EECE5512
Networked XR Systems

Recap

- XR Experiences Session
- 4D Mesh Streaming
  - Mesh fusion
  - Texture streaming
  - Splitting bandwidth between mesh and texture
  - Adaptation of mesh and texture bitrates

Lecture outline today

- Building smart glasses
    - Hardware components
    - Software components

Smart glasses as the next computing frontier

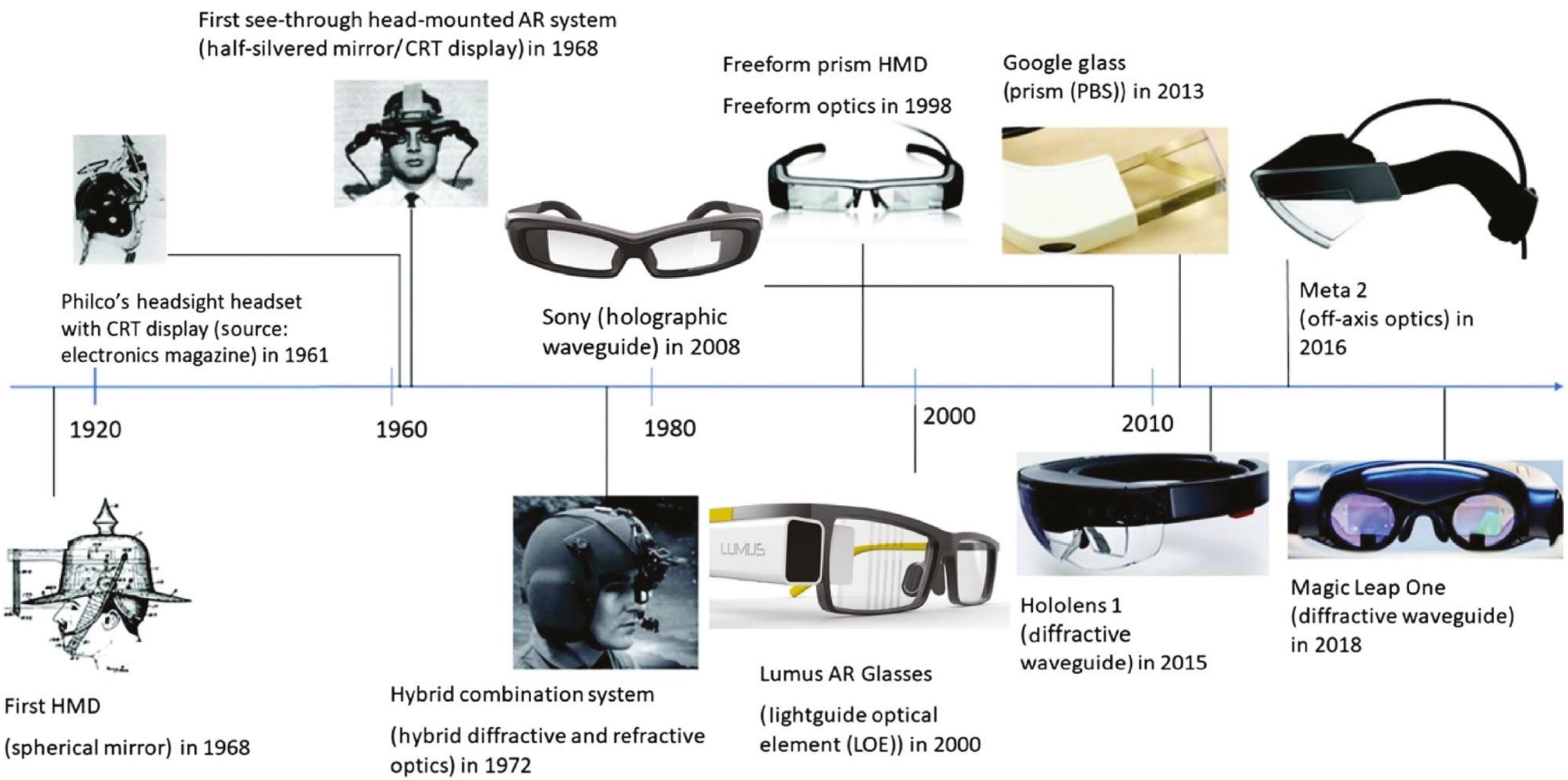
Smartphones made computing *portable*; smart glasses make it *invisible*

Smart glasses as the next computing frontier

- Always-on: Low-power sensors continuously perceive the environment
- Context-aware: Understand what the user is seeing, hearing, or doing
- Hands-free: Information and control through gaze, voice, or gesture
- Intelligent overlay: Augmented cues or AI summaries directly in view

Smart glasses as the next computing frontier

Brief history



Smart glasses as the next computing frontier

Brief history

Year	Device	Key Idea	Limitation
2013	**Google Glass**	Early AR with camera & HUD	Privacy concerns, limited UX
2016	**Microsoft HoloLens**	Mixed Reality, spatial mapping	Bulky, enterprise-only
2021	**Meta Ray-Ban Stories**	Lifestyle + camera + voice	No display overlay
2023	**Brilliant Frame / Omi AI Glass**	AI assistant & display integration	Battery, compute limits
2024→	**Emerging AI Glasses**	Multimodal LLMs + edge AI	Integration + power efficiency

Each generation tackled a different problem: visibility, comfort, and intelligence, but none fully solved them all.

