Enabling the Immersive Era of Computing

Sarita Adve

University of Illinois at Urbana-Champaign

sadve@illinois.edu

illixr.org

w/ many collaborators acknowledged on slides

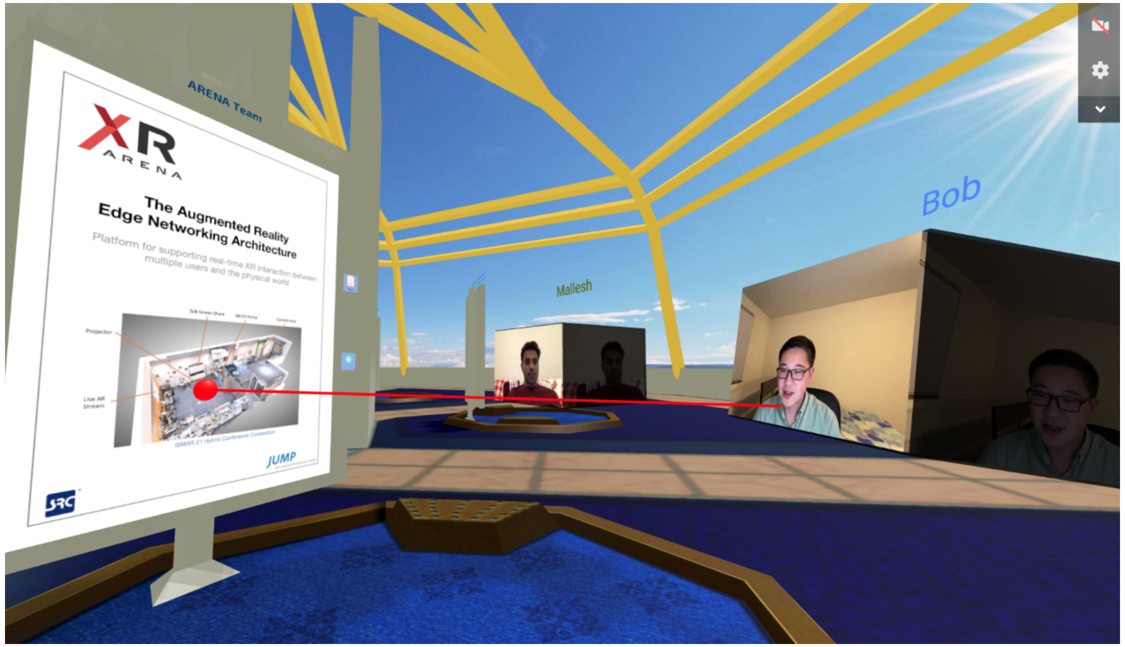
This work is supported in part by the Applications Driving Architecture (ADA) center (JUMP center co-sponsored by SRC & DARPA), the DARPA DSSOC program, and the National Science Foundation


Spoons
Elements of Computer Music


Joseph
Finn Sinclair
Jeffrey Zhang
Muhammad Huzaifa
Sarita Adve
Hengzhi Yuan
Rishi Desai
Fang Lu
Samuel Grayson
Ying Jing
Jae Lee
Jumbo Jiang

Meta avatars on Unity

ARENA [Rowe, CMU]



Immersive Computing =

Seamless integration of the physical and the virtual

Real time, mobile, comfortable all day

Virtual, augmented, mixed reality (VR, AR, MR) ⟶ Extended reality (XR)

Metaverse, digital twins, spatial computing, …

Will transform most human activities

speech therapists, special educa
Valerie Plesch for The New York Times

illiXR

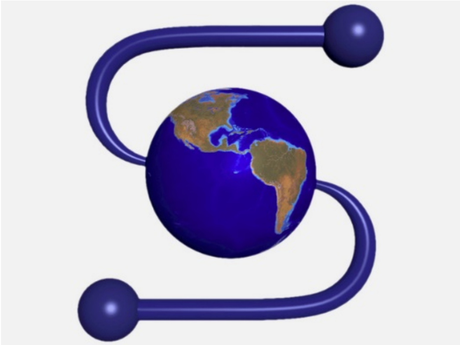
New Era of Computing

Each era was transformative

Immersive

Mobile

Web, Cloud



Personal

Mainframe

Each era changed how we design, program, and use computers

Immersive Computing =

Seamless integration of the physical and the virtual

Real time, mobile, comfortable all day

Hardware, software, applications ecosystem

Sensors, displays, headsets, wearables, edge and cloud backends, networking

A broad systems problem

Immersive Computing for Architects

Orders of magnitude gap

in power, performance, quality-of-experience

between current and desired systems

Approximate	Current	Desired
Res (Mpixels)	**7**	**200**
Power (W)	**~7**	**0.1**
Weight (g)	**500**	**10**
…	**…**	**…**

Huzaifa et al., Micro Top Picks'22

XR Systems: Challenges

Orders of magnitude gap

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

Approximate	Current	Desired
Res (Mpixels)	**7**	**200**
Power (W)	**~7**	**0.1**
Weight (g)	**500**	**10**
…	**…**	**…**

Diverse expertise

Graphics, vision, audio, video, optics, haptics, …

Cross-layer system co-design

Hardware, compiler, OS, algorithm. Device, edge, cloud

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?

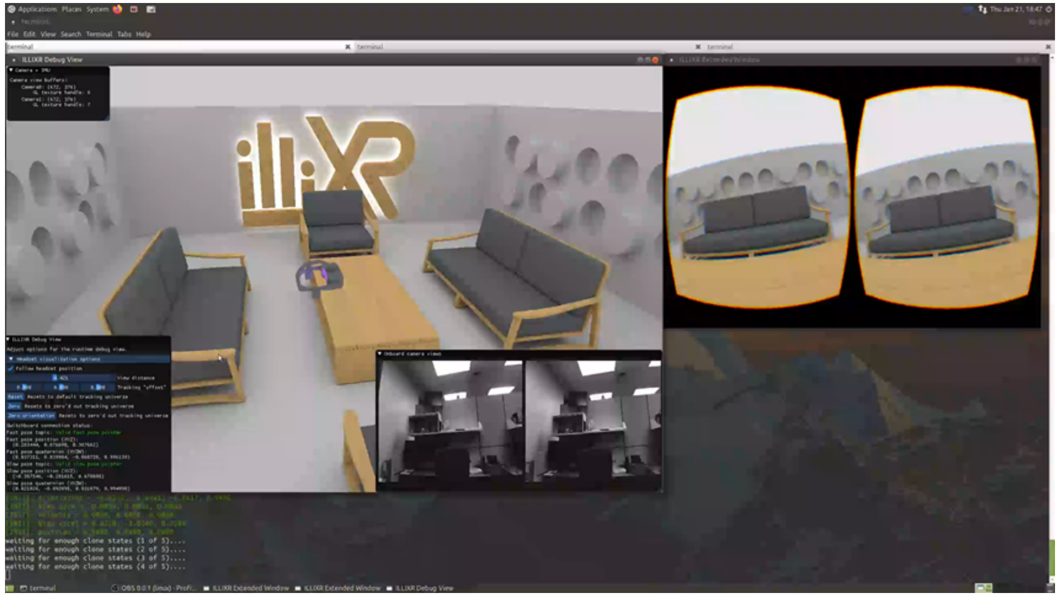
ILLIXR: Illinois Extended Reality Testbed

ILLIXR: Open-source full system XR testbed

State-of-the-art XR components w/ modular runtime

OpenXR compatible

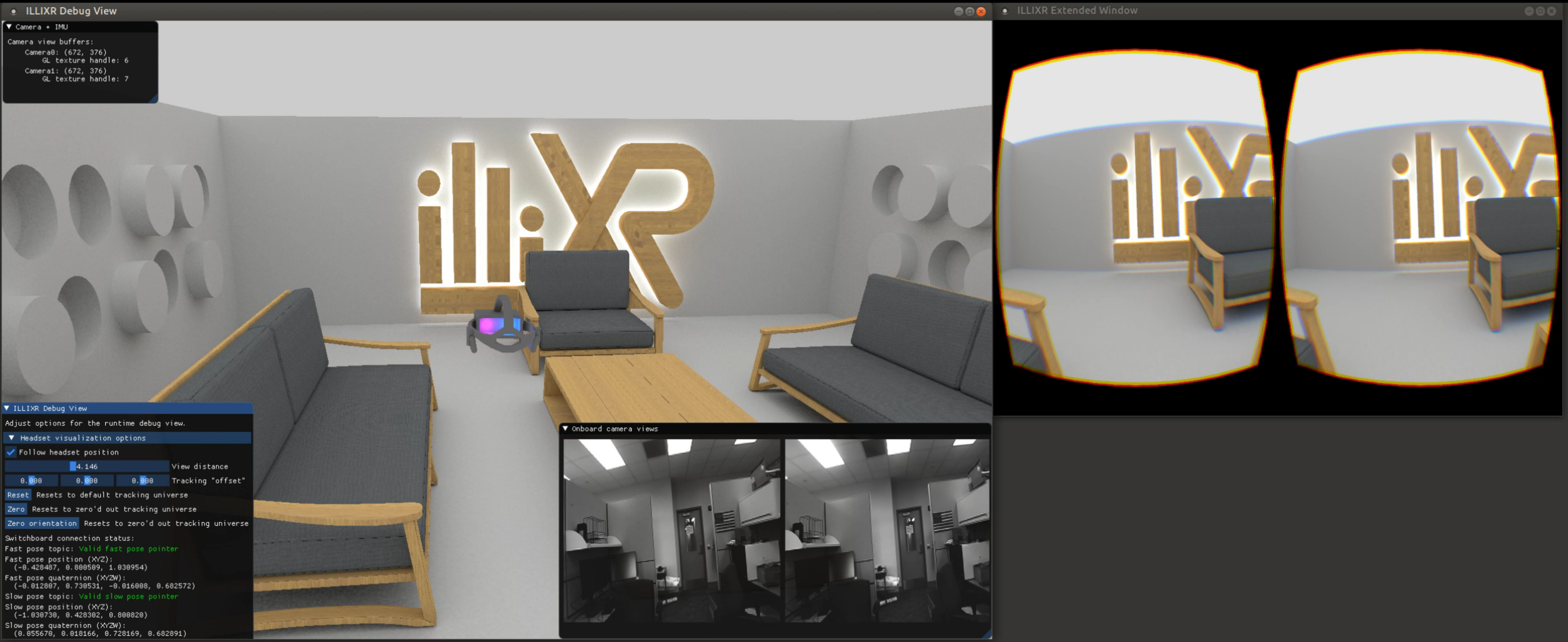
Extensive characterization and use for research

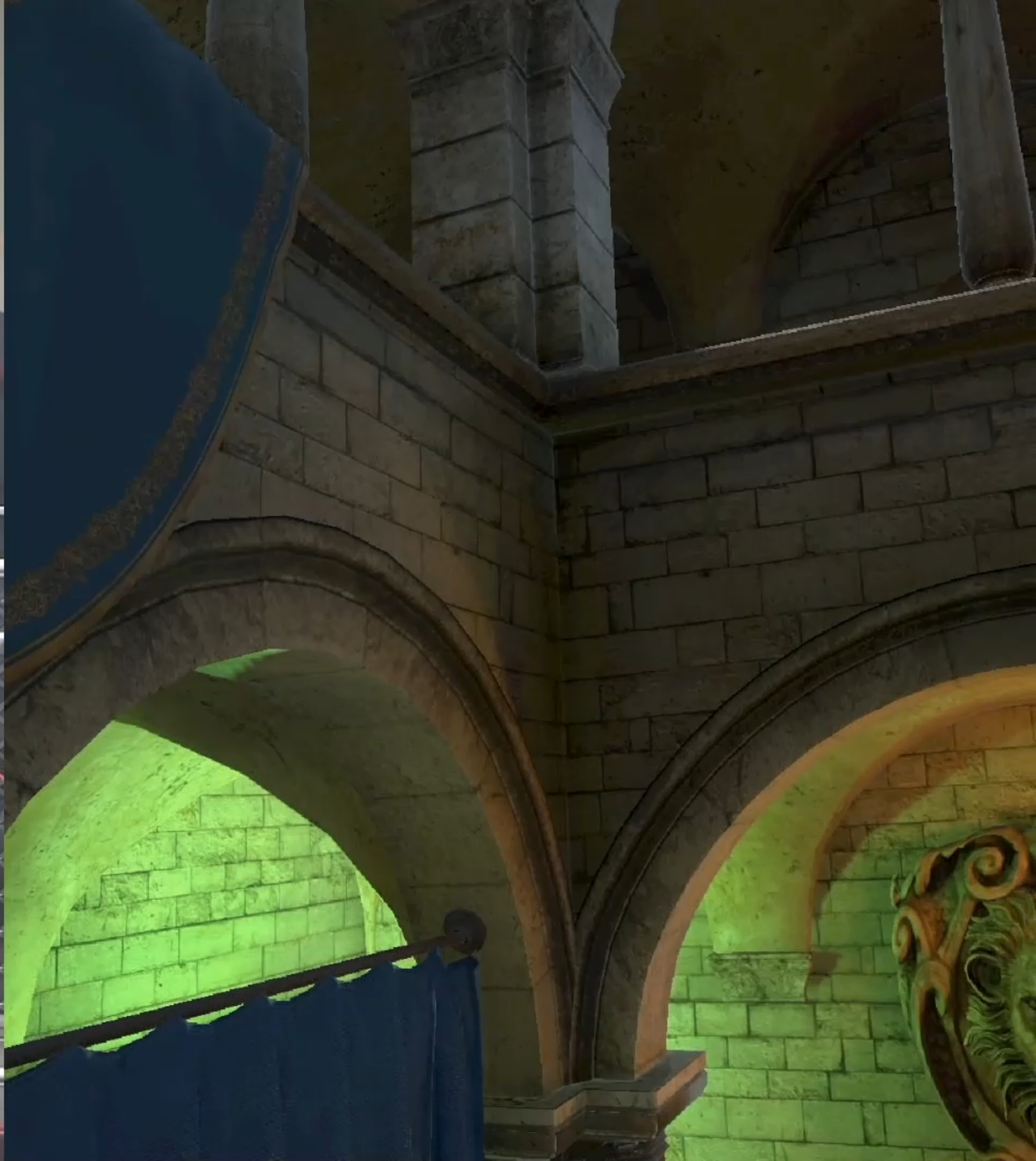


illixr.org

Huzaifa et al., IISWC'21 best paper,

IEEE Micro Top Picks'22


ILLIXR Debug View
Camera + IMU
Camera view buffers:
Camera0: (672, 376)
GL texture handle: 6
Camera1: (672, 376)
GL texture handle: 7
ILLIXR Debug View
Adjust options for the runtime debug view.
Headset visualization options
Follow headset position
4.146
View distance
0.000
0.000
0.000
Tracking "offset"
Reset
Resets to default tracking universe
Zero
Resets to zero'd out tracking universe
Zero orientation
Resets to zero'd out tracking universe
Switchboard connection status:
Fast pose topic: Valid fast pose pointer
Fast pose position (XYZ):
(-0.428487, 0.800509, 1.030954)
Fast pose quaternion (XYZW):
(-0.012807, 0.730531, -0.016008, 0.682572)
Slow pose topic: Valid slow pose pointer
Slow pose position (XYZ):
(-1.030730, 0.428302, 0.800820)
Slow pose quaternion (XYZW):
(0.055670, 0.018166, 0.728169, 0.682891)
Onboard camera views
ILLIXR Extended Window

