UsAR Mirror

Design Presentation

Steven Lee, Anna Paek, **Shengxi Wu**

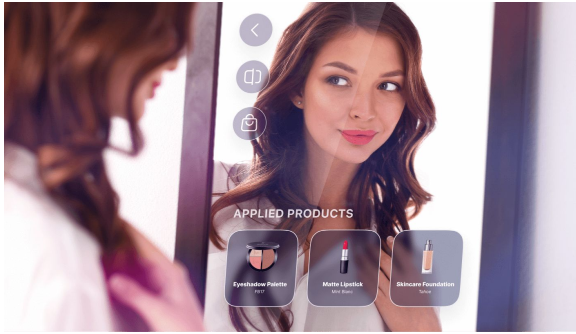
Use case

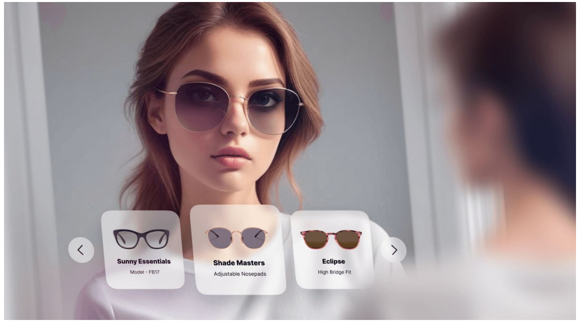
Problem:

- Applying makeup is time-consuming, and there’s no way to preview the final look before application.
- Mirrors provide a limited view, making it difficult to see the sides of the face.
- Existing apps apply 2D filters, which don’t adapt well to different angles or facial movements.

Solution:

- An AR-enabled display that uses RGB and depth sensors to generate real-time, 3D makeup previews.
- Dynamic rendering allows users to view themselves from multiple angles.
- Gesture recognition enables hands-free control, allowing users to adjust makeup styles, switch views, and navigate the interface effortlessly.





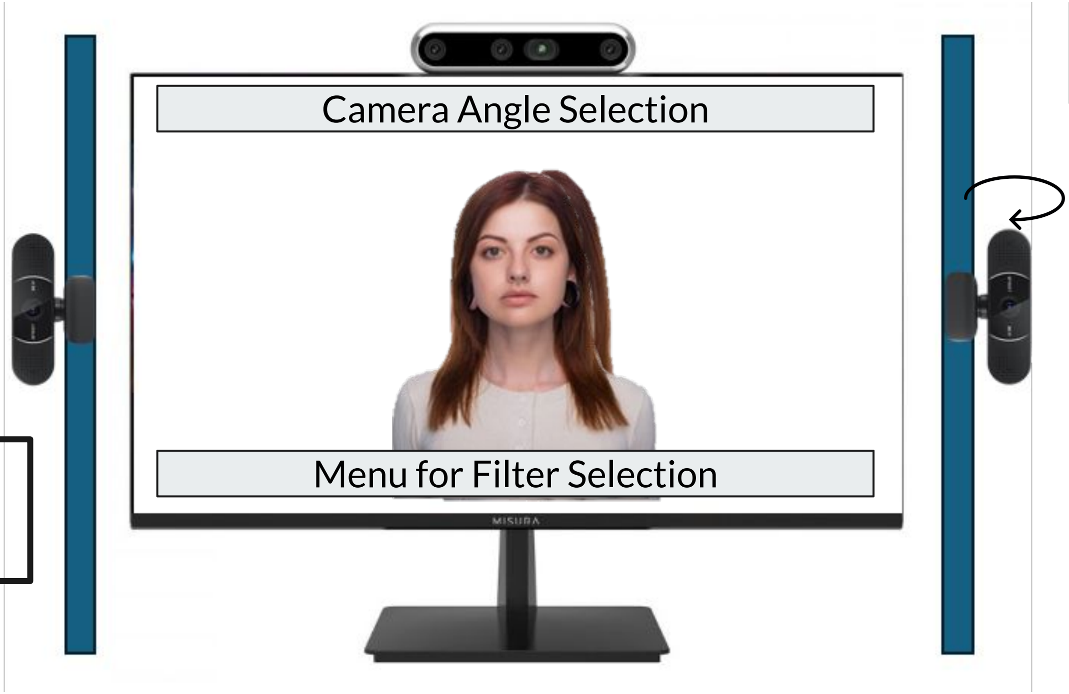
Use Case Requirements		Hardware	Software	Design Requirements
1	Accurate Interaction in real time	✓	✓	• ≤ **200 ms delay** in camera movements, making selection • ≤ **1s** delay to generate 3D face model per user • ≥ **15 FPS** for displaying AR Filters • **2%** deviation for AR Filter against head position
2	Freely select target view of themselves	✓		• Arduino, stepper motor, rotary push button for camera control • Up/down ≤ **11.8 in** (~width of display) • Pan/rotate ≤ **90 degrees** • ≤ **5 degrees** deviation from desired angle
3	Screenshot and save photos of themselves on the display		✓	• Include a “capture” option to take screenshots • Screenshots saved automatically to a default directory (or user has an option to make one) • PNG or JPEG for image quality
4	Navigate the menu or make a selection using hand motions		✓	• Swipe **up/down/left/right** • ≤ **200 ms delay** • Detection range of **0.5-2 m** • **90%** accuracy in gesture cue detection

