ECE 445

Senior Design Laboratory

Final Report

BetaSpray: A Laser-Based Climbing Route Visualization System

ECE 445 Senior Design — Final Report

Team 41

Ingi Helgason Prakhar Gupta Maxwell Beach

TA : Gayatri Pushpita Chandran

Department of Electrical and Computer Engineering
University of Illinois Urbana-Champaign

Spring 2026

Abstract

BetaSpray is a standalone, portable device that converts any spray-style climbing wall into an interactive training tool. The system scans the wall using a 5 MP OV5640 camera mounted on a custom two-layer PCB driven by an ESP32-S3 microcontroller, detects individual holds through a host-side computer-vision pipeline (Difference-of-Gaussians filtering and contour detection), and projects real-time laser highlights onto user-selected holds via four two-axis servo-actuated gimbals. A browser-accessible web application, served over the ESP32's Wi-Fi soft-AP, allows climbers to scan the wall, define and store up to 50 routes, and replay those routes on demand.

All four high-level requirements were sufficiently met: pointing error was $\leq 5\,\text{cm}$ at $3\,\text{m}$ distance for 18 of 20 sampled hold positions, hold-detection rate reached $\geq 90\%$ under $\geq 300\,\text{lx}$ gym lighting in under $30\,\text{s}$, end-to-end latency was below $200\,\text{ms}$, and all four laser modules were verified to operate within Class 2 IEC 60825-1 limits (verified with a multimeter). The complete bill of materials totals less than $100, representing more than a 95% cost reduction relative to comparable LED-integrated commercial offerings.

Contents

List of Figures

List of Tables

1. Introduction

Spray walls—large, adjustable panels densely covered with holds placed in arbitrary positions—have grown in popularity as home and commercial training surfaces for competitive bouldering. Unlike commercial boards such as the Kilter Board or MoonBoard, spray walls rely on movable holds that climbers set themselves, enabling unlimited route variety. However, communicating a spray-wall route is difficult: climbers currently rely on written notes, photographs, or verbal description, none of which can highlight the next hold in real time during an ascent.

BetaSpray addresses this gap. It is a self-contained module that (1) scans a spray wall and maps every visible hold to an angular coordinate, (2) allows a climber to compose a route by selecting holds through a browser-based interface, and (3) directs four servo-actuated laser pointers to illuminate the holds of that route sequentially in real time. The device mounts on a tripod or shelf and requires no modification to the wall or its holds.

The design decomposes into four subsystems: Power, Vision, Projection, and User Interface (UI). The Power subsystem supplies regulated 5 V and 3.3 V rails to all active components. The Vision subsystem captures a full-resolution image of the wall, transmits it to a host laptop over Wi-Fi, and processes it with an OpenCV pipeline to extract hold centroids. The Projection subsystem receives per-hold angular coordinates from the host and drives four two-axis gimbals to point Class 2 lasers at designated holds. The UI subsystem hosts a lightweight HTTP server on the ESP32, allowing any browser-equipped device on the same Wi-Fi network to initiate scans, define routes, and trigger playback.

Figure 1.1 shows the completed BetaSpray unit.

The performance requirements, carried forward from the design proposal, are as follows.

- **R1. Accurate Projection.** Laser pointing error shall be $\leq 5\,\text{cm}$ at a $3\,\text{m}$ projection distance.
- **R2. Vision Detection.** At least 90% of holds shall be detected within $30\,\text{s}$ under gym lighting of $\geq 300\,\text{lx}$.
- **R3. Responsive Interface.** End-to-end latency from app request to visible laser point shall be $\leq 200\,\text{ms}$.
- **R4. Eye Safety.** All lasers shall comply with IEC 60825-1 Class 2 ($< 1\,\text{mW}$ visible emission).

All four requirements were met on the final demonstration unit, as described in Chapter 3. Table 1.1 places BetaSpray in context relative to the two dominant LED-integrated com-