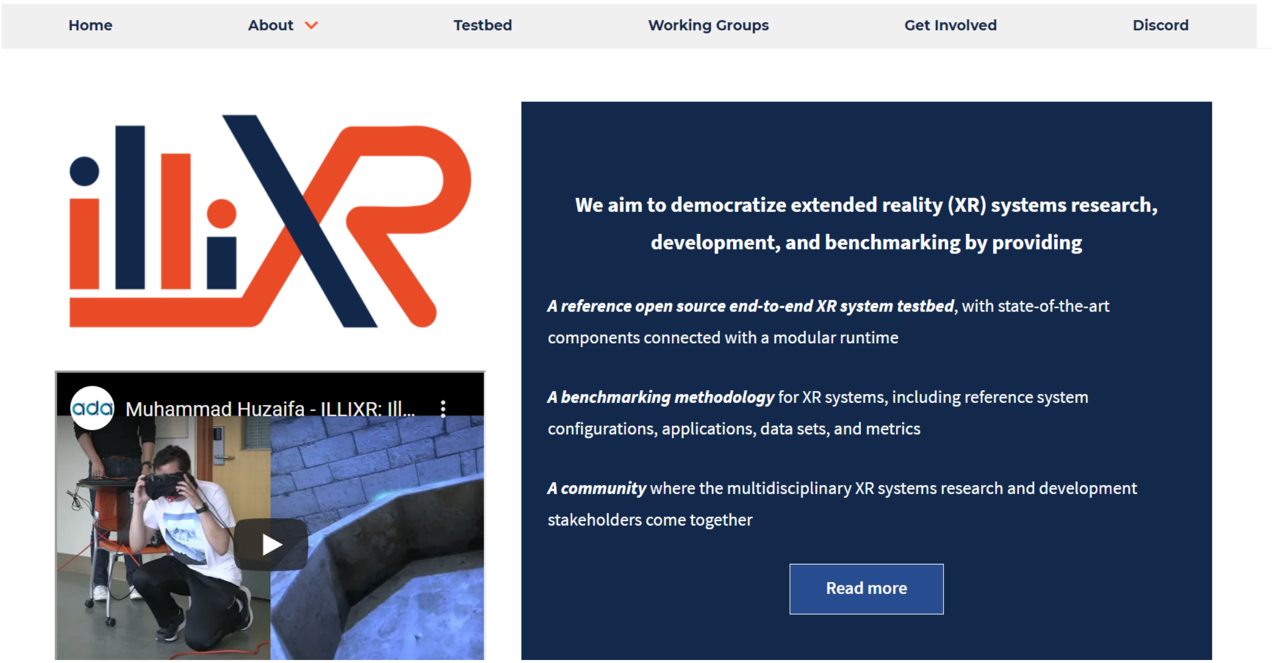

SERVER LOG
FIRST PERSON VIEW
ROUTER
DEBUG VIEW
SERVER
BACKPACK PC

ILLIXR Consortium

ILLIXR Consortium w/ industry + academic partners

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

illixr.org



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

Now funded by NSF CISE community research infrastructure progam

Join us: illixr@cs.illinois.edu, illixr.org, discord, weekly meetings

ILLIXR Deep Dive

Team ILLIXR

ILLIXR students and developers

- Madhuparna Bhowmik
- Henry Che
- Rishi Desai
- Steven Gao
- Samuel Grayson
- Qinjun Jiang
- Muhammad Huzaifa
- Xutao Jiang
- Ying Jing
- Jae Lee
- Jeffrey Liu
- Fang Lu
- Yihan Pang
- Joseph Ravichandran
- Giordano Salvador
- Bill Sherman
- Finn Sinclair
- Rahul Singh
- Boyuan Tian
- Lauren Wagner
- Henghzi Yuan
- Jeffrey Zhang

Consultations

- Ameen Akeel
- Wei Cui
- Aleksandra Faust
- Liang Gao
- Rod Hooker
- Matt Horsnell
- Amit Jindal
- Steve LaValle
- Steve Lovegrove
- David Luebke
- Andrew Maimone
- Vegard Oye
- Maurizio Paganini
- Martin Persson
- Archontis Politis
- Eric Shaffer
- Paris Smaragdis
- Chris Widdowson

Founding consortium members: Arm, Meta Reality Labs, Micron, NVIDIA

Founding sponsor: ADA research center, a DARPA/SRC JUMP center

Team ILLIXR

ILLIXR students and developers

- Madhuparna Bhowmik
- Henry Che
- Rishi Desai
- Steven Gao
- Samuel Grayson
- Qinjun Jiang
- **Muhammad Huzaifa**
- Xutao Jiang
- Ying Jing
- Jae Lee
- Jeffrey Liu
- Fang Lu
- Yihan Pang
- Joseph Ravichandran
- Giordano Salvador
- Bill Sherman
- Finn Sinclair
- Rahul Singh
- Boyuan Tian
- Lauren Wagner
- Henghzi Yuan
- Jeffrey Zhang

one

anini

n

tis

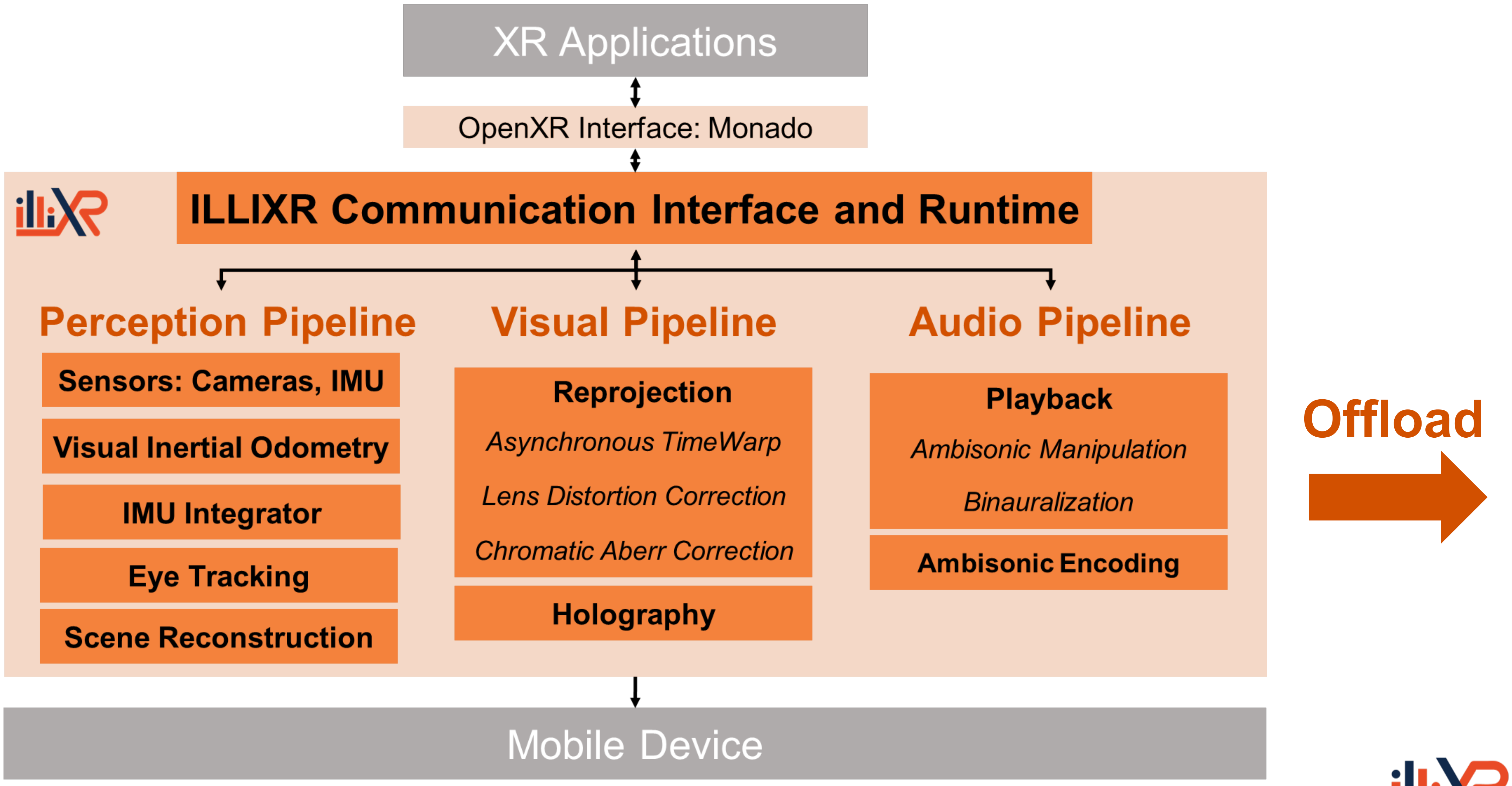
dis

son

Founding consortium members: Arm, Meta Reality Labs, Micron, NVIDIA

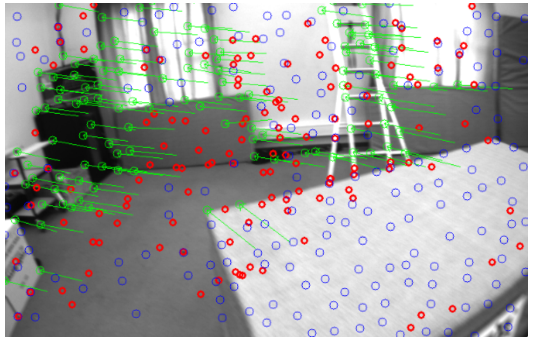
Founding sponsor: ADA research center, a DARPA/SRC JUMP center

ILLIXR Overview

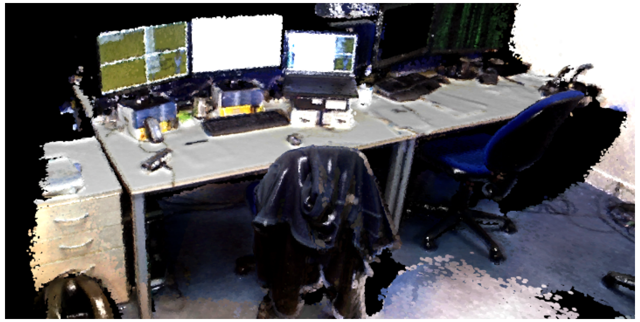


Perception Pipeline

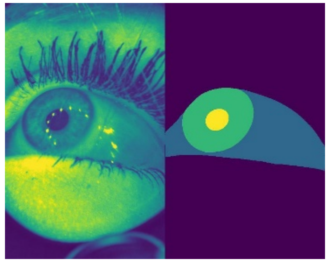
- Sensors: Camera, Inertial Measurement Unit (IMU)
- Visual Intertial Odometry (VIO)
  - Provides position and head orientation (pose)



- IMU Integrator
  - Provides high frequency pose estimates
- Pose Predictor
  - Extrapolates pose to future timestamp

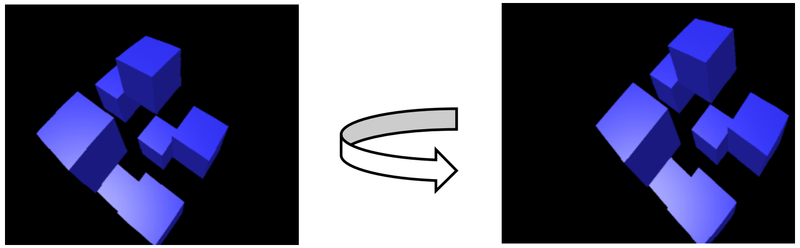


- Scene Reconstruction
  - Uses RGB-Depth camera to build dense 3D map of world
- Eye Tracking

