OpenRPT: Open Remote Physical Therapy

Ziliang Zhang, Louis Lu, Dr. Hung-Wei Tseng,

ESCALAB, 2021 UCR

ESCAL
Extreme Storage & Computer Architecture Lab

UC RIVERSIDE

Outline

- Background
- Why Building OpenRPT
- Interaction Design
- Implementation Details
- Evaluation and Future Work
- Demo Q&A

Background

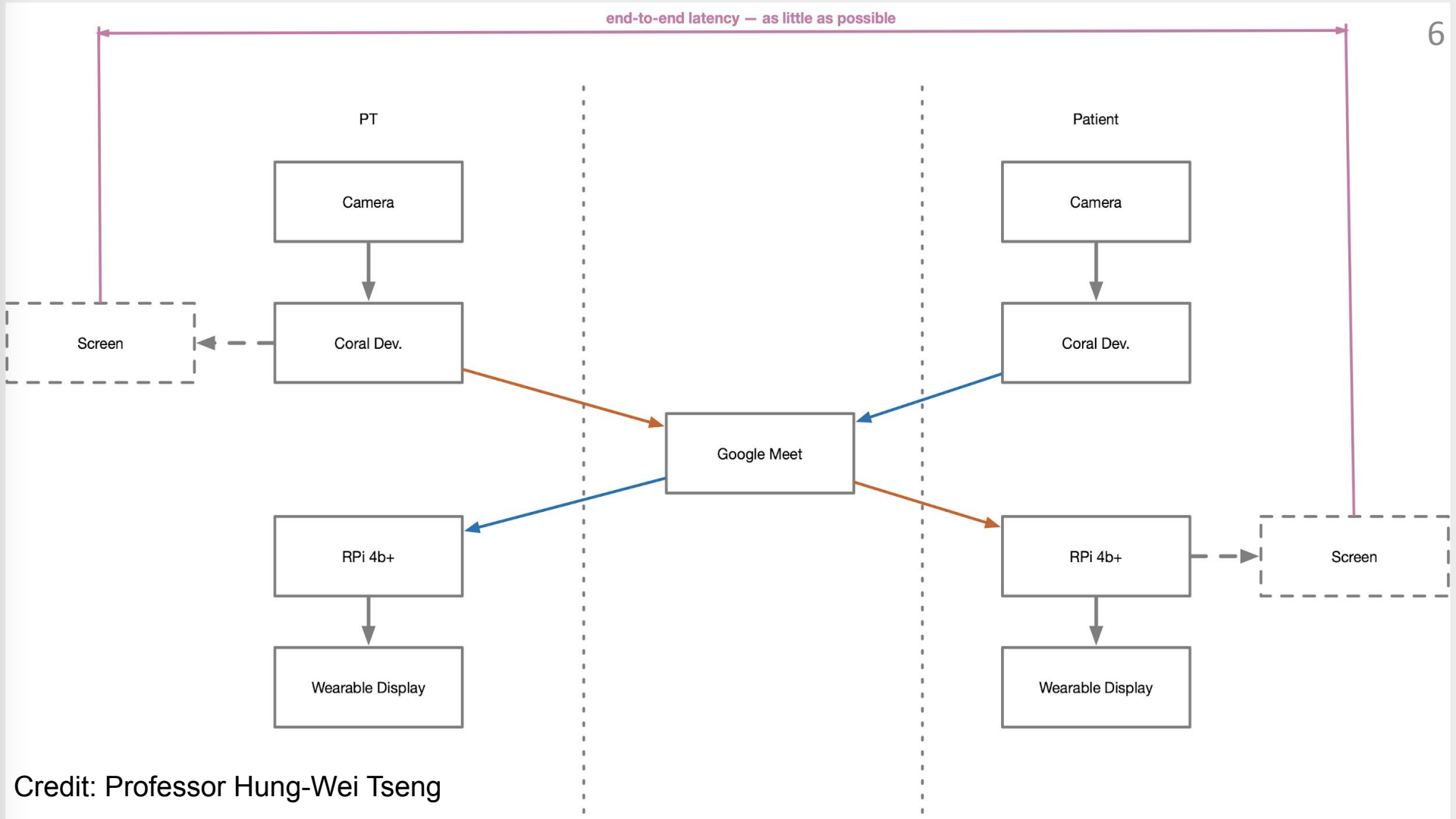
- 2020 Pandemic Constraints: needfinding remote rehabilitation training
- In-Home Therapy Sessions Problems:
  - Patients need to hold phones and talk to the therapist simultaneously
  - Therapists can only measure angles and positions with eyes and experience

