Systems 2030: The Extended Reality Case

Sarita Adve

University of Illinois at Urbana-Champaign
sadve@illinois.edu
w/ many collaborators acknowledged on slides

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

Explosion of accelerators in SoCs (System-on-Chip)

How to design specialized accelerators?
How to design specialized system?
Communication?
Software?
Applications?

Source: Brooks, Wei group, http://vlsiarch.eecs.harvard.edu/accelerators/die-photo-analysis
Rise of the Edge

Challenging performance demands
Stringent resource constraints
End-to-end quality metrics

VR@Illinois

Chowdhari et al., EarthSense robots

EPOCHS project, IBM, Columbia, Harvard, Illinois
Scalable, Generalizable
Specialization Techniques

Scalable
Application-driven, end-to-end quality driven, HW-SW-App co-designed system
Systems 2030
Scalable, Generalizable
Specialization Techniques

BUT What Application?

Application-driven, end-to-end quality driven, HW-SW-App co-designed system

Systems 2030
Systems 2030: The Extended Reality Case

Extended Reality (XR) = Virtual, augmented, mixed, … reality

Pervasive: Science, medicine, entertainment, education, …

Challenging: Orders of magnitude power, performance, quality gap to reach ideal

Diverse: Involves graphics, vision, audio, video, robotics, optics, haptics, …

Full stack: Challenges span hardware, compiler, OS, algorithm

Flexible: User-driven, end-to-end quality of experience (QoE) metrics

Great driver for research for Systems 2030
# XR Requirements

<table>
<thead>
<tr>
<th></th>
<th>VR</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HTC Vive Pro</td>
<td>Ideal</td>
</tr>
<tr>
<td>Resolution (Mpixels)</td>
<td>4.6</td>
<td>200</td>
</tr>
<tr>
<td>Field of view (degrees)</td>
<td>110</td>
<td>Full: 165x175</td>
</tr>
<tr>
<td>Refresh rate (Hz)</td>
<td>90</td>
<td>90-144</td>
</tr>
<tr>
<td>Motion to photon lat (ms)</td>
<td>&lt; 20</td>
<td>&lt; 20</td>
</tr>
<tr>
<td>Power (W)</td>
<td>N/A (server)</td>
<td>1 - 2</td>
</tr>
<tr>
<td>Silicon area (mm²)</td>
<td>N/A (server)</td>
<td>100-200</td>
</tr>
<tr>
<td>Weight (g)</td>
<td>470</td>
<td>100-200</td>
</tr>
</tbody>
</table>

*Orders of magnitude gap in power, performance, area, weight, QoE*
Challenges for XR Systems Research

• Needs expertise from many domains

• Closed interfaces
  – Recent: OpenXR opened application-runtime interface

• State-of-the-art closely guarded by industry
  No open-source benchmarks or systems
ILLIXR: Illinois Extended Reality Testbed [Huzaifa et al., ‘20]

- ILLIXR: First open-source full system XR testbed
- State-of-the-art XR components integrated with modular and extensible runtime
- OpenXR compatible
- Several QoE metrics
- Runs on desktops, embedded systems

Soon: Community Consortium
- Industry + academic partners
  - ARM, Facebook, Micron, NVIDIA, …
- Standardize benchmarking, QoE metrics, …

A new playground for systems 2030 and XR research

illixr.github.io
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
- 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

External Consultations

- Joseph Ravichandran
- Giordano Salvador
- Finn Sinclair
- Boyuan Tian
- Lauren Wagner
- Henghzi Yuan
- Jeffrey Zhang
- 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
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
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

• Wei Cui
  • Aleksandra Faust
  • Liang Gao
  • Matt Horsnell
  • Amit Jindal
  • Steve LaValle
  • Steve Lovegrove

• Wei Cui
  • Vesgaard Oye

External Consultations

• Joseph Ravichandran
  • Giordano Salvador
  • Finn Sinclair
  • Boyuan Tian
  • Lauren Wagner
  • Henghzi Yuan
  • Jeffrey Zhang

• Wei Cui
  • Aleksandra Faust
  • Liang Gao
  • Matt Horsnell
  • Amit Jindal
  • Steve LaValle
  • Steve Lovegrove