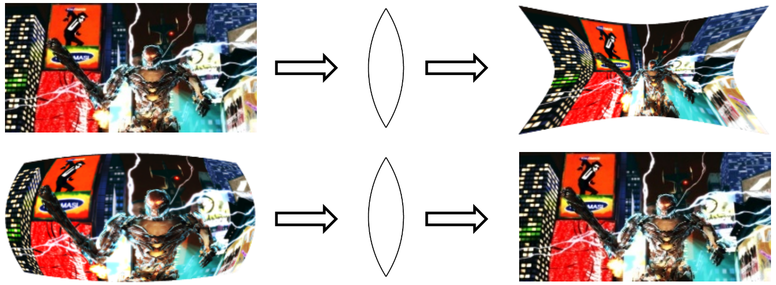


Visual Pipeline

- 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

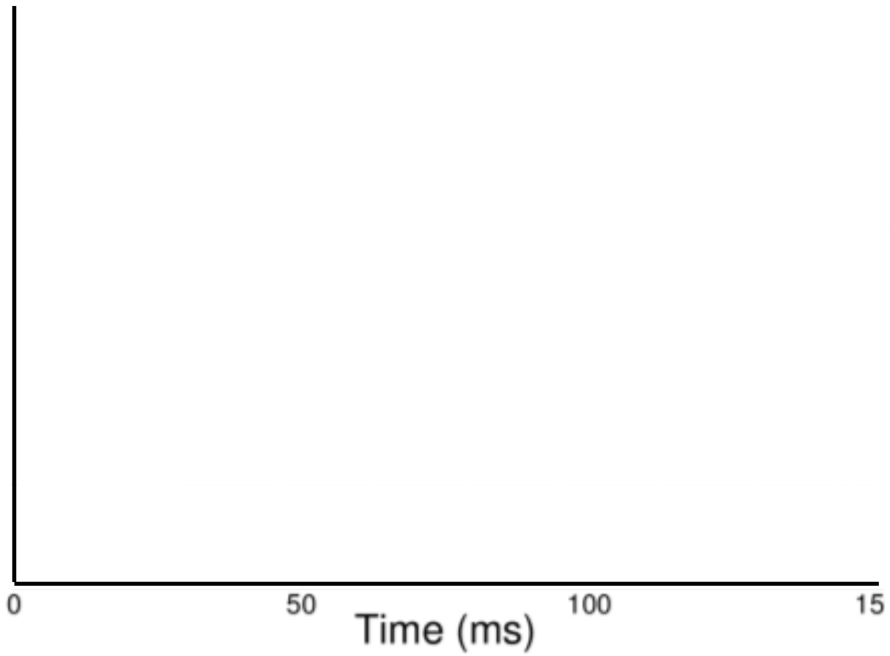
Audio Pipeline

- 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

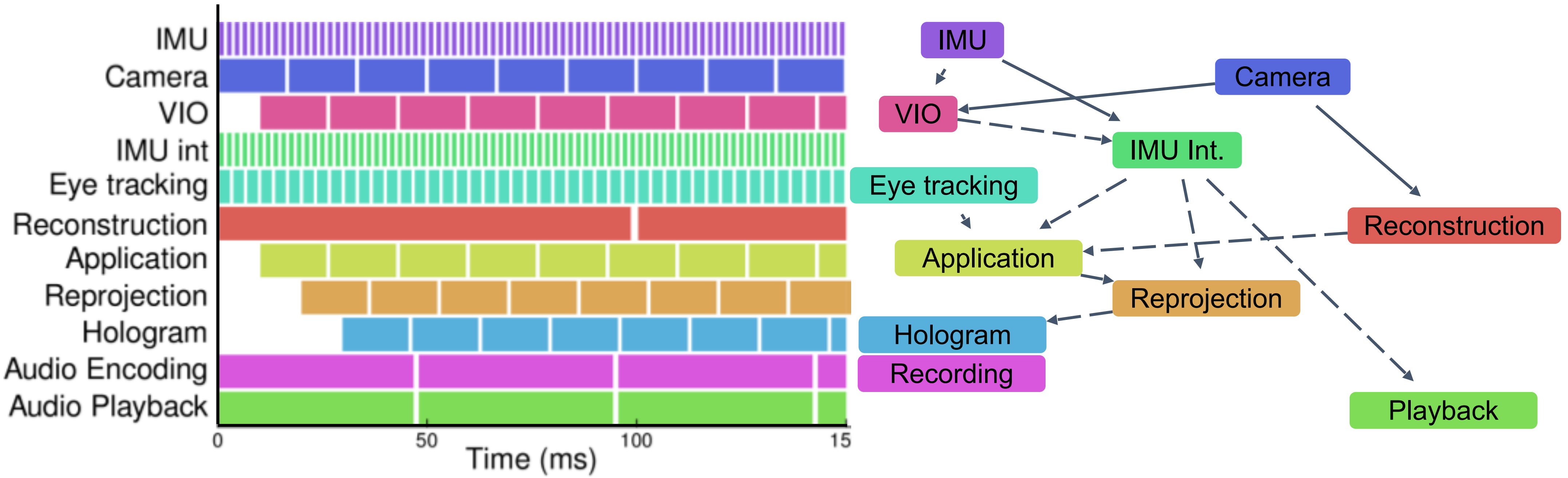
BUT XR is not just a collection of components

It is a SYSTEM

XR System Dataflow

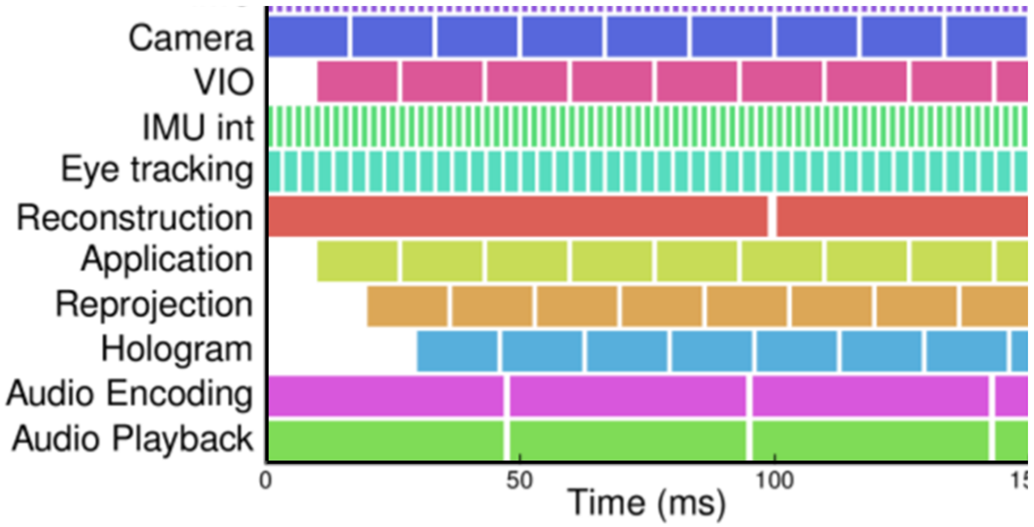


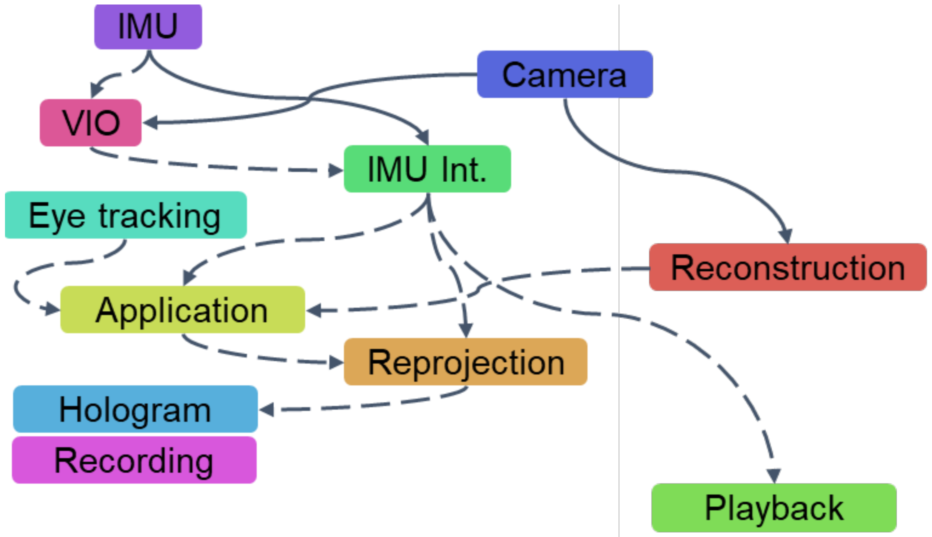
XR System Dataflow



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

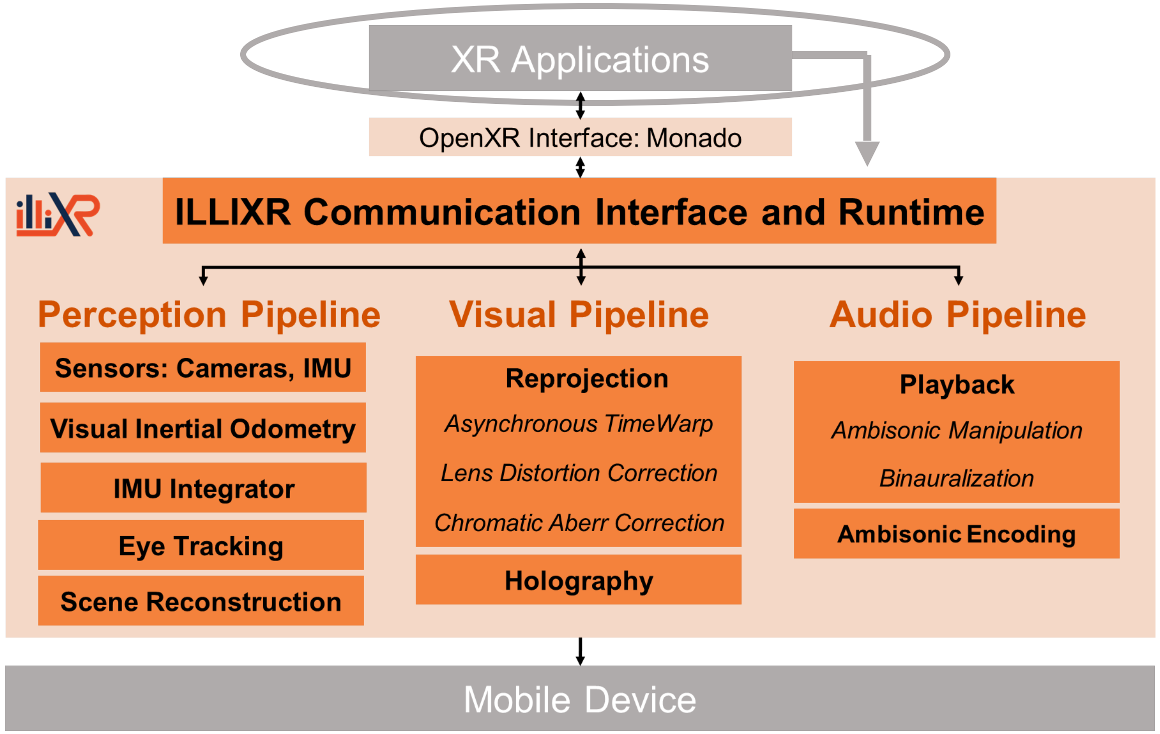
Component specifies event streams to publish, subscribe

Synchronous or asynchronous consumers

Copy-free, shared memory implementation

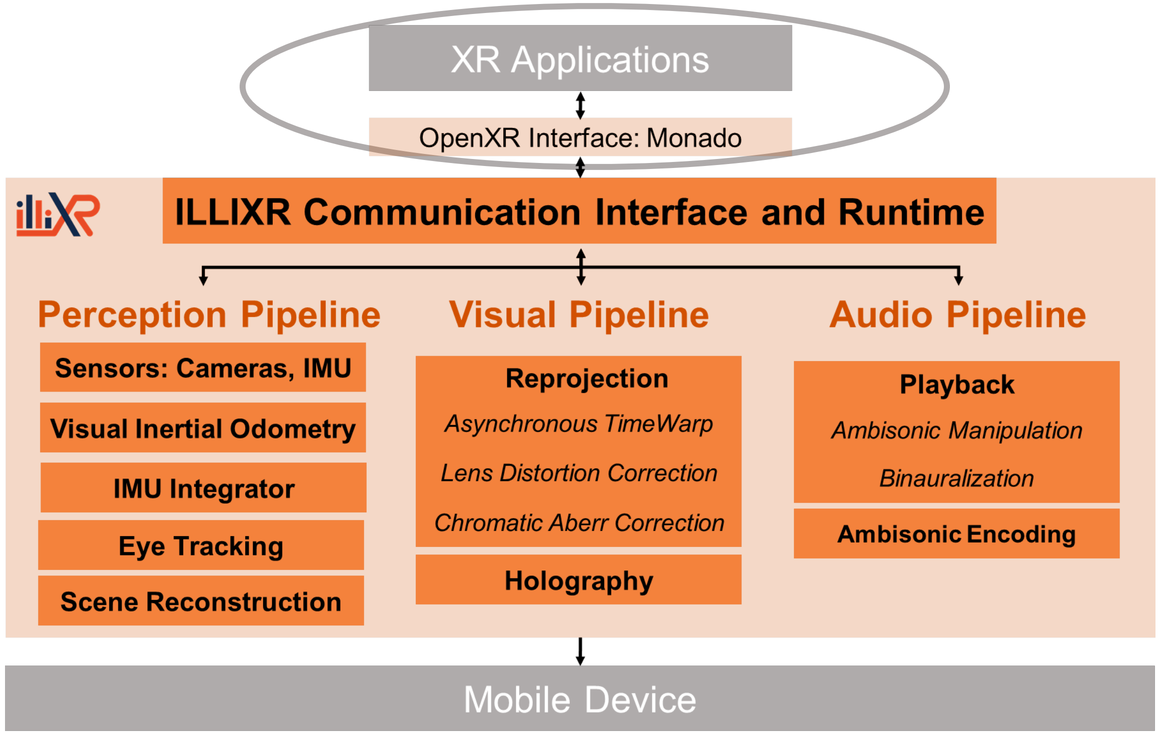
End-to-end system balances flexibility with efficiency

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
- Soon: Unity, Unreal development platforms

End-to-End Quality Metrics

- Motion-to-photon latency
  - Time from head motion to display (currently w/o display latency)
  - Target: < 20ms for VR, < 5ms for AR/MR

