An Open-Source Testbed to Democratize Extended Reality Research, Development, and Benchmarking

Sarita Adve and Muhammad Huzaifa

ILLIXR project
University of Illinois at Urbana-Champaign
illixr@cs.illinois.edu

This work is supported in part by DARPA, NSF, and the Applications Driving Architecture (ADA) Research center (JUMP center co-sponsored by SRC & DARPA)

Extended Reality (XR): The Next Interface

Virtual, Augmented, Mixed Reality

The next computing interface

Will transform science, medicine, education, …

BUT orders of magnitude gap

in power, performance, quality-of-experience

between current and desired systems

Approximate	Current	Desired
Res (Mpixels)	**4**	**200**
Power (W)	**10**	**0.1**
Weight (g)	**500**	**10**
…	**…**	**…**

XR Systems: Challenges

Orders of magnitude gap

Power, performance, quality-of-experience (QoE)

Approximate	Current	Desired
Res (Mpixels)	**4**	**200**
Power (W)	**10**	**0.1**
Weight (g)	**500**	**10**
…	**…**	**…**

Diverse expertise

graphics, vision, audio, video, optics, haptics, …

Cross-layer system co-design

hardware, compiler, OS, algorithm

Complex metrics

multiple, user-driven, end-to-end QoE metrics

Closed systems, few participants

No open reference systems or benchmarks

Large barrier to entry for open R&D

How can we democratize XR systems research, development, benchmarking?