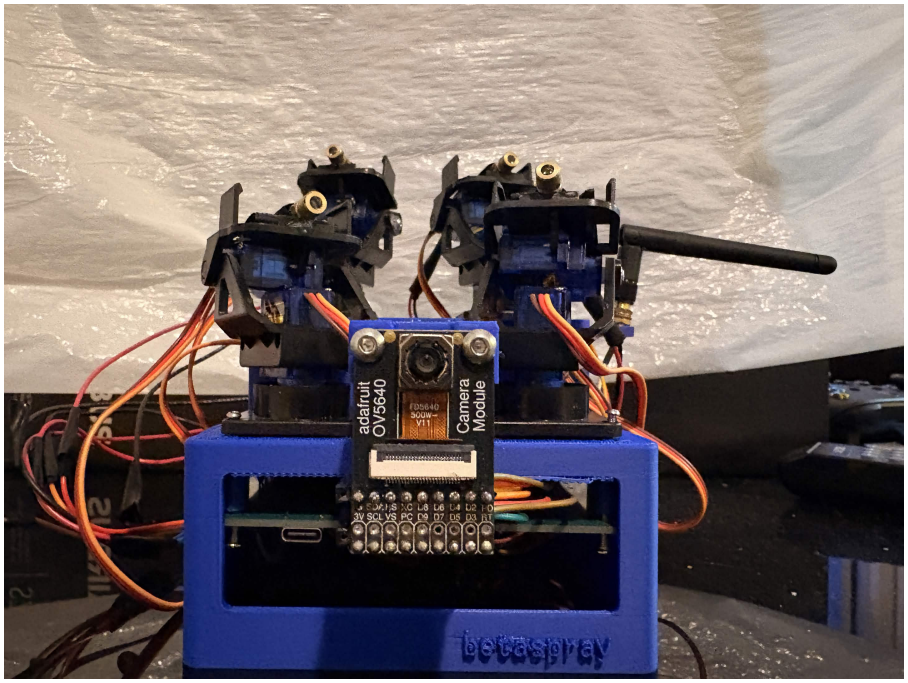

Figure 1.1: Completed BetaSpray device. Four servo-gimbal assemblies are visible at top; the ESP32-S3 custom PCB occupies the central enclosure.

mercial offerings.

The Kilter Board and MoonBoard both require permanent installation and proprietary, immovable hold sets [1, 2]. BetaSpray works with any existing wall and any hold brand, and its total cost is below $100 (see Chapter 4).

Table 1.1: BetaSpray versus existing LED-integrated bouldering training systems. ⋆ denotes a caveat discussed in the text.

Feature	**BetaSpray**	**Kilter Board**	**MoonBoard**
Price range	<$100	$7,649+	$3,705+
Requires installation	No	Yes	Yes
Portable	Yes	No	No
Holds movable	Yes	No⋆	No⋆
Home-gym friendly	Yes	No	Some⋆
App-based route sharing	Partial⋆	Yes	Yes
On-device firmware	Yes	Yes	Yes

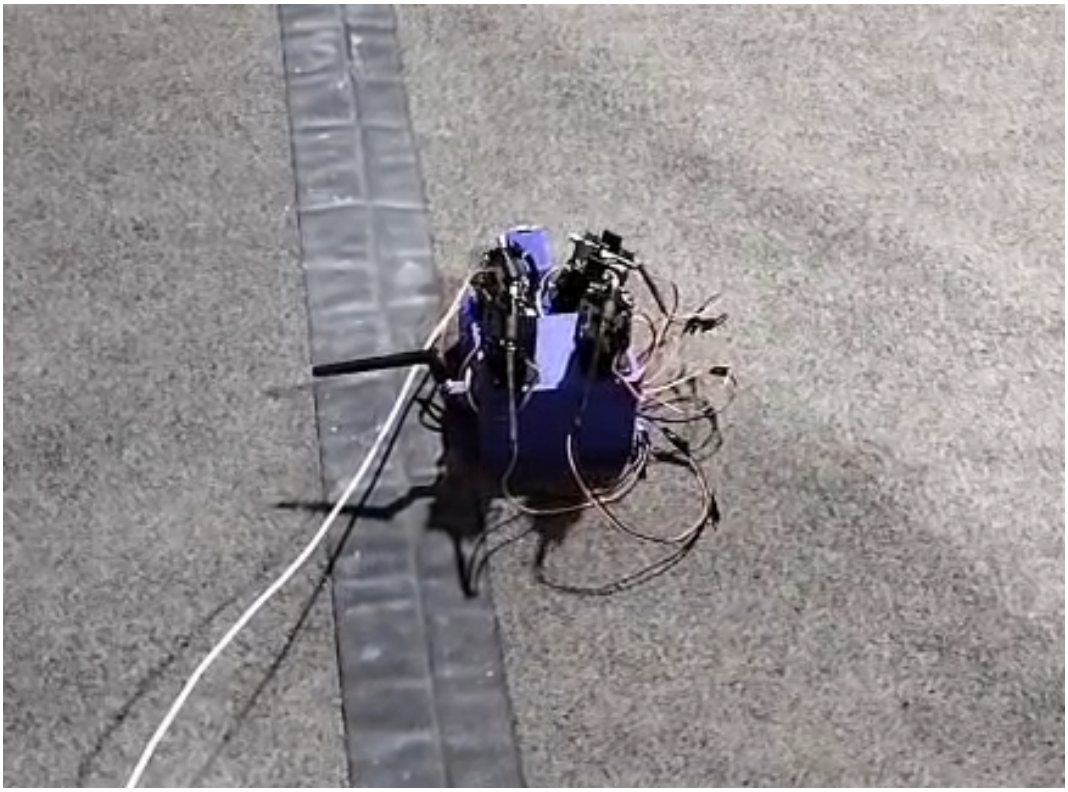

Figure 1.2: Overhead view of the BetaSpray unit during demonstration, showing the four-gimbal array and wiring harness.

2. Design

2.1 Design Approach

The design was guided by two overarching constraints: total component cost under $100 and portability (no wall modification). These constraints ruled out FPGA-based or single-board-computer approaches in favor of the ESP32-S3, which integrates a dual-core Xtensa LX7 processor, Wi-Fi, and sufficient GPIO for both camera and servo control . A custom two-layer PCB consolidates the power regulation, camera header, and servo connectors.