- Image quality: SSIM and FLIP

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

ILLIXR Components Today

	Component	Algorithm	Implementation
Perception Pipeline	Camera Camera	ZED SDK Intel RealSense SDK	C++ C++
	IMU IMU	ZED SDK Intel RealSense SDK	C++ C++
	VIO VIO	OpenVINS Kimera-VIO	C++ C++
	IMU Integrator IMU Integrator	RK4 GTSAM	C++ C++
	Eye Tracking	RITnet	Python, CUDA
	Scene Reconstruction Scene Reconstruction	ElasticFusion KinectFusion	C++, CUDA, GLSL 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++

ILLIXR Findings

Evaluation Methodology

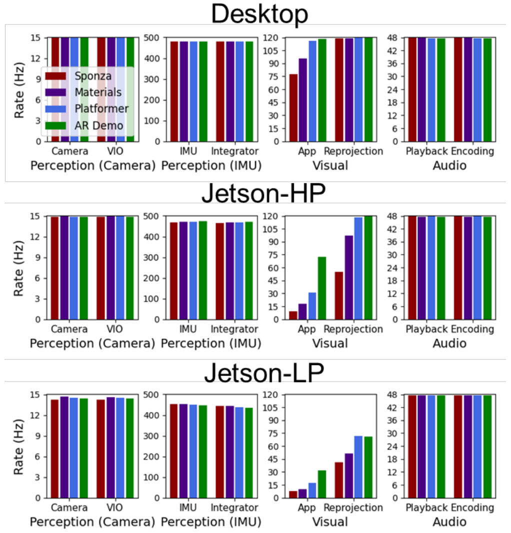
Component	Parameter	Range	Tuned	Deadline
Camera (VIO)	Frame rate Resolution Exposure	15 – 100 Hz VGA – 2K 0.2 – 20 ms	15 Hz VGA 1 ms	66.7 ms – –
IMU (Integrator)	Frame rate	≤ 800 Hz	500 Hz	2 ms
Display (Visual pipeline + Application)	Frame rate Resolution Field-of-view	30 – 144 Hz ≤ 2K ≤ 180°	120 Hz 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

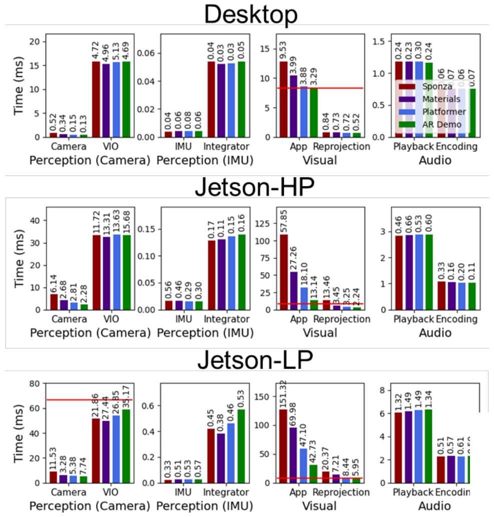
High ⟷ Graphics intensity ⟷ Low

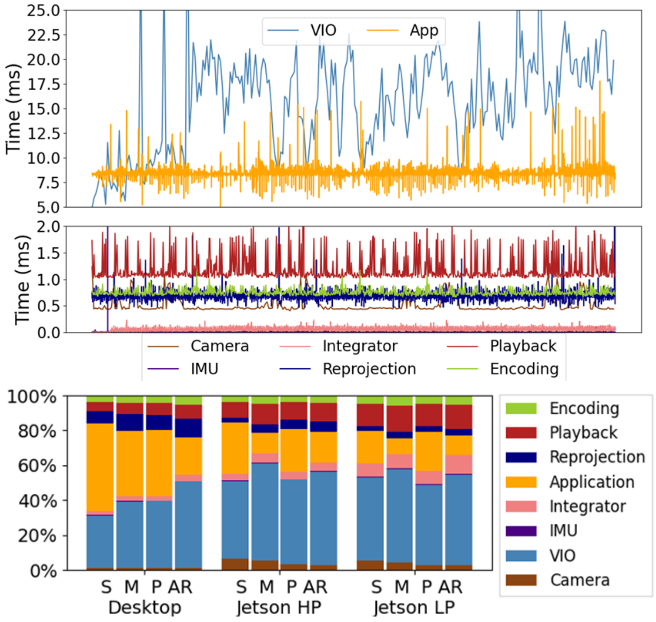
Results Summary

Frame Rate



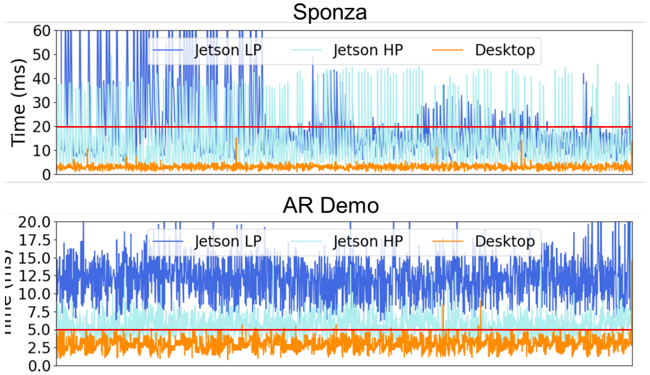
Execution Time & Distribution



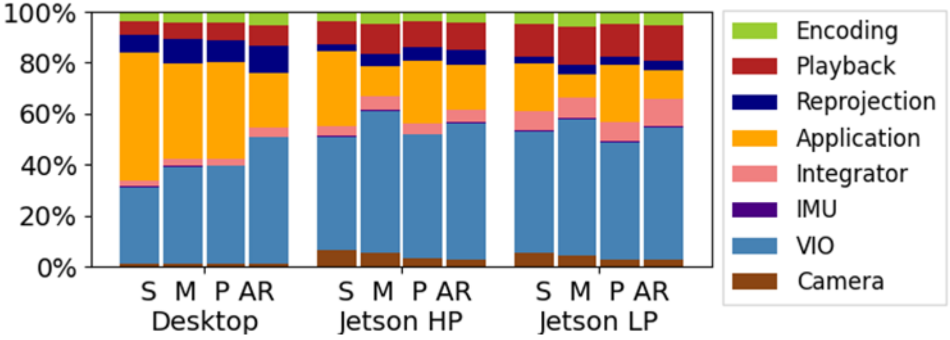


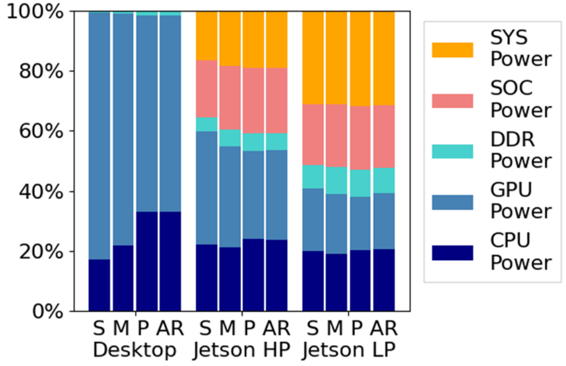
Quality of Experience

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



Power

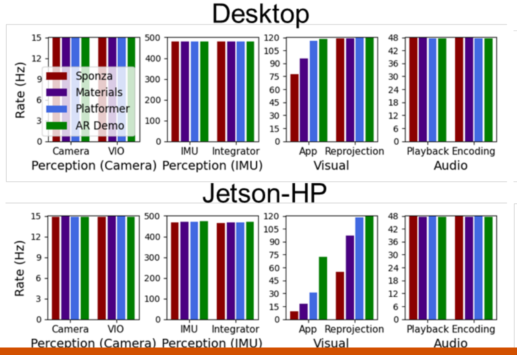




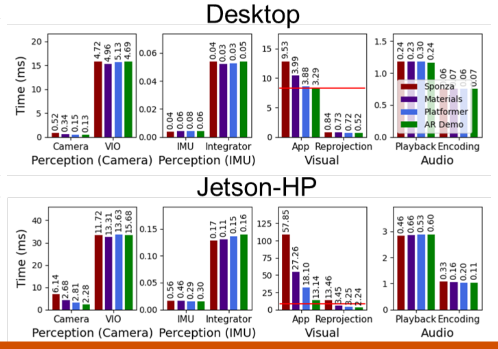
Platform	SSIM	1-FLIP
Desktop	0.83 ± 0.04	0.86 ± 0.05
Jetson-HP	0.80 ± 0.05	0.85 ± 0.05
Jetson-LP	0.68 ± 0.09	0.65 ± 0.17

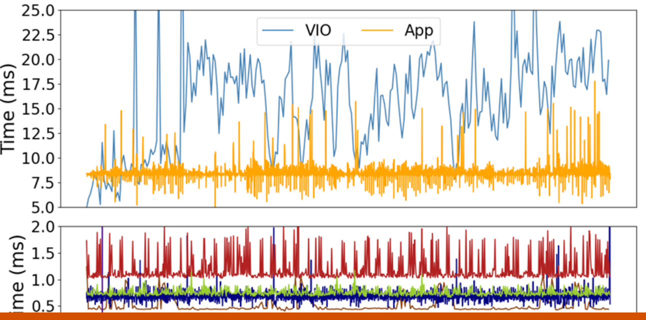
Results Summary

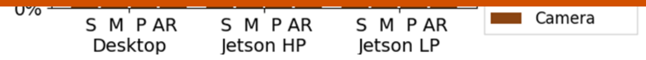
Frame Rate



Execution Time & Distribution



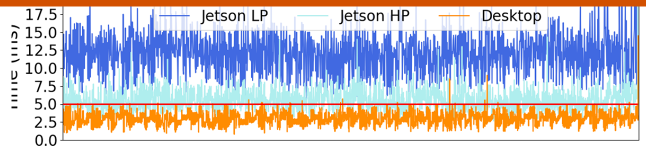




Quality of Experience

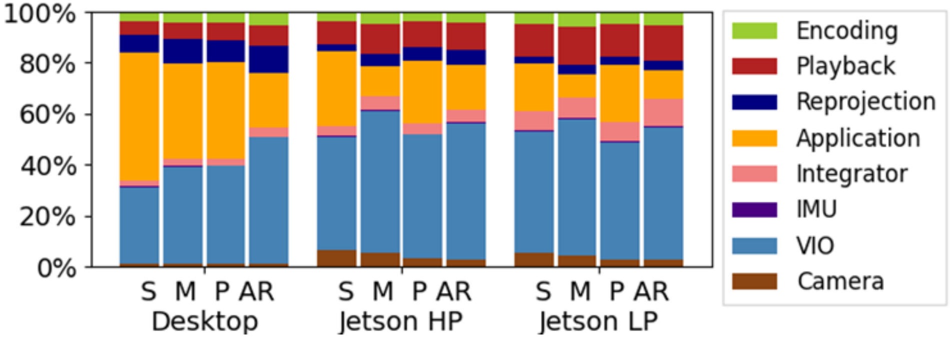
Application	Desktop	Jetson-HP	Jetson-LP
Sponza	3.1 ± 1.1	13.5 ± 10.7	19.3 ± 14.5
Materials	3.1 ± 1.0	7.7 ± 2.7	16.4 ± 4.9
Platformer	3.0 ± 0.9	6.0 ± 1.9	11.3 ± 4.7
AR Demo	3.0 ± 0.9	5.6 ± 1.4	12.0 ± 3.4

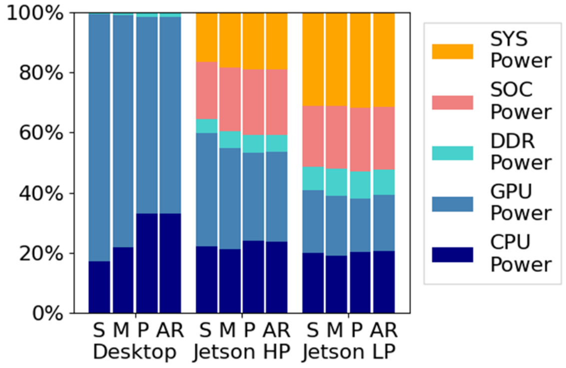
First published performance/power/QoE results for end-to-end XR system



Platform	SSIM	1-FLIP
Desktop	0.83 ± 0.04	0.86 ± 0.05
Jetson-HP	0.80 ± 0.05	0.85 ± 0.05
Jetson-LP	0.68 ± 0.09	0.65 ± 0.17

Power





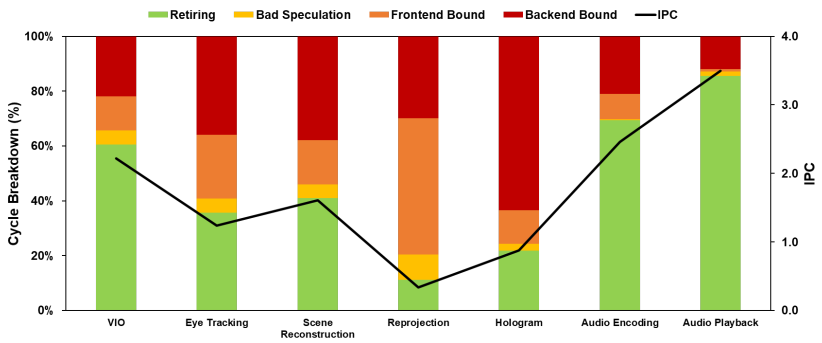
Results Summary and Implications for System Designers