ILLIXR: Illinois Extended Reality Testbed

ILLIXR: Open-source full system XR testbed

State-of-the-art XR components

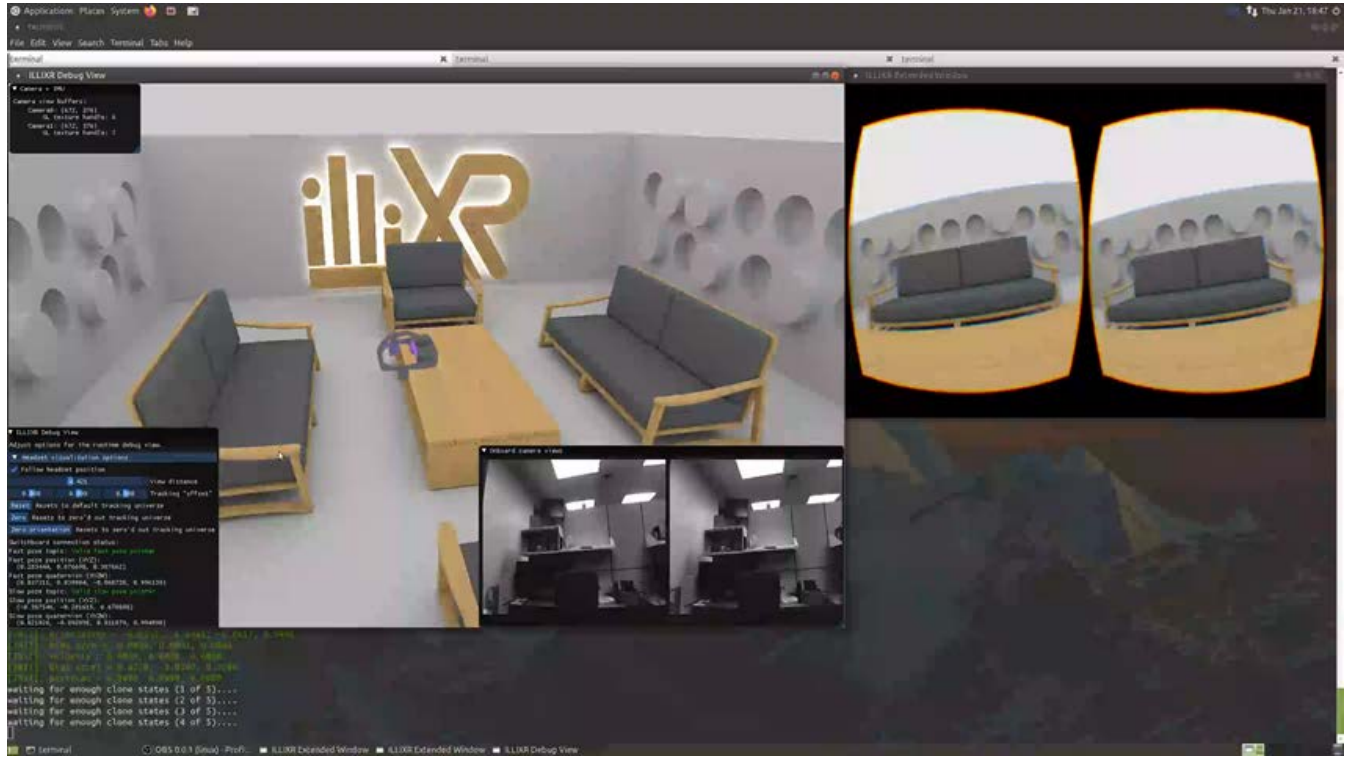
Modular & extensible runtime

OpenXR compatible

Several QoE metrics

Runs on desktop & embedded systems

Extensive characterization and use for research



illixr.github.io

Huzaifa et al., ‘20

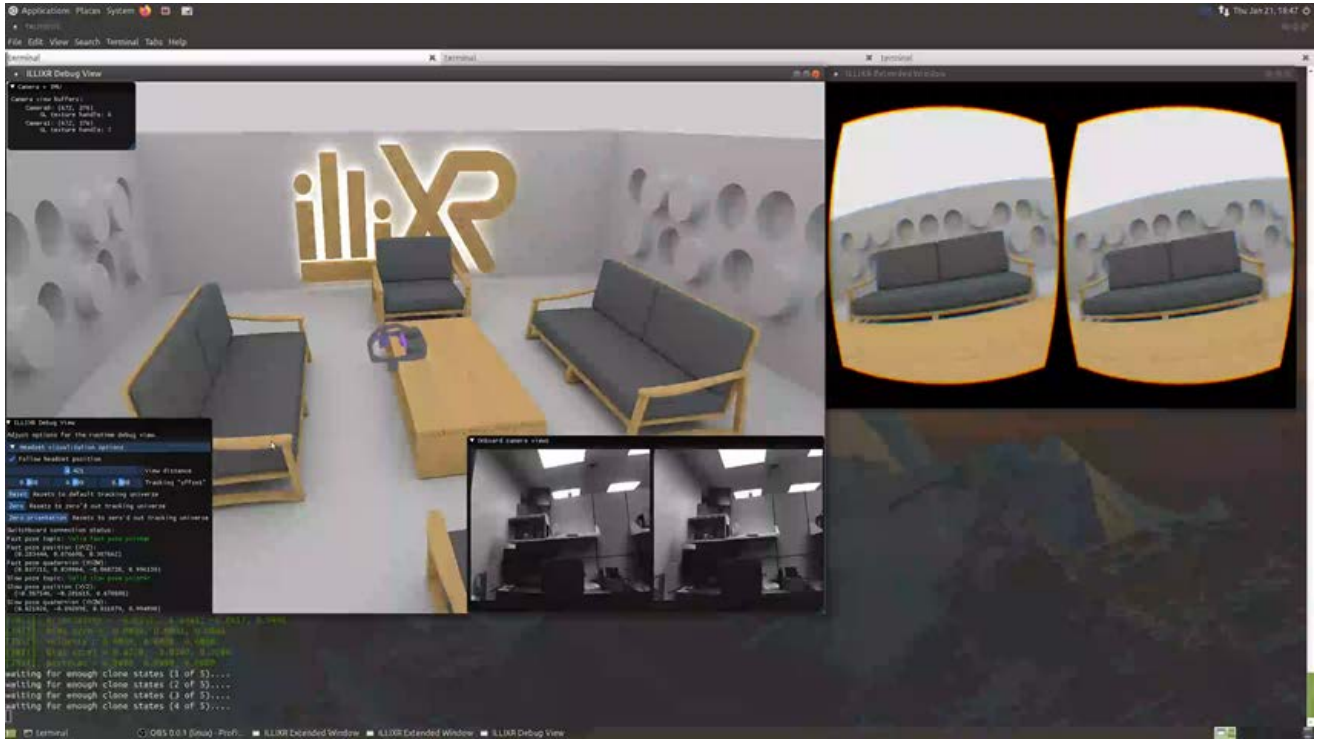
ILLIXR Consortium

ILLIXR Consortium w/ industry + academic partners

- ARM, Facebook, Micron, North Star, NVIDIA, …

Goals

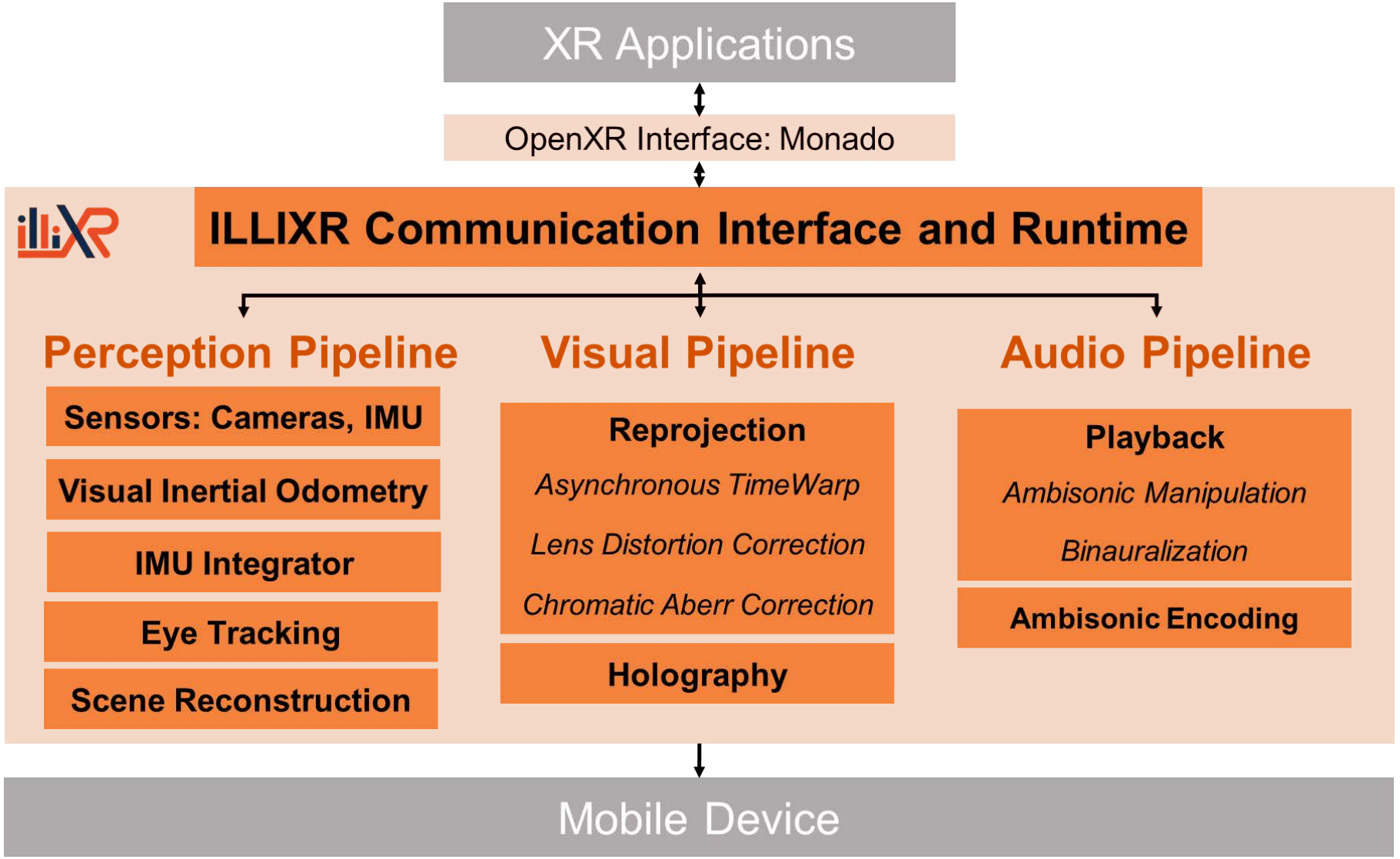
- Reference open source testbed
  - Components and interfaces
  - Modular, extensible runtime
  - Telemetry
- Benchmarking methodology
  - Applications, data sets
  - System configurations
  - Metrics



- Build XR systems research and development community

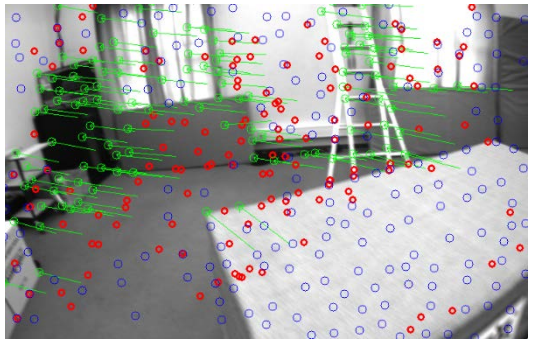
Join us: illixr@cs.illinois.edu, illixr.org

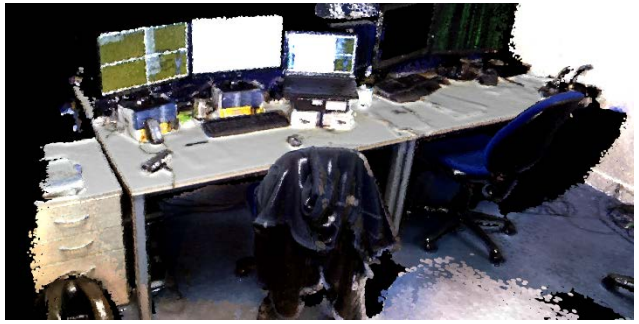
ILLIXR Overview

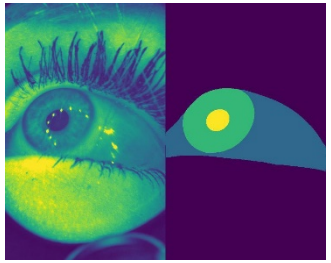


Perception Pipeline

- Sensors: Camera, Inertial Measurement Unit (IMU)
- Visual Intertial Odometry (VIO)
  - Provides position and head orientation (pose)
- IMU Integrator
  - Provides high frequency pose estimates
- Pose Predictor
  - Extrapolates pose to future timestamp
- Scene Reconstruction
  - Uses RGB-Depth camera to build dense 3D map of world
- Eye Tracking

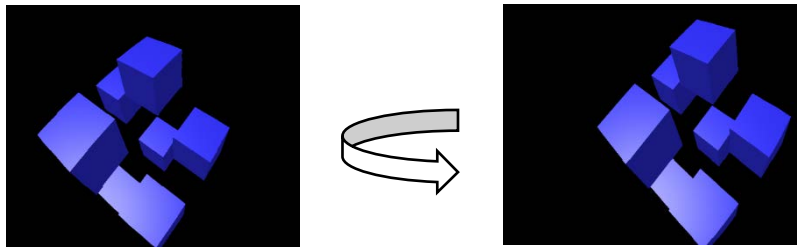




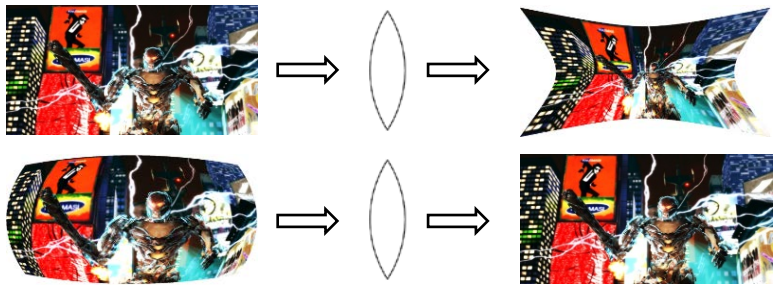


Visual Pipeline

- Asynchronous reprojection
  - Warp rendered frame to account for head movement during rendering
  - Uses latest pose estimate and prediction
  - Cuts motion-to-photon latency



- Lens distortion and chromatic aberration correction
  - Corrects for distortion due to curved lenses