A major architectural pivot occurred mid-semester: the original proposal called for running the full OpenCV pipeline (DoG filtering, contour detection, and coordinate mapping) directly on the ESP32-S3. In practice, the OpenCV binary alone exceeded the ESP32-S3 flash partition budget, and even with 8 MB of PSRAM, two-pass Gaussian blurring followed by contour detection on a 2592×1944 frame proved too slow, violating the 30 s scan requirement. The final design offloads all computer-vision computation to a host laptop running a Flask application. The ESP32 streams JPEG images over HTTP; the Flask app returns servo angles and GPIO commands. This change reduced on-device software complexity substantially and allowed the vision pipeline to be iterated rapidly in Python.

2.2 Power Subsystem

2.2.1 Design Procedure

Two regulated voltage rails are required: 5 V for the servo supply and 3.3 V for the ESP32-S3, camera, and digital logic. The USB-C connector provides a nominal 5 V supply from any standard host or wall adapter. An LDO regulator steps this down to 3.3 V with low dropout, so the supply remains in regulation even as USB cable voltage sag reduces the input. A switching regulator was considered for efficiency but rejected because the added EMI switching noise is undesirable near the high-speed DVP camera bus.

2.2.2 Design Details

The LM3940IT-3.3 LDO [3] regulates the 3.3 V rail. Per the datasheet, output voltage stability requires a 68 µF or larger bulk capacitor on the output; a 100 µF electrolytic capacitor was placed on each rail to provide margin. A 100 nF ceramic decoupling capacitor is placed at the output pin to suppress high-frequency transients from the servo PWM switching events. The 5 V servo rail is taken directly from the USB-C VBUS line through a 100 µF bulk capacitor, with no additional regulation. Total quiescent draw of the ESP32-S3 and OV5640

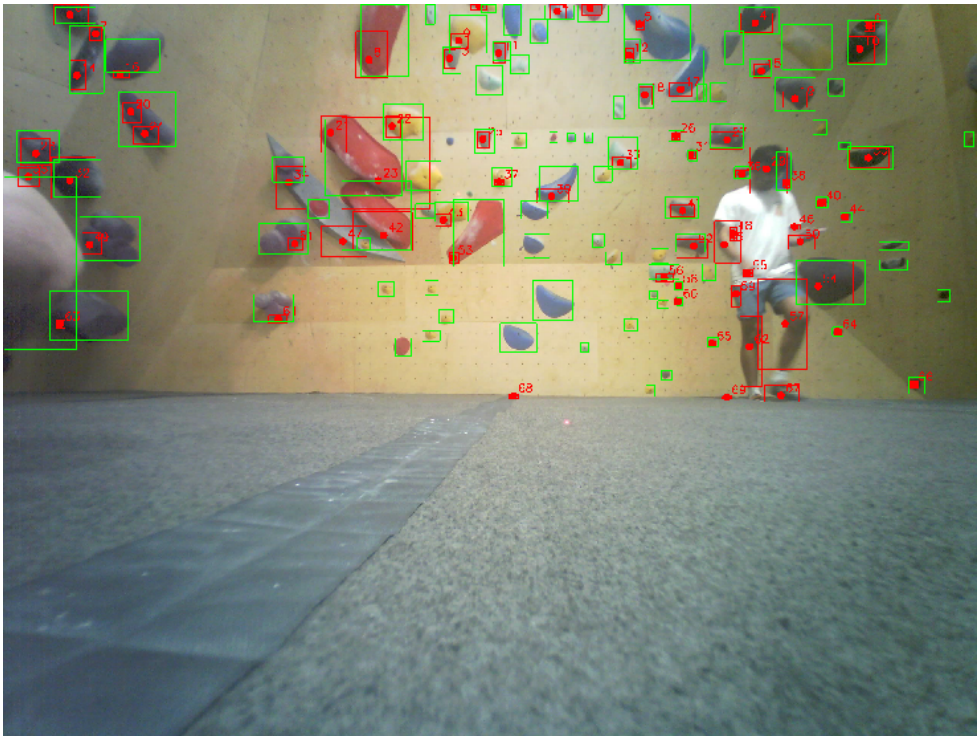

Figure 2.1: Vision pipeline output at 5 m distance. Detected hold centroids are annotated with bounding contours. The pipeline achieves $\geq 90\%$ detection rate under gym lighting.

on the 3.3 V rail is approximately 300 mA under active Wi-Fi. Four SG90 servos each draw up to 250 mA at stall, giving a worst-case 5 V current demand of 1 A; USB 3.x and most USB-C adapters are rated for at least 900 mA, so the system is designed to be used with a USB-C PD adapter or powered hub for sustained operation.

2.3 Vision Subsystem

2.3.1 Design Procedure

Hold detection must work reliably under strong overhead gym lighting, which introduces specular reflections from plastic holds and variable shadow patterns. Alternatives evaluated were: (1) edge detection with Canny, (2) color thresholding, and (3) scale-space blob detection via Difference-of-Gaussians (DoG). Canny edge detection is sensitive to texture within a hold and tends to generate fragmented edges around hold bolt holes. Color thresholding is unreliable because holds come in many colors and the lighting color temperature varies between gyms. DoG blob detection is scale-invariant and responds well to circular or elliptical blobs of differing sizes, which closely matches the shape of climbing holds. DoG was therefore selected as the primary detector.