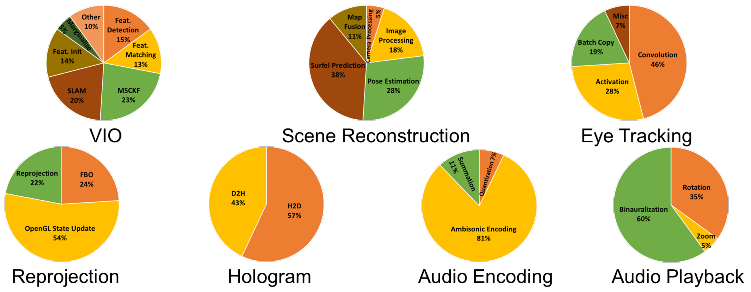
- Substantial performance, power, QoE gap
  - ⇒ Need to specialize hardware, software, *system*
- No application component dominates all metrics
  - ⇒ Must consider all application components in *system* together
- Power consumption goes beyond CPU, GPU, DDR
  - ⇒ Must consider *system*-level hardware 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

Results Summary and 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*

Standalone Components





Task	Time	Computation	Memory Pattern
Feature detection Detects new features in the new camera images	15%	Integer stencils per each pyramid level	Locally dense stencil; globally mixed dense and sparse
Feature matching Matches features across images	13%	Integer stencils; GEMM; linear algebra	Locally dense stencil; globally mixed dense and sparse; mixed dense and random feature map accesses
Filter Estimates 6DOF pose using camera and IMU measurements	62%	Gauss-Newton refinement; QR decomposition; GEMM; linear algebra	Mixed dense and sparse feature map and filter matrix accesses
Other Miscellaneous tasks	10%	Gaussian filter; histogram	Globally dense stencil

Task	Time	Computation	Memory Pattern
FBO FBO state management	24%	Framebuffer bind and clear	Driver calls; CPU-GPU communication
OpenGL State Update Sets up OpenGL state	54%	OpenGL state updates; one drawcall per eye	Driver calls; CPU-GPU communication
Reprojection Applies reprojection transformation to image	22%	6 matrix-vector MULs/vertex	Accesses uniform, vertex, and fragment buffers; 3 texture accesses/fragment

Task	Time	Computation	Memory Pattern
Hologram-to-depth Propagates pixel phase to depth plane	57%	Transcendentals; FMADDs; TB-wide tree reduction	Dense row-major; spatial locality in pixel data; temporal locality in depth data; reduction in scratchpad
Sum Sums phase differences from hologram-to-depth	< 0.1%	Tree reduction	Dense row-major; reduction in scratchpad
Depth-to-hologram Propagates depth plane phase to pixel	43%	Transcendentals; FMADDs; thread-local reduction	Dense row-major; no pixel reads; pixels written once

Task	Time	Computation	Memory Pattern
Camera Processing Processes incoming camera depth image	5%	Bilateral filter; invalid depth rejection	Dense sequential accesses to depth image
Image Processing Pre-processes RGB-D image for tracking and mapping	18%	Generation of vertex map, normal map, and image intensity; image undistortion; pose transformation of old map	Globally dense; local stencil; layout change from RGB_RGB → RR_GG_BB
Pose Estimation Estimates 6DOF pose	28%	ICP; photometric error; geometric error	Photometric error is globally dense; others are globally sparse, locally dense
Surfel Prediction Calculates active surfels in current frame	38%	Vertex and fragment shaders	Globally sparse; locally dense
Map Fusion Updates map with new surfel information	11%	Vertex and fragment shaders	Globally sparse; locally dense

Task	Time	Computation	Memory Pattern
Normalization INT16 to FP32	7%	Element-wise FP32 division	Dense row-major
Encoding Sample to soundfield mapping	81%	$Y[j][i] = D \times X[j]$	Dense column-major
Summation HOA soundfield summation	11%	$Y[i][j] += X_k[i][j]\ \forall k$	Dense row-major

Task	Time	Sub-task	Computation	Memory Pattern
Rotation Soundfield rotation using pose	35%	**Psychoacoustic filter** Applies frequency-domain filter	FFT; frequency domain convolution; IFFT	Butterfly pattern for FFT/IFFT; dense row-major sequential accesses for convolution
		HOA rotation Rotates virtual channels	Transcendentals; FMADDs	Sparse column-major accesses; some temporal locality
Zoom Soundfield zoom using pose	5%	—	Linear algebra	Dense column-major sequential accesses
Binauralization HRTF application	60%	—	Identical to **psychoacoustic filter**	Identical to **psychoacoustic filter**

Results Summary and Implications for System Designers

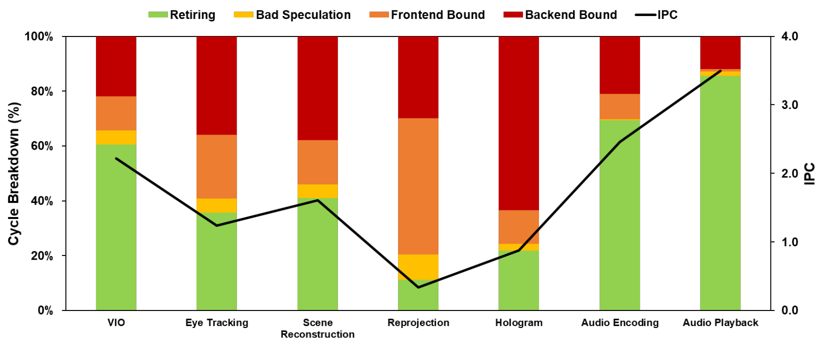
- Need to specialize hardware, software, *system*
- Must consider all application components in *system* together
- Must consider *system*-level hardware components; e.g., display and I/O
- Need to partition, allocate, and schedule *system* resources
- Must look at entire *system* to make QoE-driven tradeoffs

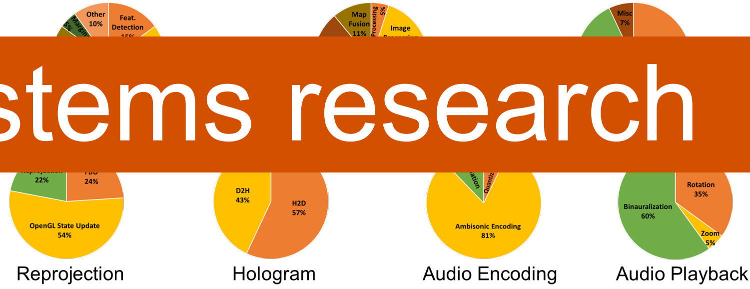
- Abundance of tasks and no single task dominates
  - ⇒ Need *automated* techniques to determine what to accelerate

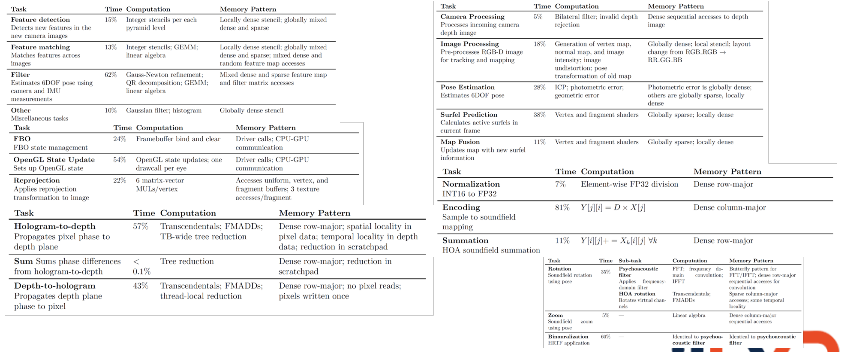
ILLIXR = Rich playground for systems research