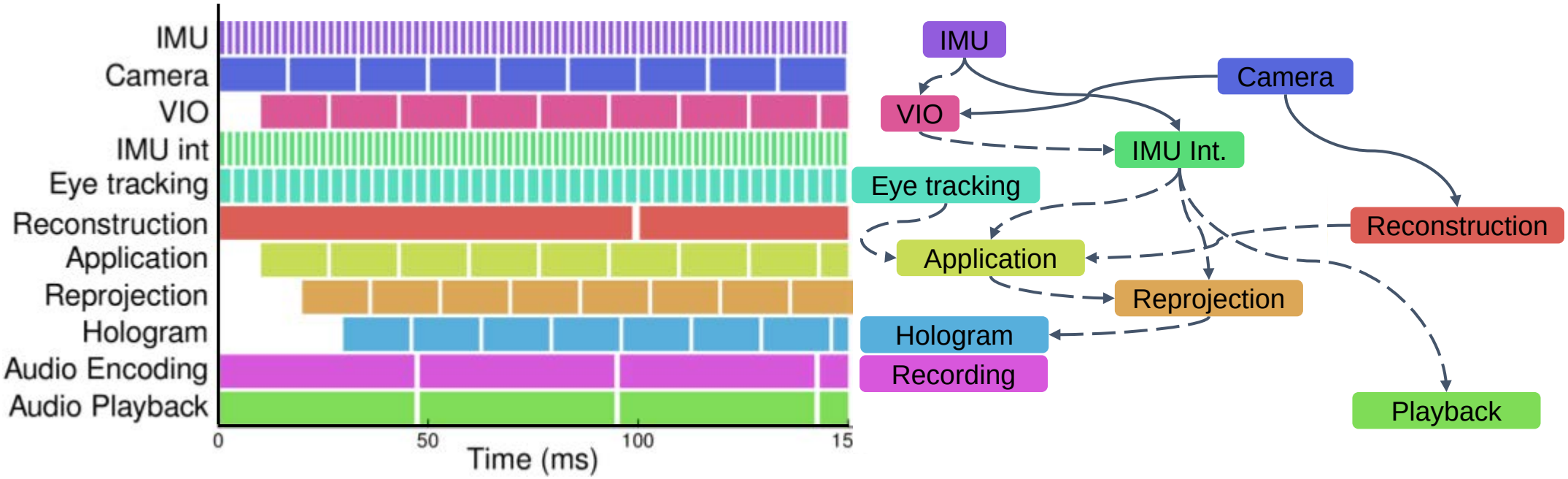
- Computational holography
  - Vergence-accommodation conflict (VAC): eyes focused at fixed point, converge at different points
  - Computational displays w/ multiple focal planes can fix VAC: compute per-pixel phase shift

Audio Pipeline

- Audio encoding
  - Encodes multiple sound sources into Higher Order Ambisonics (HOA) soundfield

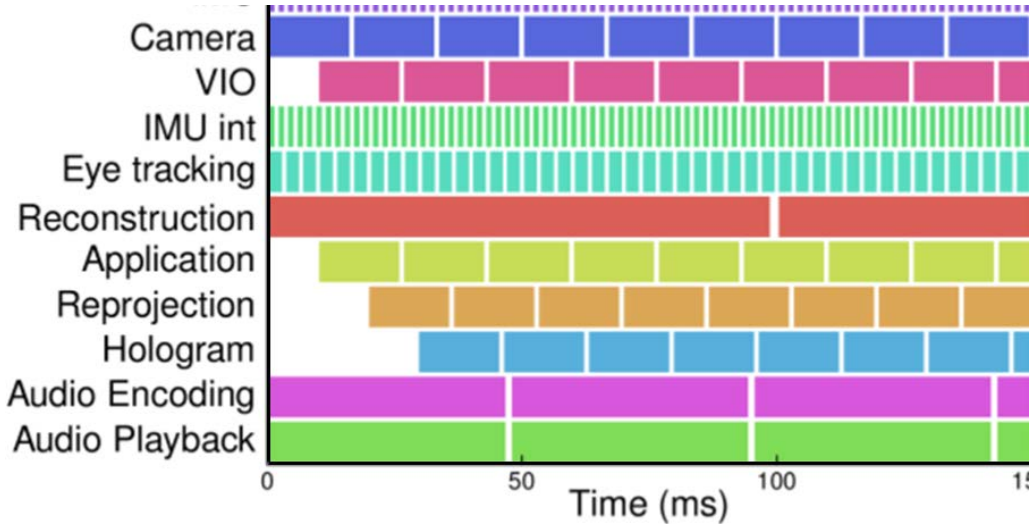
- Playback
  - Rotates and zooms HOA sound field for user's latest pose
  - Performs binauralization to account for user's ear, head, nose

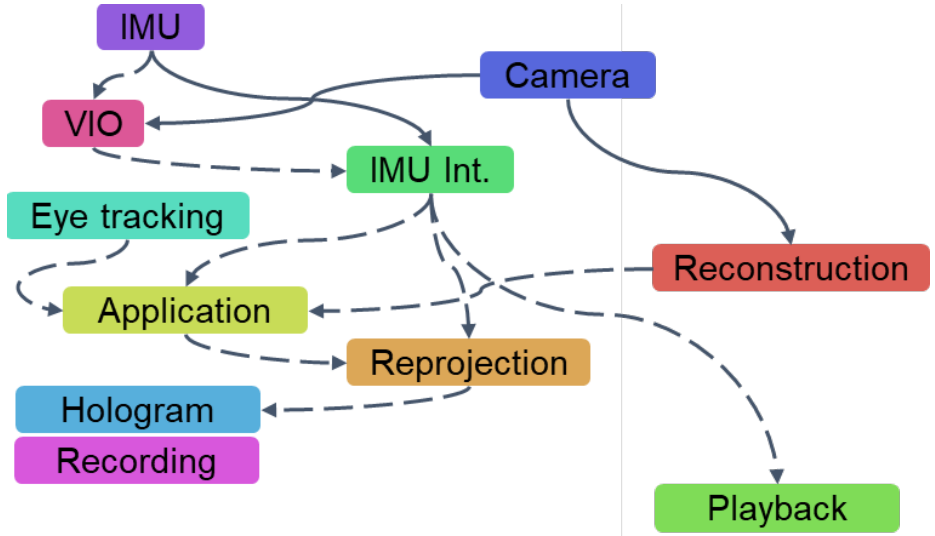
XR System Dataflow



Different components at different frequencies
Multiple interacting pipelines
Synchronous and asynchronous dependences
Multiple quality of experience metrics

ILLIXR Runtime





Modular, flexible architecture

ILLIXR components are plugins

Separately compiled, dynamically loaded

Easily swap/add new components, implementations

Efficient, flexible communication interface

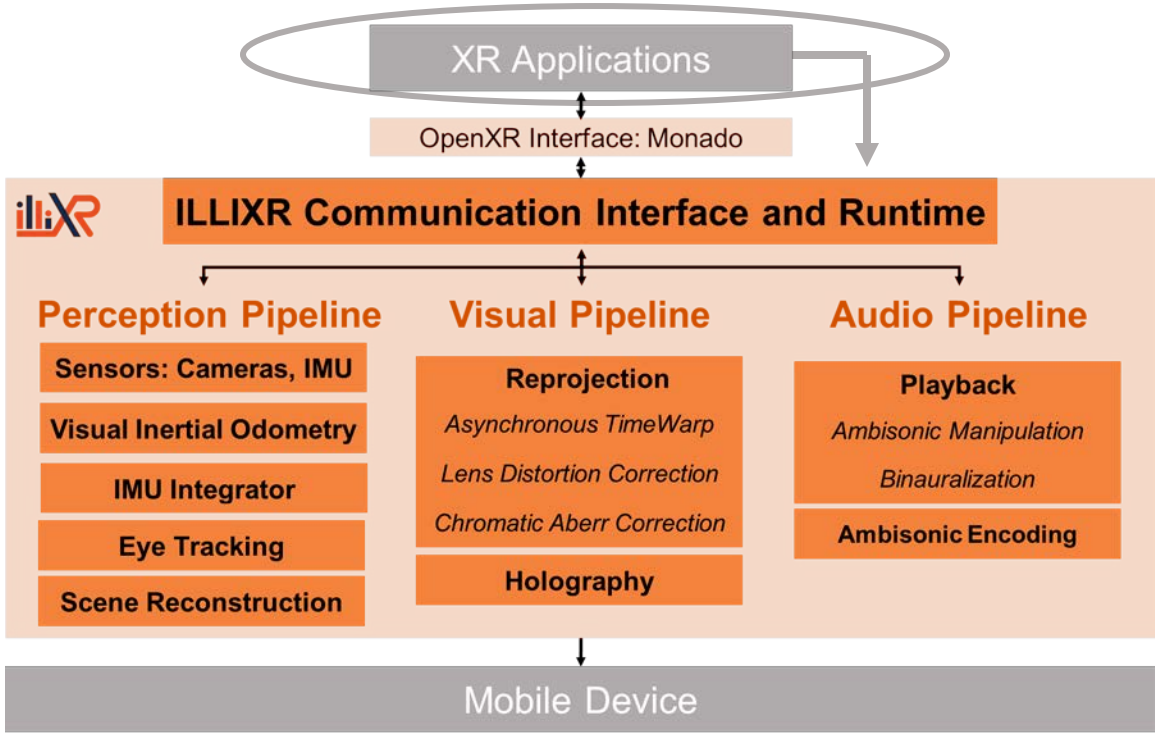
Component specifies event streams to publish, subscribe

