR system that offloads head tracking (VIO) to a remote server. The E2E system enables the first results on the E2E user experience and power savings of an XR system with offloaded head tracking. We release our implementation as open-source [1] to motivate future work in this area to use E2E system.
- We present a user study to quantify the impact of offloading head tracking on the visual E2E user experience in an immersive VR environment, with 30 users comparing a total of 246 RemoteVIO experiences with on-device head tracking under a variety of network conditions. We find that RemoteVIO provides acceptable E2E experience for expected network latencies (up to about 200 milliseconds (ms)) for 99% of the trials. Stressing the network latency further increases the number of unacceptable experiences – latencies of 400ms or beyond show unacceptable experiences for 19% of the trials.
- We study the effectiveness of the standard tracking metrics of ATE and Relative Pose Error (RPE) as proxies for E2E user experience. We find that ATE and RPE do not correlate with the E2E user experience from our studies. For a more thorough evaluation of these metrics, we perform a sweep of various

[1] https://github.com/ILLIXR/ILLIXR



(a) On-device VIO (b) RemoteVIO

Figure 1: (a) An end-to-end XR system with on-device VIO. We highlight the head tracking flow which produces high-frequency high-accuracy poses to the rest of the system. (b) RemoteVIO where VIO is offloaded to a remote server. We implement plugins on the client and server to handle data transmission, reception, and camera frame (de)compression.

network latency values with both widely used datasets and new datasets we record specifically for VR, providing further validation of the limitations of these metrics.

- We present the first quantification of power savings from offloading VIO, including the overhead of communication on 5G and WiFi networks and that of H.264 compression on commodity hardware accelerators [59]. We find that, for the cases we studied, RemoteVIO reduces power consumption for the CPU by up to 52% (avg 42%), for CPU+network by up to 39% (avg 20%), and for the full system by up to 20% (avg 13%).

2 Background

2.1 Head Tracking

Head tracking is a crucial component in XR that enables immersive and responsive user experiences by continuously monitoring the position and orientation (the 6-Degree-of-Freedom or 6DOF pose) of the user's head. To achieve accurate and high-frequency tracking, XR systems typically rely on a fusion of estimates from multiple tracking techniques that operate on one or multiple sensors. A typical head tracking flow in XR is presented in Figure 1a. VIO algorithms use inertial measurement unit (IMU) data (acceleration and angular velocity) and camera images to generate a 6DOF pose. VIO offers good accuracy, but its frequency is constrained by the high computation latencies and the relatively low camera frequencies of tens of Hertz (Hz). Relying solely on VIO is therefore insufficient to meet the demands of the more frequent pose queries in an XR system (e.g., 120Hz or 144Hz for asynchronous reprojection or timewarp to produce the final images for display). In contrast, the inertial based techniques such as IMU integration are lightweight and able to work at hundreds of Hz, yet can have bad drifting error without timely correction from the VIO poses. Combining VIO and IMU integration enables an XR system to meet both the accuracy and frequency requirements for pose queries.

Additionally, downstream XR components frequently require future pose estimates. For instance, rendering and reprojection are performed based on the anticipated pose at the moment the image will be displayed. Pose prediction algorithms address these needs by estimating the user's future position and orientation, leveraging current velocity or acceleration data under the assumption that human movements maintain momentum briefly [29].

2.2 ILLIXR

RemoteVIO builds upon ILLIXR, an open-source full-system XR research testbed [1, 38, 39]. ILLIXR implements a typical E2E XR workflow all the way from the sensor inputs to the final display. It models computation units such as sensor capturing, head tracking, application rendering, and other major functionality in XR as separately compiled, dynamically linked plugins. The communication between plugins are centrally managed by ILLIXR's runtime.

For applications, ILLIXR supports a native interface and the recent OpenXR standard (implemented with Monado [56]), thereby supporting various applications and game engines such as Godot [46] and Unreal [28]. For live experiments, ILLIXR supports several sensors for live inputs, computes on desktop and embedded class platforms (Section 4), and sends rendered pixels to the display of commercial headsets. Thus, ILLIXR offers a unique open platform for end-to-end XR research with the opportunity to study the impact of system optimizations on live XR experiences.

3 RemoteVIO

In this section, we first highlight the challenges in designing an XR system with head tracking offloaded and our solutions to each of them in RemoteVIO. We next describe our system design and implementation of RemoteVIO as an end-to-end system.

3.1 Challenges and Solutions

Challenge 1: Network Latency. XR, especially head tracking, is very sensitive to latency. Long and unpredictable network delays in wireless connections can make offloaded head tracking feel unresponsive and degrade the user experience.

Solution 1a. On-device IMU Integration As discussed in Section 2.1, the head tracking flow uses both slow but accurate VIO and fast but error-prone IMU integration. Our design, illustrated in Figure 1b, moves the compute-intensive VIO component to the remote server ❹ while keeping the lightweight IMU integration ❿ on the device. When a new VIO pose is not available, the IMU integrator keeps generating estimated poses at hundreds of Hz IMU frequency ❽ to satisfy the high-frequency pose requests from the downstream system. It does so by integrating the IMU-reported acceleration and angular velocity ❼ into translational and rotational changes, and adds the changes on top of the last VIO pose. When the new VIO pose is received from the network, the IMU integrator updates its base pose with the new VIO pose ❿ and biases, which helps maintain the accuracy of its generated poses. This design results in the client getting VIO poses with much larger delays (than on-device VIO) due to the network latency. The IMU integration thus has to operate over a longer period, commensurate with the network latency. The key question is whether the extended

integration period will affect E2E user experience in a perceptible way.[2]

Solution 1b. Low-Pass Filtering We observe that when the IMU integrator updates its base pose whenever a new VIO pose becomes available, a small, rapid, and unintended movement—known as jitter—often occurs in the trajectory. This happens because right before and after a new VIO pose is received, the IMU integrator estimates two adjacent poses: one based on the older VIO pose and a longer period of IMU data, and the other based on the newer VIO pose, using a shorter period of IMU data. Due to network latency, the difference between the two poses can increase because of the longer accumulation of IMU integration errors, resulting in more noticeable jitters. To mitigate the jitters, we apply a low-pass filter [16] to the IMU integrated poses Ⓒ which effectively smooths out the jitter with negligible overhead.

Challenge 2: Network bandwidth. Cameras on current XR devices typically capture stereo images of reasonable resolution at frequencies in the 10s of Hz for tracking [9, 53, 74]. Transmitting them on the network can easily consume tens to hundreds of Mbps of bandwidth (e.g., 115Mbps for the stereo EuRoc dataset[13]). This single XR component could therefore use much of the available WiFi bandwidth and overload 4G or 5G networks.

Solution 2. Compression of camera images. Our design therefore includes an encoder Ⓐ (decoder Ⓑ) to compress (decompress) camera data before (after) transmission (reception). Specifically, we perform H.264 compression/decompression on the raw camera stream using Gstreamer [71], accelerated by NVIDIA NVENC and NVDEC [59]. The compression and decompression functions reside in the device_TX and server_RX plugins (discussed later) respectively (Ⓐ and Ⓑ in Figure 1b). We control the compression ratio by adjusting the compression bitrate. We conducted a bitrate sweep for EuRoC datasets (results omitted here due to space constraints) and found that a target bitrate of 2Mbps (4Mbps for stereo data) provides a balance between low bandwidth and small tracking errors using ATE. Therefore, we use 2Mbps bitrate for other experiments. Previous studies in robotics and autonomous vehicles have found negligible impact of moderate compression on odometry accuracy [61, 62]. However, our key concerns are whether compression leads to user-perceptible degradation and whether the codec adds significant latency and power overhead. Again, using an E2E system for our design and evaluation enables understanding the impact of the compression algorithm on final user experience and potential power benefits.