Solution Approach

3D Face Reconstruction
*To support **accurate multi-angle** rendering*

User Interface (UI)
Enhances usability & accessibility

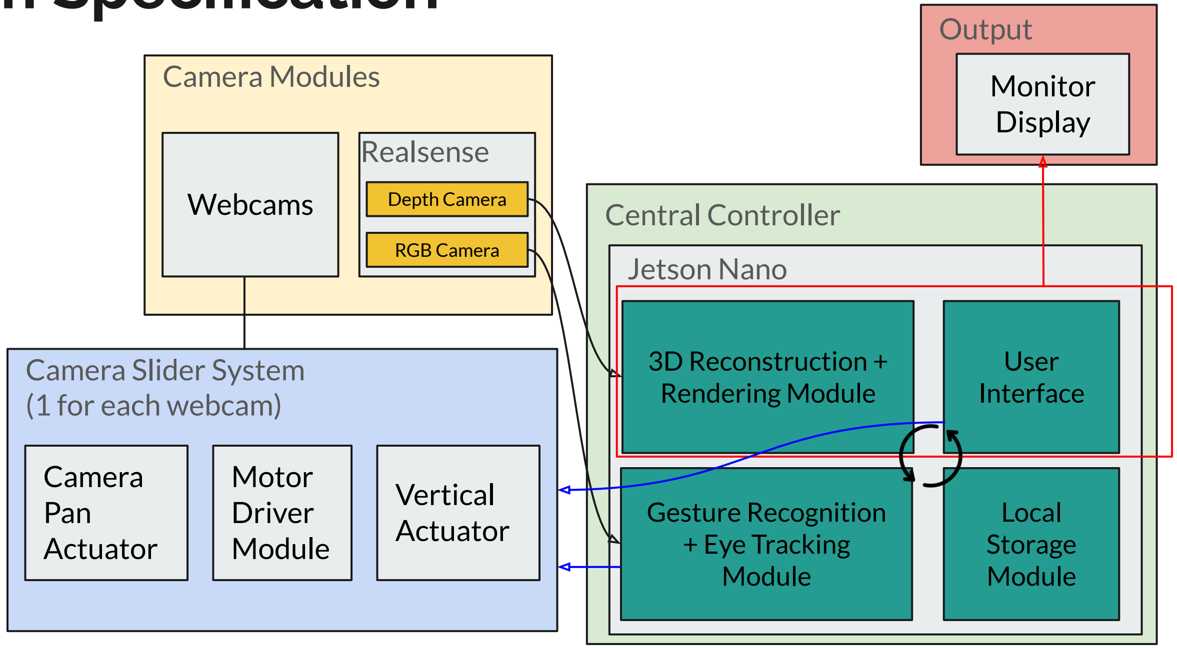
Gesture Recognition
***Touch-free interaction** - improves safety in public space*



