UNDERGRADUATE RESEARCH EXPERIENCE (UREx)
PROJECT REPORT

Portable Automated Tyre Tread Depth Measurement System for Fleet Maintenance Optimisation

Report written by:

Advait Bagri

College of Design and Engineering
National University of Singapore

November 2026

Abstract

SMRT inspects more than 1,000 buses, and each inspection includes measuring tyre tread depth by hand. This is slow, depends on the person holding the gauge, and leaves no record. We are building a handheld device that measures tread depth with a laser and a camera. A green line laser is shone across the tread at an angle, and a camera looks straight down at it. Inside a groove the line appears shifted, and the size of the shift gives the depth. This report covers our first semester: the problem, a review of existing methods, our choice of image sensor, the electronics, a 3D-printed holder, a custom circuit board, and the measurement software. The final housing is not yet assembled, so we tested our software on a computer simulation of the device and on 86 hand-held photos. In simulation, our improved software reduced the average depth error in difficult conditions from 1.52 mm to 0.25 mm. Accuracy on real tyres has not been measured yet; closing this gap between simulation and reality is the main task for next semester.

Preface and Acknowledgements

The work in this report was done jointly by the three members of the UREX team, two of whom will continue the project next semester and carry out the remaining work described in Section 9. We thank our supervisor for her guidance throughout the semester. Our code, simulation data and design files are at https://github.com/Advait-B/urex.

Contents

1 Problem Specification **3**

1.1 Background . . . 3

1.2 User needs and requirements . . . 3

2 Literature Review **3**

2.1 Drive-over systems . . . 4

2.2 Handheld device with linear sensors . . . 4

2.3 Single camera and line laser . . . 4

2.4 What is missing . . . 5

3 System Overview **5**

3.1 Architecture . . . 5

3.2 Choosing the image sensor: linear CCD or CMOS camera . . . 5

3.3 How the measurement works . . . 7

4 Hardware **8**

4.1 Bench set-up . . . 8

4.2 3D-printed holder . . . 8

4.3 Circuit board . . . 9

5 Measurement Software **10**

5.1 From photo to depth . . . 10

5.2 Calibration . . . 11

6 Testing the Software in Simulation **11**

6.1 Why we used a simulation . . . 11

6.2 How the simulated photos are made . . . 11

6.3 Test sets and scoring . . . 11

6.4 Algorithms we developed . . . 12

6.5 Results . . . 13

7 Current Status and the Simulation-to-Reality Gap **14**

8 Conclusion **14**

9 Future Work **14**

1 Problem Specification

1.1 Background

SMRT runs more than 1,000 buses. Inspecting one bus takes 45–60 minutes, and much of that time is spent measuring tread depth with a hand gauge at several points on every tyre. Manual gauging has three problems. It is slow. It is inconsistent, because a gauge that is placed slightly off the bottom of a groove gives a wrong reading. And it leaves no history, so nobody can see how quickly each tyre is wearing.

Tyres are also a large running cost. If every reading were saved with the bus, wheel and tyre brand, SMRT could compare real wear rates across its fleet and choose tyres based on evidence rather than manufacturers' claims.

1.2 User needs and requirements

We considered three users. A bus checker wants to press a device against a tyre, get a reading automatically, and be told when a reading is bad and must be repeated. A fleet manager wants pass/fail results and wear trends on a dashboard. SMRT wants a low-cost device that gives the same result no matter who uses it, needs no fixed equipment in the depot, and cuts inspection time to under 10 minutes per bus. Table 1 lists the requirements that follow from these needs.

Table 1: Key requirements.

Requirement	Target
Repeatability	± 0.1 mm on the same tyre
Pass/fail limit	Fail if any groove is shallower than 1.6 mm
Speed	Under 0.5 s per press; a whole wheel in under 5 s
Handling	One-handed, under 2 kg, three contact feet so it sits the same way each time
Robustness	Detects bad readings and asks for a retake; works in sunlight and dim light
Connectivity	Sends results to an app and the cloud; keeps working offline

Scope. This report covers the first milestone: a repeatable depth reading on a known tyre, with the device connected to a laptop.

2 Literature Review

Figure 1 shows how the four automatic systems we reviewed are set up, and Table 2 summarises them.

The manual gauge. A digital tread gauge costs about S$13 and reads to 0.01 mm. However, it measures only one point at a time, the probe must be pushed into each groove by hand, and the reading depends on where the probe is placed on the curved tread.

2.1 Drive-over systems

Wang et al. [12] built a platform that the vehicle drives over. Two line lasers shine up through a slot onto the tyre, and two cameras below the platform photograph the laser lines by way of a mirror (Figure 1a). Each groove is measured from its floor to a line joining its two edges. The error was below 0.2 mm. A patent by Bosch [3] uses the same idea at a larger scale (Figure 1b): several modules, each with a laser and a camera, sit side by side under a slot in the road, so tyres of any width can be measured. Both systems are accurate, but both need equipment built into the ground, which SMRT wants to avoid.