3.2 End-to-End System Design

Our system, RemoteVIO, employs a client-server model as shown in Figure 1b. VIO is relocated to the server while other components remain on the device (client). The client establishes two connections to the server – one for sending IMU data and camera images and the other for receiving poses – using TCP sockets with the default Linux settings and the TCP_NODELAY flag enabled.[3]

We develop plugins to handle the transmission and reception on both the device-side (device_TX ❶ , device_RX ❾) and the server-side (server_TX ❺ , server_RX ❸) as in Figure 1b.

The device_TX ❶ plugin collects and serializes the IMU data and camera frames ❷ using Protocol Buffers [2]. It then sends the serialized message to the server via a TCP socket. The server_RX ❸ plugin receives the message, deserializes it, and streams the IMU data and camera frames to VIO ❹ to calculate the pose. The new pose is handled by server_TX ❺ , which serializes the pose ❻ and sends it to device_RX ❾ over the network. Finally, device_RX ❾ deserializes the pose and sends it for use in IMU integrator ❿ .

We use open-source state-of-the-art CPU implementations for VIO and IMU integrator – OpenVINS [29] and GTSAM [21, 26] respectively. We expect the results to be extendable to other algorithms and implementations as well.

To create a complete XR experience, we implement our design on top of ILLIXR [38, 39]. We configure our system with the following critical components in addition to all of the components directly associated with head tracking. First, the system runs applications that produce the rendered images for display. We use two game engines – Unreal [28] and Godot [46] – which interface with ILLIXR through the Monado OpenXR implementation[56]. Second, the system executes asynchronous reprojection, also known as timewarp [75]. This process uses the predicted pose (discussed in Section 2.1) for the next vsync (display time) to reproject the rendered frame, correcting for the latency that has occurred since the renderer received its initial pose estimate. Reprojection is crucial as it significantly reduces visual latency, making it a vital component for the XR visual subsystem by continually updating the frame to align with the user's latest pose.

4 Evaluation Methodology

4.1 Experimental Platforms

High-end and embedded class devices. We conduct our experiments on two hardware platforms as the client. We use a high-end desktop PC with an Intel(R) Core i9-10900K CPU and a discrete NVIDIA GeForce RTX 3090 GPU. The powerful desktop provides sufficient resources for the on-device VIO configuration so that any difference in user experience or quality metrics with RemoteVIO can be attributed to offloading. For power measurements, however, we use an embedded class system, NVIDIA Jetson AGX Xavier development board [58], hereafter referred as "Jetson," to represent a device with a power profile similar to modern headsets (Section 4.4).

Cloud-mode and Local-mode. We configure the *cloud-mode* to represent the realistic scenario for a system with VIO offloaded. The XR client is either the desktop PC or the Jetson. The offload server runs on a GPU-enabled server in the Amazon Web Services (AWS) data center [10] that is closest to our client location, a distance of around 300 miles. The client is connected to either WiFi or 5G. For WiFi, we connect to our on-site network, which delivers around 600 Mbps and 400 Mbps for upload and download respectively as measured through Speedtest and 21.5ms mean packet round-trip-time (RTT) as measured through ping [54].

[2]The work in [22] uses similar observations but differs from our work as described in Section 6.

[3]TCP is an appropriate choice for the communication protocol as VIO requires reliable, in-order delivery of the compressed image stream. There are many TCP variants and UDP-based protocols that provide similar service, which may be useful to explore in the future particularly when low latency transport is required and packet dropping is tolerable. As demonstrated here, however, the default TCP algorithm in Linux (the CUBIC algorithm) is sufficient for our purposes.

For 5G experiments, we USB-tether a 5G-capable phone (Samsung S22 or Motorola Edge plus 2022) and share its network with the XR device. The phone is connected to TMobile's 5G Ultra Capacity which is based on the mid-band 5G, delivering about 70 Mbps upload and 300 Mbps download with an average of 63.7ms RTT to the cloud server.

To systematically and repeatably study the impact of network latency and bandwidth on RemoteVIO, we also set up a proxy system as follows. Both the client and the server run on the same high-end desktop PC and communicate through sockets binding to localhost. We either use the Linux Traffic Control (TC) to emulate the communication network based on fixed latency and bandwidth constraints or DChannel's trace-driven 5G emulator [66] to emulate 5G based on real-world traces. We call this mode the *local-mode*.

The user study in Section 4.2 is conducted under eight network conditions: live WiFi and 5G in cloud-mode, and six network conditions in local-mode to ensure a more controlled experimental environment. Specifically, we use TC to set four levels of fixed network RTT of 50ms, 100ms, 200ms, and 400ms. Previous work has shown that the RTT from user clients to the nearest data centers (on WiFi or cellular) is within 200ms on most places on earth in most situations [20]. Our user study thus provides a good coverage of the spectrum of common network conditions, as well as more extreme network scenarios. Additionally, we use the DChannel emulator [66] to emulate a 5G network and examine the impact of 5G network dynamicity on RemoteVIO. The emulator delays traffic based on time-varying RTT and bandwidth values from a trace. For our study, we select the LowBand-Stationary trace, representing Verizon's 5G network in a stationary indoor setting with full signal strength—typical of XR end-user conditions. From this trace, we extract two 100-second segments with different RTT profiles to represent good (5G-low) and poor (5G-high) conditions. Figure 2 shows the RTT and upload bandwidth over time for both traces.

Sensors and Display. We use a Valve Index headset [74] as the display and connect it to our desktop PC and the Jetson [4]. For experiments with live sensor inputs, we attach a ZED mini camera [68] to the front of the headset. As the user walks around in our laboratory space, the camera feeds the system with IMU samples at 500Hz and stereo grayscale images of 672 × 376 pixels at 30 frames per second (FPS) in real-time, and the headset displays frames at 144Hz.

4.2 User Study

To understand users' perception of offloading VIO in the immersive VR environment, we conducted an IRB-approved user study with a total of 30 participants (13 females and 17 males, aged 21-57, recruited via IRB-approved advertisement on our campus social media and mailing lists). 8 participants have no prior XR experience, while 19, 2, and 1 interact with XR rarely (a few days per year), occasionally (a few days most months), and frequently (several days most weeks), respectively. All participants have normal or corrected-to-normal vision.

[4]Ideally, experiments would be conducted directly on a standalone XR headset. However, state-of-the-art XR headsets are closed and proprietary, preventing customized software from running directly on their hardware. Additionally, collecting power measurements on these platforms is not feasible due to proprietary restrictions.

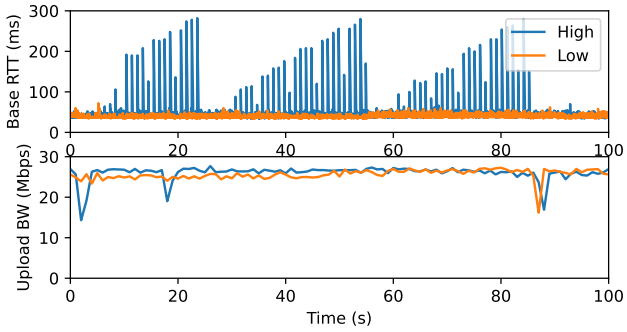


Figure 2: Network RTT and upload bandwidth for traces emulating 5G under low and high variability. Mean RTT for 5G-low is 39.36ms. 5G-high has a 49.84ms mean with spikes up to 282ms.

During the study, participants wore a Valve Index headset configured as described in Section 4.1 and were presented with two VR applications built on Unreal: Zen Garden (featuring a beautiful mansion and natural environment that the user could walk in) [24] and Flight Simulator (developed within the lab, with the user controlling an airplane). We adopted a Comparative Category Rating (CCR) method [36]. For each application, participants had a random subset of eight trials[5] where each trial contained two experiences under two setups: **baseline experience** performed VIO on the client XR device without offloading and **RemoteVIO** offloaded VIO under one of eight network conditions (Section 4.1. The participants were initially instructed to perform movements across all three axes of rotation and translation, followed by freely exploring Zen Garden and piloting in Flight Simulator. After each trial, participants were asked to fill out a questionnaire to assess their perception of the differences between the two experiences. The questionnaire contains the following questions:

- Q1: Did you find any difference in the two XR experiences? (Yes/No)
- Q2: Which experience did you like better? (Experience 1/2)
- Q3: For the experience ranked worse, how significant is the degradation in comparison to the better experience?
  - (a) Only slightly worse or almost the same
  - (b) Worse but the experience is acceptable to me
  - (c) Much worse and not acceptable to me
- Q4: What caused the perceived degradation? (Jitters, lag, jerkiness, drift, lost tracking, others)[6]

Importantly, participants were kept unaware of which experience was the baseline and which was RemoteVIO to avoid biasing their responses. The order of the two experiences in each trial was randomized as well for this purpose.