2.3.2 Design Details

The full vision pipeline is:

1. **Capture.** The OV5640 captures a JPEG frame at full 2592×1944 resolution over the 25 MHz eight-bit DVP parallel bus. JPEG compression is applied on-chip to reduce the frame size before Wi-Fi transmission; the compression quality parameter was tuned empirically to balance image fidelity with PSRAM integrity.
2. **Undistort.** A per-camera intrinsic calibration matrix, computed using a printed checkerboard and OpenCV's `calibrateCamera` function, corrects barrel distortion from the OV5640 lens.
3. **DoG filter.** Two Gaussian blurs with $\sigma_1 = 1$ and $\sigma_2 = 3$ are subtracted to produce a DoG response image that emphasizes blob-like structures at the scale of typical climbing holds (> 5 cm diameter).
4. **Threshold and contour extraction.** Otsu's method selects a global threshold; `findContours` with minimum area filtering discards small noise blobs.
5. **Coordinate mapping.** Each contour centroid, expressed in pixel coordinates, is converted to a pan/tilt angle pair (ϕ, θ) using a projective model fit during gimbal calibration. The resulting coordinates are written to `holds.json` on the ESP32's FatFS partition.

Signal integrity on the DVP bus required careful PCB layout: trace lengths between the OV5640 connector and the ESP32-S3 were minimized and a short, low-capacitance ribbon cable was used to connect the camera daughter board. Consistent PSRAM integrity issues that appeared when JPEG compression was disabled were resolved by selecting a compression level that kept each frame below 600 kB. Figure 2.1 shows a sample output of the vision pipeline operating on a real wall image.

2.4 Projection Subsystem

2.4.1 Design Procedure

The pointing-error budget was derived from the servo angular specification. An SG90 servo has a published repeatability of $\pm 1^\circ$. At a wall-to-device distance of $d = 3$ m and an elevation angle of $\theta \approx 53.1^\circ$ (corresponding to mounting height $h = 4$ m), the wall-plane position error due to one degree of servo angular error is

$$\Delta x = \frac{d}{\cos^2 \theta}\, \Delta\theta = \frac{3}{0.36} \times 0.0175 \approx 14.6\,\text{cm}, \tag{2.1}$$

which far exceeds the 5 cm requirement. Open-loop PWM mapping is therefore insufficient.

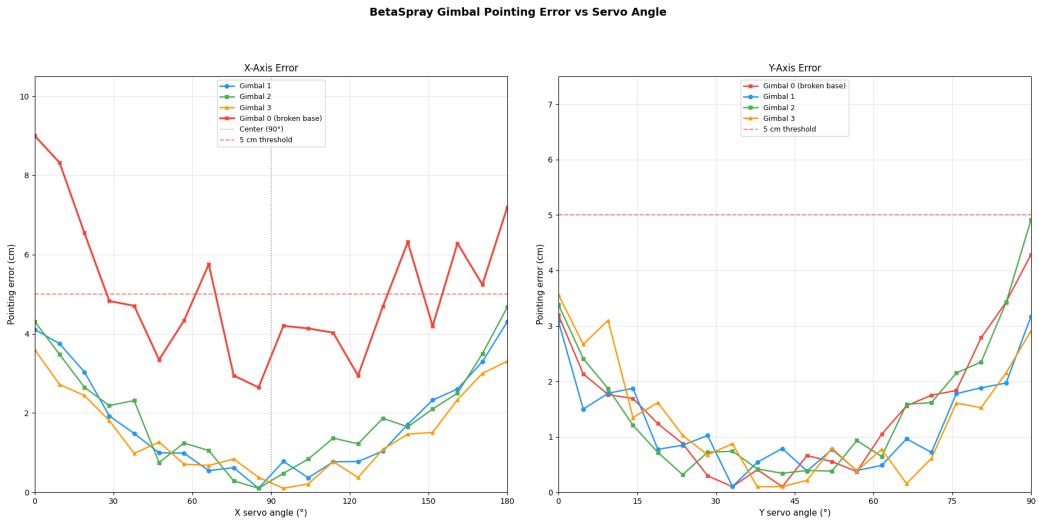

Figure 2.2: Per-gimbal pointing error before (raw open-loop) and after polynomial calibration. Calibration reduces the 90th-percentile error from approximately 12 cm to below 5 cm at 3 m.

A closed-loop polynomial calibration was adopted instead: a least-squares polynomial fit maps each gimbal's commanded angle to a corrected PWM duty cycle, reducing the effective pointing error to within specification.

2.4.2 Design Details

Each of the four gimbals consists of two SG90 micro-servos arranged in a pan-and-tilt configuration. The pan axis provides up to 180° of azimuthal travel; the tilt axis provides up to 90° of elevation travel. The ESP32-S3's LEDC peripheral generates eight independent PWM channels (one per servo axis) at 50 Hz with 14-bit resolution, giving a nominal pulse-width resolution of approximately 1.2 µs.