Smart glasses as the next computing frontier

Market Overview: AR vs XR vs AI Glasses

Category	Focus	Example Devices	Primary Tech
AR (augmented reality)	Overlay visuals on real world	HoloLens, Magic Leap	Spatial mapping, 3D rendering
XR (extended reality)	Unified spectrum of AR/VR/MR	Quest 3, XReal Air	Visual immersion + tracking
AI Glasses	Contextual intelligence + voice assistant	Meta Ray-Ban, Brilliant Frame, Oshen	Multimodal LLMs, on-device AI

Market trends

- Shift from *graphics-heavy AR* → *intelligence-driven AI*
- Emphasis on *battery life*, *connectivity*, and *comfort*
- Forecast: consumer AI glasses > 10 M units by 2028

Transition: AR glasses focus on visuals; AI glasses focus on cognition. The sweet spot is ***spatial intelligence,*** a mix of both

Smart glasses as the next computing frontier

Pros

- Integrates a display
- Some are suitable for everyday use
- Long-lasting battery
- Some embeds a computer for stand-alone operation
- Take calls, hand-free (if available)
- Access to AI assistant (if available)
- Integrated Camera (if available)
- Integrated CPU (if available)

Cons

- Some offer ridiculously small displays
- Some are monochrome, which can be seen as a limitation.

Use-Cases

- Productivity tool
- Remote support
- HUD for real-life
- Acces to contextual information

Brands/Models

- Vuzix z100, Shield, Blade
- TCL RayNeo X2

https://medium.com/antaeus-ar/introduction-to-smart-glasses-2024-tech-review-d871e7d95cb8

Building smart glasses

- What do you need to build your own smart glasses?

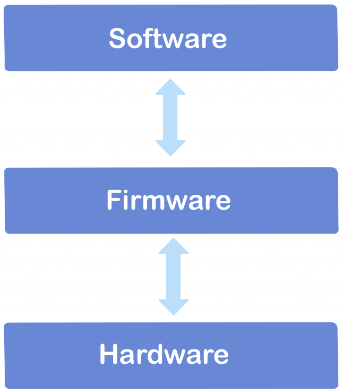
Building smart glasses

- System Architecture: Hardware → Firmware → Software

•**Hardware layer**
  •Sensors (RGB/Depth cams, IMU, mic array, touch pads)
  •Compute (SoC, memory, power IC, antennas)
  •Display (waveguide or micro-projector)

•**Firmware layer**
  •Sensor drivers + RTOS tasks
  •Power management loops & event handlers
  •Communication protocols (BLE/Wi-Fi)

•**Software layer**
  •OS (Android AOSP / custom Linux)
  •Middleware (OpenXR, custom SDK)
  •Applications: navigation, AI assistant, telepresence



Most of the glasses today have vertically tightly integrated full stack

Building smart glasses

- Design challenges

🧩 **1. Size and Form Factor**

•Must fit all components into < 15 mm temple thickness

•Optics and battery compete for space

⚡ **2. Power and Battery Life**

•Continuous sensing drains 300–500 mW

•Need efficient sleep states and power gating

🔥 **3. Thermal Management**

•Heat near temple area causes discomfort

•Passive spreaders and thermal simulation critical

🕶️ **4. Usability and Ergonomics**

•Weight distribution, nose bridge pressure, balance

•Must look and feel like everyday eyewear

Hardware components: Sensing & Input

- Types of Cameras
  - RGB cameras: Standard visual capture for AI perception
  - Stereo cameras: Two lenses for depth estimation and 3D mapping
  - Depth sensors (ToF / Structured Light): Direct distance measurement
  - Event cameras: Capture *changes* in light at microsecond precision
- Design Factors
  - Placement: typically in temple or bridge area
  - Trade-offs: Resolution vs. power, privacy vs. utility
  - Rolling vs. global shutter impacts motion artifacts

Sony IMX686 CMOS sensor

https://www.sinoseen.com/smart-glasses-mipi-camera-module-with-novatek-nt99141-auto-white-balance-hd

Hardware components: Sensing & Input

Microphones & Audio Array: Spatial Audio

Microphone Subsystems

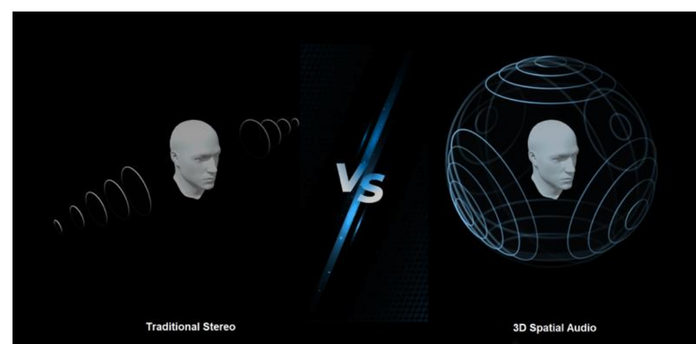
- Mono/Dual mic: Basic voice commands (e.g., ‘Hey Meta’)
- Far-field mic array: Beamforming and noise suppression
- Bone conduction mic: Captures speech via vibration, even in noise