Why Building OpenRPT

- Why we set these goals
  - Users need to have a natural speech environment synchronized with body motions
  - Patients want to know what the therapists are posing, and can pose accordingly.
  - Therapists want to know how the patient pose with precise readings of key angles and displacement in body parts.

Why Building OpenRPT

- Latency: 150ms in single round-trip
- Patients Display: Therapist's pose with skeleton - need to know what pose they need to perform.
  - Easy to Setup
  - Mobility
- Therapists Display: Patient's pose with skeletons and angles and distance measurement
  - If the patient performs correct pose.
  - Records/measures the patient's improvements



Credit: Professor Hung-Wei Tseng

Interaction Design

- Why we make the design decisions
  i. Google Coral DevBoard
  ii. Coral DevBoard capture Camera Feed Directly from wired camera
  iii. Raspberry pi receiving video feed from other side DevBoard
  iv. Raspberry pi holding a server for user control
  v. Raspberry pi output Google Meet Feed to wearable display
  vi. Using a on-glasses wearable display for movement

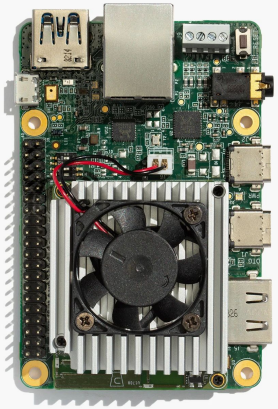
Interaction Design

- Limitation of Coral DevBoard
  - The DevBoard's Memory is 1GB
  - Camera module compatibility is low

- Limitation of Raspberry Pi 4
  - The Raspberry pi 4 is slow with on-board ML
  - Has to connect extra modules with Rpi4 to produce same quality of ML - power hungry
  - Has no built-in speaker and microphone to receive audio

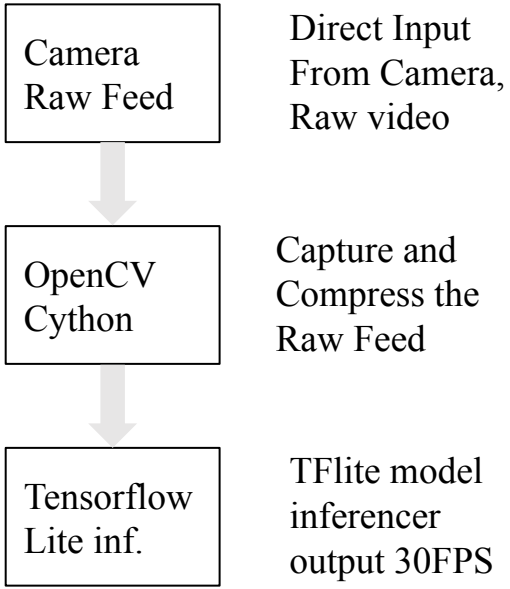
Implementation Details

- Coral DevBoard (**1GB**)
  - Has a camera direct connection to the board
  - Process Input Camera Feed using Cythonized code and output Pose Estimation video to v4l2loopback device
  - Model: mobilenet quant decoder 720p and 480p
  - v4l2 device directs the video feed to headless chromium, which will launch Teleconferencing Software without an external monitor
  - Teleconferencing Software carries the video to other side