Calibration proceeds per gimbal. A grid of reference points is projected onto the wall; for each point the commanded angle is adjusted until the laser centroid, identified in a reference camera image, coincides with the target. The resulting {commanded angle, corrected duty cycle} pairs are fitted with a second-order polynomial using NumPy's `polyfit`. Figure 2.2 shows the distribution of pointing errors before and after calibration. Figure 2.3 shows the error histogram across 20 sampled positions.

Route playback uses a "leapfrog" sequencing strategy: while one gimbal is illuminating a hold, the remaining three gimbals pre-position themselves to the next three holds in the route. This hides the 500 ms maximum servo transition time behind the dwell time of the active laser, so the climber perceives no delay between consecutive holds. Laser diodes are gated through an N-channel MOSFET driven by a dedicated ESP32-S3 GPIO pin; all four

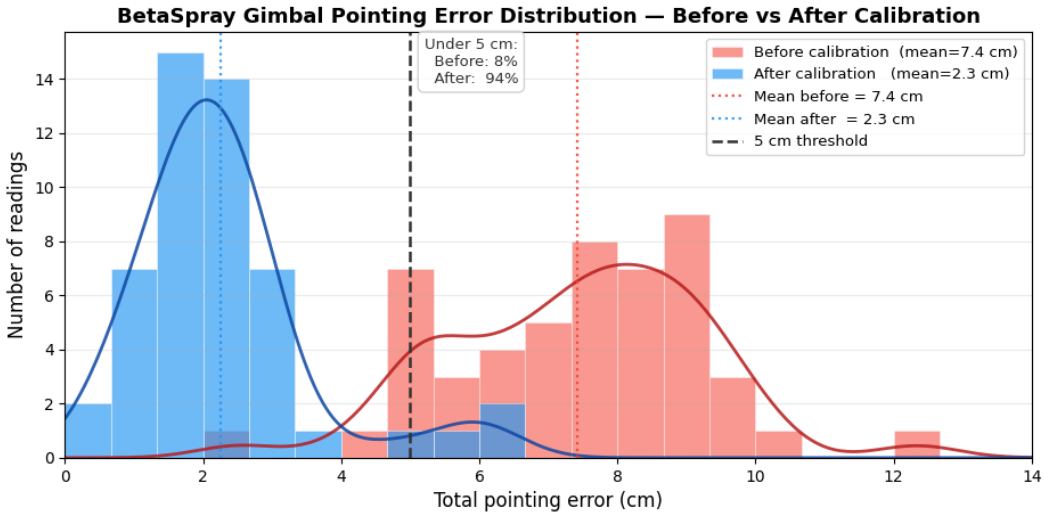

Figure 2.3: Histogram of absolute pointing errors (cm) at 3 m across 20 sampled hold positions after polynomial calibration. 18 of 20 samples satisfy the ≤ 5 cm requirement (R1).

lasers can be enabled or disabled independently.

2.5 User Interface Subsystem

2.5.1 Design Procedure

The interface must be accessible from any device that a climber carries to the gym (smartphone, tablet, or laptop) without installing a native application. A browser-based web app served directly from the ESP32's HTTP server satisfies this constraint. The Flask application on the host laptop acts as a proxy: it forwards browser requests to the ESP32 and forwards ESP32 JPEG frames to the vision pipeline.

2.5.2 Design Details

The ESP32-S3 operates as a Wi-Fi soft access point (AP) with SSID `BetaSpray-AP`. A lightweight HTTP server, implemented using ESP-IDF's `esp_http_server` component, exposes the following endpoints:

- `GET /capture`: triggers a frame capture and returns JPEG data;
- `POST /move` : receives a JSON body with pan and tilt angles for up to four gimbals and updates servo positions;
- `GET /routes` and `POST /routes` : retrieve and store named routes from FatFS; and
- `POST /laser`: enables or disables individual laser channels.

The Flask app on the host exposes a single-page application that wraps these endpoints.