• Wei Cui
  • Vesgaard Oye

• Martin Persson
  • Archontis Politis
  • Eric Shaffer
  • Paris Smaragdis
  • Chris Widdowson

• Martin Persson
  • Archontis Politis
  • Eric Shaffer
  • Paris Smaragdis
  • Chris Widdowson
ILLIXR Overview

Illuxr Communication Interface and Runtime

Perception Pipeline
- Sensors: Cameras, IMU
- Visual Inertial Odometry
- IMU Integrator
- Eye Tracking
- Scene Reconstruction

Visual Pipeline
- Reprojection
  - Asynchronous TimeWarp
  - Lens Distortion Correction
  - Chromatic Aberr Correction
- Display: Hologram

Audio Pipeline
- Playback
  - Ambisonic Manipulation
  - Binauralization
- Ambisonic Encoding

XR Applications
OpenXR Interface: Monado

Mobile Device
Perception Pipeline

• Sensors: Camera, Inertial Measurement Unit (IMU)
  – Provides position and orientation of user’s head (pose)

• Visual Intertial Odometry (VIO)
  – Provides position and orientation of user’s head (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 (TimeWarp)
  – 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

• Adaptive display: hologram
  – Vergence-accommodation conflict (VAC) causes fatigue, headache
    • Eyes focused (accommodated) at fixed point, converge at different points
  – Computational displays w/ multiple focal planes can fix VAC
  – Computational holography: 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
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

• 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

<table>
<thead>
<tr>
<th>Component</th>
<th>Algorithm</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perception Pipeline</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera Camera</td>
<td>ZED SDK</td>
<td>C++</td>
</tr>
<tr>
<td>IMU IMU</td>
<td>ZED SDK</td>
<td>C++</td>
</tr>
<tr>
<td>VIO VIO</td>
<td>OpenVINS</td>
<td>C++</td>
</tr>
<tr>
<td>IMU Integrator IMU Integrator</td>
<td>RK4 GTSAM</td>
<td>C++</td>
</tr>
<tr>
<td>Eye Tracking</td>
<td>RITnet</td>
<td>Python, CUDA</td>
</tr>
<tr>
<td>Scene Reconstruction</td>
<td>ElasticFusion KinectFusion</td>
<td>C++, CUDA, GLSL C++, CUDA</td>
</tr>
<tr>
<td><strong>Visual Pipeline</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reprojection</td>
<td>VP-matrix reproject w/ pose</td>
<td>C++, GLSL</td>
</tr>
<tr>
<td>Lens Distortion</td>
<td>Mesh-based radial distortion</td>
<td>C++, GLSL</td>
</tr>
<tr>
<td>Chromatic Aberration</td>
<td>Mesh-based radial distortion</td>
<td>C++, GLSL</td>
</tr>
<tr>
<td><strong>Audio Pipeline</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio Encoding</td>
<td>Ambisonic encoding</td>
<td>C++</td>
</tr>
<tr>
<td>Audio Playback</td>
<td>Ambisonic manipulation, binauralization</td>
<td>C++</td>
</tr>
</tbody>
</table>
Evaluation Methodology

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

- 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
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
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

<table>
<thead>
<tr>
<th>Application</th>
<th>Desktop</th>
<th>Jetson-hp</th>
<th>Jetson-lp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sponza</td>
<td>3.1 ± 1.1</td>
<td>13.5 ± 10.7</td>
<td>19.3 ± 14.5</td>
</tr>
<tr>
<td>Materials</td>
<td>3.1 ± 1.0</td>
<td>7.7 ± 2.7</td>
<td>16.4 ± 4.9</td>
</tr>
<tr>
<td>Platformer</td>
<td>3.0 ± 0.9</td>
<td>6.0 ± 1.9</td>
<td>11.3 ± 4.7</td>
</tr>
<tr>
<td>AR Demo</td>
<td>3.0 ± 0.9</td>
<td>5.6 ± 1.4</td>
<td>12.0 ± 3.4</td>
</tr>
</tbody>
</table>
Motion-to-Photon Latency