2.2 Handheld device with linear sensors

Alves et al. [1] built a portable device that is pressed against the tyre (Figure 1c). It uses eight linear CCD sensors, each a single row of 3,694 pixels, lit by two rows of LEDs. An STM32 microcontroller reads the sensors and sends the data to an Android phone, which converts it to depth. They reported an error of about 0.115 mm. This is the closest work to ours.

2.3 Single camera and line laser

Lung et al. [7] showed that one camera and one line laser are enough to measure tread depth (Figure 1d). This method, called laser triangulation, is also the basis of our device. Its accuracy depends on finding the centre of the laser line to a fraction of a pixel, which is a well-studied problem [2, 6, 9].

Table 2: Summary of existing methods.

Method	Set-up	Portable	Reported accuracy
Digital gauge	Probe pushed into one groove	Yes	0.01 mm, one point
Wang et al. [12]	2 lasers, 2 cameras; drive-over	No	Error below 0.2 mm
Bosch [3]	Laser–camera modules in a road slot	No	About $\pm$0.1–0.2 mm
Alves et al. [1]	8 linear CCDs and LEDs; handheld	Yes	0.115 mm
Lung et al. [7]	1 camera, 1 line laser	Bench	Feasibility shown

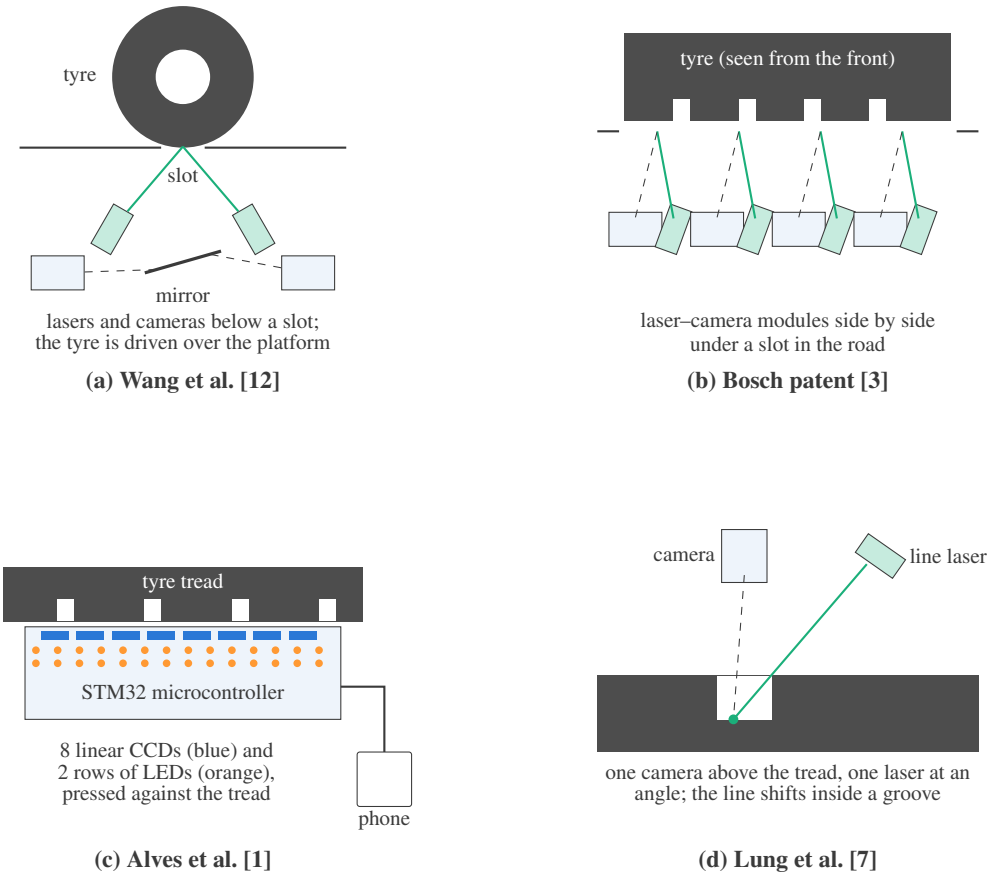


Figure 1: Set-ups of the four automatic systems reviewed.

2.4 What is missing

The accurate systems need fixed equipment, and the portable ones were tested only under controlled conditions. None of the papers explains what happens when a photo is poor, for example because of glare, a wet patch or a crooked press, and none lets the device refuse a bad reading. For a legal pass/fail decision, a confident wrong answer is worse than a request to measure again.