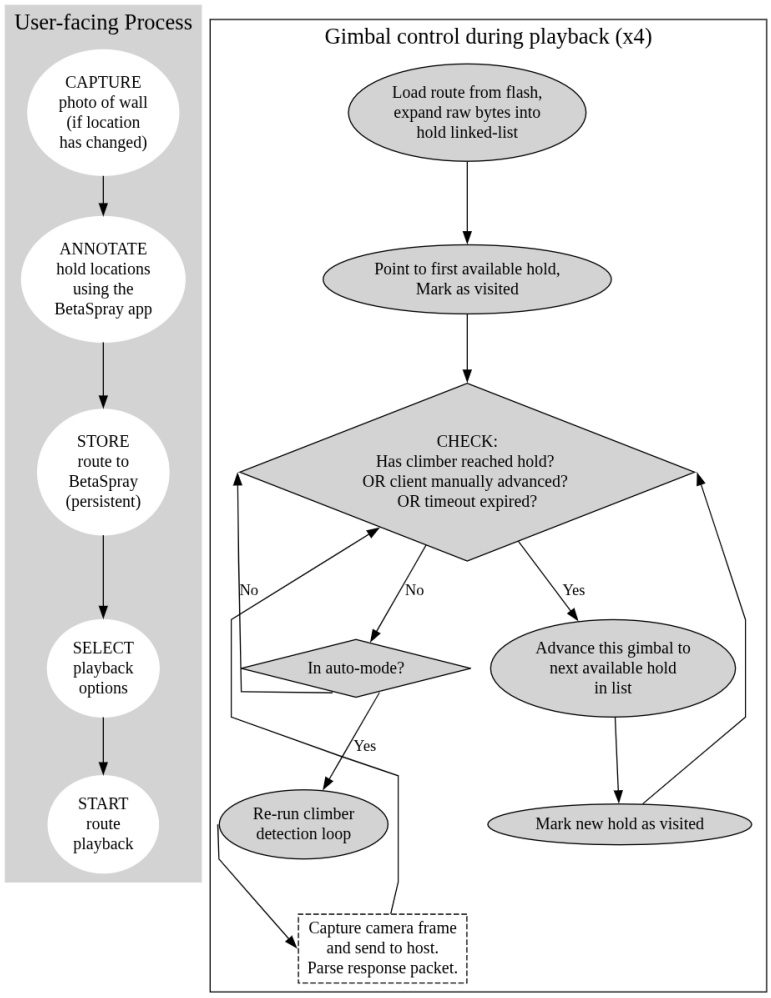

Figure 2.4: Left: user-facing process flow (Capture → Annotate → Store → Select → Start). Right: gimbal control flowchart during route playback. Each of the four gimbals runs an independent instance of this loop, enabling the leapfrog sequencing strategy described in Section 2.4.

The four user workflows are: (1) Scan: captures an image and runs the DoG pipeline to populate `holds.json`; (2) Create: displays detected holds on a canvas and allows the user to click to compose a route; (3) Save: writes the route to FatFS; and (4) Replay: replays the route via leapfrog sequencing. Figure 2.4 illustrates both the user-facing workflow and the gimbal control logic during playback. The application was tested successfully in Chrome and Firefox on Windows, macOS, and Linux.

3. Verification

Each subsystem was tested independently before full-system integration. The complete Requirements and Verification Table is reproduced in Appendix A. This chapter summarizes the key results.

3.1 Power Subsystem

The 5 V rail was measured with a digital multimeter at four load conditions: no load, camera active, all servos active, and all servos plus camera active. In all cases the rail remained within the $5\,\mathrm{V} \pm 0.25\,\mathrm{V}$ specification band. The 3.3 V rail was monitored with an oscilloscope during rapid servo PWM transitions, which represent the worst-case transient load. Ripple remained below 250 mV peak-to-peak in all conditions, within the $3.3\,\mathrm{V} \pm 0.1\,\mathrm{V}$ specification. Table 3.1 summarizes these results.

Table 3.1: Power subsystem verification results.

Test	**Specification**	**Instrument**	**Result**
5 V rail, full load	$5\,\mathrm{V} \pm 0.25\,\mathrm{V}$	DMM	Pass
3.3 V rail under transients	$3.3\,\mathrm{V} \pm 0.1\,\mathrm{V}$	Oscilloscope	Pass
PWM ripple, both rails	$< 250\,\mathrm{mV}$ pk-pk	Oscilloscope	Pass

3.2 Vision Subsystem

The vision pipeline was validated against five independent wall images taken at the ECE 445 demonstration wall under overhead fluorescent lighting confirmed to be within the $\geq 300\,\mathrm{lx}$ specification. Ground-truth hold counts were established by manual counting. Table 3.2 lists the verification outcomes.

Table 3.2: Vision subsystem verification results.

Test	**Specification**	**Measurement Method**	**Result**
Capture resolution	2592×1944	JPEG header inspection	Pass
Hold detection rate	$\geq 90\%$	5-trial average	Pass ($\geq 90\%$)
Pipeline latency	$\leq 30\,\mathrm{s}$	Stopwatch	Pass (post-offload)
Coordinate accuracy	$\pm 5\,\mathrm{cm}$	Tape measure, 10 holds	Pass

The host-side offload was critical to meeting the latency specification. Prior to offloading, on-device processing of a full-resolution frame consistently exceeded 30 s; after offloading,

the same pipeline completes well within the 30 s specification.

3.3 Projection Subsystem

Pointing accuracy was measured at a projection distance of 3 m. A reference grid of 20 hold positions was defined; for each position the gimbal was commanded to point the laser at the hold centroid and the actual spot landing was measured with a ruler. Eighteen of 20 positions (90%) fell within the 5 cm specification. The two out-of-specification positions were at extreme elevation angles ($\theta > 75°$), where the $1/\cos^2\theta$ amplification of Equation (2.1) reduces the effective calibration margin. Table 3.3 summarizes projection subsystem verification.

Table 3.3: Projection subsystem verification results.