We categorized the responses to the questionnaire and assigned a numeric rating to the RemoteVIO experience under test:

- Rating 4: No worse than the baseline or indistinguishable: when the user answers "No" to Q1; or when the user answered "Yes" to Q1, but deemed the RemoteVIO experience to be better in Q2.

[5]We limited the number of trials per participant to keep the study at a reasonable length and prevent motion sickness due to prolonged exposure to virtual content.
[6]The different artifacts are explained in the questionnaire. Research staff are available for questions during the study as well.

Rating 3: Slightly worse than or almost the same as the baseline: when the user answered "Yes" to Q1 and (a) to Q3.
Rating 2: Worse than the baseline but the experience is acceptable: when the user answered "Yes" to Q1 and (b) to Q3
Rating 1: Much worse than the baseline and the experience is unacceptable: when the user answered "Yes" to Q1 and (c) to Q3.

During the trials with live WiFi or 5G, we monitored the network condition and noticed one case of unusual latency spike with WiFi (1566ms RTT on average with a maximum of 3783ms). The user experience for this trial was rated as much worse and not acceptable, motivating for better adaptive solutions for edge cases.

Overall, we conducted 248 trials with a total of 496 experiences (248 each for RemoteVIO and baseline). Of the 496 experiences, users reported "Lost tracking" in two cases. This was anomalous (and unrelated to offloading). As these experiments can not be repeated to determine/fix the source of the anomaly, we dropped the two trials from our reported results in Section 5.1.

4.3 Standard Tracking Accuracy Evaluation

While user studies are the gold standard for evaluating XR systems [3, 50, 77], they are expensive, time-consuming, and require significant engineering to build usable E2E systems. Consequently, we aim to assess the impact of offloading on standard metrics to determine if these metrics can serve as proxies for user experience when evaluating the performance of offloading head tracking.

Metrics. We report Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) (separately, for translation and rotation)[81], standard metrics used in the literature to evaluate the accuracy of head tracking [11, 22, 35, 40, 49, 63, 69, 76]. ATE takes the Root Mean Square Error (RMSE) of positional difference between the estimated and ground truth trajectories at corresponding time steps. RPE, focusing on the relative movement, is measured by taking the RMSE of the difference between the motion vector of corresponding pose pairs. For each experiment, we collect the IMU integrated poses at the IMU frequency and calculate the ATE and RPE with respect to the ground truth. We use the evo package [31] to align the trajectories and then measure the ATE and RPE.

Datasets. We use the three Vicon Room 1 and three Vicon Room 2 datasets from the widely used EuRoC MAV Dataset [13] recorded with drones, referred to as V1/V2_{01,02,03} respectively. The 01, 02, and 03 each represent a certain level of tracking difficulty (easy, medium, difficult) based on the drone speed, moving patterns, lighting conditions, blurriness in images, etc. These datasets contain stereo grayscale images of 752 $\times$ 480 pixels at 20 FPS and synchronized IMU measurements at 200Hz. Because the EuRoC datasets are recorded by drones and may not represent realistic movements of a person wearing an XR headset, we also record six additional datasets in two different computer labs using a ZED mini camera attached to the front of a Valve Index VR headset. The ZED mini camera captures stereo images of 672 $\times$ 376 pixels at 30 FPS and IMU measurement at 500Hz. The recorded datasets cover common use scenarios in VR: (1) "walk" where the user is walking slowly and looking around at an interesting site; (2) "static" where the user is sitting and looking at some monitor (e.g., website browsing, video gaming); (3) "game" where the user is playing some active VR game that involves frequent body movements (e.g., fitness games, dodging a bullet); (4) "slow" where the user is moving slowly and smoothly in both rotational and translational motions; (5) and (6) "fast1" and "fast2" where the user's movements are intense and fast in order to stress the system intentionally.

Thus, in total, we use 12 datasets – 6 from EuRoC and 6 from XR experiences. However, the latter six lack the high quality ground truth information usually obtained from motion capture systems. For assessing tracking accuracy for these trajectories in offloaded configurations, we use the poses generated by the on-device IMU integrator, without offloading, as a proxy for the ground truth.

Latency Sweep We perform a network latency sweep from 0 to 1000ms over all the 12 datasets in the local-mode.

We find that the ATE/RPE measurements are not affected by the XR application that runs on the device, so for our dataset experiments we use GLdemo, a lightweight OpenGL-based VR application available in the ILLIXR project.

Correlating with user experience To understand any potential relationship between ATE/RPE and user experience, we record IMU, camera streams, and poses from our RemoteVIO user experience trials. We replay the IMU and camera streams on the baseline system (on-device VIO) to collect a reference trajectory which we use as a proxy for ground truth and compute ATE/RPE for the user study trials. We plot the ATE/RPE against the corresponding user ratings in a scatter plot and calculate Spearman's correlation coefficient [65] to evaluate the relationship between these metrics and actual user experience.

	WiFi	5G
Airplane Mode (Network disabled)	0.658W	
Standby Mode (Network enabled, no traffic)	0.819W	0.909W
Activation Mode (Transmitting small traffic)	0.943W	1.328W
Offloading Mode (RemoteVIO running in cloud-mode)	1.253W	1.734W

Table 1: Wireless power consumption across network modes.

4.4 Power Consumption

We conduct power measurements on the Jetson using GLdemo and Platformer (a stylized, vibrant environment with floating platforms and simple geometries) [30] with nine datasets (six EuRoC datasets and three VR datasets) for repeatability. For each combination, we measure the power consumption of the Jetson with VIO running onboard or offloaded to the AWS server. To resemble the power envelope of commercial XR devices, we configure the Jetson to operate at a CPU frequency of 2.2 GHz, a GPU frequency of 1.3 GHz, a memory controller frequency of 1.06 GHz, and display frequency of 60Hz.

During these experiments, we continuously monitor the Jetson's power rails to measure the power consumption of various subsystems, including the CPU, GPU, DDR (DRAM), SoC (on-chip microcontrollers, encoder and decoder accelerators), and SYS (display ports, storage, and I/O) [39, 60]. The aggregate of these values yields the total power consumption.

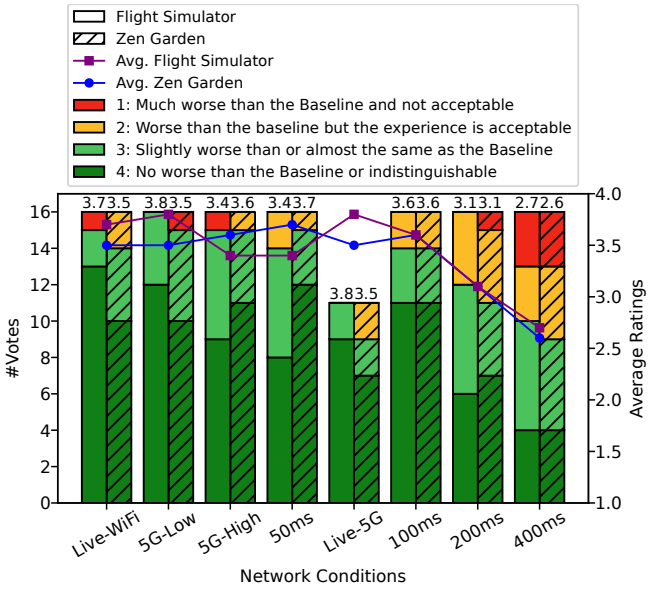


Figure 3: Users' ratings for RemoteVIO compared to the Baseline at different network conditions for Flight Simulator (solid left bars) and Zen Garden (right hatched bars). On top of each bar is the average rating for that application at that network condition. Higher numbers indicate better experiences. Additionally, the average ratings are plotted as line graphs, showing very similar average ratings when the average RTT is below 100ms. However, as the network RTT increases beyond 100ms, the ratings worsen significantly.

