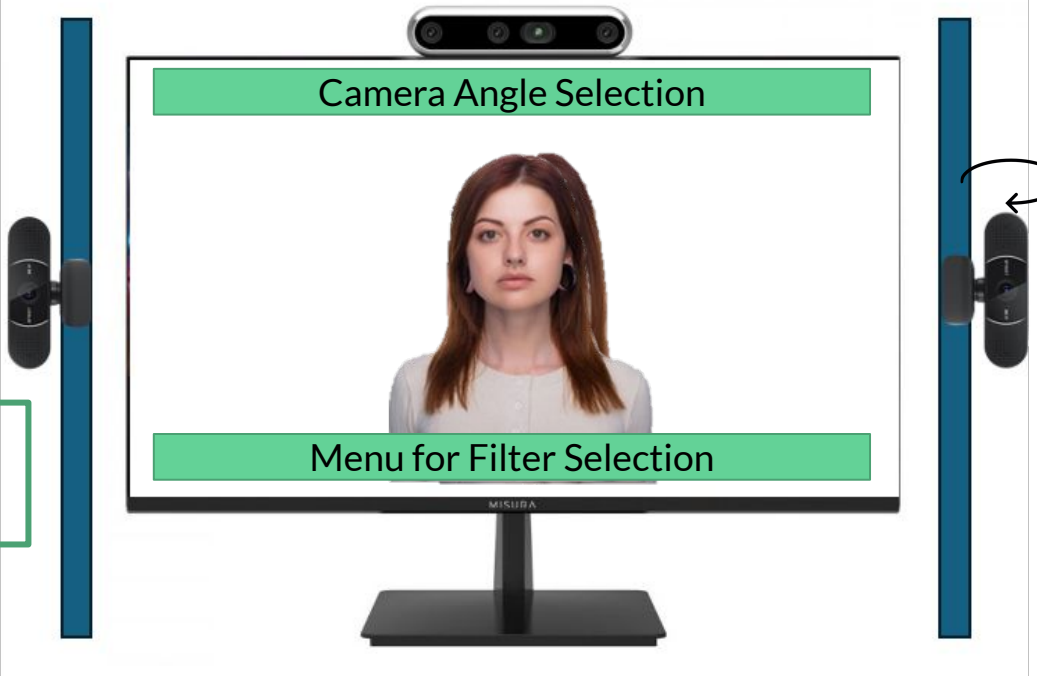
Use-Case & Design Requirements

Use-Case Requirements	Design Requirements
Accurate interaction in real time up to 200 ms delay	• **≤ 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
Freely select target view of themselves within 90 degrees	• 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
Screenshot and save ~100 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
Navigate the menu or make a selection using 4 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*

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



User Interface (UI)
*Enhances **usability** & **accessibility***

Eye level matching
***Reduce distorted perspective** - ensures natural & comfortable viewing*

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

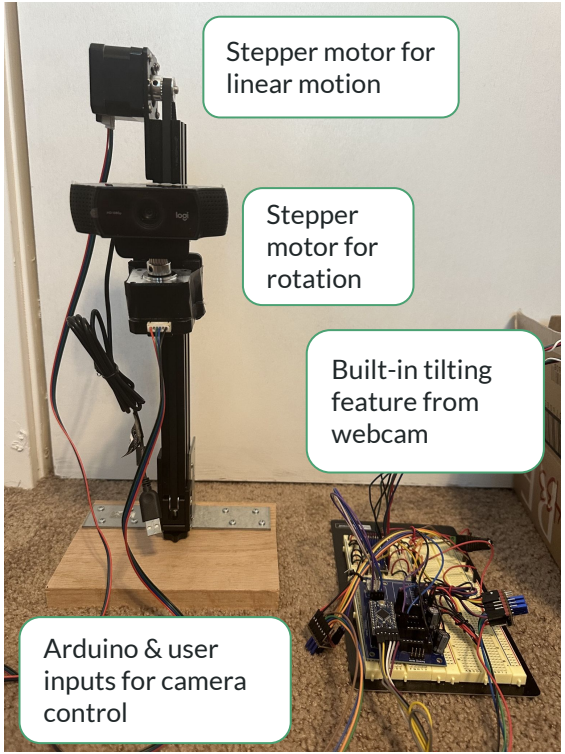
Impact of Solution Approach

Public Health, Safety, Welfare	Social Factors	Economic Factors
• **Touch-free interaction**– reduces the spread of germs • **Eye-tracking**– minimize eye strain & fatigue	• Transforms are accurate & image filters are reasonable as possible– **not promote unrealistic beauty standards**/negative self-perception	• **Within $600** • Simple camera control system design • Easy to carry, package, ship
Global Factors	**Cultural Factors**	**Environmental Factors**
• **Robust** and **secure**– adheres to industry standards • Able to deploy in different regions	• Considers **different religious values** in makeup styles	• **Affordable** & **reusable**– commonly found parts (off-the-shelf computer monitors & webcams) and basic depth sensors