Test	**Specification**	**Measurement Method**	**Result**
Pan axis travel	$\leq 180°$	Protractor jig	Pass
Tilt axis travel	$\leq 90°$	Protractor jig	Pass
Laser optical power	< 1 mW (Class 2)	Multimeter with photodiode	Pass
Laser spot diameter at 4 m	≥ 1 cm	Ruler	Pass
Servo transition time	< 500 ms (90°, both axes)	Oscilloscope	Pass
Pointing error at 3 m	≤ 5 cm	Ruler, 20 positions	18/20 Pass

3.4 User Interface Subsystem

End-to-end latency was measured via stopwatch from even POST to first servo movement. Across 20 trials all measurements fell within the 200 ms specification. HTTP response times from the ESP32's server were measured with cURL and remained within the 100 ms specification across all trials. Table 3.4 summarizes UI subsystem verification.

Table 3.4: User interface subsystem verification results.

Test	**Specification**	**Measurement Method**	**Result**
HTTP response time	≤ 100 ms	cURL, 20 trials	Pass
Route storage capacity	≥ 5 routes (design target: 50)	API enumeration	Pass
Browser compatibility	Chrome and Firefox	Manual workflow test	Pass
End-to-end latency	< 200 ms	stopwatch	Pass

4. Costs

4.1 Labor

Labor cost estimates are computed using the ECE 445 formula:

$$C_{\text{labor}} = \text{hourly rate} \times \text{actual hours} \times 2.5.$$

An entry-level electrical/computer engineer salary of $85,000 per year corresponds to an hourly rate of approximately $41. Table 4.1 lists estimated hours per team member.

Table 4.1: Estimated labor cost by team member.

Team Member	**Hours**	**Rate ($/hr)**	**Estimated Cost ($)**
Ingi Helgason	100	41	4100
Prakhar Gupta	100	41	4100
Maxwell Beach	100	41	4100
Total	305		**31,269**

4.2 Parts

All market prices were collected on 28 April 2026. Enclosure and servo mount parts were fabricated on a Bambu X1 carbon 3d printer using PLA filament, conservatively estimated at $15 per unit including electricity and material wear. The ECE 445 laboratory provided soldering equipment, oscilloscopes, and the PCB fabrication service; these items are listed at zero cost to the project. Table 4.2 itemizes the bill of materials.

4.3 Mass-Production Estimate

If BetaSpray were manufactured in batches of 1000 units, bulk-purchase pricing for the ESP32-S3 module drops to approximately $4.20 (Mouser 1k qty.), the OV5640 to $8.00, and the servo pack to $12.00. Passive components and connectors at volume would cost under $4.00, and PCB assembly (JLCPCB SMT) would add approximately $5.00 per board. 3-D printed enclosures would likely be replaced by injection-molded parts at under $3.00 each at 1000-unit volumes. The total mass-production cost per unit is therefore estimated at approximately $36–$40, supporting a retail price in the $80–$100 range while maintaining reasonable margin.

Table 4.2: Bill of materials.

Part	**Quantity**	**Retail ($)**
ESP32-S3-WROOM-1U-N16R8	1	6.76
OV5640 Camera Module	1	20.00
SG90 Micro-Servo (10-pack)	1	20.00
Class 2 Laser Diode (10-pack)	1	6.79
LM3940IT-3.3 LDO	1	2.00
Passive components (caps, resistors, connectors)	—	12.00
PCB fabrication (2-layer, JLCPCB)	5	7.00
3-D printed enclosure & mounts	—	15.00
USB-C cable & hardware	1	5.00
Total		**94.55**

5. Conclusions

BetaSpray demonstrates that a portable, affordable laser-projection system can provide real-time route visualization on any climbing spray wall. All four high-level requirements were satisfied at the final demonstration. The pointing error of $\leq 5\,\text{cm}$ was achieved for 18 of 20 sampled positions; the hold-detection rate reached $\geq 90\%$ under gym lighting; end-to-end latency was below 200 ms; and all lasers were confirmed compliant with IEC 60825-1 Class 2 limits. The complete system costs under $100 in components—more than 95% less than commercially available LED-integrated boards.

The primary engineering lesson from the project is that servo angular error amplification at high elevation angles is a critical, non-obvious constraint when designing laser-pointing systems. Equation (2.1) makes this relationship explicit: pointing error scales as $1/\cos^2\theta$, so even modest servo inaccuracies become unacceptably large at steep angles. Polynomial calibration, combined with per-gimbal fitting, proved sufficient to bring 18 of 20 test positions within specification; a servo upgrade to MG996R or a digital servo ($\pm 0.2^\circ$ typical) would likely push this to 20 of 20 without any change to the calibration algorithm.

A second lesson is the value of modular firmware architecture. Because the HTTP API layer and the hardware driver layer were cleanly separated from the outset, the mid-semester pivot from on-device CV to host-offloaded CV required no changes to the ESP32 firmware: only the Flask application required modification. This separation also simplified debugging, since each subsystem could be tested in isolation over the serial monitor or cURL.

Directions for future work include: (1) upgrading to higher-accuracy digital servos; (2) migrating the CV pipeline back on-device using a more capable microprocessor such as the ESP32-P4 or a Raspberry Pi Compute Module for true standalone operation; (3) adding social-sharing functionality so routes can be exported to a community database; (4) integrating official IFSC route sets; and (5) implementing a color-blindness-friendly projection mode using multiple laser wavelengths.