Since our intended scenario is a wireless headset, we need to characterize the power consumption of data transmission over the wireless network. To achieve this goal, we tether the Jetson to a Motorola Edge plus 2022 and read the phone's power rails [57] at various modes as shown in Table 1.[7] To isolate the wireless communication power, we maintain the phone in an idle state without running any applications and ensure that the display remains off. Each measurement lasts for one minute, during which we sample instantaneous voltage and current every 100ms. Subsequently, we calculate the average power consumption in Watts (W).

Since current XR devices require access to the Internet to use most features, we safely assume that the radio will be kept on and transmitting data even without VIO offloading. Therefore, the wireless power consumption for an XR device without VIO offloading will be similar to the power in activation mode, as shown in Table 1. To calculate the full system power for on-device VIO, we add the activation mode power to the power measured on the Jetson. For the scenarios where VIO is offloaded, we measure the wireless power in parallel as we measure the power on the Jetson.[8]

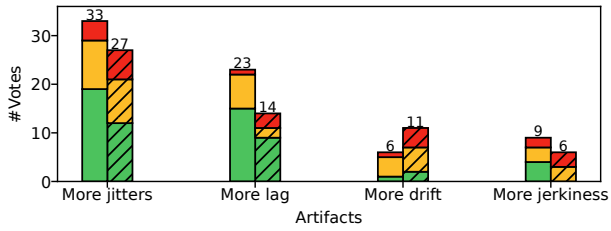


Figure 4: Users' votes on the artifacts that degrade their experience. The legend is the same as Figure 3.

5 Results

5.1 User Experience From a User Study

5.1.1 Impact of Network Latency. Figure 3 reports the results of our user study described in Section 4.2. It covers the outcomes of 246 total trials comparing RemoteVIO with the baseline system (on-device VIO) across two applications and eight different network scenarios involving 30 users. For each network condition, except Live-5G, we conduct 16 trials per application. Only 11 trials are completed for Live-5G due to the loss of access to TMobile's 5G Ultra Capacity device midway through the study. We also report the average ratings on top of each bar. Higher numbers mean worse ratings of the experience.

Figure 3 shows that of all 246 RemoteVIO experiences, 209 (85%) are deemed indistinguishable from, similar to, or only slightly worse than the baseline experience (dark and light green bars at the bottom, rating 4 and 3); 27 (11%) are worse but the experience is still acceptable (yellow bars, second from the top, rating 2); and only 10 (4%) are deemed unacceptable (red bars on top, rating 1). Of the 10 unacceptable experiences, 8 occur at high RTT – one at 200ms, six at 400ms, and one for Live-WiFi where the average/peak application latency is 1566ms/3783ms (an unusually high network congestion scenario).

Considering the impact of latency more specifically, for average network RTT no more than 200ms (5G-low, 5G-high, 50ms, 100ms, 200ms, and 54 live cases), 211 out of 214 experiences (99%) are acceptable. The experiences preserved acceptable quality even under scenarios where there are dynamic variations of RTT, such as 5G-low, 5G-high, and live WiFi and 5G. A one-way ANOVA test [55] shows no significant difference in their mean ratings, with a significance level of 5%. In contrast, experiences with 400ms or higher RTT (including one Live case) see dramatic decrease in acceptable cases (79%). The one-way ANOVA test [55] has a p-value of less than 0.02 when including user responses with network RTT greater than or equal to 400ms, indicating that the ratings are indeed statistically much worse and the experience is degraded significantly at 400ms and above.

Overall, our user study shows that RemoteVIO provides a satisfactory experience until 200ms RTT even with significant variability

[7]The Jetson has no built-in wireless capability but can be equipped with a Wireless Network Card to enable access to WiFi; however, this makes it hard to isolate network power from the rest of the system power. Therefore, we choose to use the tethered phone for both WiFi and 5G power measurements.

[8]We report only the 5G power numbers in the paper. For WiFi, the offloading mode has an average incremental power consumption of 0.3 W over activation mode, compared to 0.4 W for 5G, indicating a lesser impact on overall power savings.

in the latency (as in the 5G traces), and starts to degrade significantly at 400ms and beyond.

5.1.2 Reasons for Experience Degradation. In the user study questionnaire, we ask participants to identify the reasons that contribute to the degradation of their experience when they rate one experience as worse than another. Figure 4 presents the responses. The reasons in decreasing order of the total votes (combined across all applications) are *more jitters* (46%), *more lag* (29%), more drift (13%), and more jerkiness (12%).

The results have several implications. First, jitters and lag are the most perceptible artifacts when VIO is offloaded. Future efforts to improve systems with offloaded head tracking should prioritize reducing these two artifacts. Second, the effects of these artifacts on user experience are not captured well by commonly used metrics such as ATE and RPE, as will be quantitatively shown in Section 5.2.1. Therefore, developing a quantitative metric that better correlates with user experience and can accurately detect and quantify these artifacts is necessary. Finally, we observe that for a majority of cases where users report experiencing "more jitters" and "more lag," the rating for the overall experience is "slightly worse than or almost the same as the Baseline." This suggests that users have a relatively high tolerance for jitters and lag. It would be valuable to validate this finding using a broader range of applications, and to determine the threshold of perceptibility and tolerance for these artifacts, in a way that can be captured by the quantitative metrics.

5.2 Standard VIO Metrics

5.2.1 Correlation between ATE/RPE and user experience. Figure 5 provides scatter plots showing the users' ratings of RemoteVIO experiences alongside the corresponding computed ATE/RPE (using the methodology in Section 4.3).

The figures show no obvious correlation between ATE/RPE and user ratings of their experiences, with a Spearmans' correlation coefficient of less than 0.4 for all cases. For example, a user rating of 4 (unacceptable) sees ATE value of 8.9cm/1.6 degree for a 400ms RTT case, while a user rating of 1 (the best) shows much higher ATE value of 14cm/2.8 degree for a 50ms RTT case. The Live-WiFi unacceptable case in Figure 3 with unusually congested latency scenario shows ATE (RPE) values of only 6.7cm (2.14cm) and 0.9 degree (0.67 degree). This in fact corresponds to the lowest level of ATE and RPE we observe from the latency sweep in the next section and clearly demonstrates the unsuitability of ATE and RPE as metrics for XR systems research.

5.2.2 Impact of Latency on ATE/RPE. Although the previous section already calls to question the validity of ATE/RPE as proxies for user experience in evaluating RemoteVIO, we report the impact of network latency on these widely used metrics for completeness. We use the local-mode with TC injected latency and perform a latency sweep from 0 to 1000ms. Also, we use both the widely adopted EuRoC datasets and our own recorded datasets to better represent XR use cases. Though we expect network latencies to typically remain below 200ms [20, 66], the higher latency sweep enables understanding of the robustness of the system and the metrics' behavior in presence of system stressors such as remote server congestion or high network RTT due to device mobility. As discussed in Section 4.3, we do the latency sweep for 12 datasets, during which we observe the visual quality on a monitor. [9]

Figure 6 shows the ATE, both translational and rotational, for different injected latencies in local-mode (Section 4.1) for all 12 datasets. (RPE is omitted for space and has similar trends.) We first observe that the ATE values at 0ms (i.e., no injected latency) are dataset dependent and cover a wide range from 3.47cm (V2_02) to 17.8cm (slow1). The visual experience, however, is similarly smooth and pleasant for all cases. Thus, the results so far do not provide a clear ATE threshold that can be used to discriminate between acceptable and unacceptable visual experiences.