- 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*

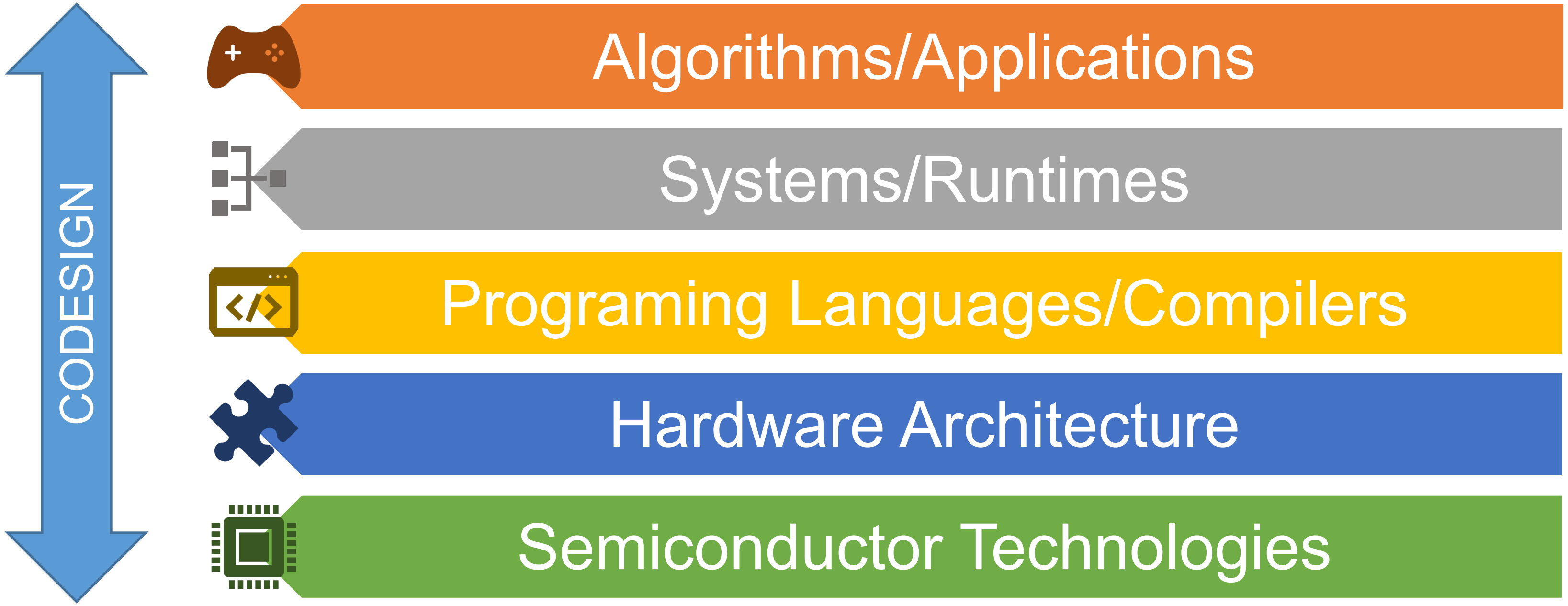
Standalone Components



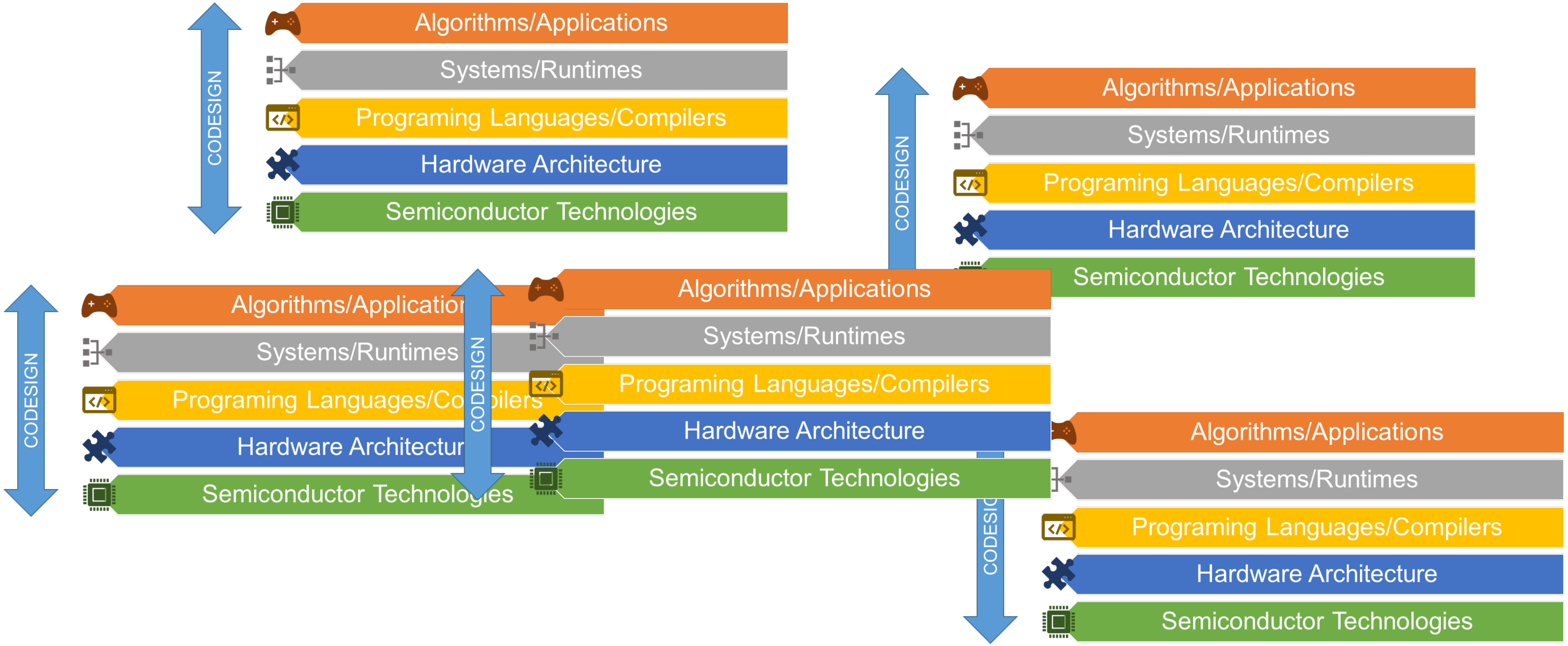




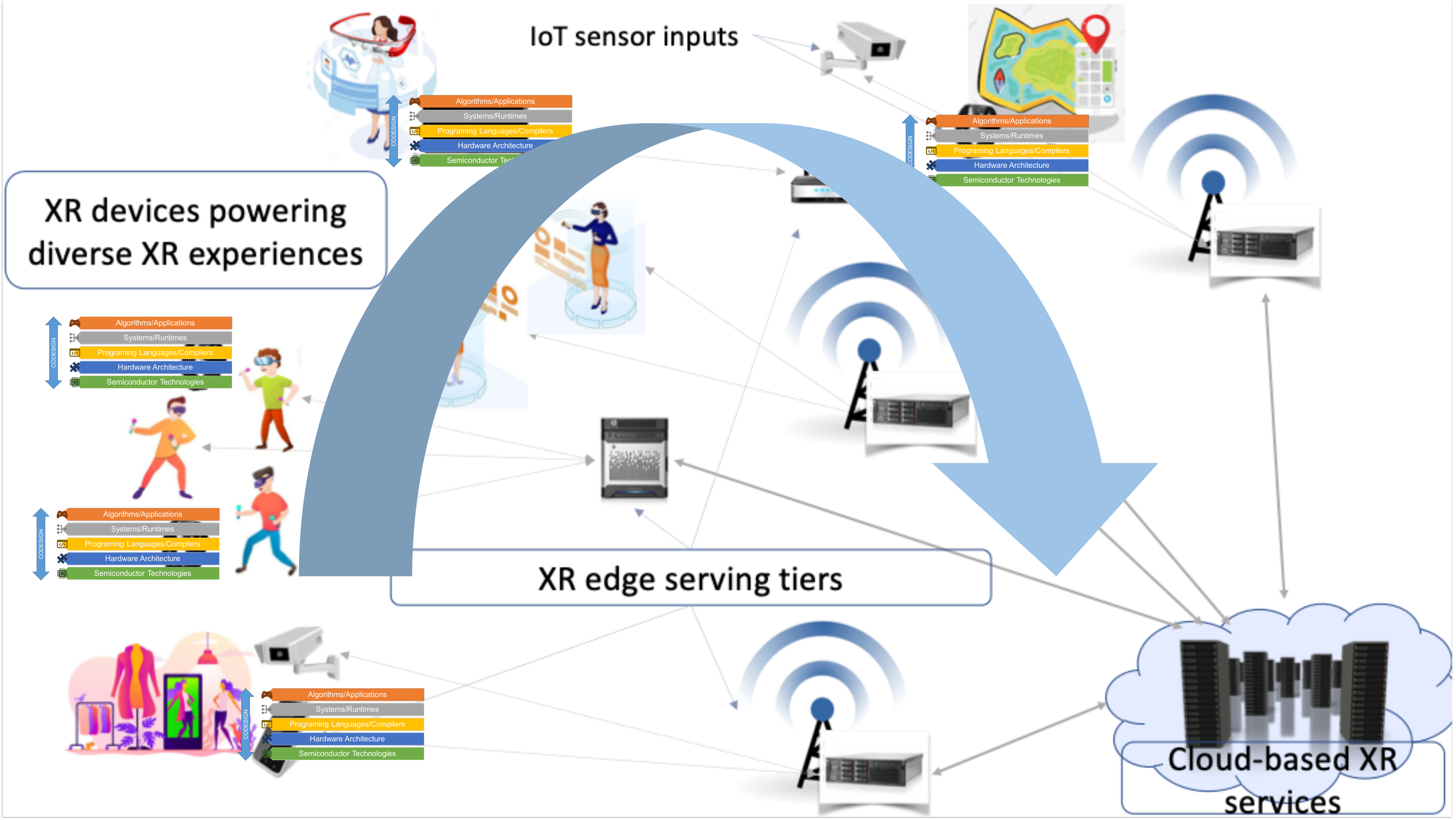
A New Style of Research



A New Style of Research

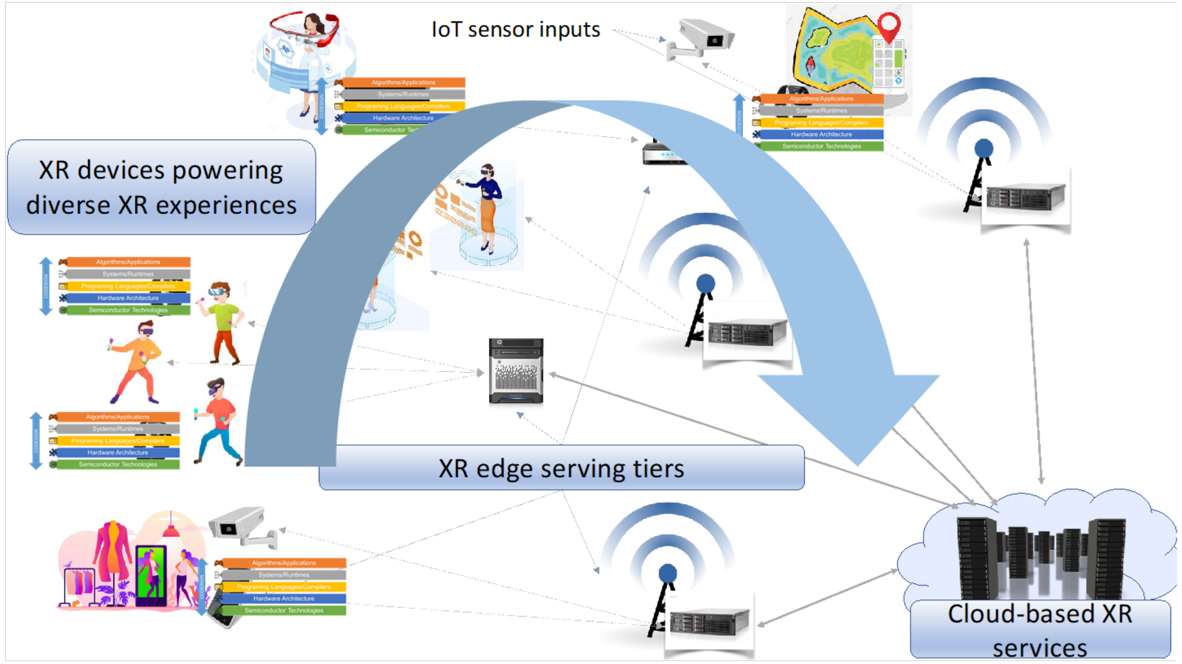


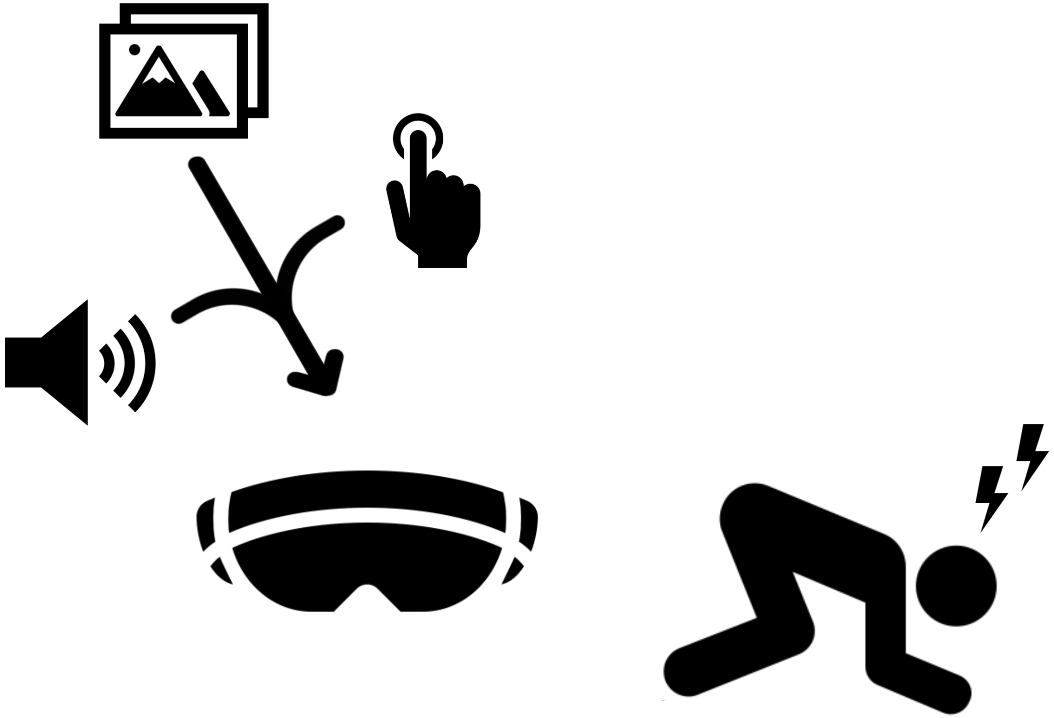
A New Style of Research



Distributed system figure from Gavrilovska

A New Style of Research





End-to-end QoE-driven, full system codesign

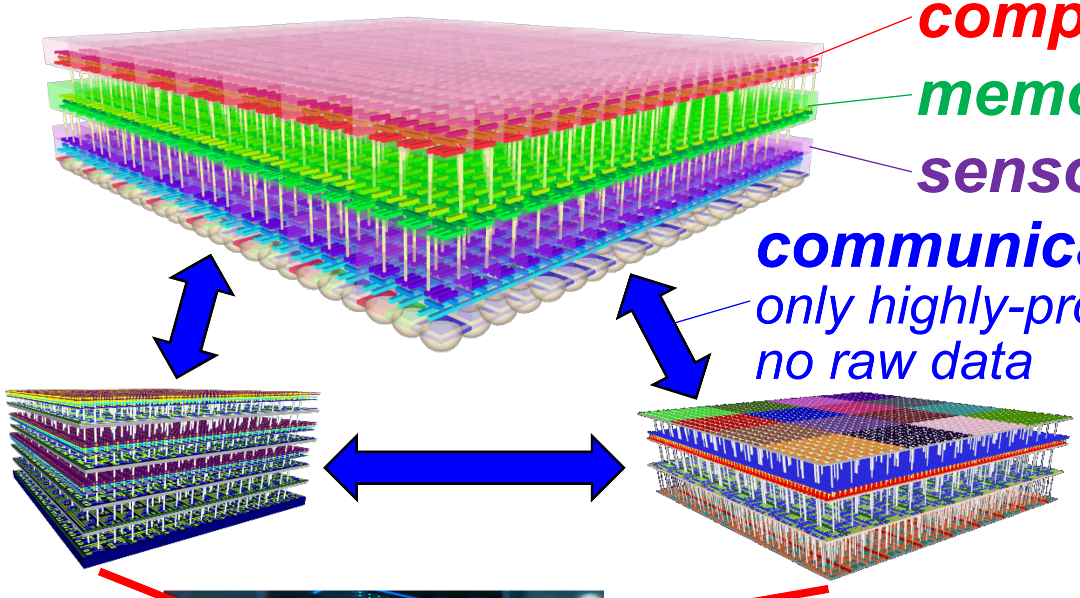
Research with ILLIXR

3D-Integrated Sense/Compute/Memory/Communication for XR

w/ D. Brooks, G. Hills

Enables ultra-low latency "sense-to-processed information" architectures + alleviates data communication bottlenecks

Network of 3D Integrated Circuits:
all 3D ICs have local sense/compute/memory



computation

memory

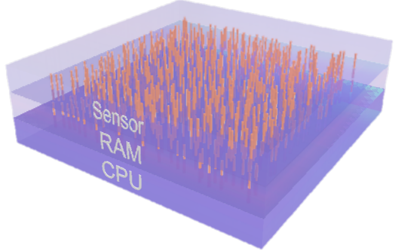
sensors

communication:

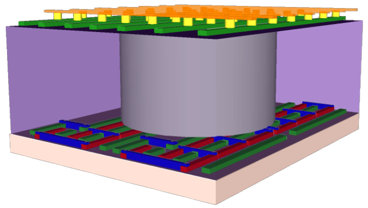
only highly-processed information
no raw data

driving application: ILLIXR

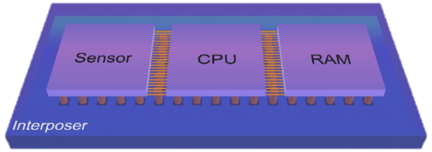
Design Space Exploration:
many options for 3D Integration



monolithic 3D
densest connectivity



3D chip stacking
denser connectivity



2.5-D: interposer + chiplets
dense connectivity

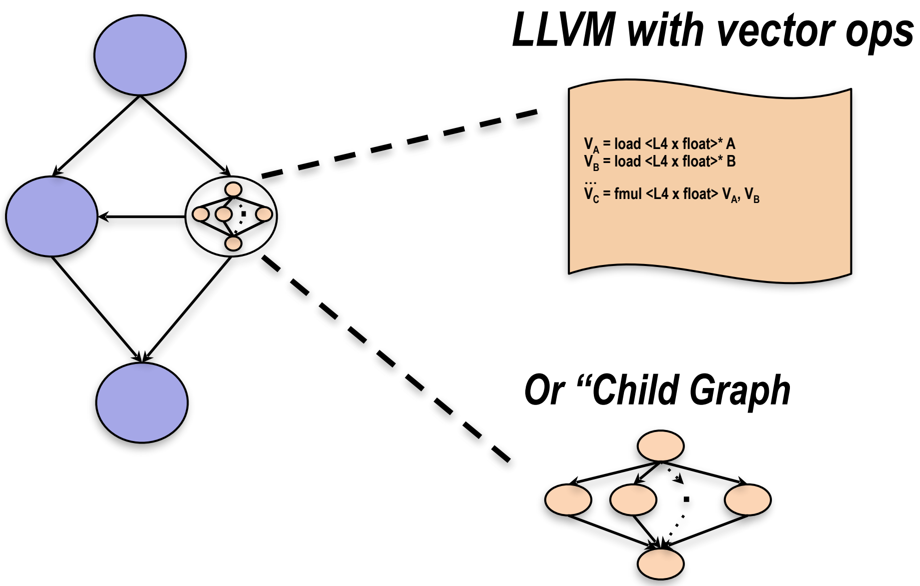
⋮

Representing Heterogeneous Parallelism in Software

w/ V. Adve and S. Misailovic

HPVM: Heterogeneous Parallel Virtual Machine [PPoPP18, OOPSLA19, PPOPP21]