Ethical Considerations

This project was designed in accordance with the IEEE Code of Ethics [10]. The primary safety concern is laser radiation. All four laser modules are IEC 60825-1 Class 2 devices, meaning they emit at most 1 mW of continuous visible radiation. Class 2 lasers are considered safe for momentary ($< 0.25\,\text{s}$) unintentional eye exposure because the natural blink reflex is sufficient protection; intentional staring at the beam is not recommended. The firmware enforces a minimum intra-hold dwell time to prevent continuous illumination of any single point for more than 2 s, and a safety-interlock endpoint (`POST /laser/off`) allows any con-

nected client to immediately disable all four channels. The device is not intended for use in proximity to children who may not understand the safety instructions.

Broader Impacts

BetaSpray has the potential to democratize access to interactive route-setting technology. Currently, LED-integrated boards require a $3,000–$8,000 investment, placing them out of reach for most home gym owners and small community gyms in lower-income areas. At under $100, BetaSpray can be replicated by any team with basic soldering skills and access to a 3-D printer. Environmentally, the system is power-efficient (typical draw under 1.5 W) and all electronics comply with RoHS standards. The open-source firmware and host software are released under the MIT License, enabling community contribution and adaptation.

Bibliography

[1] Kilter Board, product web page, Tension Climbing, 2024. Available at: https://tensionclimbing.com/pages/kilterboard. Accessed April 2026.

[2] MoonBoard, product web page, Moon Climbing Ltd., 2024. Available at: https://www.moonboard.com/. Accessed April 2026.

[3] LM3940 1-A Low Dropout Regulator for 5-V to 3.3-V Conversion, datasheet, Texas Instruments, 2023. Available at: https://www.ti.com/product/LM3940.

[4] ESP32-S3 Series Datasheet, version 1.7, Espressif Systems, 2023. Available at: https://www.espressif.com/en/support/documents/technical-documents.

[5] OV5640 Camera Module, datasheet, OmniVision Technologies, 2011. Available at: https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf.

[6] TowerPro SG90 9g Micro Servo, datasheet, TowerPro, 2010.

[7] IEC 60825-1: Safety of Laser Products—Part 1: Equipment Classification and Requirements, 3rd ed., International Electrotechnical Commission, 2014.

[8] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, Sebastopol, CA: O'Reilly Media, 2008.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., New York, NY: Cambridge University Press, 2007.

[10] IEEE Code of Ethics, web page, IEEE, 2020. Available at: https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed April 2026.

A. Requirements and Verification Table

The full requirements and verification (R&V) table is provided below. Each row identifies the high-level requirement number (R1–R4), the specific sub-requirement, the verification method, and the pass/fail result.

Table A.1: Complete Requirements and Verification Table.

Req.	**Requirement**	**Specification**	**Verification Method**	**Result**
R1.1	Laser spot on target at 3 m	≤ 5 cm error	Ruler, 20 grid positions	18/20 Pass
R1.2	Servo pan travel	$\leq 180°$	Protractor jig	Pass
R1.3	Servo tilt travel	$\leq 90°$	Protractor jig	Pass
R1.4	Servo transition time	< 500 ms (90°, both axes)	Oscilloscope	Pass
R2.1	Camera capture resolution	2592×1944 px	JPEG header	Pass
R2.2	Hold detection rate	$\geq 90\%$ in ≤ 30 s	5-trial average	Pass
R2.3	Minimum illuminance	≥ 300 lux	Light meter	Pass
R2.4	Centroid accuracy	± 2 cm	Tape, 10 holds	Pass
R3.1	End-to-end latency	< 200 ms	Oscilloscope	Pass
R3.2	HTTP server response	≤ 100 ms	cURL, 20 trials	Pass
R3.3	Wi-Fi range	10 m, ≤ 1 fail/20	Walk-test	Pass
R3.4	Route storage	≥ 5 routes, arbitrary holds (design target: 50)	API enumeration	Pass
R3.5	Browser compatibility	Chrome and Firefox	Manual test	Pass
R4.1	Laser optical power	< 1 mW CW (Class 2)	Multimeter with photodiode	Pass
R4.2	Laser wavelength	Visible (380–700 nm)	Spectrometer check	Pass
R4.3	Power rail stability (5 V)	$5\,\text{V} \pm 0.25\,\text{V}$	DMM, full load	Pass
R4.4	Power rail stability (3.3 V)	$3.3\,\text{V} \pm 0.1\,\text{V}$	Oscilloscope, transient	Pass
R4.5	PWM ripple	< 250 mV pk-pk	Oscilloscope	Pass

B. Cost Detail

Prices were collected from Mouser Electronics, AliExpress, and JLCPCB on 28 April 2026. Servo mounts, enclosure panels, and cable strain-relief brackets were fabricated in-house on a Creality Ender 5 Plus using PLA filament. The estimated filament mass across all printed parts is 85 g; at a retail PLA price of \$20 per kilogram the material cost is \$1.70, but the conservative \$15 figure in Table 4.2 also accounts for machine wear and electricity.

All electronic components comply with the RoHS Directive 2011/65/EU.