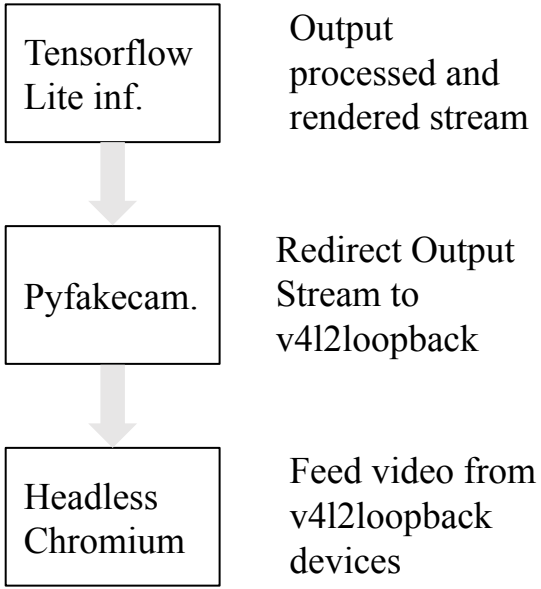
Implementation Details

- Software Stack
  - Reason for use OpenCV and TFlite
  - Input Part - OpenCV Cython: raw video to H264
  - Processing Part - Cythonized Tensorflow lite inferencer: mobilenet TFlite model to produce 30FPS



Implementation Details

- Output Part - Pyfakecamera Python redirect to v4l2loopback device
- Headless Chromium: receives input from v4l2loopback device, launching without using external monitor.



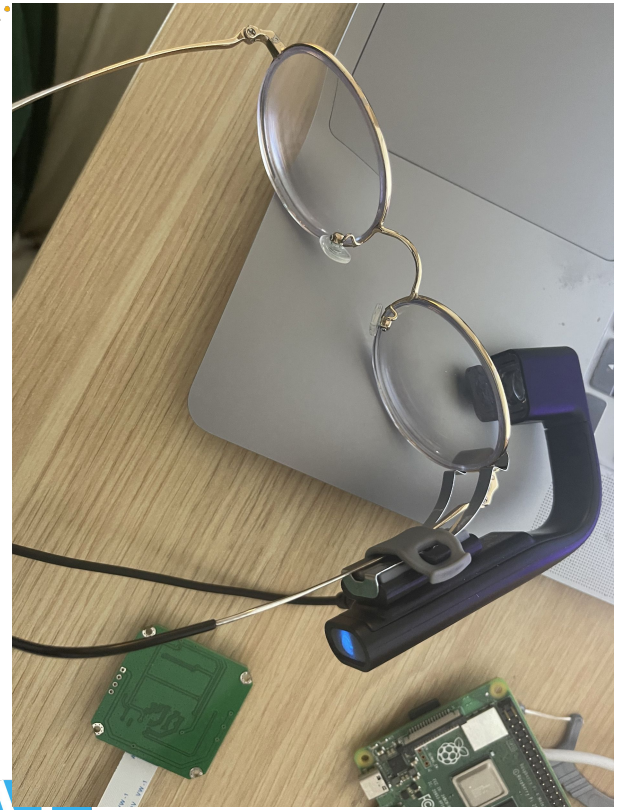
Implementation Details

- Raspberry Pi (4GB)
  - Initial Setup with Bluetooth PAN Wifi from the user computer or smartphone, propagate the Internet Connection by using PAN to DevBoard
  - Server to the router app (frontend, written in flask), receive user credentials and request, launch Google Meet on both Raspberry pi and Coral DevBoard

Implementation Details

- Teleconferencing Software (Google Meet)
  - Can carry the video feed from different sides devBoard. Only output to the different side raspberry pi
  - Has an encrypted connection that is private to therapist and patient
- Wearable Display
  - Small on-glasses monitor with 4-inch size
  - Battery powered, getting input from raspberry pi