3 System Overview

3.1 Architecture

In Figure 2, a microcontroller board (NUCLEO-C542RC) triggers the camera, switches the laser and passes each photo to a laptop over USB. The camera module (OpenMV Cam H7 Plus) captures and compresses each image.

3.2 Choosing the image sensor: linear CCD or CMOS camera

Our first design used a linear CCD, following Alves et al. [1]. We later changed to a CMOS camera.

Linear CCD. A linear CCD is a single row of light-sensitive cells. After each exposure, the charge

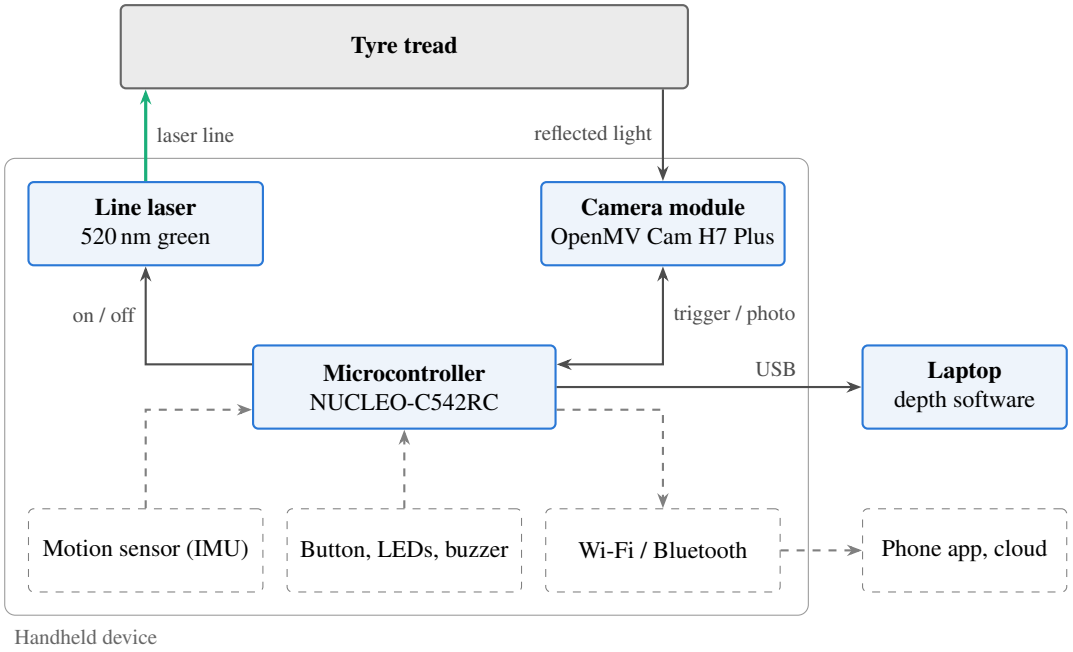


Figure 2: System architecture. Solid boxes: implemented. Dashed boxes: planned for later milestones.

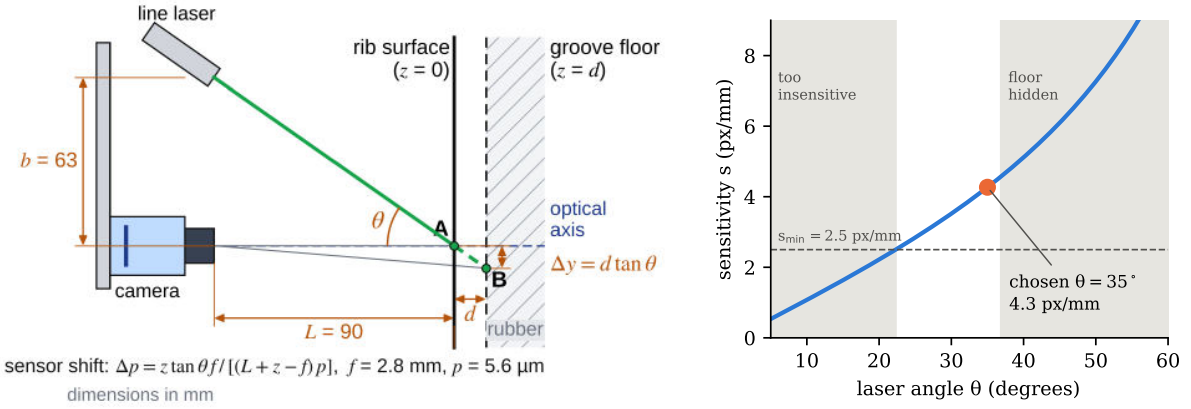
in every cell is shifted out to one amplifier, and the microcontroller's analogue-to-digital converter (ADC) reads the result as a brightness profile. We designed a bench circuit for the Toshiba TCD1209DG (2,048 pixels) [10]. It needs a 12 V supply, six clock signals at 5 V, a logic chip (74HCT04) to raise the microcontroller's 3.3 V timer signals to 5 V, and an op-amp stage to bring its 4–7 V output into the ADC's 0–3.3 V range. Its main weakness is that it sees only one line of the tyre, so it cannot tell the laser from a reflection, or notice that the device is tilted.