Audio Output

- Open-ear speakers: Sound without blocking the environment
- Bone conduction speakers: low-leakage listening

Spatial Audio

- IMU + HRTF mapping recreates 3D sound positioning
- Enhances immersion for AR/VR experiences



Hardware components: Sensing & Input

Touch & Gesture Interfaces: Capacitive, Wrist, EMG

Touch and Gesture Inputs

- Capacitive touch pads: Swipe/tap on temple for control
- Wrist-based gestures: Detect muscle activity via radar or IMU
- EMG (Electromyography): Reads muscle signals for finger gestures
- In-air gestures — Optical hand tracking (limited by power and FOV)

Design Factors

- Minimal user friction — intuitive, no learning curve
- Robust in motion or noisy environments
- Always secondary to *voice and gaze* for hands-free UX

Hardware components: Sensing & Input

What other sensors could you think of adding to the glasses that would be useful?

Hardware components: Computation & Processing

- Main SoC (System-on-Chip), Common Platforms
  - Qualcomm: Flagship AR/VR chipset – multi-ISP, AI engine, 5G modem
    - XR2 / XR2 Gen 2 (headsets), AR2 Gen2 (glasses, RB-Meta)
  - TI OMAP / AM62x: Legacy low-power ARM SoCs used in embedded prototypes
  - ESP32 / ESP-S3: Wi-Fi + BLE MCUs for lightweight AI glasses or IoT control
  - nRF52 / nRF54 series: Ultra-low-power BLE controllers for wearables – Brilliant Labs Frame glasses use this
  - Silicon Labs MG24 series: Integrated BLE + ML accelerator for sensor fusion
  - Custom ASICs: Purpose-built NPUs for mesh/vision/audio compression

Hardware components: Computation & Processing

Any other co-processors?

- Sensor Fusion MCU: Aggregates IMU, camera, mic data
- AI Accelerators: Edge TPUs or NPUs for on-device inference
- Audio DSPs: Noise suppression and keyword spotting
- PMIC Microcontrollers: Manage battery and charging loops

Hardware components: Computation & Processing

Memory modules

Layer	Function	Example
On-chip SRAM	CPU/GPU cache (ultra-fast, few MB)	L1/L2 cache
LPDDR RAM	Active application memory	LPDDR4 / LPDDR5
Flash Storage	OS, firmware, AI models	eMMC / UFS
Off-chip SD / EEPROM	Logs or user data	MicroSD card

RB-Meta

- **Internal Storage:** 32 GB of flash storage
- **RAM:** 2 GB of LPDDR4x RAM

Hardware components: Computation & Processing

PCB Design – Layout & Stack-UP

- Form-Factor Constraints
  - Must fit within temple arms (< 10 mm width) and bridge section
  - Requires flex + rigid-flex PCBs to bend around curves
  - Typical board thickness: 0.4 – 0.8 mm

Typical Stack-Up (4 – 6 Layers)

$\boxed{1}$ Signal Layer 1 – High-speed lines (CPU↔Memory, MIPI)
$\boxed{2}$ Ground Plane – Continuous return for signal integrity
$\boxed{3}$ Power Plane – SoC core, I/O rails (1.1 V, 1.8 V, 3.3 V)
$\boxed{4}$ Signal Layer 2 – Sensors, I^2C, SPI, UART
$\boxed{5}$ RF Layer – BLE/Wi-Fi antenna feed
$\boxed{6}$ Ground Plane / Shield – Thermal and EMI return

Routing Challenges

- Trace length matching for DDR signals (±5 mil tolerance)
- RF isolation between Wi-Fi/BLE antennas and high-speed lines
- EMI shielding for camera and display lines
- Tight space (< 10 cm temple PCB) demands flex PCB folds

Hardware Components: Power & Connectivity

- Power Subsystem: Battery, Regulation, Current Draw

Subsystem	Active	Standby
Compute (SoC + RAM)	350 mA	< 10 mA
Sensors + MCU	80 mA	2 mA
Display	100 mA	0 mA
Wireless	150 mA	5 mA

Charging Methods

•**USB-C:** standard for data + charging (5 V, 1 A typical)

•**Pogo pins:** mechanical contacts in frames or docks

•**Magnetic dock:** safer for wearables, avoids connector wear

Hardware Components: Power & Connectivity

Most consumer AI glasses offload heavy computing to a paired phone via BLE or Wi-Fi Direct