We next try to determine how *increases* in ATE (rather than absolute values) correlate with rising latency and impact on visual quality. We see that the increase in ATE with increasing latencies is also dataset dependent. For example, the ATE value increases extremely slowly for V1_01 (by 1cm and 0.01 degree respectively when the injected latency is 1000ms) and much faster for fast2 (by almost 10cm and 2.2 degrees when the injected latency is 1000ms). Other datasets fall in between the two extremes.[10] Nevertheless, Figure 6 shows that, overall, the range of ATE at different latencies stays mostly within the range seen at 0ms. However, the visual experience indicates that latencies above 200ms typically result in noticeable degradation. This degradation in the visual quality is corroborated by the user study results, but is not revealed by obvious changes in ATE.

The results in this section show that we cannot reliably use ATE or RPE to predict the user experience in an XR system. That is, there is no clearly identifiable ATE/RPE threshold or ATE/RPE-indicated latency value that guarantees a smooth user experience. This is because ATE and RPE focus more on the overall similarity of trajectories, averaging performance over the entire path, whereas artifacts like jitters are sporadic, subtle, and often missed by these metrics. These results underscore the need for full E2E system evaluation with a user study to understand the impact of offloading VIO on XR user experience.

5.3 Power Savings from Offloading VIO

Figure 7 shows the power consumption of the Jetson for two applications and nine datasets, running the baseline or on-device VIO (left bar) and RemoteVIO offloading to AWS over 5G (right bar). Results with RemoteVIO offloading to WiFi are similar. Each bar shows power consumption distribution across different hardware components. At the top of the right bars, we show the percentage reduction in power for RemoteVIO for the full system and the CPU+network component (separated by /).

RemoteVIO leads to a significant reduction in power consumption, with an overall system decrease of 13% and an average reduction of 30% in the CPU and network (5G) components. The most substantial impact is observed in the CPU, with an average reduction of 42%, which aligns with expectations since VIO is primarily CPU-bound.

[9] Viewing a dataset experiment on a headset usually causes discomfort because the sensor data is independent of the user's motion. Our observations of the visual quality on a monitor are to provide an informal assessment of the impact of latency for an experiment (the formal user study is already reported in Section 5.1).

[10] One exception is for "static" where the rotational error increases by 5 degrees from 0ms to 1000ms.

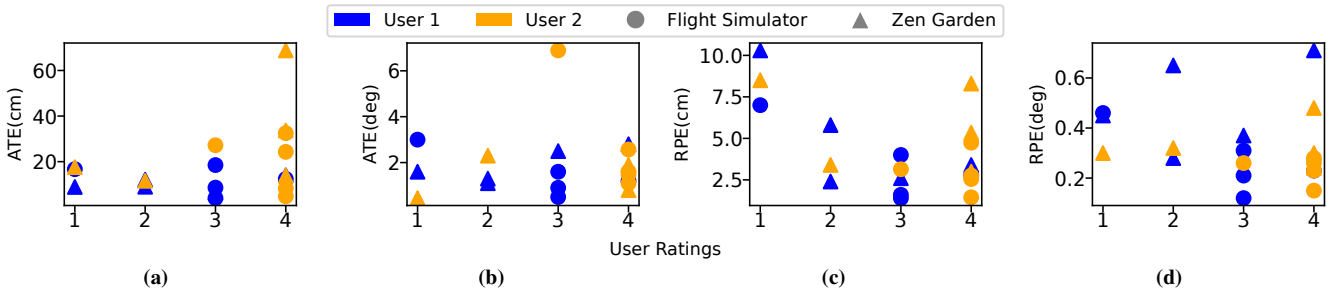


Figure 5: The relationship between two users' ratings of the experiences (in blue and orange) and (a) translational ATE, (b) rotational ATE, (c) translational RPE, (d) rotational RPE, for two applications (circle for FlightSimulator, triangle for ZenGarden). The scatter plots show no strong monotonic relationship (Spearman's correlation coefficient < 0.4) between the metrics and the user experience, suggesting ATE/RPE are inappropriate for evaluating head tracking in XR systems.

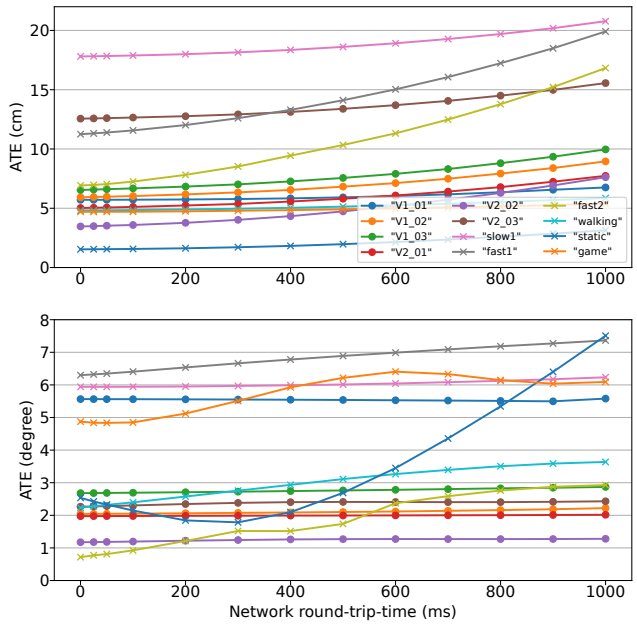


Figure 6: ATE (translational and rotational) across datasets with increasing emulated network RTT in Local-mode.

In addition to the reduction in CPU power consumption, a slight increase in power usage for the GPU and other components is observed in some cases. This occurs because offloading VIO frees up CPU resources for system-level tasks, such as orchestrating memory transfers and synchronization between hardware components. As a result, the performance of other components is enhanced. For example, we observe an 18% increase in the rendering frame rate for Platformer, and the GPU power consumption increases accordingly by 175mW on average. The SYS component also exhibits a small average power increase of 75mW. While we lack visibility into the specific components within SYS, it's reasonable to speculate that the increased GPU and SoC (discussed in the next paragraph) power contribute to this overall rise.

RemoteVIO also introduces power usage for image compression and wireless communication. The power for compression is categorized under SoC. Across all experiments, there is a 4.2% increase in SoC power with an absolute value of 73 mW. The wireless communication power is measured as described in Section 4.4. Incrementally beyond the activation mode (Section 4.4), RemoteVIO adds on average around 0.4 W to 5G power (and 0.3 W to WiFi power), which is small compared to the total power savings of 2.2 W. Thus, RemoteVIO can still bring considerable power savings even accounting for compression and wireless communication power.

5.4 Summary of Results and Implications

In summary, our comprehensive evaluation of RemoteVIO addresses the two questions raised in Section 1: satisfactory user experience can be maintained under typical network conditions (i.e., network RTT below 200ms with variability) and offloading head tracking is an effective approach for reducing power consumption on XR devices. However, we observe significant degradation in user experience when network RTT exceeds the typical range. This decline is not well reflected in the latency sweep experiments with datasets and quantitative metrics. As highlighted in Section 5.2, the increases in ATE and RPE with rising network RTT are minimal (also reported in [22] and exhibit variability across different datasets. Such behaviors of ATE and RPE may lead researchers to overly optimistic conclusions about their system designs. Furthermore, ATE and RPE fail to capture key artifacts, such as jitters and lag, that contribute to user experience degradation. These findings emphasize the importance of conducting end-to-end system evaluations and user studies to ensure the practicality of XR system optimizations.

6 Related Work

Section 6.1 focuses on studies related to offloading head tracking in XR, highlighting how our approach differs from those works. Section 6.2 briefly discusses work on offloading other XR components.