Synchronous or asynchronous consumers

Copy-free, shared memory implementation

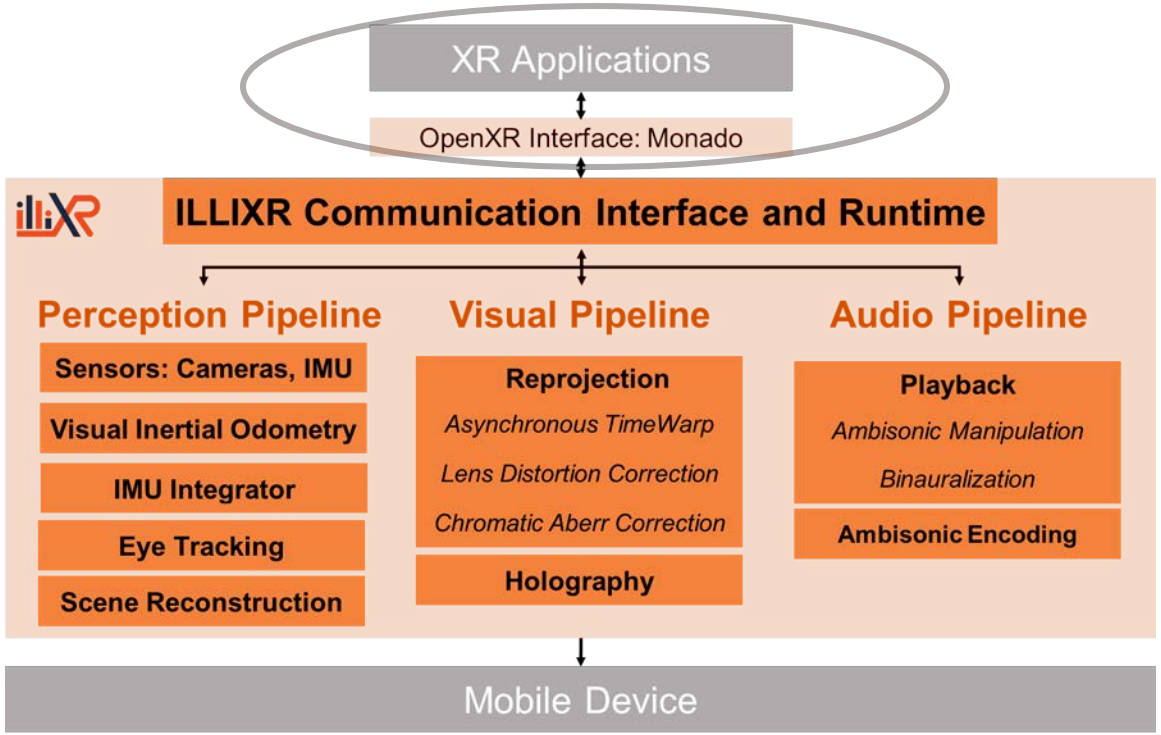
End-to-end system balances flexibility with efficiency

ILLIXR Applications



Can write XR applications directly to ILLIXR

ILLIXR Applications



Can write XR applications directly to ILLIXR

ILLIXR supports OpenXR applications

- Uses Monado implementation of OpenXR
- Today: Godot game engine with many apps
- Soon: Unity, Unreal, …

End-to-End Quality Metrics

- Motion-to-photon latency
  - Time from head motion to display (currently w/o display latency)

- Image quality: SSIM and FLIP

- Pose: Average Trajectory Error and Relative Pose Error

+ Extensive telemetry: Frame rates, missed frames, time distributions, power, …

ILLIXR Components Today

	Component	Algorithm	Implementation
Perception Pipeline	Camera	ZED SDK	C++
	Camera	Intel RealSense SDK	C++
	IMU	ZED SDK	C++
	IMU	Intel RealSense SDK	C++
	VIO	OpenVINS	C++
	VIO	Kimera-VIO	C++
	IMU Integrator	RK4	C++
	IMU Integrator	GTSAM	C++
	Eye Tracking	RITnet	Python, CUDA
	Scene Reconstruction	ElasticFusion	C++, CUDA, GLSL
	Scene Reconstruction	KinectFusion	C++, CUDA
Visual Pipeline	Reprojection	VP-matrix reproject w/ pose	C++, GLSL
	Lens Distortion	Mesh-based radial distortion	C++, GLSL
	Chromatic Aberration	Mesh-based radial distortion	C++, GLSL
	Adaptive Display	Weighted Gerchberg-Saxton	CUDA
Audio Pipeline	Audio Encoding	Ambisonic encoding	C++
	Audio Playback	Ambisonic manipulation, binauralization	C++

Evaluating and Using ILLIXR

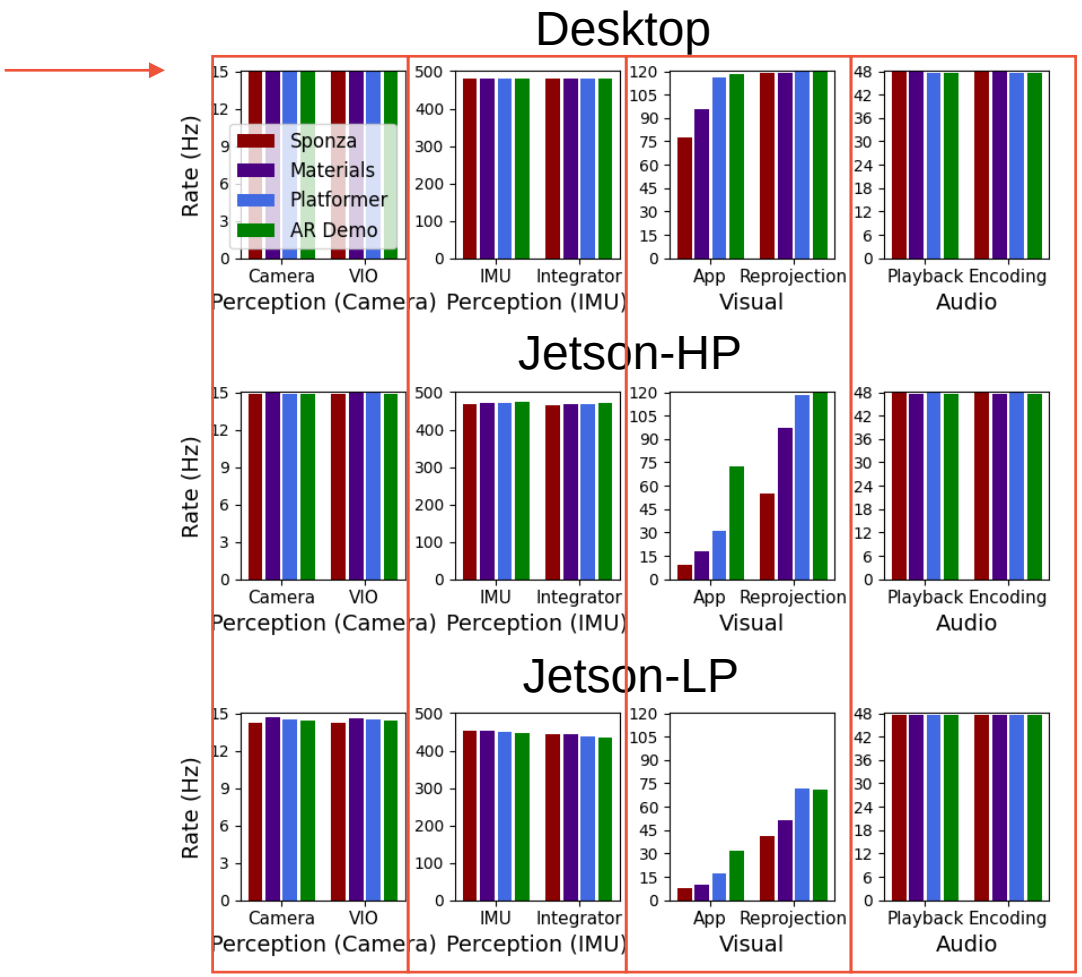
- Research, development, benchmarking
- Testbed provides full visibility into XR system
- Insights for system designers