CMOS camera. A CMOS sensor is a grid of pixels. Each pixel collects charge in proportion to the light falling on it, has its own amplifier, and is converted to a number from 0 to 255 on the chip, so the camera outputs digital images directly with no analogue circuit. If a pixel collects too much light it saturates, so we keep the laser power and exposure low enough to avoid this. Silicon is most sensitive at about 500–600 nm, so a green 520 nm laser is seen well, and a 520 nm filter in front of the lens blocks most other light. Taking one photo with the laser off and one with it on, and subtracting them, leaves only the laser line.

Our choice. A CMOS camera sees the whole scene: the laser line, how it bends across each groove, and the surroundings. This lets the software check whether a photo is usable before measuring, which became the most important problem in our software (Section 6), and it needs simpler electronics. We therefore made the CMOS camera our main design and kept the CCD circuit as a possible lower-cost option.

3.3 How the measurement works

Figure 3a shows the geometry. The camera looks straight down at the tread. The laser is mounted a distance b to the side and tilted by an angle θ, so that its beam meets the camera's line of sight at the tread surface, a distance L from the lens.



(a) Geometry: a surface d deeper moves the spot by $d\tan\theta$. **(b)** Choosing θ: grey regions break a constraint.

Figure 3: Measurement geometry and choice of laser angle.

When the laser enters a groove of depth d, it travels further before it hits rubber. Because it travels at an angle, it also moves sideways, by $\Delta y = d\tan\theta$. The lens shrinks this sideways move by the ratio $f/(L+d-f)$, where f is the focal length. Dividing by the pixel size p gives the shift of the line in pixels:

$$\Delta p = \frac{d\tan\theta\, f}{(L+d-f)\,p} \qquad \Longrightarrow \qquad d = \frac{\Delta p\, p\,(L-f)}{f\tan\theta - \Delta p\, p}. \tag{1}$$

This relationship is not a straight line, so instead of a single conversion factor we calibrate the device by photographing a flat plate at known heights (Section 5.2).

Choosing $\theta = 35°$ and b = 63 mm. The sensitivity s is how many pixels the line moves per millimetre of depth. Setting $d \to 0$ in Equation (1),

$$s(\theta) = \frac{f\tan\theta}{(L-f)\,p}, \tag{2}$$

which grows with θ, so we want θ as large as possible. Two constraints limit it:

1. Resolution. If the line can be located to $\delta p = 0.25$ px, then to resolve 0.1 mm we need $s \geq 0.25/0.1 = 2.5$ px/mm. With f = 2.7 mm, p = 5.6 μm and L = 81.8 mm this gives $\theta \geq 22.3°$.
2. Shadowing. Where a groove wall crosses the beam (for example in zig-zag grooves), it hides a strip of the groove floor $d\tan\theta$ wide. For the deepest groove (d_{max} = 16 mm) to still show some floor in

the narrowest groove ($w_{min} = 12$ mm), we need $d_{max} \tan\theta \le w_{min}$, so $\theta \le \arctan(12/16) = 36.9°$.

The problem is therefore

$$\max_{\theta} \; s(\theta) \quad \text{subject to} \quad 22.3° \le \theta \le 36.9°. \tag{3}$$

Since s increases with θ, the best value is at the upper limit. We chose $\theta = 35°$, leaving about $2°$ of margin for printing and alignment errors (Figure 3b). This gives $s = 4.3$ px/mm. The laser is mounted at height $H = 90$ mm, just above the camera lens so the two bodies do not collide. For its beam to cross the camera axis exactly at the tread, its sideways offset must be

$$b = H\tan\theta = 90\tan 35° \approx 63 \text{ mm}. \tag{4}$$

4 Hardware

4.1 Bench set-up

On the bench, the camera is connected to the microcontroller by four wires: two for serial data, one for 5 V power and one for ground. The laser needs 7–10 V, more than the board supplies, so a small boost converter raises 5 V to 9 V, and a transistor controlled by the microcontroller switches the laser on and off.