6.1 Offloading Head Tracking in XR

SLAM-share [22] proposes offloading substantial portions of simultaneous localization and mapping (SLAM) to the cloud in a multi-user AR scenario. The SLAM-share design shares similarities with ours in utilizing IMU integration on the device to compensate for network latency. However, its evaluation is conducted by calculating ATE on a standalone SLAM system without considering the user experience. The results indicate that ATE increases from 5.91 cm to 6.58 cm as network RTT rises from 0 to 1000 ms, based

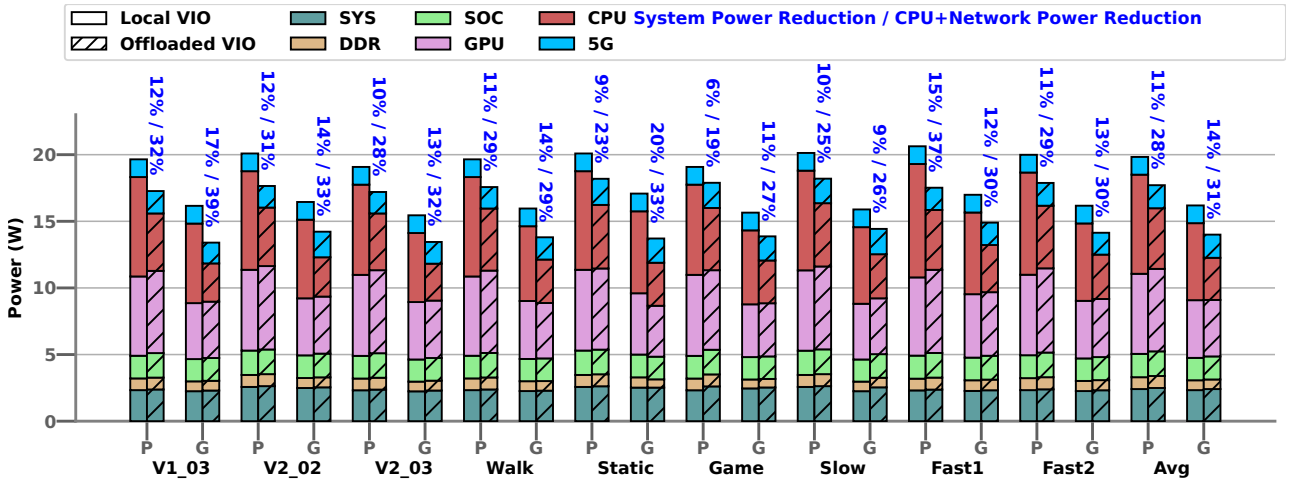


Figure 7: Power consumed on the Jetson with Baseline or on-device VIO (left bar) and RemoteVIO (right bar), for two applications, Platformer (P) and GLdemo (G), and nine datasets. Each bar shows the division of power into different hardware components. On top of the right hatched bars is the System Power Reduction and the CPU+Network Power Reduction, separated by /.

on a dataset from the robotics domain. The findings align closely with our own results on similar datasets (the six EuRoC datasets) in Figure 6. Yet, we show that ATE can behave very differently on datasets that are XR specific (the six datasets we collected). The user study on the E2E RemoteVIO further demonstrates that the experience is significantly degraded when the network RTT goes beyond 200ms and ATE is not well correlated with the user experience. Therefore, user studies with E2E XR systems are crucial. (This observation applies to non-offloading research as well, as ATE is also used in other studies that evaluate on-device tracking or pose prediction techniques for XR [11, 35, 40, 49, 63, 76].) Our study further distinguishes itself by evaluating the power consumption of the resulting E2E system on real hardware. In contrast, SLAM-share speculates the power savings based on the CPU usage and does not account for the communication and codec overhead.

Several other studies [12, 19, 34, 44] have also explored offloading tracking tasks, such as SLAM or VIO. However, none evaluates the end-user experience. Additionally, the network conditions assumed are often constrained, with the server and client usually located close to each other and sharing low network RTT and high bandwidth. In contrast, our evaluation considers challenging real-world scenarios with live networks to cloud servers. Our offloading solution can also be seamlessly deployed on edge servers.

6.2 Offloading Other Tasks in XR

Besides head tracking, there is prior work on offloading other heavy components in XR to the edge or cloud to gain computational and power efficiency. Some popular targets for offloading are rendering [4, 42, 45, 48–50, 52], object detection [8, 14, 17, 18, 32, 43, 47, 51, 78–80], depth estimation [43, 44], and 3D scene reconstruction [41]. These components introduce unique challenges and power-saving opportunities compared to head tracking. For instance, offloading rendering is bandwidth-intensive due to high frequency and large frame sizes, and primarily reduces GPU power usage. Offloading different components can also be complementary; freed CPU resources from offloading head tracking, for example, can be reallocated to mesh decoding required by scene reconstruction offloading. Unfortunately, only a few existing works have properly evaluated user experience (e.g., [4, 50]). Instead, most evaluations rely on task-specific quantitative metrics obtained from incomplete systems, which may not adequately reflect the impacts on the user experience, as demonstrated by our work. Meanwhile, given the extreme power constraints of mobile XR devices, it is also important to ensure that the power overhead of offloading, along with the additional on-device computations required to mask offloading impacts [45, 49, 50], does not violate the power budget of the device. Our work offers an E2E system and a methodology to facilitate such work.

7 Conclusion and Future Work

This paper presents RemoteVIO, the end-to-end XR system that offloads the most computationally intensive part of head tracking. Our extensive user study demonstrates new results – offloaded head tracking provides satisfactory user experience for a range of normal wireless network conditions; however, contradicting previous work that does not measure E2E impacts, we also show that the user experience degrades with increasing network latencies (lower than identified in the prior work). Of equal importance, we find that widely used standard quantitative metrics are not well correlated with user experience. We also show power savings measured on real hardware, a significant result since power consumption is a first order obstacle to integrating more compute in modern headsets and a key motivator for offloading.

Future directions include addressing privacy concerns emerging from transmitting camera images of users' surroundings over the network, co-designed offloading of various XR components, supporting multiple users on the XR server, evaluating AR and MR workloads, and implementing adaptive techniques to enhance system robustness. Each of these directions presents distinct performance tradeoffs. We believe an E2E system will be essential for understanding these tradeoffs. To facilitate progress, we have made RemoteVIO open-source with modular and easily extendable offloading interfaces. Nevertheless, building a usable end-to-end XR system and performing

extensive user studies for new optimizations is difficult. We therefore advocate the development of XR-specific metrics, input datasets, and application benchmarks that can reliably predict how optimizations affect user experience. These tools are crucial for advancing practical XR research and enabling real-world applications.

8 Acknowledgments

We thank our shepherd Mario Montagud Climent and other anonymous reviewers for their valuable feedback. This work is supported in part by the IBM-Illinois Discovery Accelerator Institute (IIDAI), the National Science Foundation under grants 2120464 and 2217144, and gifts from Cisco and T-Mobile.