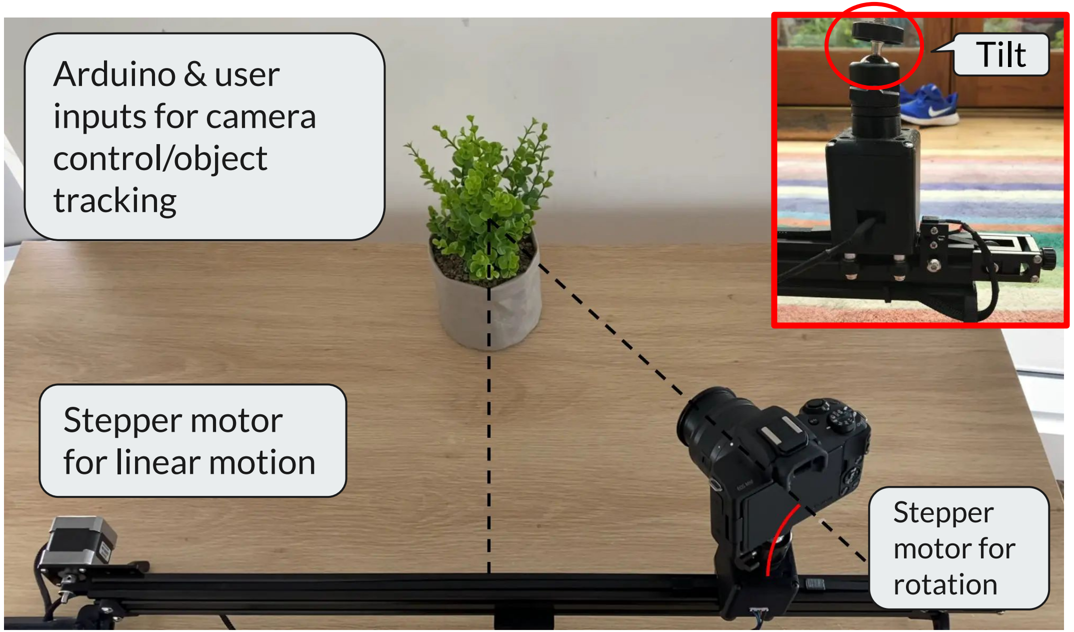
Camera Rig
Multi-angle tracking** supports **better accessibility** for individuals with **limited mobility

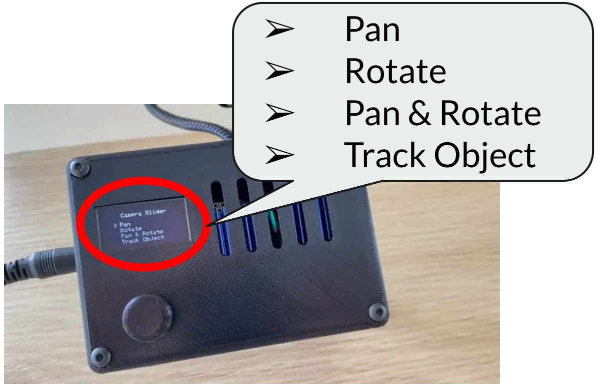
Eye level matching
Reduce distorted perspectives - ensures Natural & Comfortable Viewing