Interface	Range	Power	Typical Use
Wi-Fi (2.4/5 GHz)	100 m	High	Streaming & updates
BLE 5.x / BLE Audio	10–30 m	Low	Phone pairing, sensors
Wi-Fi Direct / P2P	50 m	Med	Device-to-device sync
Cellular (LTE/5G)	Km+	High	Standalone connectivity

5G module – NBIoT (< 100Kbps)

Sequans GM02SP 5G modem on ESP32S3

Hardware Components: Power & Connectivity

Most consumer AI glasses offload heavy computing to a paired phone via BLE or Wi-Fi Direct

- **Antenna Placement & EMI Coexistence**
  - Small frame = limited antenna length
  - Metal hinges and batteries detune signals
  - Multiple radios (Wi-Fi + BLE + GPS + 5G) → mutual interference

5G module – NBIoT (< 100Kbps)

Sequans GM02SP 5G modem on ESP32S3

Hardware Components: Optics and Display

- **Optical Engines: Waveguide, MicroLED, LCoS, OLED**
- **Core Display Technologies**
  - **Waveguide + Micro-projector**
    - Light injected via coupler → guided → exits into eye
    - Lightweight, transparent, low FOV (~30°–50°)
  - **MicroLED:**
    - Ultra-bright, high efficiency, micro-scale pixels
    - Ideal for daylight visibility
  - **LCoS (Liquid Crystal on Silicon):**
    - Reflective display using polarization
    - Good color accuracy, limited brightness
  - **OLED (Organic LED):**
    - Self-emissive, high contrast
    - Common in early AR prototypes

Hardware Components: Optics and Display

- **Optical Engines: Waveguide, MicroLED, LCoS, OLED**
- **Core Display Technologies**
  - **Waveguide + Micro-projector**
    - Lig uid
    - Lig FO
  - **MicroL**
    - Ult mic
    - Ide
  - **LCoS (I n):**
    - Ref zat
    - Go brig

es

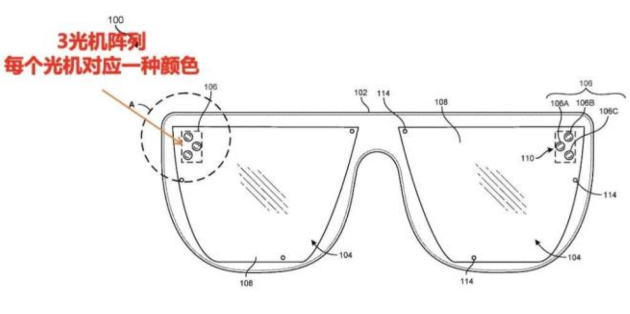
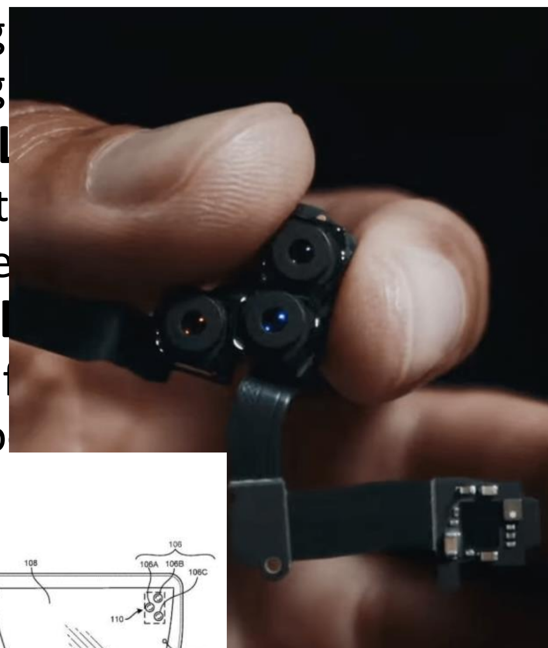


FIG. 1



Meta Orion: 3 microLEDs

Hardware Components: Optics and Display

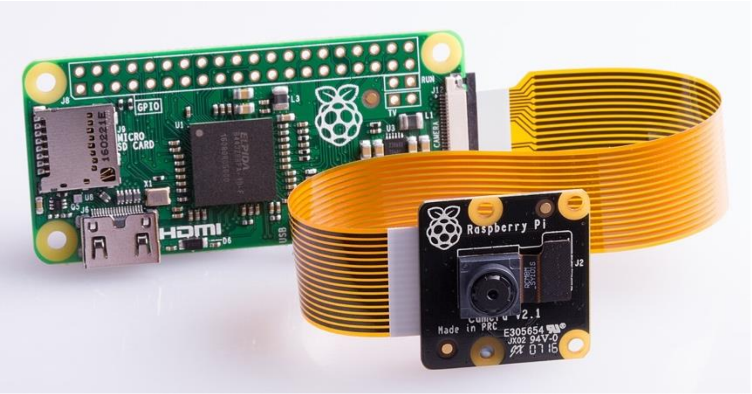
- Eye Comfort & Calibration
- Human-Factor Design
  - IPD (Interpupillary Distance): Align optical centers (typically 60–70 mm)
  - Vergence–Accommodation Conflict:
    - Eye focuses (accommodation) at screen distance
    - But vergence cues suggest object depth → fatigue
  - Solutions:
    - Varifocal lenses, light-field displays, or foveated rendering
  - Brightness Management:
    - Adaptive brightness for indoor/outdoor comfort
    - Maintain ≥ 1000 nits for outdoor readability