Compiler IR and Hardware Virtual ISA



Model: Hierarchical dataflow graph with side effects

Captures

- coarse grain task parallelism
- streams, pipelined parallelism
- nested parallelism
- SPMD-style data parallelism
- fine grain vector parallelism

& data communication

Supports high-level optimizations as graph transformations

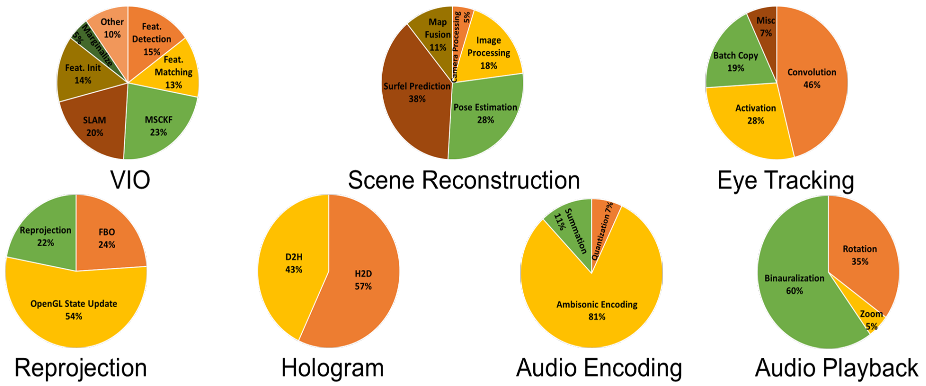
Targets: CPUs, vector extensions, GPUs, FPGAs, domain specific accelerators [so far, SoC; now distributed system]

Representing ILLIXR in HPVM

For code generation, automated accelerator selection, approximation, resource mapping, distributed systems, …

Automated Selection, Generation of Accelerator HW & SW

w/ V. Adve, D. Brooks, V. Reddi, G.-Y. Wei



Task	Time	Computation	Memory Pattern
Feature detection Detects new features in the new camera images	15%	Integer stencils per each pyramid level	Locally dense stencil; globally mixed dense and sparse
Feature matching Matches features across images	13%	Integer stencils; GEMM; linear algebra	Locally dense stencil; globally mixed dense and sparse; mixed dense and random feature map accesses
Filter Estimates 6DOF pose using camera and IMU measurements	62%	Gauss-Newton refinement; QR decomposition; GEMM; linear algebra	Mixed dense and sparse feature map and filter matrix accesses
Other Miscellaneous tasks	10%	Gaussian filter; histogram	Globally dense stencil

Task	Time	Computation	Memory Pattern
FBO FBO state management	24%	Framebuffer bind and clear	Driver calls; CPU-GPU communication
OpenGL State Update Sets up OpenGL state	54%	OpenGL state updates; one drawcall per eye	Driver calls; CPU-GPU communication
Reprojection Applies reprojection transformation to image	22%	6 matrix-vector MULs/vertex	Accesses uniform, vertex, and fragment buffers; 3 texture accesses/fragment

Task	Time	Computation	Memory Pattern
Hologram-to-depth Propagates pixel phase to depth plane	57%	Transcendentals; FMADDs; TB-wide tree reduction	Dense row-major; spatial locality in pixel data; temporal locality in depth data; reduction in scratchpad
Sum Sums phase differences from hologram-to-depth	< 0.1%	Tree reduction	Dense row-major; reduction in scratchpad
Depth-to-hologram Propagates depth plane phase to pixel	43%	Transcendentals; FMADDs; thread-local reduction	Dense row-major; no pixel reads; pixels written once

Task	Time	Computation	Memory Pattern
Camera Processing Processes incoming camera depth image	5%	Bilateral filter; invalid depth rejection	Dense sequential accesses to depth image
Image Processing Pre-processes RGB-D image for tracking and mapping	18%	Generation of vertex map, normal map, and image intensity; image undistortion; pose transformation of old map	Globally dense; local stencil; layout change from RGB_RGB → RR_GG_BB
Pose Estimation Estimates 6DOF pose	28%	ICP; photometric error; geometric error	Photometric error is globally dense; others are globally sparse, locally dense
Surfel Prediction Calculates active surfels in current frame	38%	Vertex and fragment shaders	Globally sparse; locally dense
Map Fusion Updates map with new surfel information	11%	Vertex and fragment shaders	Globally sparse; locally dense

Task	Time	Computation	Memory Pattern
Normalization INT16 to FP32	7%	Element-wise FP32 division	Dense row-major
Encoding Sample to soundfield mapping	81%	$Y[j][i] = D \times X[j]$	Dense column-major
Summation HOA soundfield summation	11%	$Y[i][j] += X_k[i][j] \; \forall k$	Dense row-major

Task	Time	Sub-task	Computation	Memory Pattern
Rotation Soundfield rotation using pose	35%	**Psychoacoustic filter** Applies frequency-domain filter	FFT; frequency domain convolution; IFFT	Butterfly pattern for FFT/IFFT; dense row-major sequential accesses for convolution
		HOA rotation Rotates virtual channels	Transcendentals; FMADDs	Sparse column-major accesses; some temporal locality
Zoom Soundfield zoom using pose	5%	—	Linear algebra	Dense column-major sequential accesses
Binauralization HRTF application	60%	—	Identical to **psychoacoustic filter**	Identical to **psychoacoustic filter**

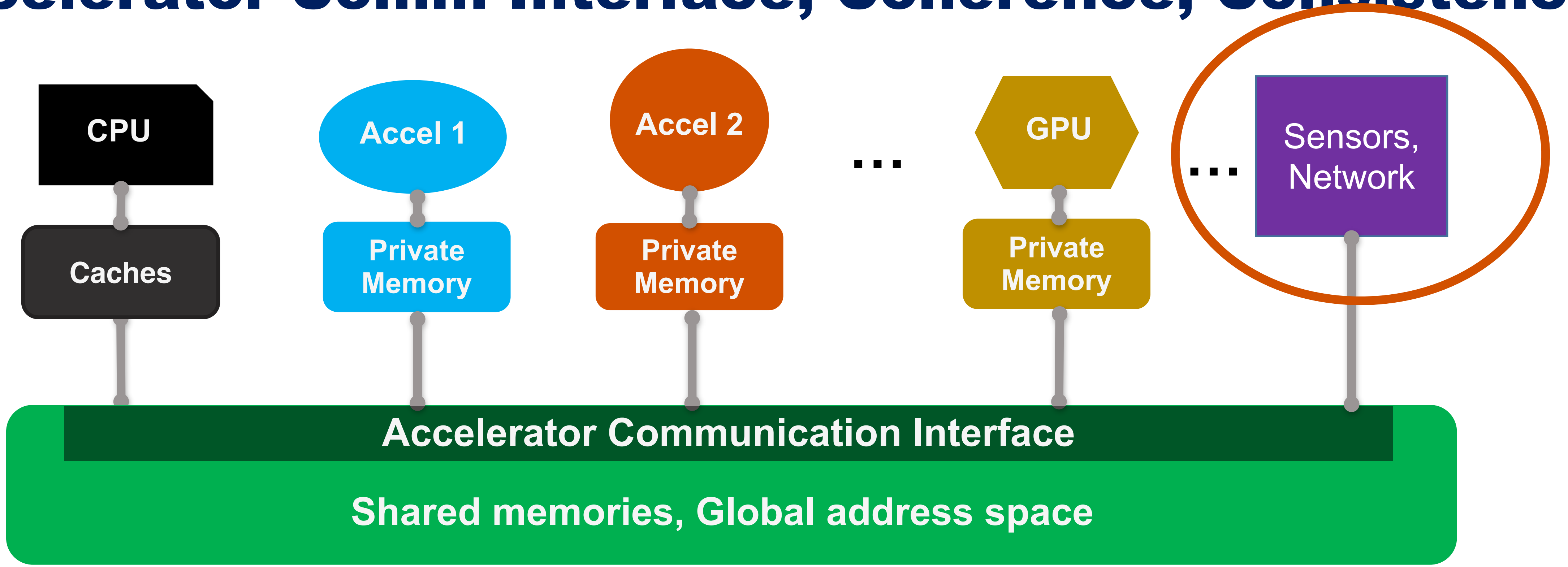
Manual identification of common compute, memory patterns

⇒ Cross-component co-design allows hardware, computation, and data reuse w/ large benefits

Automated design space exploration to identify profitable acceleration, generate HW+SW

- Use HPVM's parallelism representation
- Recent results for automated design space exploration w/ loop, task, streaming parallelism
  - ~2X better performance for same area vs. using sequential LLVM representation [in review]
- Ongoing: Compiler analysis and transformations for common patterns and optimizations, code generation, resource mapping

Accelerator Comm Interface, Coherence, Consistency



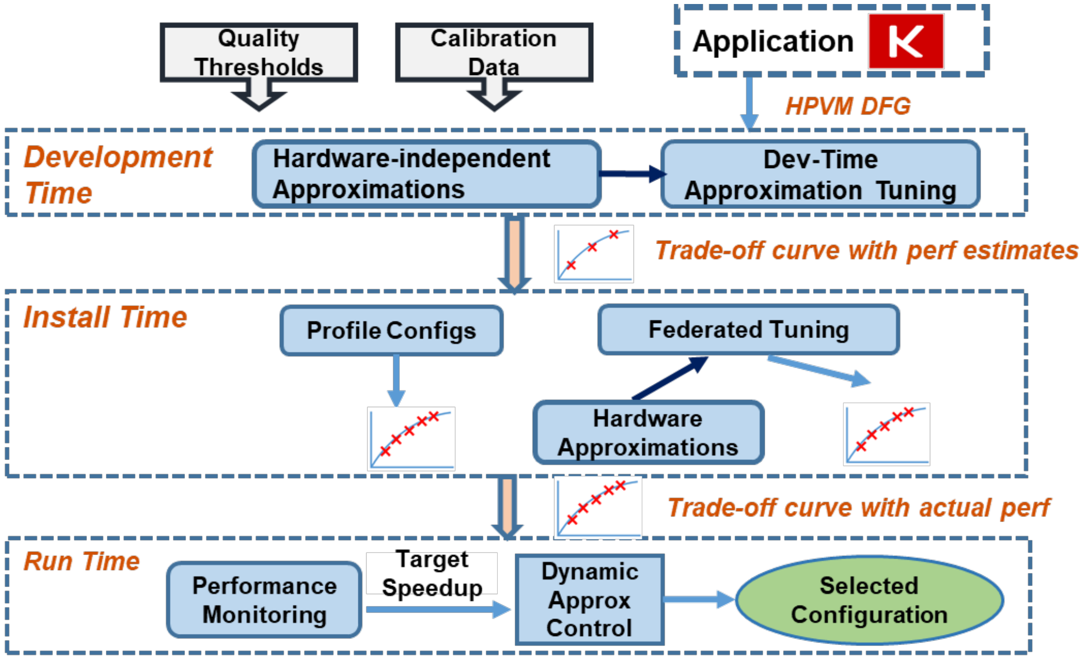
- How should heterogeneous parallel accelerators, sensors, network i/f, … communicate w/ each other?
- Programmable, shared hardware ⇒ shared memory
  - Coherence, consistency, communication
  - Build on Spandex heterogeneous coherence interface for coherence specialization [ISCA18, TACO’22]

Automated Approximation Selection

w/ V. Adve and S. Misailovic

ApproxTuner [PPoPP21]

Combines multiple software and hardware approximations for tensor operations



Uses predictive models to compose accuracy impact of multiple approximations

3-phase approximation tuning

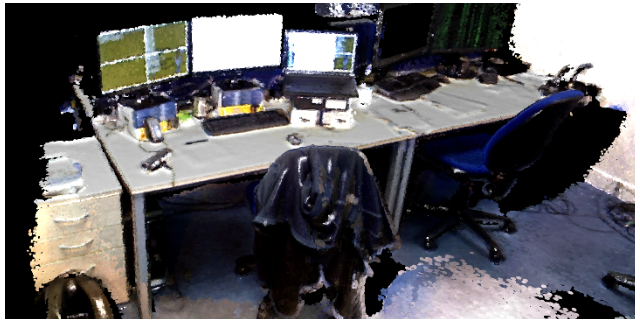
- Development-time preserves hardware portability via ApproxHPVM IR
- Install-time allows hardware-specific approximations
- Run-time allows dynamic approximation tuning

Approximations for ILLIXR

Build on ApproxTuner for QoE-driven automated selection

End-to-End Cross-Component Co-Design

- Scene reconstruction
  - Co-design with other upstream and downstream components
  - Co-design Hardware + System software + Algorithm
  - So far 69X better energy/frame w/ only SW (vs. InfiniTAM)
  - Hardware accelerator in progress

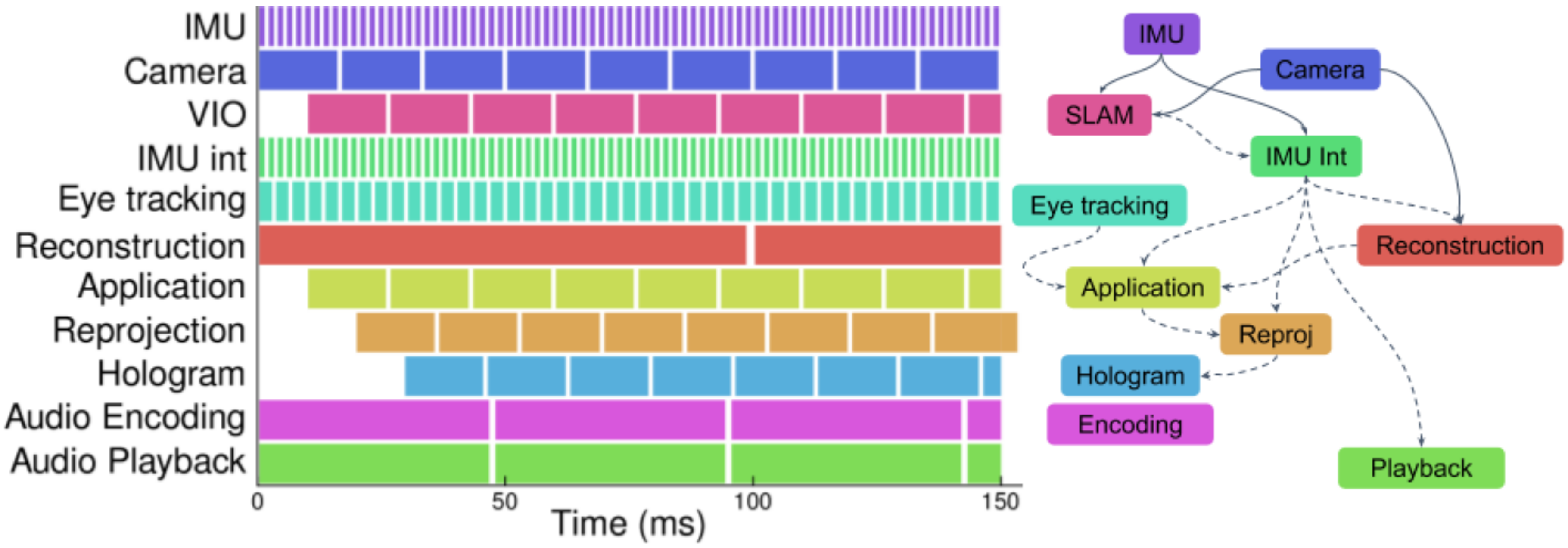


- Eye tracked foveated rendering (w/ NVIDIA)
  - How to trade off accuracy among components?