Complete Solution – Gesture Input + Camera Control

Camera command

“ru 100”



Complete Solution – 3D Rendering + Filter

Without makeup filter

With makeup filter

Gesture Recognition Testing Metrics

Metric	Measured Result	Target / Expectation	Status
Pose Recognition Latency	**~25 ms** (from camera feed -> pose)	≤ 200 ms delay	✅ Fast
Input Accuracy	**≥ 95% correct** (in all possible inputs)	≥ 90% correct	✅ Accurate
Keypoint Estimation Error	**~35 px** (max error from stationary pose)	Target: ≤ 20 px	⚠️Noisy

Gesture Recognition Trade-Offs

Feature / Metric	Position-based inputs	Velocity-based inputs
Noise	⚠️ Noisy keypoints due to OpenPose limitations	❌ Much noisier velocities calculated from keypoints
Latency	✅ Low latency, use keypoint estimates as-is	❌ Needs additional filtering from keypoint estimates
Robustness	✅ Position based button input more robust to noisy pose estimates	❌ Difficult to denoise velocities of keypoints
Ease of Use	✅ Better visual feedback (draw position on screen), simply hover over button.	❌ Difficult for user to perceive gesture velocities

Camera Control System Performance

Metric	Measured Result	Target / Expectation	Status
Speed (steps per second)	**48 steps/sec**	50 steps/sec	⚠️ Slightly lower due to load
Distance per step (inches)	**0.031 inches/step**	1/32 inches/step	✅ Meets the requirements
Maximum motor current	**1.1 A**	1.2 A	✅ Safe operating range
Full travel time (up & down)	**12 sec**	10 sec	⚠️Slight delay due to friction
Camera accuracy (degrees)	**3.7 degrees/step**	3.75 degrees/step	✅ Within expected precision

Camera Control System Design Trade-offs

Design Element	Choice/Approach	Pros	Cons
Stepper Motor Driver	TMC2209 instead of TMC2208	Quieter, smoother operation	Requires rewiring + $6 more expensive
Control by Ticks vs. Position	Control by Ticks (Steps)	Less complex integration between gesture recognition and camera system	Less smooth movement and flexibility in choosing position
Speed vs. Precision	Slower Motor Speed for Smoothness	Smoother and more precise motion	Takes longer to move the camera

AR Filter + Rendering Performance Metrics

Metric	Measured Result	Target / Expectation	Status
3D Face Model Generation Delay	**~20 ms** (sparse landmarks)	Target: ≤ 50 ms	✅ Fast
6DoF Head Pose Identification	**~ 2 ms** (solvePnP + depth for 68 points)	Target: ≤ 150 ms	✅ Fast (with very sparse point cloud)
AR Filter Rendering Frame Rate	**~160 FPS**	Target: ≥ 15 FPS	✅ Real-time
Drift Over Movement Range	**Some visible jitter, ~3–5 px**	Target: ≤ 5 px	✅ Minor drift
Pose Estimation Error	**~10-15 px**	Target: ≤ 5 px	⚠️Less stable

↑
Influenced by calibration error

AR Filter + Rendering Trade-offs

Feature / Metric	dlib	OpenCV DNN + LBF	MediaPipe Face Mesh
Detection Backend	HOG or CNN (CPU)	OpenCV DNN (with LBF regressor)	BlazeFace (GPU-accelerated)
Landmark Output	68-point	68-point	468-point dense mesh
Performance (Jetson)	❌ CPU-bound (~4s)	⚠️ Partial GPU (~20ms)	✅ Fully GPU (likely faster)
Robustness (Pose/Lighting)	Good (CNN)	Limited	Excellent under varied conditions
Ease of Integration (C++)	Easy	Easy	⚠️ Complex (graph-based, Bazel build)
Customizability	✅ Full access to data and flow	✅ Simple to inject into any pipeline	❌ Difficult — GL/stream sync difficult (easier with Python API)

Project Management

- Need to work on: UI, integration, testing