Firmware & Low-Level Software

- Boot Process & RTOS: From Power-On to App-Ready
- **Boot Sequence**
  1. Power-On Reset: PMIC stabilizes voltage
  2. Bootloader: verifies firmware and loads kernel
  3. RTOS init: creates tasks, interrupt priorities
  4. Driver init: camera, IMU, display, radio
  5. Main app launch: user UI or AI assistant
- **Real-Time Operating System (RTOS)**
  - Common choices: FreeRTOS, Zephyr, ThreadX
  - Task scheduling via preemptive round-robin
  - Inter-task sync with semaphores and queues

Firmware & Low-Level Software

- **Core Driver Modules**
  - Camera driver: controls exposure, frame buffer, MIPI interface
  - IMU driver: SPI/I^{2}C interface for accelerometer & gyro
  - Display driver: MIPI-DSI or SPI update loop with DMA transfer
  - Wireless stack: BLE, Wi-Fi, TCP/IP
  - Audio driver: I^{2}S streaming + noise reduction
- **Driver Design Facotrs**
  - Non-blocking IO, interrupt-driven architecture
  - Unified API layer for upper-level apps
  - Power state awareness



Firmware & Low-Level Software

- Low-power firmware design
  - Event-driven loops instead of polling
  - Fine-grained power domains (SOC, RF, sensors)
  - Tickless RTOS scheduling
  - Use DMA to avoid CPU wake-ups
  - Cache sensor frames in RAM before processing

Example: Wake-on-Voice Pipeline
DSP listens for keyword → triggers main MCU → boots AI engine

Firmware & Low-Level Software

- Most of the embedded devices do not have displays

Debug Interfaces
•**UART Logs:** printf style for early boot visibility
•**JTAG / SWD:** step through firmware line-by-line
•**Serial Console:** interactive command shell
•**Logic Analyzer / Oscilloscope:** timing verification

Software Architecture

- Operating system stack

Platform	Pros	Cons
Android XR	Mature ecosystem, OpenXR ready	Heavy for low power MCUs
Embedded Linux	Flexible driver model	Requires more RAM
Bare Metal + RTOS	Fast boot, ultra-low power	Limited app ecosystem

Proprietary – Mentra OS

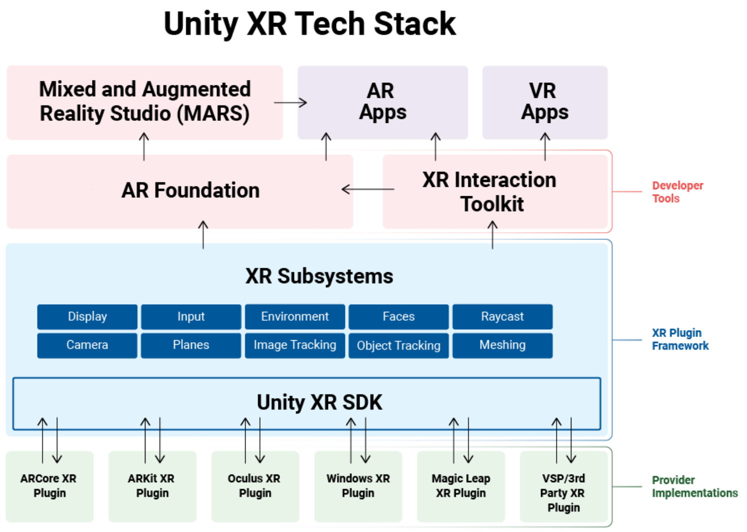
Software Architecture

- **Middleware for XR & AI Integration**
  - **Key Frameworks**
    - **OpenXR:** Industry standard for cross-device XR apps
    - **Academic platforms: ARENA / ILLIXR:** Open-source research stacks for AR/VR systems
    - **Custom SDKs:** OEM optimized pipelines for low latency – we will see one in homework 4
  - **Functions**
    - Sensor abstraction & spatial tracking
    - Scene graph management
    - Input (unified voice, gesture, gaze)

Middleware turns raw sensors into usable spatial data

Software Architecture

- **Developer Choices**
  - **Unity / Unreal Engine:** 3D content rendering, XR SDK integration
  - **WebXR / WebGPU:** Lightweight, cross-platform XR apps via browser
  - **Native C++ / Java / Kotlin SDKs:** Custom low-level control
- **App Lifecycle**
  - Sensor input → AI perception → AR rendering → User interaction
  - Background tasks handled via services for always-on operation