Implementation Details

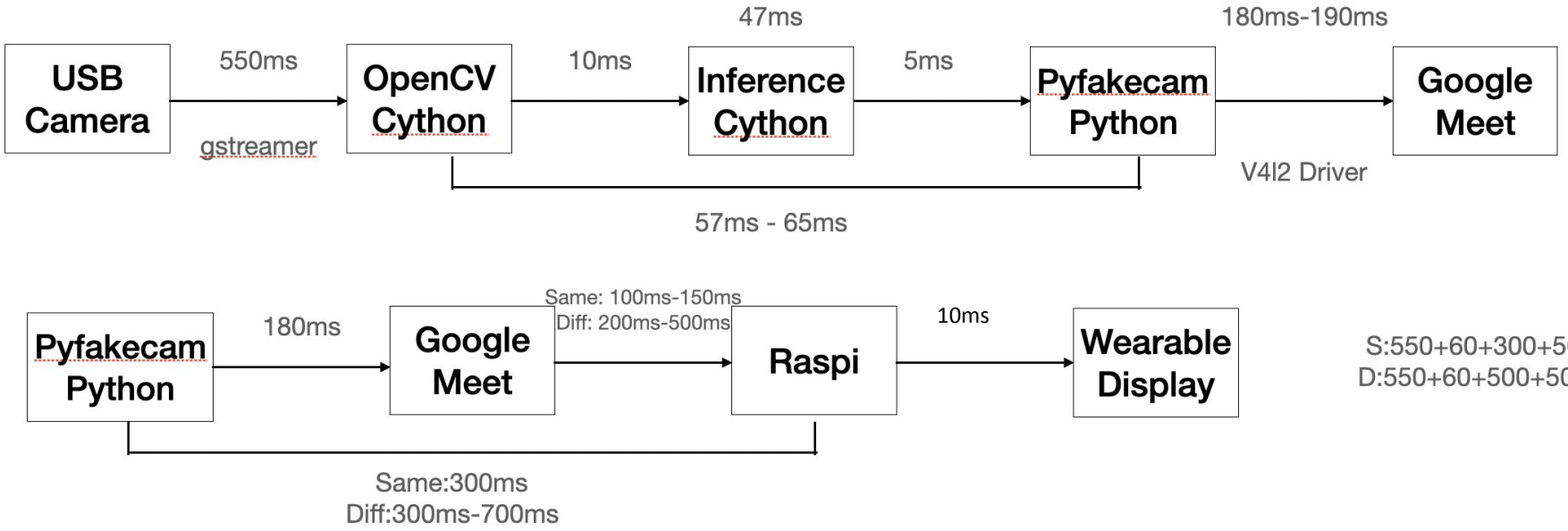


Evaluation

- Experiment Environment:
  - Homebase Wifi Connection 100mb/s speed, user device MacBook Pro 2019, Communicating to another homebase wifi environment
  - USB camera with 720p video recording capabilities
- Video Quality:
  - Raw Video captured: 1280x720 raw video, compressed to H264 format
  - Transmitting video quality option: 1280x720, 640x480
- Audio Quality:
  - Mic from raspberry pi. Earphones using the AUX jack on the raspberry pi 4

Evaluation

- Latency Breakdown (real scenarios 1160ms)



Future Work

- System Improvement - overheat problem if left on for more than 2 hour
- Latency reduction
  - Target: 150ms single trip according to ZOOM guideline
  - Potential improvement: camera module and video redirection, inferencing using Cython module (or C++)

Demonstration

- New Setup Walkthrough
- Live Demo Chat Session

Questions?

Reference

[1] Gonzalez-Franco M, Gilroy S, Moore JO. Empowering patients to perform physical therapy at home. Annu Int Conf IEEE Eng Med Biol Soc. 2014;2014:6308-11. doi: 10.1109/ EMBC.2014.6945071. PMID: 25571439.

[2]Google Coral PoseNet, https://github.com/ google-coral/project-posenet

[3]v4l2 device Loopback, https://github.com/ umlaeute/v4l2loopback

[4]Sharing Internet by Bluetooth, https:// www.raspberrypi.org/forums/viewtopic.php? t=223029#p1400465

[5]Coral DevBoard, https://coral.ai/

[6]Headless Chrome/ Headless Chromium, https://chromium.googlesource.com/chromium/ src/+/lkgr/headless/README.md

[7]Flask, https://flask.palletsprojects.com/en/ 1.1.x/