    Disciplined end-to-end accuracy driven approximation w/ Aprox
  - Foveated video image quality metrics

QoE-Driven Scheduling

w/ P. B. Godfrey, R. Mittal



ILLIXR task graph is a DAG with multiple critical paths and QoE constraints

Scheduler goal: Determine frame rates and schedule to meet QoE for given hardware mapping

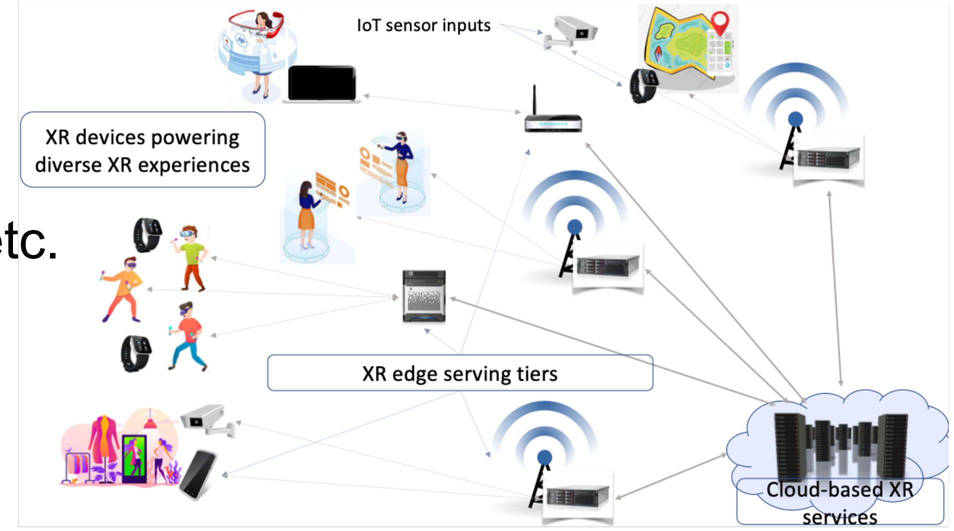
Preliminary results: Lower MTP than Linux baseline on single core CPU

Ongoing: Multiple hardware targets for given task, hardware and software approximations

Offloading to Remote Servers

w/ A. Gavrilovska, Godfrey, Hassanieh, Intel

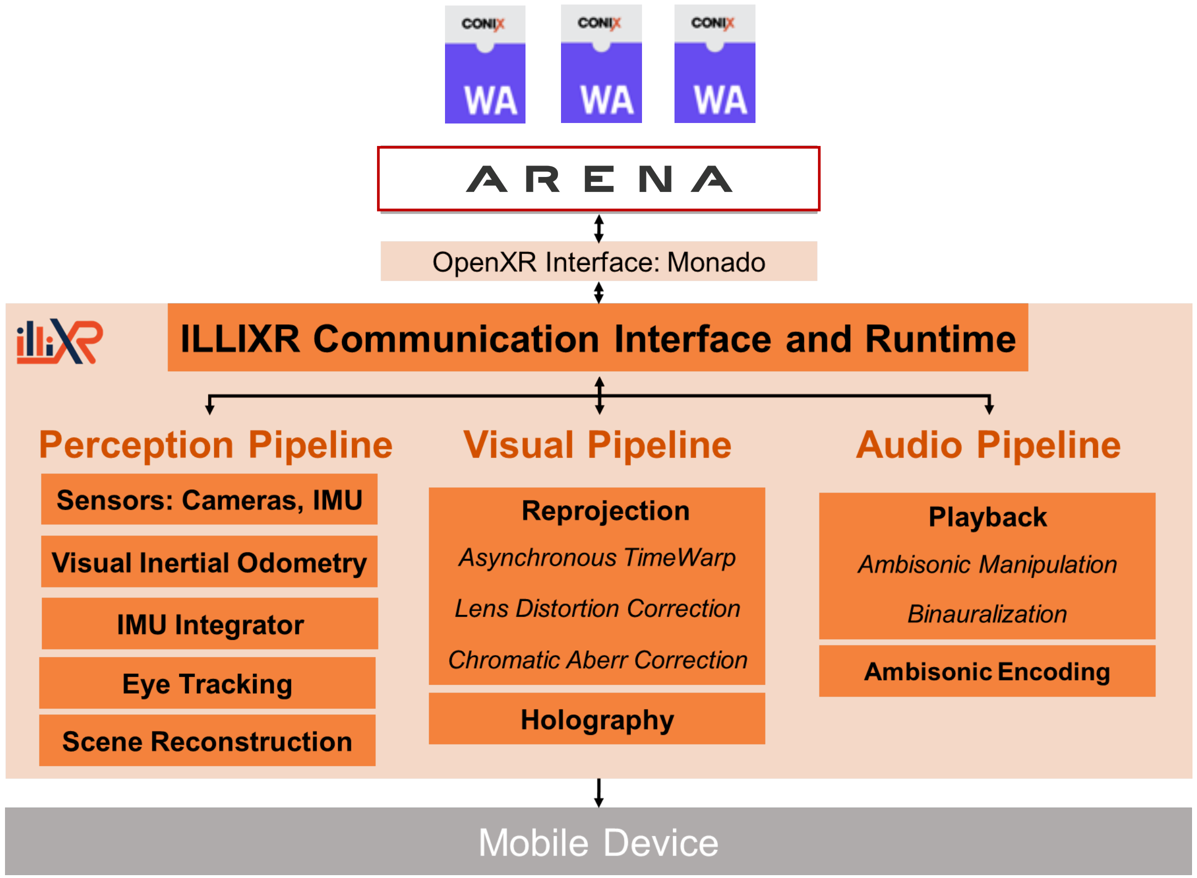
- Offloading computation to remote compute
  - Recent support in ILLIXR
  - What to offload, when, where?
    - Depends on compression, transmission energy, etc.
    - Integrate with scheduler
  - Impact of network
    - Intel's Wireless Time Sensitive Networking
    - mmWave
  - Impact on accelerator design, algorithm, scheduler



Multi-User Immersive Systems

w/ A. Gavrilovska, Nahrstedt, Rowe

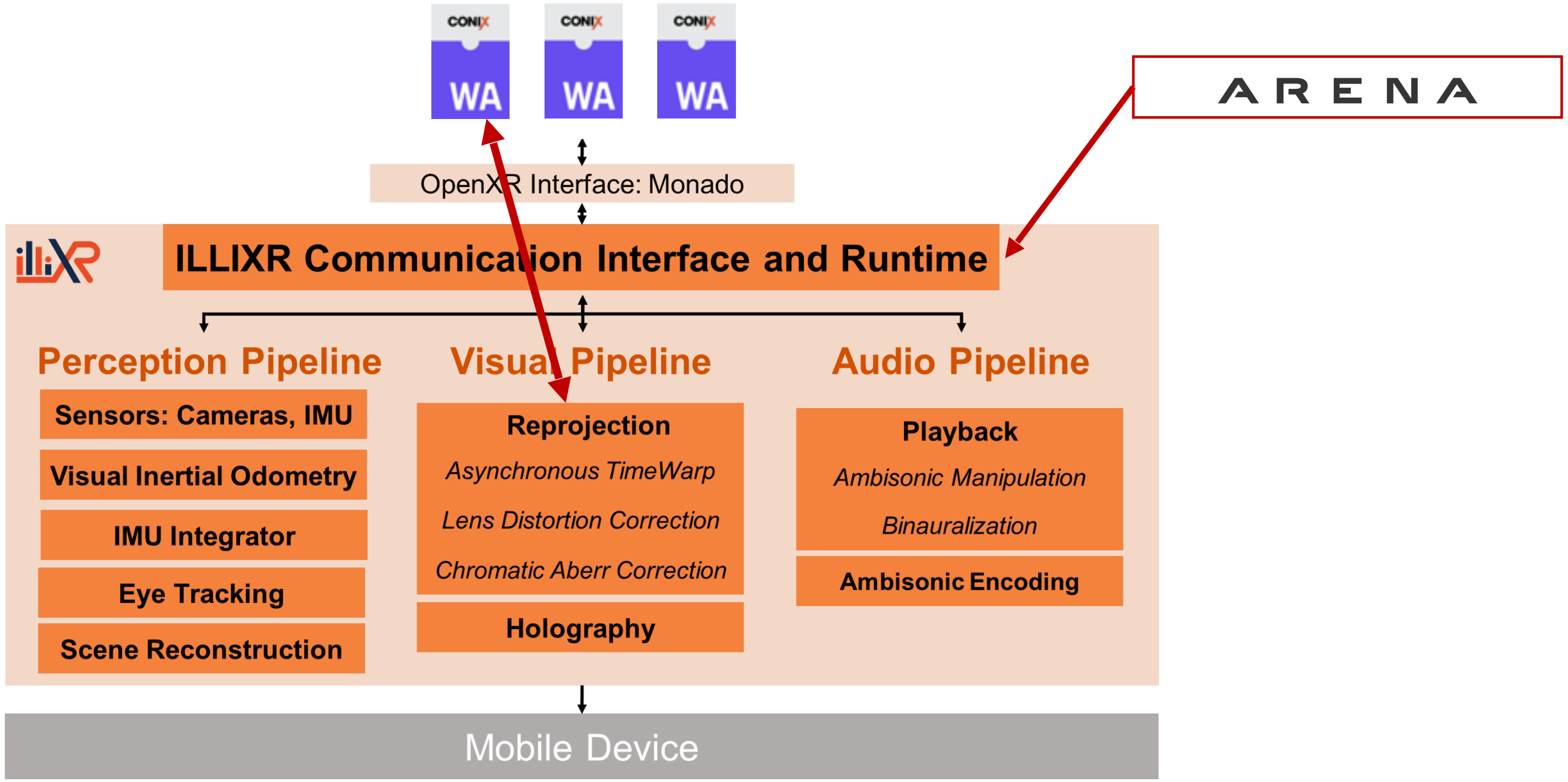
- Multiuser XR experiences
  - Devices, edge, cloud distributed computing
  - Step 1: ILLIXR + CMU's ARENA for distributed services



Multi-User Immersive Systems

w/ A. Gavrilovska, Nahrstedt, Rowe

- Multiuser XR experiences
  - Devices, edge, cloud distributed computing
  - Step 1: ILLIXR + CMU's ARENA for distributed services



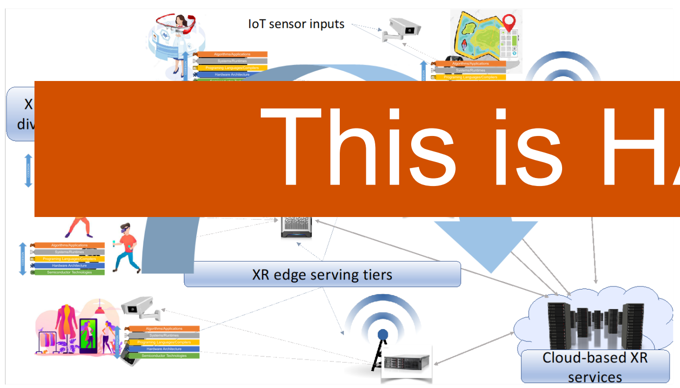
And More

- Eye tracking + Holograms [Sivasubramanium et al., Micro’21]
- **Security and Privacy**
- 360 Video streaming
- Multiparty AR programming stack
- Displays
- On-sensor computing
- QoE metrics
- XR algorithms
- …

A New Immersive Era

Will transform how we design, program, and use computers

We need new style of research



This is HARD

End-to-end QoE-driven, full system codesign

Build systems

- Chips, compilers, runtimes, apps

User studies

Large teams

We need new style of reviewing

- ILLIXR paper rejected four times from top conferences

We need new style of funding

- We were fortunate to be part of the DARPA/SRC funded ADA center, DARPA DSSOC project IBM/Pradip Bose + 3 univs, (recently) NSF CISE Community Research Infrastructure

ILLIXR: Illinois Extended Reality Testbed

ILLIXR is a rich playground for immersive systems research

Consortium for immersive systems research, development, and benchmarking

Join us: illixr@cs.illinois.edu, illixr.org, discord, open meetings on Wed@11a CT