Example User Interactions

- **On-Device AI: Intelligence at the Edge**
- **Key Functions**
  - Voice recognition, object detection, translation
  - Summarization & question answering (micro-LLMs)
  - Visual captioning & scene understanding
  - Activity classification (walking, typing, driving)
- **Hardware Examples**
  - Qualcomm Hexagon NPU, Edge-TPU, ESP32-S3 AI engine
  - Quantized models (<100 MB) optimized for real-time use

Example User Interactions

- **LLM & Multimodal Integration**

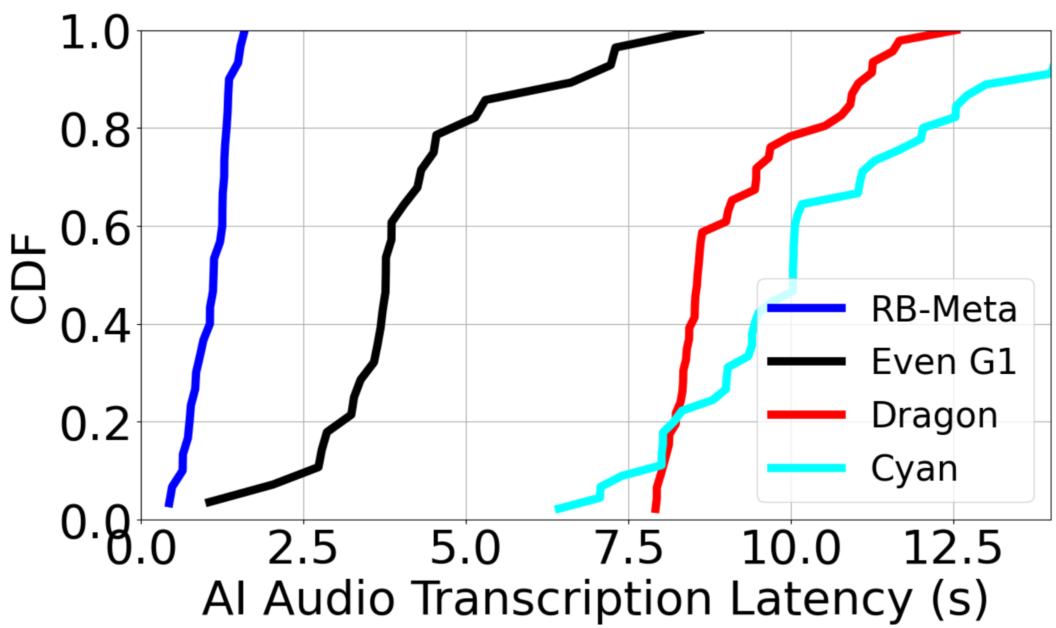
1	Capture: Camera/mic/IMU feed
2	Pre-process: Compress & tokenize locally
3	LLM inference: Edge or cloud model
4	Response: Text-to-speech, overlay

- **Trends**
  - **Multimodal LLMs (GPT-4V, Gemini, Claude)** process image + audio
  - **Context windows** personalized to daily usage
  - **Hybrid execution:** small local model + remote reasoning

An early study of smart glasses performance

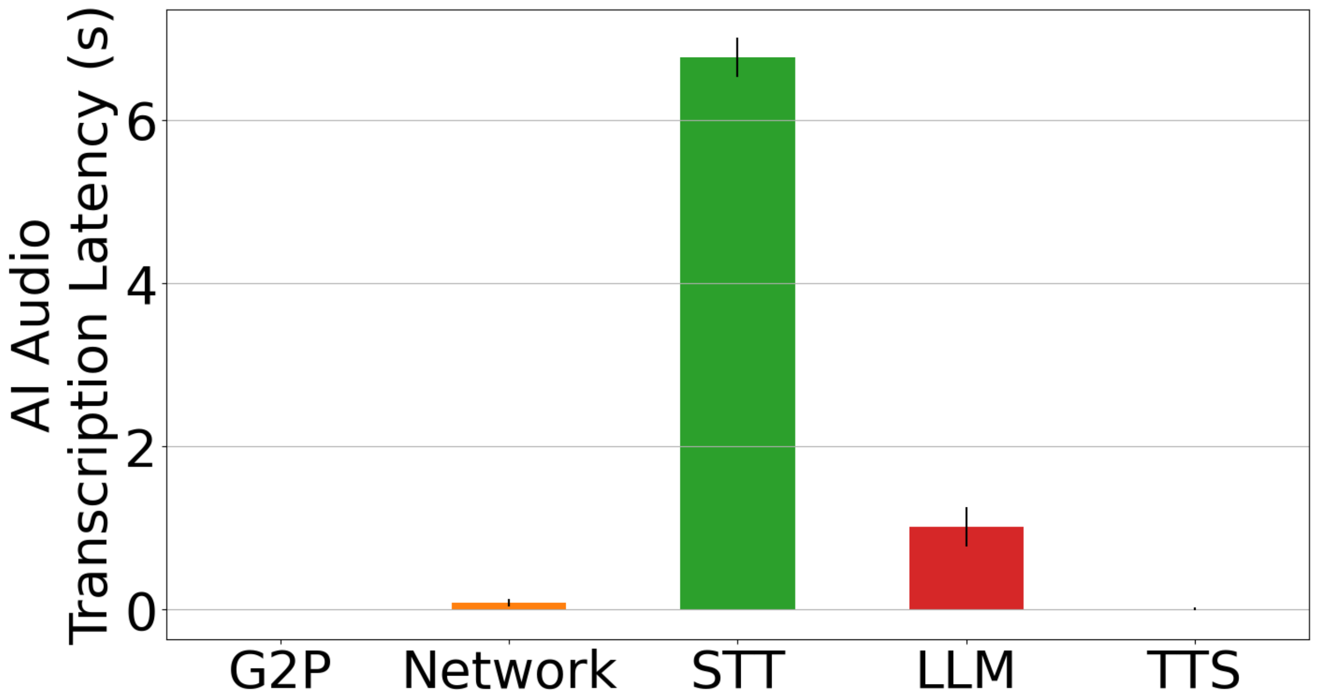
We studied some of these glasses across a range of metrics