Sponza

AR Demo

Extremely unpleasant experience on Jetson
## Image Quality

<table>
<thead>
<tr>
<th>Platform</th>
<th>SSIM</th>
<th>1-FLIP</th>
<th>ATE/degree</th>
<th>ATE/meters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop</td>
<td>0.83 ± 0.04</td>
<td>0.86 ± 0.05</td>
<td>8.6 ± 6.2</td>
<td>0.33 ± 0.15</td>
</tr>
<tr>
<td>Jetson-hp</td>
<td>0.80 ± 0.05</td>
<td>0.85 ± 0.05</td>
<td>18 ± 13</td>
<td>0.70 ± 0.33</td>
</tr>
<tr>
<td>Jetson-lp</td>
<td>0.68 ± 0.09</td>
<td>0.65 ± 0.17</td>
<td><strong>138 ± 26</strong></td>
<td>13 ± 10</td>
</tr>
</tbody>
</table>

Must consider end-to-end QoE
Need better QoE metrics
Implications for Architects

• 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
Component Microarchitectural Diversity

Wide range in IPC and hardware utilization
Task Diversity

VIO
- Reprojection 22%
- FBO 24%
- OpenGL State Update 54%

Scene Reconstruction
- D2H 43%
- H2D 57%

Audio Encoding
- Ambisonic Encoding 81%

Eye Tracking
- Rotation 35%
- Binauralization 60%
- Zoom 5%

Variety (27!) of tasks and no task dominates
## Component Deep Dive

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
<th>Computation</th>
<th>Memory Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Feature detection</strong></td>
<td>15%</td>
<td>Integer stencil, once per pyramid level</td>
<td>Subtask 1: Globally dense, local Bresenham stencil</td>
</tr>
<tr>
<td>Detects new features in the new camera images</td>
<td></td>
<td></td>
<td>Subtask 2: Globally sparse feature accesses, locally dense stencil</td>
</tr>
<tr>
<td><strong>Feature matching</strong></td>
<td>13%</td>
<td>Integer stencil; GEMM; RANSAC; linear algebra</td>
<td>Subtask 1: Globally sparse, locally dense pixel accesses</td>
</tr>
<tr>
<td>Matches features across images</td>
<td></td>
<td></td>
<td>Subtask 2: dense feature map accesses</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Subtask 3: random feature map accesses</td>
</tr>
<tr>
<td><strong>Filter</strong></td>
<td>62%</td>
<td>Gauss-Newton refinement; QR decomposition; GEMM; linear algebra</td>
<td>Mixed dense and sparse feature map and filter matrix accesses</td>
</tr>
<tr>
<td>Estimates 6DOF pose using camera and IMU measurements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>10%</td>
<td>Gaussian filter; histogram</td>
<td>Globally dense stencil</td>
</tr>
<tr>
<td>Miscellaneous tasks</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Component Deep Dive

### Task | Time | Computation | Memory Pattern
--- | --- | --- | ---
Feature detection | 5% | Integer stencils per each pyramid level | Locally dense stencil; globally mixed dense and sparse
Feature matching | 13% | Integer stencils; GEMM; linear algebra | Locally dense stencil; globally mixed dense and sparse; mixed dense and random feature map accesses
Filter | 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
FBO | 24% | Framebuffer bind and clear | Driver calls CPU-GPU communication
OpenGL State Update | 54% | OpenGL state updates; one drawcall per eye | Driver calls CPU-GPU communication
Reprojection | 22% | 6 matrix-vector MULs/vertex | Accesses uniform, vertex, and fragment buffers; 3 texture fragments/fragment
Hologram-to-depth | 57% | Transcendentials; FMADDs; TB-wide tree reduction | Dense row-major; spatial locality in pixel data; temporal locality in depth data; reduction in scratchpad
Sum | < 0.1% | Tree reduction | Dense row-major; reduction in scratchpad
Depth-to-hologram | 43% | Transcendentials; FMADDs; thread-local reduction | Dense row-major; no pixel reads; pixels written once
Camera Processing | 5% | Bilateral filter; invalid depth rejection | Dense sequential accesses to depth image
Image Processing | 16% | Generation of vertex map, normal map, and image intensity; image undistortion; pose transformation of old map | Globally dense; local stencil; layout change from RGBA,RGB=R,G,G,B
Pose Estimation | 26% | RCP; photometric error; geometric error | Photometric error is globally dense; others are globally sparse, locally dense
Surfel Prediction | 36% | Vertex and fragment shaders | Globally sparse; locally dense
Map Fusion | 14% | Vertex and fragment shaders | Globally sparse; locally dense
Normalization INT16 to FP32 | 7% | Element-wise FP32 division | Dense row-major
Encoding Sample to soundfield mapping | 81% | $Y[i][t] = D \times X[i]$ | Dense column-major
Summation HOA soundfield summation | 11% | $Y[i][t] + = X_k[i][t] \forall k$ | Dense row-major
Rotation Soundfield rotation using pose | 30% | Psychoacoustic filter | FFT; frequency domain correlation; FFT
Zoom Soundfield zoom using pose | 16% | Linear algebra | Dense column-major sequential accesses
Resampling SHF application | 60% | Identical to psychoacoustic filter | Identical to psychoacoustic filter
Implications for Architects