References

[1] 2021. ILLIXR Consortium. https://illixr.org.
[2] 2022. Protocol Buffers. https://developers.google.com/protocol-buffers.
[3] Ahmad Alhilal, Ze Wu, Yuk Hang Tsui, and Pan Hui. 2024. FovOptix: Human Vision-Compatible Video Encoding and Adaptive Streaming in VR Cloud Gaming. In *Proceedings of the 15th ACM Multimedia Systems Conference* (Bari, Italy) *(MMSys '24)*. Association for Computing Machinery, New York, NY, USA, 67–77. https://doi.org/10.1145/3625468.3647612
[4] Ahmad Alhilal, Ze Wu, Yuk Hang Tsui, and Pan Hui. 2024. FovOptix: Human Vision-Compatible Video Encoding and Adaptive Streaming in VR Cloud Gaming. In *Proceedings of the 15th ACM Multimedia Systems Conference*. 67–77.
[5] Sepehr Alizadehsalehi, Ahmad Hadavi, and Joseph Chuenhuei Huang. 2020. From BIM to extended reality in AEC industry. *Automation in construction* 116 (2020), 103254.
[6] Ahmed Alnagrat, Rizalafande Che Ismail, Syed Zulkarnain Syed Idrus, and Rawad Mansour Abdulhafith Alfaqi. 2022. A review of extended reality (XR) technologies in the future of human education: Current trend and future opportunity. *Journal of Human Centered Technology* 1, 2 (2022), 81–96.
[7] Christopher Andrews, Michael K Southworth, Jennifer NA Silva, and Jonathan R Silva. 2019. Extended reality in medical practice. *Current treatment options in cardiovascular medicine* 21 (2019), 1–12.
[8] Kittipat Apicharttrisorn, Xukan Ran, Jiasi Chen, Srikanth V. Krishnamurthy, and Amit K. Roy-Chowdhury. 2019. Frugal following: power thrifty object detection and tracking for mobile augmented reality. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* (New York, New York) *(SenSys '19)*. Association for Computing Machinery, New York, NY, USA, 96–109. https://doi.org/10.1145/3356250.3360044
[9] Apple. 2023. Apple Vision Pro. https://www.apple.com/apple-vision-pro/specs/.
[10] Amazon AWS. 2024. Amazon EC2 G4 Instances. https://aws.amazon.com/ec2/instance-types/g4/.
[11] Armand Behroozi, Yuxiang Chen, Vlad Fruchter, Lavanya Subramanian, Sriseshan Srikanth, and Scott Mahlke. 2024. SlimSLAM: An Adaptive Runtime for Visual-Inertial Simultaneous Localization and Mapping. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*. 900–915.
[12] Ali J Ben Ali, Marziye Kouroshli, Sofiya Semenova, Zakieh Sadat Hashemifar, Steven Y Ko, and Karthik Dantu. 2022. Edge-SLAM: Edge-assisted visual simultaneous localization and mapping. *ACM Transactions on Embedded Computing Systems* 22, 1 (2022), 1–31.
[13] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research* (2016). https://doi.org/10.1177/0278364915620033 arXiv:http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.full.pdf+html
[14] Jacky Cao, Xiaoli Liu, Xiang Su, Sasu Tarkoma, and Pan Hui. 2021. Context-aware augmented reality with 5G edge. In *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.
[15] Leonor Adriana Cárdenas-Robledo, Óscar Hernández-Uribe, Carolina Reta, and Jose Antonio Cantoral-Ceballos. 2022. Extended reality applications in industry 4.0.–A systematic literature review. *Telematics and Informatics* 73 (2022), 101863.
[16] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 € Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639
[17] Kaifei Chen, Tong Li, Hyung-Sin Kim, David E Culler, and Randy H Katz. 2018. Marvel: Enabling mobile augmented reality with low energy and low latency. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 292–304.
[18] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. 2015. Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 155–168.
[19] Ying Chen, Hazer Inaltekin, and Maria Gorlatova. 2023. AdaptSLAM: Edge-Assisted Adaptive SLAM with Resource Constraints via Uncertainty Minimization. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 1–10. https://doi.org/10.1109/INFOCOM53939.2023.10229009
[20] The Khang Dang, Nitinder Mohan, Lorenzo Corneo, Aleksandr Zavodovski, Jörg Ott, and Jussi Kangasharju. 2021. Cloudy with a chance of short RTTs: analyzing cloud connectivity in the internet. In *Proceedings of the 21st ACM Internet Measurement Conference*. 62–79.
[21] Frank Dellaert and GTSAM Contributors. 2022. *borglab/gtsam*. https://doi.org/10.5281/zenodo.5794541
[22] Aditya Dhakal, Xukan Ran, Yunshu Wang, Jiasi Chen, and K. K. Ramakrishnan. 2022. SLAM-Share: Visual Simultaneous Localization and Mapping for Real-Time Multi-User Augmented Reality. In *Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies* (Roma, Italy) *(CoNEXT '22)*. Association for Computing Machinery, New York, NY, USA, 293–306. https://doi.org/10.1145/3555050.3569142
[23] Sanika Doolani, Callen Wessels, Varun Kanal, Christos Sevastopoulos, Ashish Jaiswal, Harish Nambiappan, and Fillia Makedon. 2020. A review of extended reality (xr) technologies for manufacturing training. *Technologies* 8, 4 (2020), 77.
[24] Epic. 2020. Epic Zen Garden. https://www.unrealengine.com/marketplace/en-US/product/epic-zen-garden.
[25] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. 2011. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture* (San Jose, California, USA) *(ISCA '11)*. 365–376. https://doi.org/10.1145/2000064.2000108
[26] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. 2015. *IMU preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation*.
[27] Meta Community Forums. 2024. Quest 3 Power Usage Tests and Approximate Battery Life. https://communityforums.atmeta.com/t5/Talk-VR/Quest-3-Power-Usage-Tests-and-Approximate-Battery-Life/td-p/1094433.
[28] Epic Games. 2024. Unreal Engine. https://www.unrealengine.com/en-US.
[29] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. 2019. OpenVINS: A Research Platform for Visual-Inertial Estimation. *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications* (2019).
[30] Godot. 2020. Platformer 3D. https://github.com/godotengine/godot-demo-projects/tree/master/3d/platformer.
[31] Michael Grupp. 2017. evo: Python package for the evaluation of odometry and SLAM. https://github.com/MichaelGrupp/evo.
[32] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2014. Towards wearable cognitive assistance. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. 68–81.
[33] David Heaney. 2024. Apple Vision Pro Battery Capacity Reveals Its True Purpose. https://www.uploadvr.com/apple-vision-pro-battery-capacity/.
[34] Jin Heo, Ketan Bhardwaj, and Ada Gavrilovska. 2023. FleXR: A System Enabling Flexibly Distributed Extended Reality. In *Proceedings of the 14th ACM Multimedia Systems Conference* (Vancouver, BC, Canada) *(MMSys '23)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3587819.3590966
[35] Tianyi Hu, Fan Yang, Tim Scargill, and Maria Gorlatova. 2024. Apple vs. Meta: A Comparative Study on Spatial Tracking in SOTA XR Headsets. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*.
[36] Zixia Huang, Ahsan Arefin, Pooja Agarwal, Klara Nahrstedt, and Wanmin Wu. 2012. Towards the understanding of human perceptual quality in tele-immersive shared activity. In *Proceedings of the 3rd Multimedia Systems Conference* (Chapel Hill, North Carolina) *(MMSys '12)*. Association for Computing Machinery, New York, NY, USA, 29–34. https://doi.org/10.1145/2155555.2155560
[37] Muhammad Huzaifa. 2023. *Design and Evaluation of Extended Reality Systems*. Ph. D. Dissertation. University of Illinois Urbana-Champaign.
[38] Muhammad Huzaifa, Rishi Desai, Samuel Grayson, Xutao Jiang, Ying Jing, Jae Lee, Fang Lu, Yihan Pang, Joseph Ravichandran, Finn Sinclair, Boyuan Tian, Hengzhi Yuan, Jeffrey Zhang, and Sarita V. Adve. 2021. ILLIXR: Enabling End-to-End Extended Reality Research. In *2021 IEEE International Symposium on Workload Characterization (IISWC)*. 24–38. https://doi.org/10.1109/IISWC53511.2021.00014
[39] Muhammad Huzaifa, Rishi Desai, Samuel Grayson, Xutao Jiang, Ying Jing, Jae Lee, Fang Lu, Yihan Pang, Joseph Ravichandran, Finn Sinclair, Boyuan Tian, Hengzhi Yuan, Jeffrey Zhang, and Sarita V. Adve. 2022. ILLIXR: An Open Testbed to Enable Extended Reality Systems Research. *IEEE Micro* 42, 4 (2022), 97–106. https://doi.org/10.1109/MM.2022.3161018
[40] Gazi Karam Illahi, Ashutosh Vaishnav, Teemu Kämäräinen, Matti Siekkinen, and Mario Di Francesco. 2023. Learning to predict head pose in remotely-rendered

virtual reality. In *Proceedings of the 14th Conference on ACM Multimedia Systems*. 27–38.

[41] Tao Jin, Edward Lu, Mallesham Dasari, Kittipat Apicharttrisorn, Srinivasan Seshan, and Anthony Rowe. 2024. MeshReduce: Split Rendering of Live 3D Scene for Virtual Teleportation. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 1186–1189.