An early study of smart glasses performance



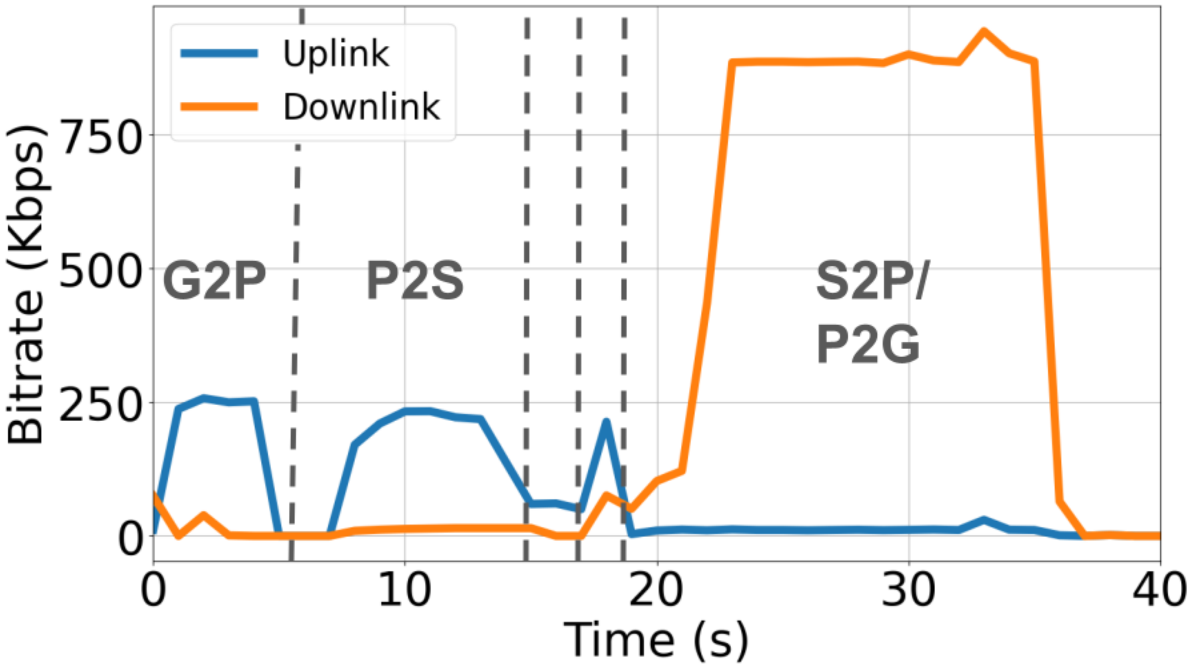
Meta glasses have the least interaction latency

