Depth Perception Haptic Feedback System (DPHS)

Christa Lawrence, Cristopher Matos,

Kathryn Pagano, Chad Pauley

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper presents the design methodology used to create a vest that will assist the visually impaired in navigating their environment. This is done using the latest computer vision software and LiDAR optical detection to determine objects and obstacles in front of the user.**

***Index Terms* — Haptic feedback, Computer Vision**

I. INTRODUCTION

The goal of this project is to develop a wearable guide system to assist the blind or visually impaired with navigating the world around them. The idea is to potentially remove the use of a cane or other similar tool while still maintaining awareness of the surroundings. While other visual impairment products exist on the market; the project differs by concept due to its distinct input features including LiDAR in conjunction with the latest technology in computer vision.

The system has various devices controlled by the central CM4 (Compute Module 4). Output features such as an external speaker and the use of haptic feedback make it a more comprehensive solution than its competitors. The input system consists of communication between a camera and two optical distance sensors programmed to identify obstacles and alert the user through a haptic feedback array (output system) in the form of a grid as well as an external speaker for more advanced operations.

Competition in the same market includes handheld ultrasonic sensors and cane modified technology that the average user would likely find difficult; or even dangerous to use. This design is meant to be more accessible, simple, and effective than what is on the current market.

Along with possessing a simple design that's easy to understand; the system was designed to be lightweight, while still allowing for a full range of motion and millisecond response time. The device was designed with an initial budget of approximately $500; relatively low for a non-manufactured device with advanced camera software and LiDAR technology.

II. OVERVIEW OF THE SYSTEM

The idea behind the concept was to implement various sensors and integrate them to collaborate within the system to effectively guide an individual, whether inside or outside, in spite of impaired or obstructed vision. To develop the device, the team faced many issues common to even the car manufacturer Tesla; such as avoiding potholes and reducing time lag between the device and method of communicating with the user without sacrificing the accuracy of the design.

The use of both sound and haptic feedback are implemented to notify the user and help them navigate their surroundings. There are a total of nine motors each controlled by a motor controller chip integrated into the significant PCB design. These nine motors are in a three by three grid that vibrates stronger in intensity the closer an object is in that detection.

As aforementioned in the introduction, the device is designed to detect outside objects such as curbs or potholes; a LiDAR is used with a geometric algorithm to detect a dropoff or step up depending on geometric calculated values. A second lidar is used facing directly in front of the user to act as an emergency device should something appear that the camera does not have enough time to detect. Different haptic vibrations and audio emitted from the speaker are used to notify the user of various events that may occur by the device such as startup, change in mode, and shutdown.

III. SPECIFICATIONS/SAFETY

The purpose of this section is to note the specifications and verify if they were met.

A. Design Process

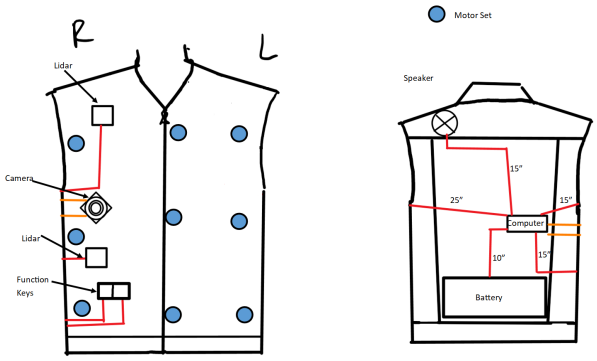

Fig. 1. Design illustration of the component positioning for the DPHS.

The specifications of this device include an affordable cost, comfortable design, ease of use, and long battery life.

B. Cost Specification

It is possible to reduce the final cost of this design by half as most of the budget for the project went into PCB design and testing; which included three different versions of a carryout board for the CM4. The cost estimate if the device were to be built again would be an estimated $265 which would reduce further in the case of being manufactured in a factory with parts bought for wholesale.

C. Comfort/Ease of Use Specification

As the device is to be worn during use, comfort was a major concern for the user. The material used for the vest is suiting fabric and has been sewn from scratch. The wiring of each component uses stranded wire instead of solid wire to promote flexibility in the vest. Having exact lengths of wire was also important as having lumps of wire in the vest with slack was undesired and thought of as uncomfortable.

Ease of use was a requirement for specification as the device was designed for the blind and visually impared. A simple set of four buttons and a power switch promotes a simple-to-use design. Audio feedback for modes and haptic motors for control in places where audio might be difficult to hear such as inside a room or outside by a street makes the system easy to understand.

D. Significant Battery Lifespan Specification

Battery life is dependent on the user's use case period. In long periods of rest without the LiDAR or camera detecting anything the device will notify with a sound and enter rest mode thereby preserving the battery.