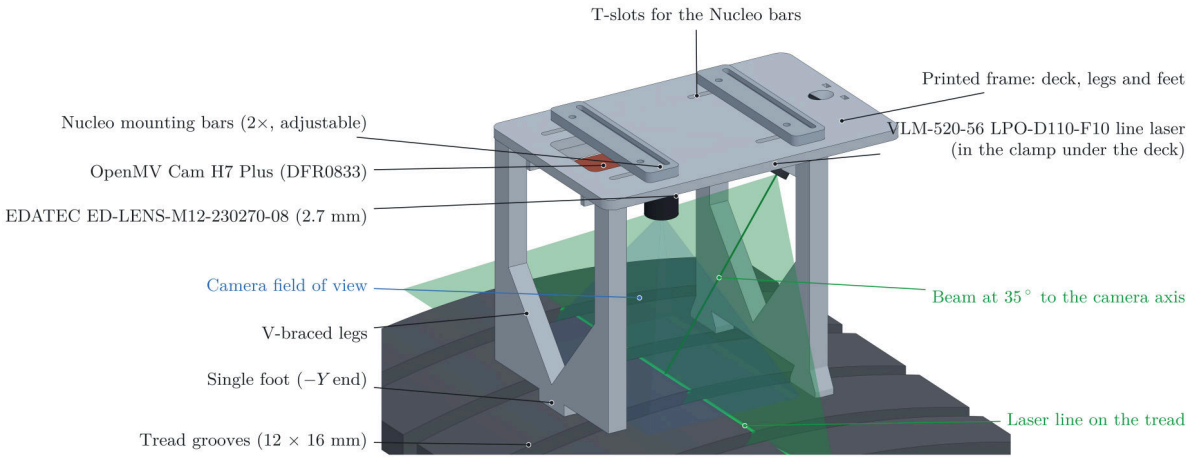
4.2 3D-printed holder

Laser triangulation only works if the camera, the laser and the tread stay in the same positions for every photo. The holder in Figure 4 keeps them fixed at the angle and offset chosen above. It is a single printed frame that stands on three feet, 110 mm apart, with a separate cap that clamps the laser and two bars that hold the microcontroller (Figure 4b). The camera hangs under the deck with its lens 85 mm above the feet.

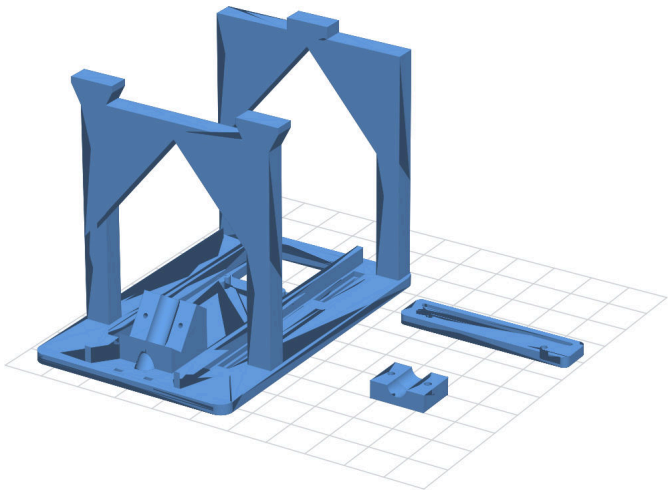
The camera and the laser are placed one behind the other along the grooves rather than across them, so straight groove walls run parallel to the beam and cannot block it. A bus tyre is curved, with a radius of about $R = 480$ mm. Between feet $2a = 110$ mm apart, the top of the tread rises

$$h = R - \sqrt{R^2 - a^2} = 480 - \sqrt{480^2 - 55^2} \approx 3.2 \text{ mm} \tag{5}$$

above the feet. This offset is the same for every photo, so the calibration removes it. One photo covers about 111×83 mm of tread, so three placements cover a whole bus tyre.



(a) Holder on a bus tyre, with the laser line (green) and camera view (blue).



(b) STL files of the printed parts: frame (printed upside down), laser cap and two controller bars.

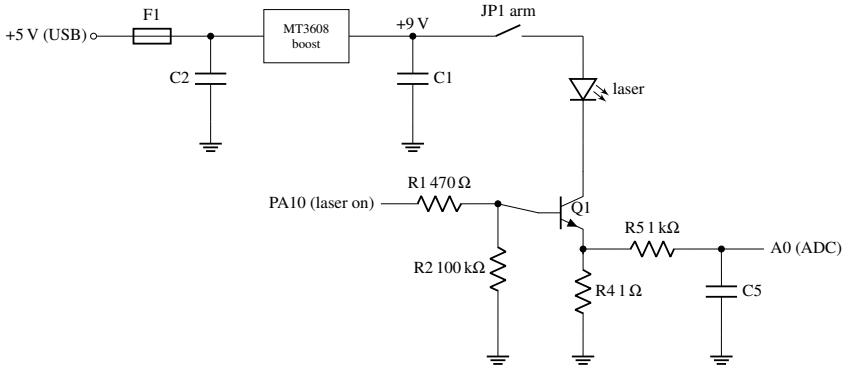
Figure 4: The intended housing.