An early study of smart glasses performance



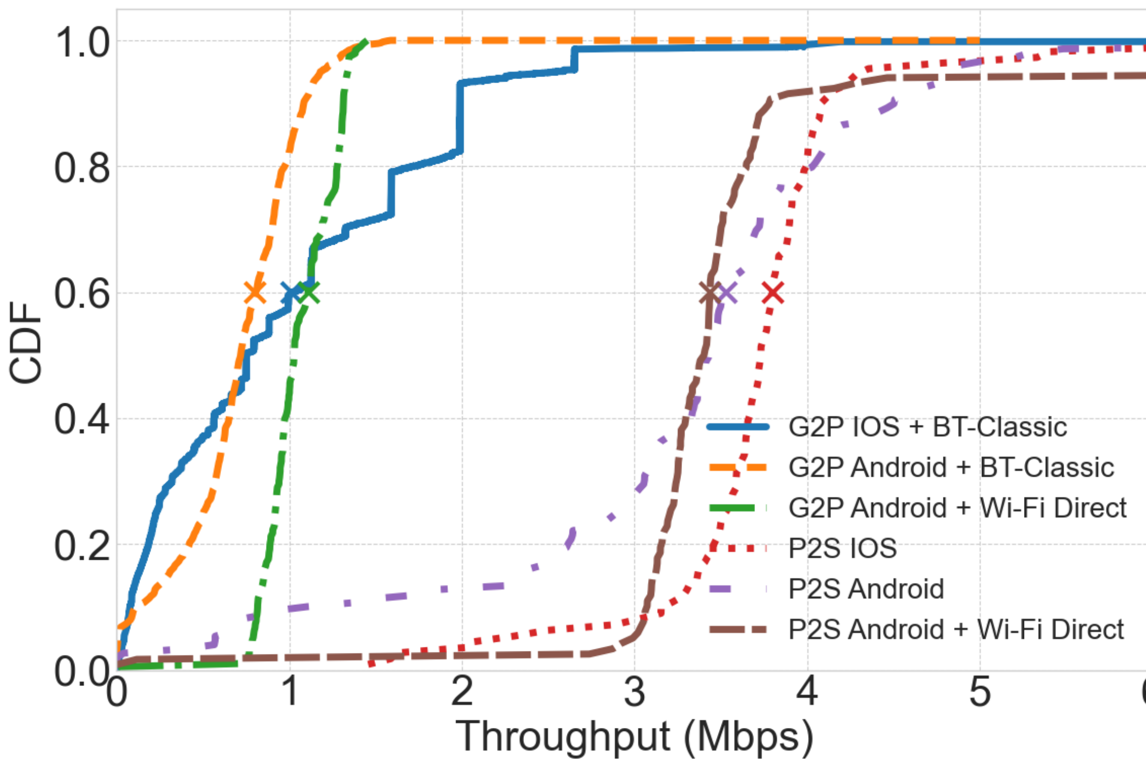
Breakdown on Cyan Glasses – with Deepgram AI models

An early study of smart glasses performance



A timeline of AI voice interaction and the corresponding bitrate

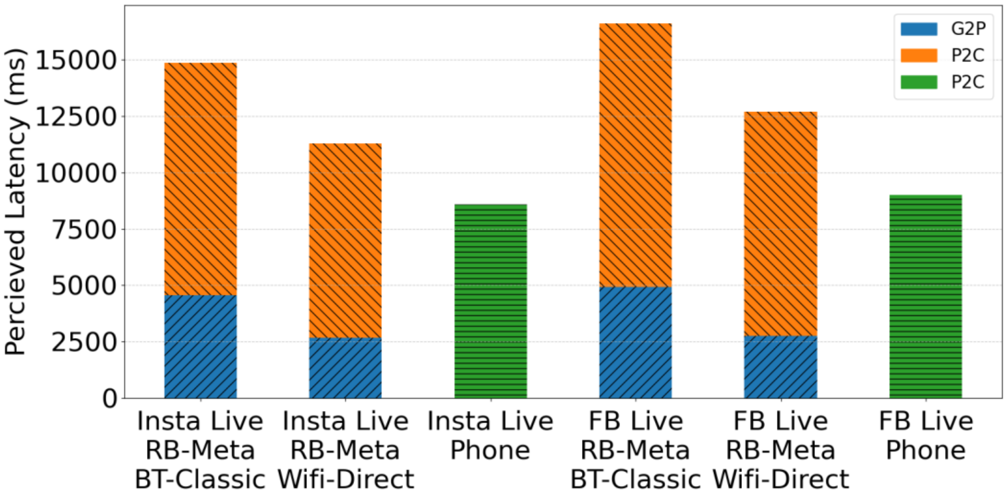
An early study of smart glasses performance

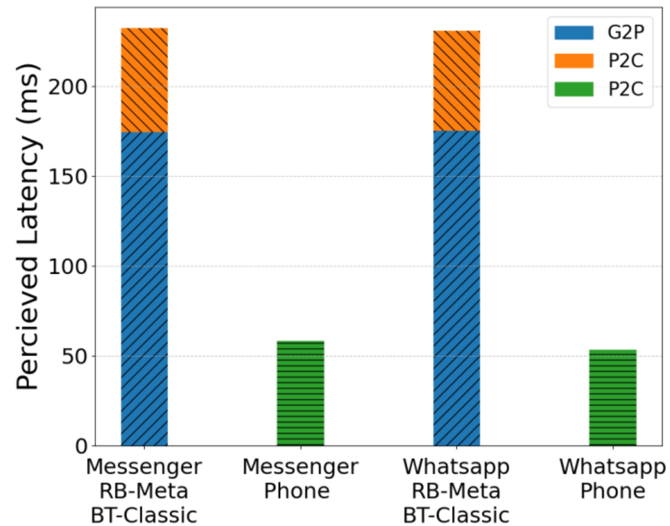


Bitrate/Throughput analysis across G2P, P2S on Android vs. iOS and BT vs. Wi-Fi

Live video streaming

An early study of smart glasses performance





G2P is the key bottleneck in Messenger and WhatsApp while P2S is the bottleneck for insta and fb

Summary of the Lecture

- Smart glass hardware
- Smart glass software

Next up: Basics of Rendering