[42] Zhihui Ke, Xiaobo Zhou, Dadong Jiang, Hao Yan, and Tie Qiu. 2023. CollabVr: Reprojection-Based Edge-Client Collaborative Rendering for Real-Time High-Quality Mobile Virtual Reality. In *2023 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 304–316.

[43] Z. Jonny Kong, Qiang Xu, and Y. Charlie Hu. 2024. ARISE: High-Capacity AR Offloading Inference Serving via Proactive Scheduling. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services* (Minato-ku, Tokyo, Japan) *(MOBISYS '24)*. Association for Computing Machinery, New York, NY, USA, 451–464. https://doi.org/10.1145/3643832.3661894

[44] Z Jonny Kong, Qiang Xu, Jiayi Meng, and Y Charlie Hu. 2023. AccuMO: Accuracy-centric multitask offloading in edge-assisted mobile augmented reality. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–16.

[45] Zeqi Lai, Y Charlie Hu, Yong Cui, Linhui Sun, Ningwei Dai, and Hung-Sheng Lee. 2019. Furion: Engineering high-quality immersive virtual reality on today's mobile devices. *IEEE Transactions on Mobile Computing* 19, 7 (2019), 1586–1602.

[46] Juan Linietsky, Ariel Manzur, and contributors. 2024. Godot Engine. https://godotengine.org.

[47] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge Assisted Real-time Object Detection for Mobile Augmented Reality. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 25, 16 pages. https://doi.org/10.1145/3300061.3300116

[48] Luyang Liu, Ruiguang Zhong, Wuyang Zhang, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Marco Gruteser. 2018. Cutting the cord: Designing a high-quality untethered vr system with low latency remote rendering. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 68–80.

[49] Xing Liu, Christina Vlachou, Mao Yang, Feng Qian, Lidong Zhou, Chendong Wang, Lifei Zhu, Kyu-Han Kim, Gabriel Parmer, Qi Chen, et al. 2020. Firefly: Untethered multi-user {VR} for commodity mobile devices. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 943–957.

[50] Edward Lu, Sagar Bharadwaj, Mallesham Dasari, Connor Smith, Srinivasan Seshan, and Anthony Rowe. 2023. Renderfusion: Balancing local and remote rendering for interactive 3d scenes. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 312–321.

[51] Jiayi Meng, Z. Jonny Kong, Y. Charlie Hu, Mun Gi Choi, and Dhananjay Lal. 2022. Do we need sophisticated system design for edge-assisted augmented reality?. In *Proceedings of the 5th International Workshop on Edge Systems, Analytics and Networking* (Rennes, France) *(EdgeSys '22)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3517206.3526267

[52] Jiayi Meng, Sibendu Paul, and Y. Charlie Hu. 2020. Coterie: Exploiting Frame Similarity to Enable High-Quality Multiplayer VR on Commodity Mobile Devices. (2020), 923–937. https://doi.org/10.1145/3373376.3378516

[53] Meta. 2024. Meta Quest 2 Tech Specs. https://www.meta.com/quest/products/quest-2/tech-specs.

[54] Microsoft. 2023. ping. https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/ping.

[55] Prabhaker Mishra, Uttam Singh, Chandra M Pandey, Priyadarshni Mishra, and Gaurav Pandey. 2019. Application of student's t-test, analysis of variance, and covariance. *Annals of cardiac anaesthesia* 22, 4 (2019), 407–411.

[56] Monado [n. d.]. Monado - Open Source XR Platform. https://monado.dev/

[57] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, et al. 2021. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 610–625.

[58] NVIDIA. 2024. NVIDIA Jetson Xavier. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/.

[59] NVIDIA. 2024. NVIDIA Video Codec SDK. https://developer.nvidia.com/video-codec-sdk.

[60] NVIDIA. 2024. Software-based power consumption modeling. https://docs.nvidia.com/jetson/archives/r34.1/DeveloperGuide/index.html#page/.

[61] Li Qingqing, J Penia Queralta, Tuan Nguyen Gia, Hannu Tenhunen, Zhuo Zou, and Tomi Westerlund. 2019. Visual odometry offloading in internet of vehicles with compression at the edge of the network. In *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 1–2.

[62] Li Qingqing, Jorge Pena Queralta, Tuan Nguyen Gia, and Tomi Westerlund. 2019. Offloading monocular visual odometry with edge computing: Optimizing image quality in multi-robot systems. In *Proceedings of the 2019 5th International Conference on Systems, Control and Communications*. 22–26.

[63] Tim Scargill, Ying Chen, Tianyi Hu, and Maria Gorlatova. 2023. SiTAR: Situated Trajectory Analysis for In-the-Wild Pose Error Estimation. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 283–292. https://doi.org/10.1109/ISMAR59233.2023.00043

[64] Robert R Schaller. 1997. Moore's law: past, present and future. *IEEE spectrum* 34, 6 (1997), 52–59.

[65] Philip Sedgwick. 2014. Spearman's rank correlation coefficient. *Bmj* 349 (2014).

[66] William Sentosa, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Haitham Hassanieh, and Bruce Maggs. 2023. DChannel: Accelerating Mobile Applications With Parallel High-bandwidth and Low-latency Channels. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, Boston, MA, 419–436. https://www.usenix.org/conference/nsdi23/presentation/sentosa

[67] Omar Sohail. 2023. Apple Vision Pro Battery Capacity Said To Be Equivalent To An External Powerbank, But Runtime Does Not Match Current Figures. https://wccftech.com/apple-vision-pro-battery-capacity-equal-to-powerbank/.

[68] Stereolabs. 2024. ZED Mini - Mixed-Reality Camera. https://www.stereolabs.com/zed-mini/.

[69] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 573–580.

[70] Dean Takahashi. 2018. Oculus chief scientist Mike Abrash still sees the rosy future through AR/VR glasses. https://venturebeat.com/2018/09/26/oculus-chiefscientist-mike-abrash-still-sees-the-rosy-future-through-ar-vr-glasses/.

[71] GStreamer Team. 2024. GStreamer: open source multimedia framework. https://gstreamer.freedesktop.org.

[72] Thomas N. Theis and H.-S. Philip Wong. 2017. The End of Moore's Law: A New Beginning for Information Technology. *Computing in Science & Engineering* 19, 2 (2017), 41–50. https://doi.org/10.1109/MCSE.2017.29

[73] Alan Truly. 2022. How long does the Quest Pro battery really last? Here's Meta's answer. https://www.digitaltrends.com/computing/how-long-does-quest-pro-battery-last-metas-answer/.

[74] Valve. 2024. Valve Index Headset. https://www.valvesoftware.com/en/index/headset.

[75] Johannes Marinus Paulus Van Waveren. 2016. The asynchronous time warp for virtual reality on consumer hardware. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. 37–46.

[76] Junyi Wang and Yue Qi. 2023. Scene-independent Localization by Learning Residual Coordinate Map with Cascaded Localizers. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 79–88.

[77] Nan Wu, Kaiyan Liu, Ruizhi Cheng, Bo Han, and Puqi Zhou. 2024. Theia: Gaze-driven and Perception-aware Volumetric Content Delivery for Mixed Reality Headsets. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*. 70–84.

[78] Wenxiao Zhang, Bo Han, and Pan Hui. 2018. Jaguar: Low latency mobile augmented reality with flexible tracking. In *Proceedings of the 26th ACM international conference on Multimedia*. 355–363.

[79] Wenxiao Zhang, Bo Han, and Pan Hui. 2022. Sear: Scaling experiences in multi-user augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 28, 5 (2022), 1982–1992.

[80] Wenxiao Zhang, Sikun Lin, Farshid Hassani Bijarbooneh, Hao-Fei Cheng, Tristan Braud, Pengyuan Zhou, Lik-Hang Lee, and Pan Hui. 2022. Edgexar: A 6-dof camera multi-target interaction framework for mar with user-friendly latency compensation. *Proceedings of the ACM on Human-Computer Interaction* 6, EICS (2022), 1–24.

[81] Zichao Zhang and Davide Scaramuzza. 2018. A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 7244–7251. https://doi.org/10.1109/IROS.2018.8593941