4.3 Circuit board

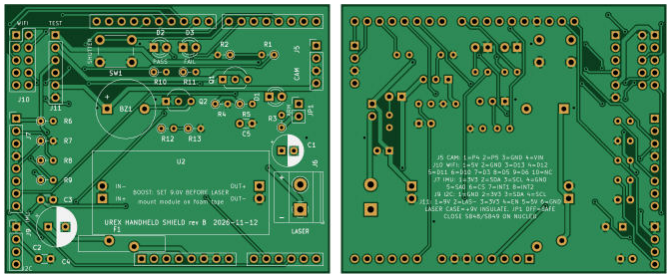
To replace the breadboard wiring, we designed a two-layer circuit board that plugs on top of the microcontroller board. Figure 5a shows its most important circuit, the laser driver. The USB 5 V supply first passes through a resettable fuse (F1), so a short circuit on our board cannot damage the microcontroller. A boost converter (MT3608) raises 5 V to the 9 V the laser needs, and capacitors C1 and C2 smooth the supply when the laser switches on. A transistor (Q1) switches the laser: when the microcontroller drives pin PA10 high, current flows through R1 into Q1's base and the laser turns on; R2 keeps it off while the microcontroller is starting up. The jumper JP1 physically disconnects the laser, so it cannot fire during testing whatever the software does.

The board also checks that the laser really turned on. All laser current flows through R4, a 1 Ω resistor, so 100 mA produces 0.1 V across it. R5 and C5 filter this voltage before the ADC reads it on pin A0. If

the reading is too low while the laser should be on, the laser is faulty, and the device asks for a retake instead of measuring a dark photo. The rest of the board carries the camera connector, connectors for the motion sensor, the Wi-Fi/Bluetooth board and an extra sensor, a shutter button, pass and fail lights, and a buzzer (Figure 5b). The design passed KiCad's automatic design checks with no errors. It has not been manufactured yet.



(a) Laser driver and laser-current check (simplified from the KiCad schematic).



(b) Board layout, top and bottom layers.

Figure 5: Circuit board design (Rev B).

5 Measurement Software

5.1 From photo to depth

Our first version of the software, which we call the baseline, works in three steps:

1. **Find the laser line.** A standard line detector [9] finds the centre of the bright line to a fraction of a pixel.
2. **Fit the tread surface.** A smooth curve $z(u) = c_0 + c_1 u + c_2 u^2$ is fitted to the line, ignoring the grooves. We use RANSAC [5], a method that fits many trial curves to small random samples of points and keeps the one that most points agree with. The depth of each groove is how far the line drops below this curve.
3. **Convert to millimetres** using the calibration, and report fail if any groove is shallower than 1.6 mm.

5.2 Calibration

Calibration converts a position in the photo into a height, using photos of a flat plate at known heights. A physical model of the camera and laser with only five unknowns stayed accurate to 0.002 mm with just three plate heights, while a polynomial fit was off by 0.12 mm.

6 Testing the Software in Simulation

6.1 Why we used a simulation

The holder in Figure 4 has not been assembled yet, so we have no photos taken with the camera and laser fixed in their correct positions. The 86 photos we do have were taken by hand, with the laser held separately. Only 19 of them are usable, and only one tyre has a known depth. We therefore wrote a simulation of the holder on a tyre, in which we know the exact depth of every groove.

6.2 How the simulated photos are made

Each simulated photo is built in four steps, using the holder's real dimensions. First, a random tyre surface is generated: a curved crown with one to three grooves of random depth (0.6–17 mm), width and wall slope. Second, 2,400 laser rays are sent out across the laser fan, and each ray is followed until it first hits the surface, so groove walls cast real shadows. Third, each hit point is kept only if the camera can see it, and is then projected onto the sensor using a camera model with lens distortion. Finally, the bright line is blurred, overlaid on a rubber texture, and given realistic sensor noise and JPEG compression. The simulation can also add difficult conditions on purpose: sunlight, wet reflections, stones, mud, paint, a tilted device, or a laser that is too dim or too bright (Figure 6).

6.3 Test sets and scoring

We created fixed sets of simulated photos: good conditions (150 photos), difficult conditions (200), worn tyres close to the 1.6 mm limit (150), and devices with small assembly differences (150). A separate set of 1,200 photos was used only for training. For N grooves with true depth d_i and measured depth $\hat{d}_i$, we report the mean absolute error and the share of large errors:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{d}_i - d_i|, \qquad \text{Gross error rate} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[|\hat{d}_i - d_i| > 1\,\mathrm{mm}\right], \tag{6}$$

where $\mathbf{1}[\cdot]$ is 1 if the condition is true and 0 otherwise. Because the software may refuse a photo, we also report the share of photos it answered.