Evaluation Methodology

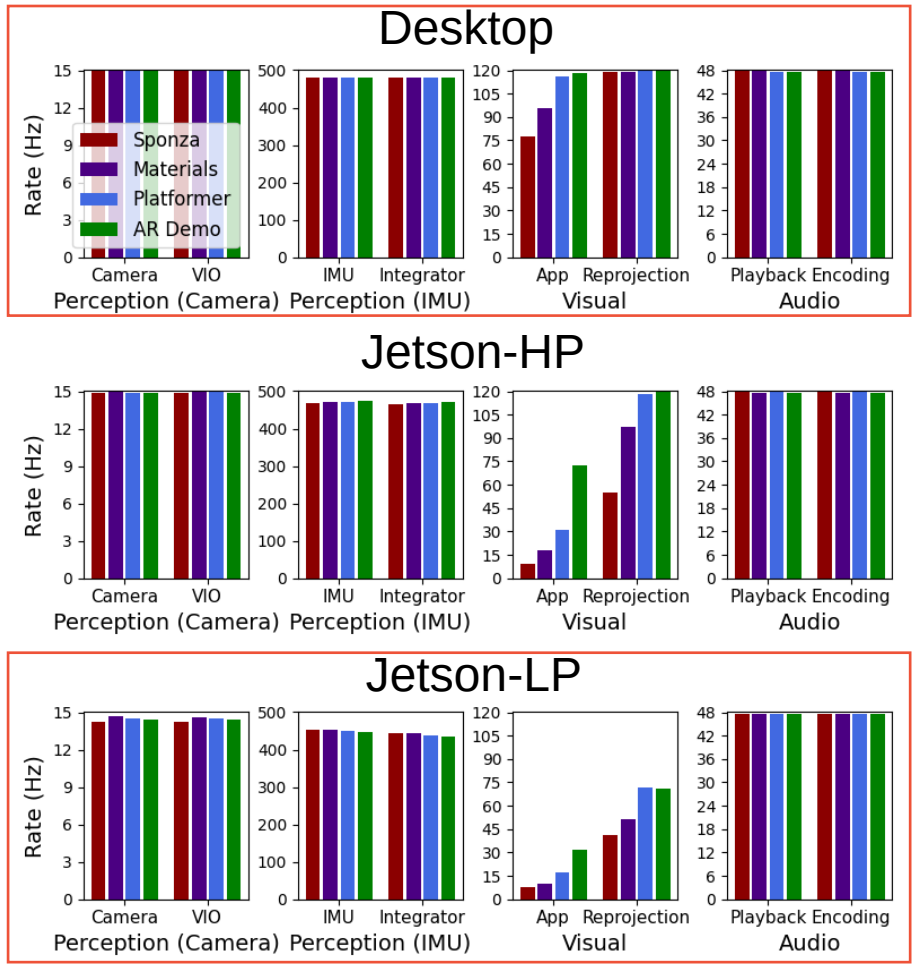
Component	Parameter	Range	Tuned	Deadline
Camera (VIO)	Frame rate Resolution Exposure	15 – 100 Hz VGA – 2K 0.2 – 20 ms	15 Hz VGA 1 ms	66.7 ms – –
IMU (Integrator)	Frame rate	≤ 800 Hz	500 Hz	2 ms
Display (Visual pipeline + Application)	Frame rate Resolution Field-of-view	30 – 144 Hz ≤ 2K ≤ 180°	120 Hz 2K 90°	8.33 ms – –
Audio (Encoding + Playback)	Frame rate Block size	48 – 96 Hz 256 – 1024	48 Hz 1024	20.8 ms –

- Platforms
  - High-end desktop machine
  - Embedded: NVIDIA Jetson-HP (high performance) and Jetson-LP (low power)
- Applications: Sponza, Materials, Platformer, AR Demo on Godot game engine

Frame Rate

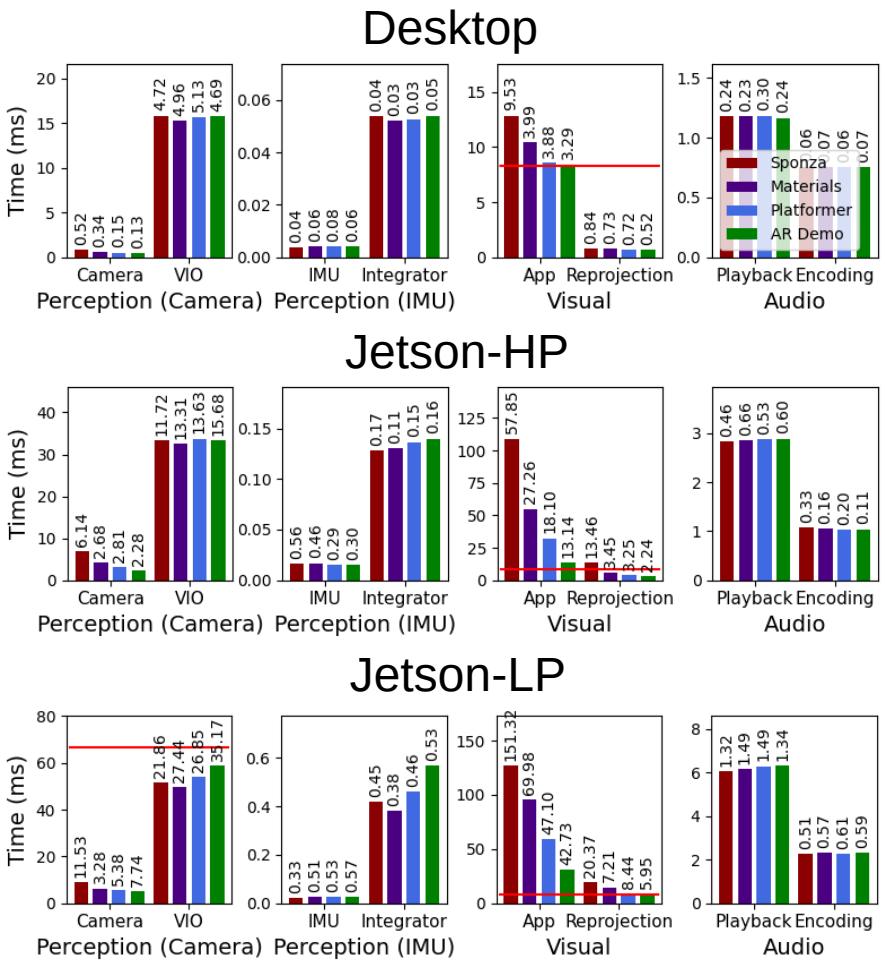

Desktop
Jetson-HP
Jetson-LP
Rate (Hz)
Sponza
Materials
Platformer
AR Demo
Camera
VIO
IMU
Integrator
App
Reprojection
Playback
Encoding
Perception (Camera)
Perception (IMU)
Visual
Audio