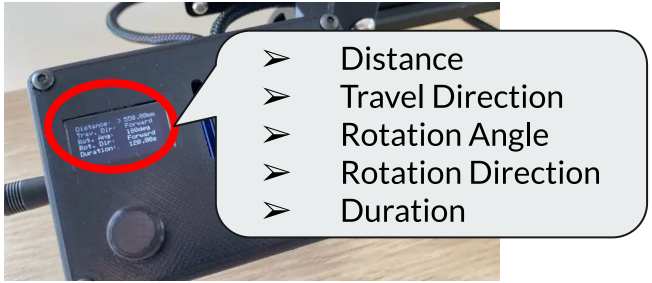
System Specification



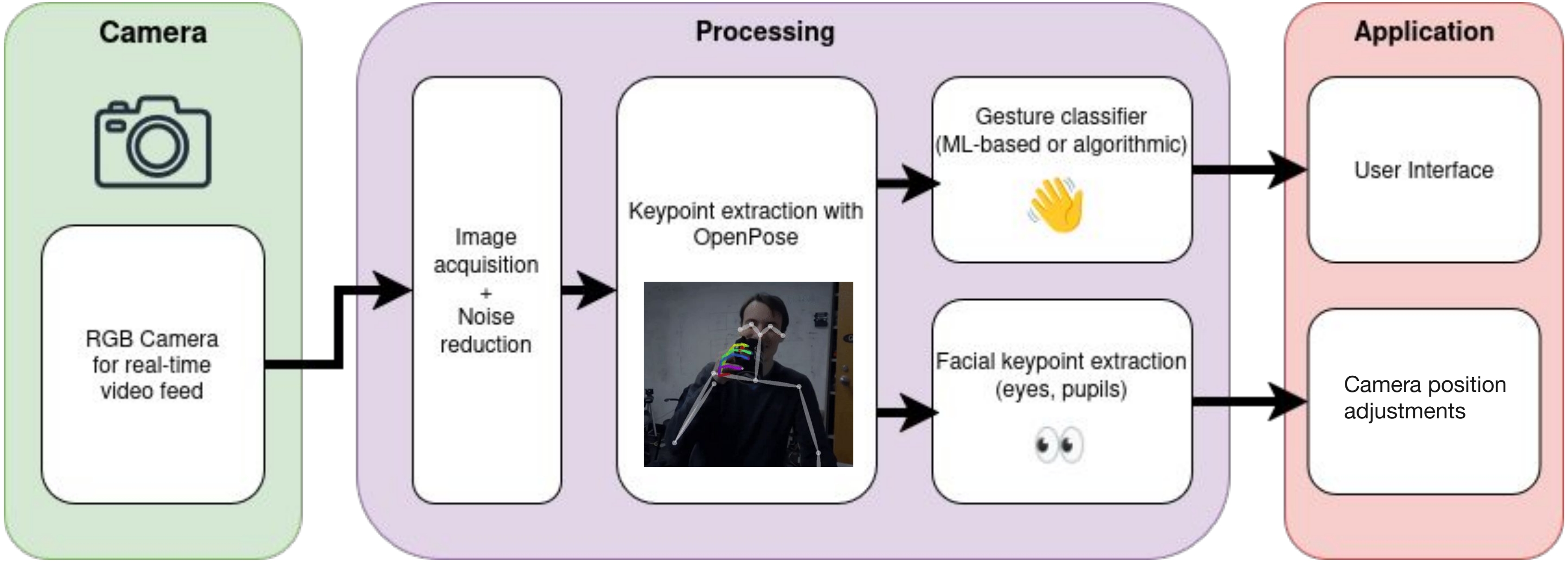
System Specification



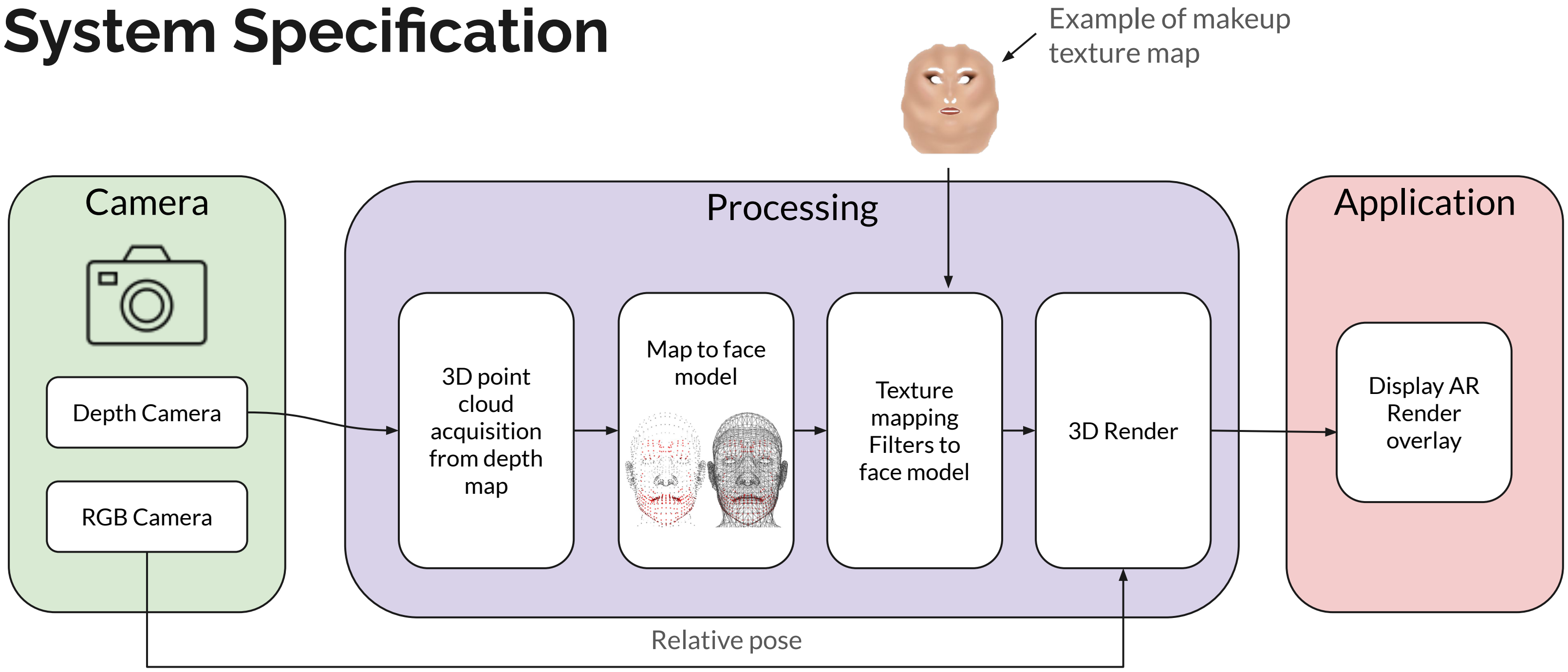




System Specification



System Specification



Implementation Plan

Things we buy:

- 23.8 Inch Monitor
- Mechanical Hardwares
- Stepper Motors
- Arduino
- Webcam *2
- Realsense Camera*
- Nvidia Jetson Orin Nano* (from Inventory*)

Things we Depend on:

- Libraries & Tools:
  - OpenCV
  - Open3D
  - OpenGL
  - OpenPose
- Mechanical Design for Camera Slider
- Filters

Things we Design & Develop:

Hardware & System Integration

- Multisystem Interfacing
- Arduino-Controlled Camera System

Perception & Interaction

- Gesture Control System
- Eye Tracking Feedback System

3D Processing & Rendering

- 3D Face Reconstruction
- AR Rendering Pipeline

User Interface

Test, Verification, Validation

Accurate Interaction in real time

Test Input:

- Continuous real-time camera feed
- Gesture input

Expected Output:

- **≤ 200 ms** delay in camera movements, making selections

Test input:

- Continuous depth map feed
- User head motion

Expected Output:

- **≤ 1s** delay to generate 3D face model per user
- **<< 200 ms** delay in identifying 6DoF head transform
- **≤ 5mm RMSE** for 3D reconstruction

Risk management:

- Include gesture cue to reconstruct face model

Test input:

- Continuous real-time camera feed
- Filter selection

Expected Output:

- **≥ 15 FPS** for displaying AR filters
- **≤ 2 mm** drift over movement range
- **≤ 2°** deviation compared to detected headpose

Test, Verification, Validation

Screenshot and save photos of themselves on the display

- **100%** success rate of screenshots saved
- Verify file creation & image quality

Risk management:

1. Screenshot commands & file creation
2. File I/O failures or storage issues → error handling/warnings

Freely select target view of themselves

- **95%** accuracy of camera orientation & frame stabilization
- **≤ 5 degrees** deviation from desired angle

Risk management:

1. User commands for camera vertical motion, rotate, pan,
2. Mechanical or software-based misalignment → feedback loop for auto-correction

Navigate the menu or make a selection using hand motions

- Clearly distinguishes between **up/down/left/right**
- Detection range of **0.5-2 m**
- 90% accuracy in gesture cue recognition

Risk management:

1. Series of hand gestures under various conditions (include reverse action)
2. Include mouse control in case gesture recognition fails

Project Management