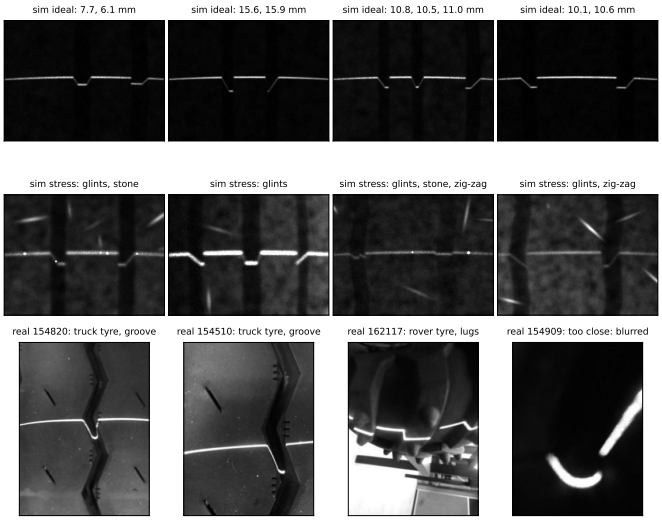


Figure 6: Top two rows: simulated photos in good and difficult conditions. Bottom row: real hand-held photos, where the laser line is thicker, often too bright, and can appear anywhere in the frame.

6.4 Algorithms we developed

We wrote and tested ten families of algorithms, each in its own branch of our code, scored with the same test program. They fall into three groups (Table 3).

Finding the line. The simplest approach looks at each image column on its own, finds the brightest pixel, and fits a curve through its neighbours to place the peak between pixels. For samples a, b, c around the brightest pixel i, a parabola gives

$$\hat{v} = i + \frac{1}{2}\,\frac{a - c}{a - 2b + c}. \tag{7}$$

We also tried a centre-of-gravity and a Gaussian fit. These are precise, but they treat every column separately, so a reflection brighter than the laser in a few columns becomes a false groove. Line tracking fixes this by using the fact that the laser is one continuous curve. Every pixel gets a score $R_c(v)$ for how line-like it is, and the Viterbi method [11] finds the left-to-right path with the best total score:

$$C_c(v) = \min_{v'} \left[C_{c-1}(v') + \mathrm{pen}(v - v') \right] - R_c(v), \tag{8}$$

where $C_c(v)$ is the cost of the best path ending at row v of column c, and pen is a penalty for moving between rows. A single reflection cannot win, because the path would have to pay to jump to it and back. We also tried two learned line finders: a small U-Net neural network [8] that marks laser pixels, and a classifier that decides which detected points belong to the line.

Measuring the depth. We rewrote the depth step to work in millimetres rather than pixels, to look specif-

ically for shallow grooves, and to take each groove floor from a cluster of points found by DBSCAN [4], so one stray point cannot set the depth. We also built the physical calibration model of Section 5.2, and an end-to-end network that predicts depth straight from the image.

Checking quality. Two small logistic-regression models estimate whether a photo shows a usable laser line, and whether a measured depth is likely to be wrong; if either is doubtful, the device asks for a retake. Finally, laser on/off subtraction uses two photos instead of one.

Table 3: Algorithms tested, with results in difficult conditions (200 simulated photos).

Step	Algorithm	MAE (mm)	Gross	Outcome
Line	Baseline line detector [9]	1.52	36 %	Reference
Line	Per-column peak (Gaussian fit)	1.09	7.3 %	Fooled by reflections
Line	Line tracking (Viterbi)	0.54	3.3 %	**Used**
Line	U-Net line finder	0.24	2.1 %	Failed on real photos
Line	Point classifier	0.27	2.4 %	Learned a shortcut
Depth	Improved depth step	0.46	12 %	**Used**
Depth	End-to-end network	0.48	10 %	Failed on other devices
Check	Quality check (16 % answered)	0.14	0 %	**Used**
Two photos	Laser on/off subtraction	0.15	0.9 %	For next semester
All	**Recommended** (89 % answered)	0.25	2.7 %	Final choice

The learned methods looked good in simulation but did not carry over. The U-Net found the line well on simulated photos but failed on real ones, and the point classifier learned that the simulated line always sits near the same row, which is not true in reality. We therefore kept them out of the final design.

6.5 Results

Our recommended software combines line tracking, the improved depth step and the quality check (last row of Table 3).

In good conditions every method was accurate to about 0.15 mm, so finding the line precisely was not the main weakness. The baseline's large errors came from three causes, each with a simple fix. It had no idea that the laser is one continuous line, which line tracking fixes. Its depth step let a single stray point set the groove floor, which the improved depth step fixes. And it could not refuse a bad photo, which the quality check fixes. Together, these reduced the mean error in difficult conditions from 1.52 mm to 0.25 mm,

while still answering 89 % of photos. On worn tyres, one groove of 1.58 mm was read as 1.60 mm and passed, so the device should also have a safety margin, passing a tyre only above about 1.8 mm.