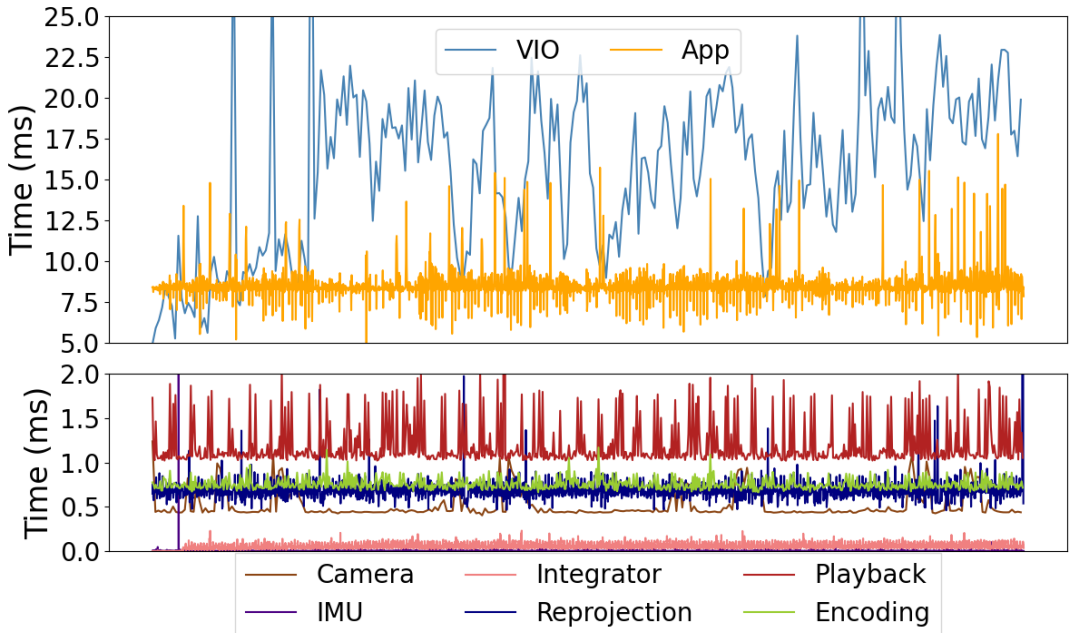
Frame Rate



- Desktop meets performance
  - But at what power cost?
- Jetson-LP can run only audio at target fps
- Gap will increase as displays and components scale

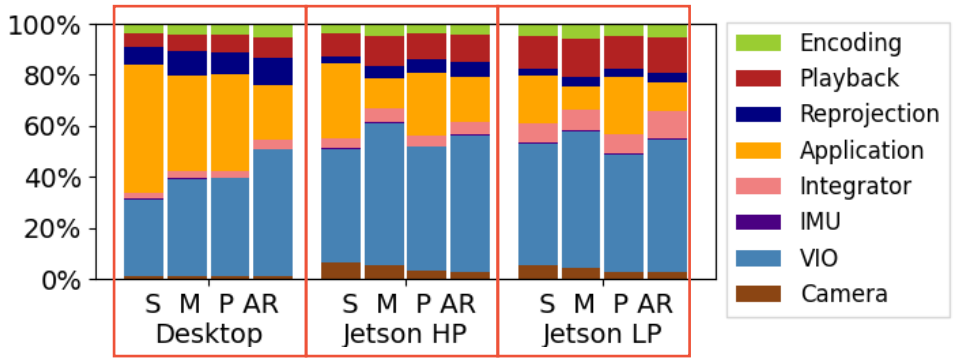
Time Per Frame





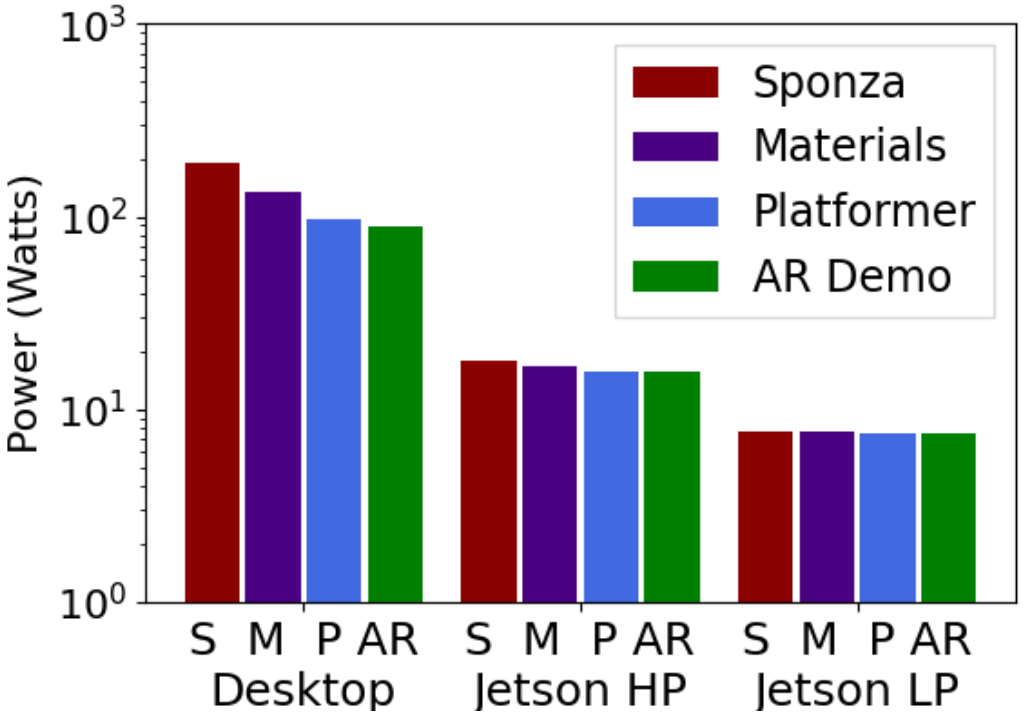
Input-dependence, scheduling, and resource contention lead to significant variability

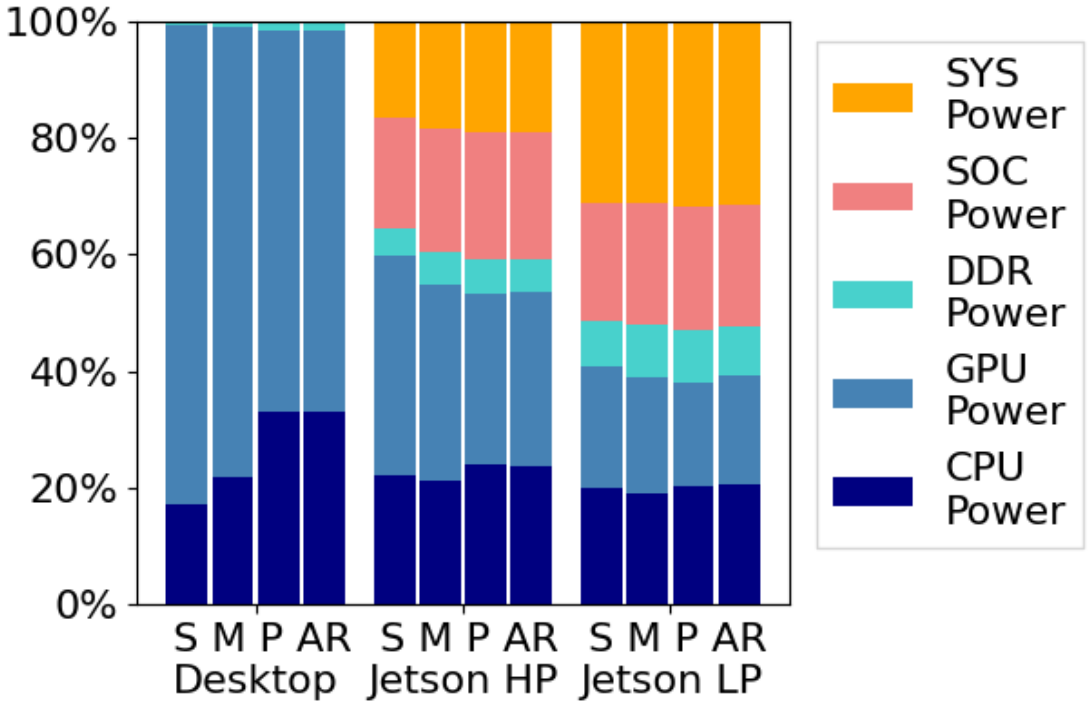
Distribution of Cycles



- Application and VIO dominate
- Reprojection and integrator take little time, but critical for QoE
- All components and metrics must be considered together