- 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 2030 research
• How should heterogeneous parallel accelerators communicate with each other?
• Programmable, shared hardware ⇒ shared memory
  – Coherence, consistency, communication
  – Build on Spandex heterogeneous coherence interface for coherence specialization [ISCA18, in review]
Representing Heterogeneous Parallelism in Software

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

Supports high-level optimizations as graph transformations
Targets: CPUs, vector extensions, GPUs, FPGAs, domain specific accelerators

Representing ILLIXR in HPVM

For code generation, automated accelerator selection, approximation, resource mapping, …
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
Automated Selection, Generation of Accelerator HW & SW

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

Manual identification of common compute, memory patterns
⇒ Cross-component co-design allows hardware, computation, and data reuse w/ large benefits

Goal: Automated design space exploration to identify profitable acceleration, generate HW+SW
  – Use HPVM’s parallelism and communication representation
  – Compiler analysis and transformations for common patterns and optimizations
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

Future: Multiple hardware targets for given task, hardware and software approximations
From Single-Device to Distributed Systems

w/ A. Gavrilovska, K. Nahrstedt

- Offload computation to edge, cloud servers
- Content streaming
- Multiparty AR/VR experiences
More Use Cases

• Security and Privacy
• 360 Video streaming
• Multiparty AR programming stack
• Displays
• On-sensor computing
• QoE metrics
• XR algorithms
• …
ILLIXR Testbed

• New components: translational reprojection (spacewarp), hand tracking, ...
• Add North Star head set
• Broaden hardware/software platforms supported
• Create and curate data sets and applications
• Incorporate research results
• …

Soon: Community Consortium
• Industry + academic partners
  – ARM, Facebook, Micron, NVIDIA, …
• Standardize benchmarking, QoE metrics, …
ILLIXR is a rich playground for research for Systems 2030


Scalable & Generalizable Specialization

Application-driven, end-to-end quality driven, HW-SW-App co-designed system specialization techniques

illixr.github.io
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

Spandex team: Johnathan Alsop, Robert Jin, Weon Tak Na, Matthew Sinclair, Zeran Zhu

HPVM team: Vikram Adve, Adel Ejjeh, Muhammad Huzaifa, Keyur Joshi, Rakesh Komuravelli, Maria Kotsifakou, Akash Kothari, Sasa Misailovic, Yasmin Sarita, Ben Schreiber, Hashim Sharif, Matthew Sinclair, Prakalp Srivastava, Elizabeth Wang, Yifan Zhao, Nathan Zhao
Some Life Stories and Lessons

• ILLIXR story
  – Born out of frustration and desire to impact something real
  – Had no clue about XR except that my colleague Steve Lavalle had recently returned from a successful stint as founding chief scientist at Oculus 😊
  – CFAR seed proposal (1 page) + Encouraging colleague + Excited student ⇒
    Detour became main research thrust w/ many students, faculty, and industry collaborators
  – 3+ years of work, real impact still to come, but already satisfying

• Memory models story
  – Frustration with HW memory models, called on SW community to fix
  – Joined Java memory model effort – didn’t know Java or PL-ese, no students, no funding
  – 5 years of work, 1 paper, but real impact

• SIGARCH chair story
  – Colleagues gathered frustrating data on diversity in architecture community
  – Joined hands with colleagues for intense activism, concrete actions
  – ~3 years of work, no papers, but real impact; e.g., CARES movement
Some Life Stories and Lessons

• Follow your passion

• Take risks. Believe in yourself.

• Impact = Change minds. Takes time and hard work. (≠ # Papers)

• It takes a village. Pay it forward.