7 Current Status and the Simulation-to-Reality Gap

The holder has not yet been printed and assembled, and the circuit board has not been manufactured, so all the results in Section 6 come from simulation.

On the real photos, our software measured 15 of the 19 usable photos and correctly refused all 55 photos with no laser line. On a truck tyre with grooves about 9 mm deep, two photos of the same spot gave similar shifts (34.3 and 36.6 pixels), so the readings repeat. However, converting them to millimetres gave 22–38 mm, two to four times too large. This is because the camera and laser were held by hand, which the holder is designed to fix. Figure 7 shows some of these photos.

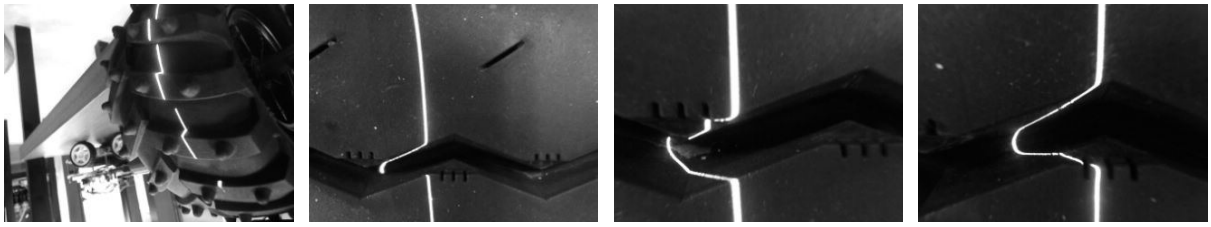


Figure 7: Photos taken with our camera and a hand-held laser. Left: a lugged tyre (28 October). Others: a truck tyre with zig-zag grooves (30 October), where the laser line shifts sideways inside each groove.

The simulation is also easier than reality (its laser line is thinner and dimmer), so its results are a best case, and the accuracy on real tyres is still unknown.

8 Conclusion

We designed a handheld laser-triangulation device for bus tyres: a CMOS camera, a laser angle and offset chosen by maximising sensitivity, a 3D-printed holder and a circuit board. In simulation, line tracking, a careful depth step and a quality check reduced the average depth error in difficult conditions from 1.52 mm to 0.25 mm.

9 Future Work

The work so far was done jointly by the whole team. Next semester, my two teammates will continue the project. Their first task is to print and assemble the holder in Figure 4 with the camera, laser and microcontroller, and to manufacture, assemble and test the circuit board in Figure 5.

With the housing built, they will close the gap between simulation and reality:

1. Calibrate the device with a flat plate at eight known heights, and check it on a block with machined steps.
2. Photograph 5–10 tyres through the holder, three times per groove, and measure each groove with a gauge. About fifty gauge readings would show the device's real accuracy.
3. Take a laser-off and a laser-on photo for every press and test the subtraction method on real photos.
4. Adjust the simulation until its photos look like the real ones, and retrain the quality check on real photos.

References

[1] Alves, D. N. A., et al. (2024). Portable device for faster tire tread depth estimation using linear CCD sensors. Brazilian Symp. on Computing Systems Engineering (SBESC), 139–144.

[2] Blais, F., & Rioux, M. (1986). Real-time numerical peak detector. Signal Processing, 11(2), 145–155.

[3] Robert Bosch GmbH (2015). Device and method for measuring the tread depth of a tire. US Patent Application US20150330773A1.

[4] Ester, M., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. KDD-96, 226–231.

[5] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus. Communications of the ACM, 24(6), 381–395.

[6] Fisher, R. B., & Naidu, D. K. (1996). A comparison of algorithms for subpixel peak detection. In Image Technology, Springer, 385–404.

[7] Lung, C.-W., Chiu, Y.-C., & Hsieh, C.-W. (2016). A laser-based vision system for tire tread depth inspection. Int. Symp. on Computer, Consumer and Control (IS3C), 850–853.

[8] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. MICCAI 2015, 234–241.

[9] Steger, C. (1998). An unbiased detector of curvilinear structures. IEEE Trans. Pattern Analysis and Machine Intelligence, 20(2), 113–125.

[10] Toshiba Electronic Devices & Storage Corp. (2026). TCD1209DG linear image sensor datasheet, Rev. 2.4.

[11] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Information Theory, 13(2), 260–269.

[12] Wang, X.-B., et al. (2019). The study on tire tread depth measurement method based on machine vision. Advances in Mechanical Engineering, 11(4).

AI Declaration

We used AI tools during the UREX project to help generate and debug code, and to refine the language of this report. All designs, results and conclusions were checked by the team, and we take full responsibility for the content.