Power





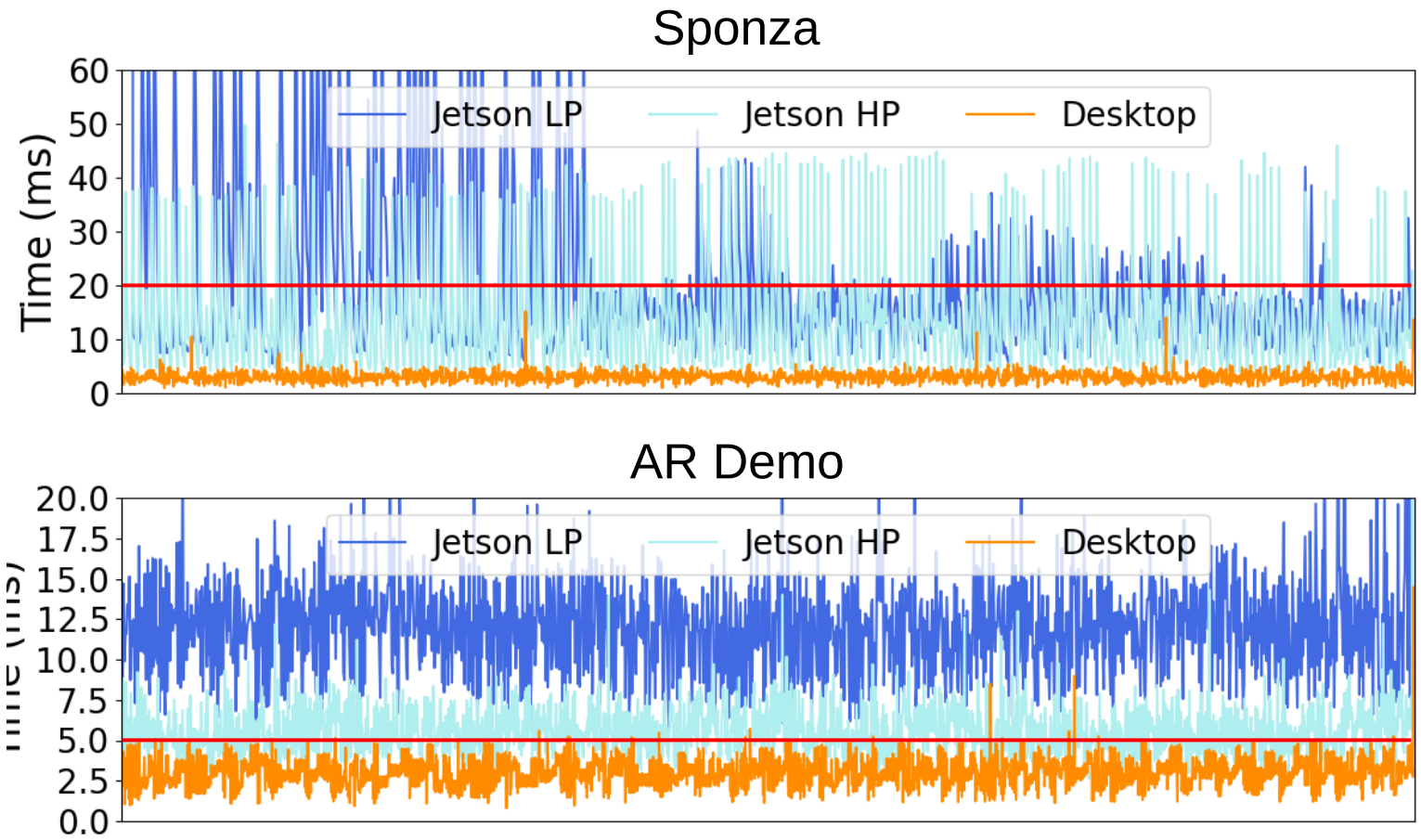
Must consider system-level components such as display and I/O

Motion-to-Photon Latency*

Application	Desktop	Jetson-HP	Jetson-LP
Sponza	3.1 ± 1.1	13.5 ± 10.7	19.3 ± 14.5
Materials	3.1 ± 1.0	7.7 ± 2.7	16.4 ± 4.9
Platformer	3.0 ± 0.9	6.0 ± 1.9	11.3 ± 4.7
AR Demo	3.0 ± 0.9	5.6 ± 1.4	12.0 ± 3.4

* w/o display latency

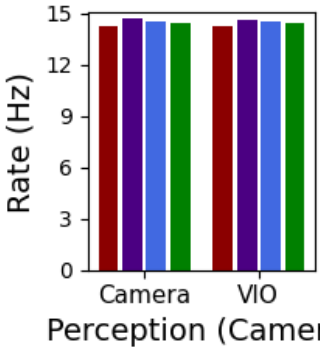
Motion-to-Photon Latency*



Unpleasant experience on Jetson

Image Quality

Platform	SSIM	1-FLIP	ATE/degree	ATE/meters
Desktop	0.83 ± 0.04	0.86 ± 0.05	8.6 ± 6.2	0.33 ± 0.15
Jetson-HP	0.80 ± 0.05	0.85 ± 0.05	18 ± 13	0.70 ± 0.33
Jetson-LP	0.68 ± 0.09	0.65 ± 0.17	138 ± 26	13 ± 10

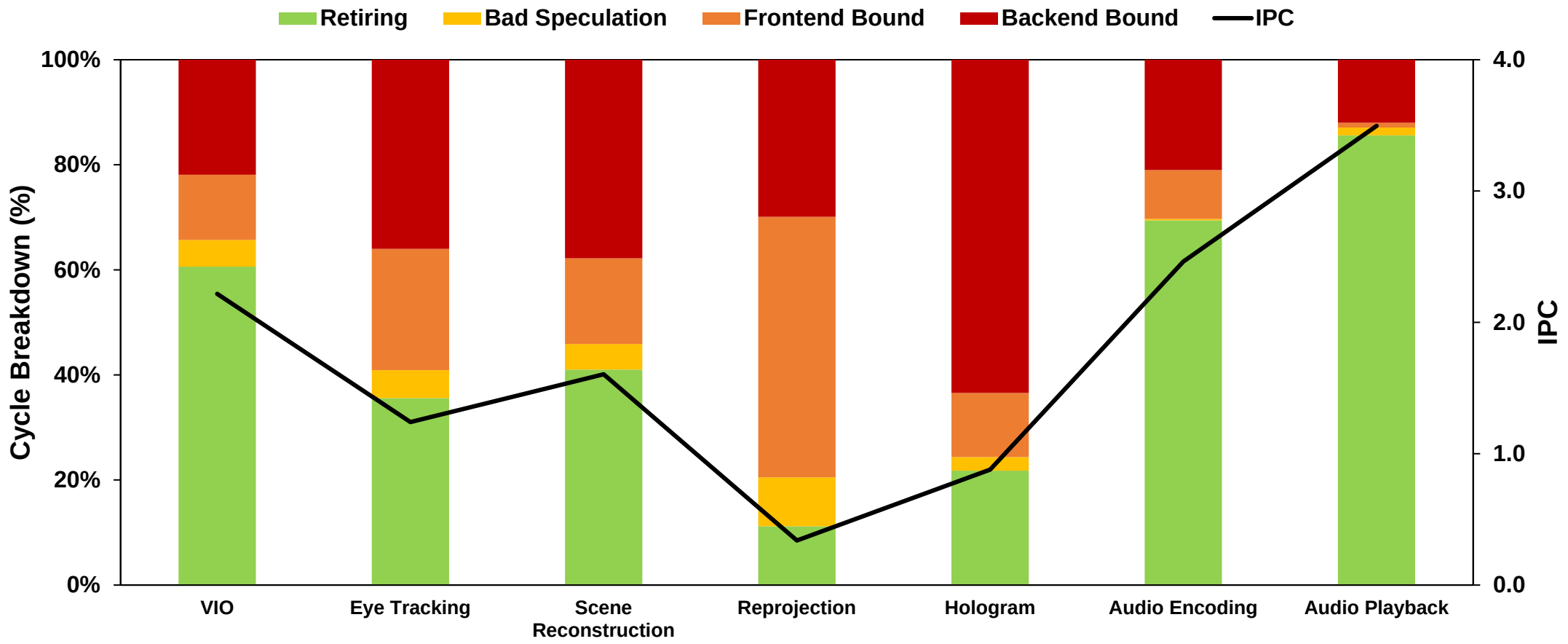


Must consider end-to-end QoE
Need better QoE metrics

Implications for System Designers

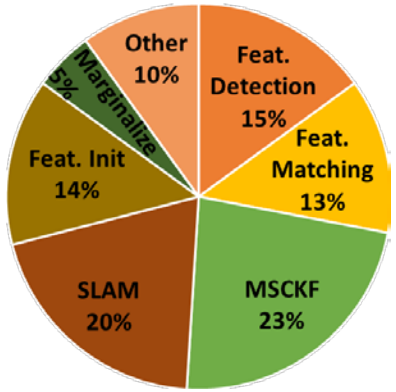
- Substantial performance, power, QoE gap
  - ⇒ Need to specialize hardware, software, *system*
- No application component dominates all metrics
  - ⇒ Must consider all application components in *system* together
- Power consumption goes beyond CPU, GPU, DDR
  - ⇒ Must consider *system*-level hardware components; e.g., display and I/O
- Significant variability
  - ⇒ Need to partition, allocate, and schedule *system* resources
- Per-component metrics do not capture QoE
  - ⇒ Must look at entire *system* to make QoE-driven tradeoffs

Microarchitectural Diversity

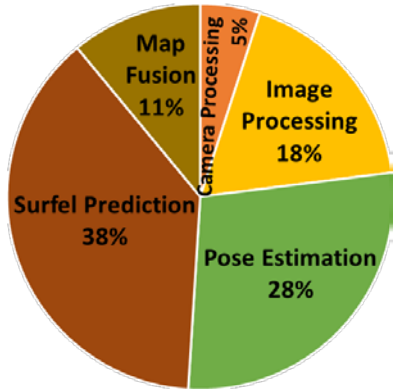


Wide range in IPC and hardware utilization

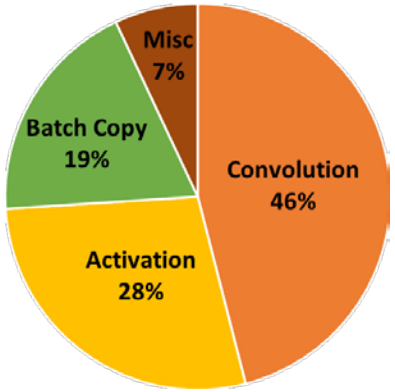
Task Diversity



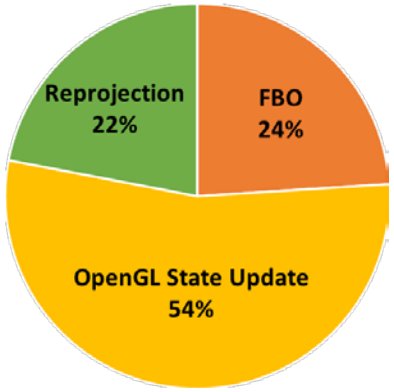
VIO



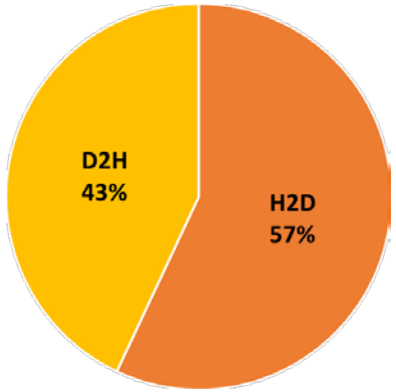
Scene Reconstruction



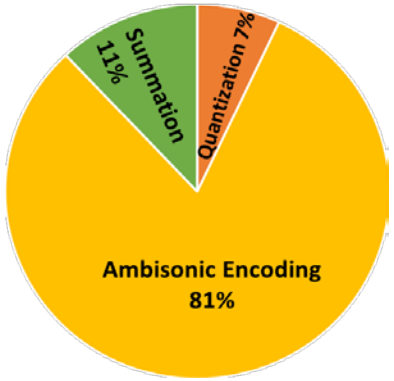
Eye Tracking



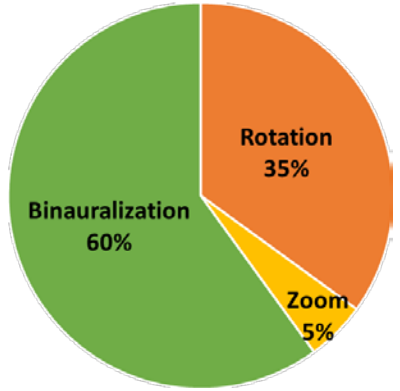
Reprojection



Hologram



Audio Encoding



Audio Playback

Variety (27!) of tasks and no task dominates

Implications for System Designers

- Need to specialize hardware, software, *system*
- Must consider all application components in *system* together
- Must consider *system*-level hardware components; e.g., display and I/O
- Need to partition, allocate, and schedule *system* resources
- Must look at entire *system* to make QoE-driven tradeoffs
- Abundance of tasks and no single task dominates
  ⇒ Need *automated* techniques to determine what to accelerate
- Impractical to build accelerator for every task
  ⇒ Must build *shared* hardware
- Diversity of compute and memory primitives
  ⇒ *Flexible* on-chip memory hierarchy
  ⇒*Flexible* accelerator communication interface
- Algorithms in flux
  ⇒ Must design *programmable* hardware
- Different algorithms have different QoE vs. resource usage profiles
  ⇒End-to-end QoE driven *approximate computing*

ILLIXR =
Rich playground for systems research

Ongoing Work

Research with ILLIXR (with many collaborators)

- Accelerators and memory system for XR
- Compiling to heterogeneous hardware
- QoE-driven scheduling
- Computation offload, content streaming, multiparty XR
- On-sensor computing
- QoE metrics

ILLIXR testbed

- New components: spatial reprojection, hand tracking, …
- North Star head set
- Off-loading, streaming, multiparty XR
- Broaden hardware/software platforms supported
- Create and curate data sets and applications
- Incorporate research results

ILLIXR consortium

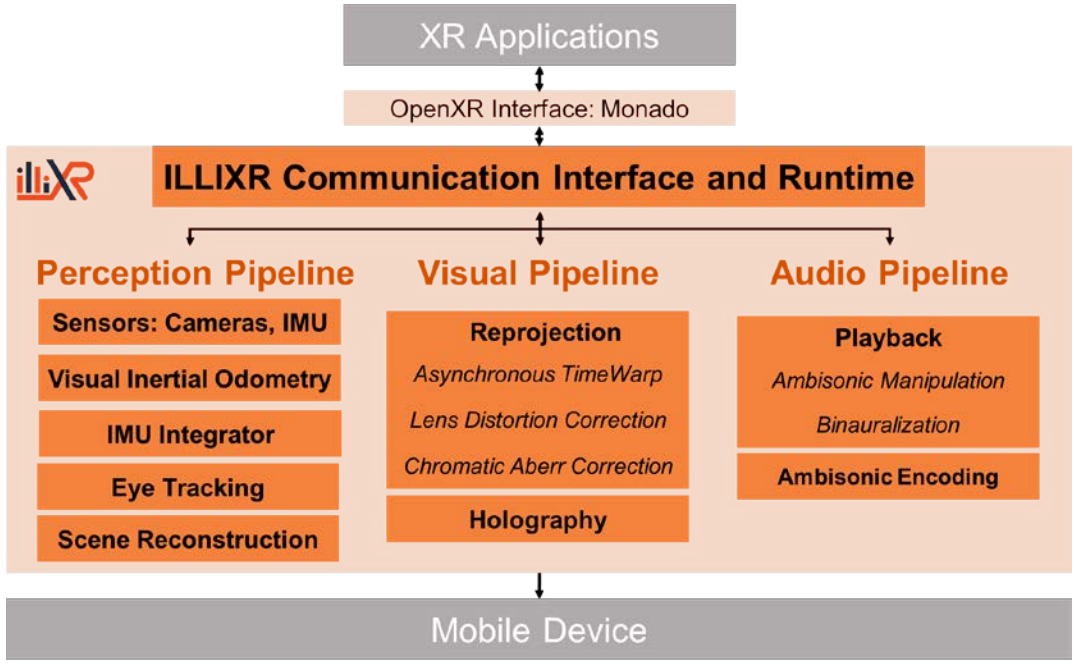
- Working groups on various topics
- Reference open source testbed and benchmarking methodology
- XR systems research and development community

ILLIXR: Illinois Extended Reality Testbed

Rich playground for XR systems research

Democratize XR systems research, development, and benchmarking

Join us: illixr@cs.illinois.edu, illixr.org



Team ILLIXR

- Rishi Desai
- Samuel Grayson
- Muhammad Huzaifa
- Xutao Jiang
- Ying Jing
- Jae Lee
- Fang Lu
- Yihan Pang
- Joseph Ravichandran
- Giordano Salvador
- Finn Sinclair
- Boyuan Tian
- Lauren Wagner
- Henghzi Yuan
- Jeffrey Zhang

External Consultations

- Wei Cui
- Aleksandra Faust
- Liang Gao
- Matt Horsnell
- Amit Jindal
- Steve LaValle
- Steve Lovegrove
- Andrew Maimone
- Vegard Oye
- Martin Persson
- Archontis Politis
- Eric Shaffer
- Paris Smaragdis
- Chris Widdowson