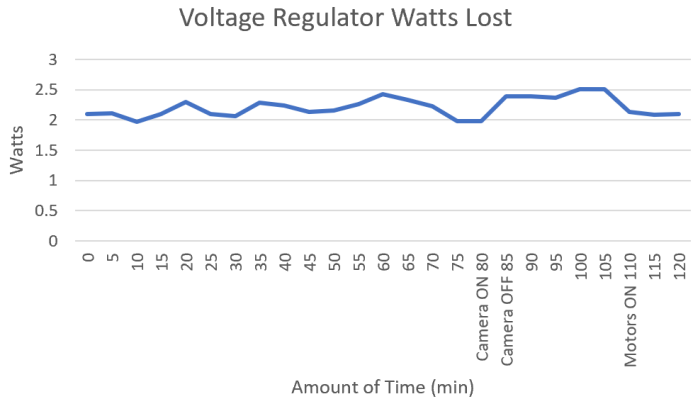

Fig. 2. Power waste by voltage regulator.

$$W = (V_{in} - V_{out}) \times I_{out} \quad (1)$$

The power wasted by the voltage regulator converting 12V to 5V is in Fig. 2. and the method of calculation can be seen in (1).

The device is intended to last about four hours with continuous use, however future versions of the device can be introduced to lithium batteries; thus significantly increasing the maximum operation time of continuous use. Currently with a single battery the expected lifespan is two hours which is within the reasoning that two batteries are used in the final design.

Product Class	**Product Class Description**	**Rated Battery Energy (Ebatt)**	**Special characteristics or battery voltage**	**Maximum UEC (kWh/yr)**
2	Low-Energy, Low-Voltage	<100 Wh	<4V	.1440*E_{batt}+ 2.95
3	Low-Energy, Medium-Voltage		4-10 V	For E_{batt} <10 Wh, 1.42 kWh/y E_{batt} ≥10 Wh, 0.0255 * E_{batt} + 1.16
4	Low-Energy, High-Voltage		>10V	0.11 * E_{batt} + 3.18

Table I. Energy usage classification

The purpose of Table 1 is to show the classification of a voltage input of 12V to 5V while maintaining regulations of having a low-energy and medium voltage. The use of low-energy is noted in the specification

requirements as the device is intended to last an amount of time that would be ideal for a daily driver. The medium voltage classification is the minimum voltage needed to maintain a computational device with enabled accelerometer, lidars, camera, and accelerometer.

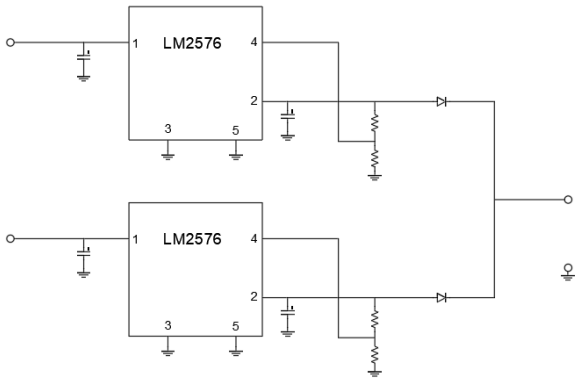

Fig. 3. Voltage regulator design.

The voltage regulator changes the 12V battery output into a steady 5V source. The excess voltage is expressed in wattage Fig. 2 then converted to heat.

D. Safety of the User

Fig. 4. Construction and wiring of the device.

It can be noted from Fig.4 that thicker than necessary wire was used for each component as safety of the user as well as possible static from the clothing was a concern. The device functions using low voltage; medium energy specifications to prevent any static buildup and prevent shock in the worst case that a wire becomes exposed over a long period of time for the user as the device itself is considered a daily driver.

IV. SOFTWARE DESIGN

User experience played a pivotal role in early design discussions--especially with respect to hardware placement, feedback signal design, and response time. Feedback motors were distributed symmetrically across the torso in an attempt to create a sense of embodiment. The number of depth sensing devices required was minimized as much as possible. Additionally, program speed was increased as much as possible to ensure user safety.

A. Computer Vision

The inclusion of CV as a sensing device allowed for the reduction in the number of depth sensors necessary to navigate in a scene. Reducing the total number of sensing devices allowed for an improved ability to conceal the hardware, distribute devices more efficiently while preventing interference from haptics, and granting the user a greater ROM. Further motivation to implement CV was provided by an increase in research and development in monocular depth estimation for embedded devices [2], and the prospect in adopting existing models in conjunction with LiDAR in a visual-tactile sensory substitution device.

As inference is an expensive operation in terms of both memory footprint and response time, an SoC capable enough to run such models was required. In addition to inference, the SoC would be responsible for controlling the entirety of the system. Multiple such devices and hardware accelerators exist on the market. Early investigations compared different devices by inference benchmarks, availability of hardware documentation, and price. Additional research revealed significant work in monocular depth estimation targeted for mobile devices. The final selection was determined after inspecting the results of the Conference on Computer Vision and Pattern Recognition, CVPR, Mobile AI monocular depth estimation challenge. The results indicated that the Raspberry Pi 4 is capable of running depth inference in 100 ms with minimal losses in accuracy [10].

The Raspberry Pi Compute Module 4 was chosen for its extensive documentation, support, and slim form factor in addition to considerations noted above. The device was chosen over the NVIDIA Jetson Nano because of their comparable performance on MobileNet benchmarks, especially when models are executed from frameworks such as Tensorflow TRT and Tensorflow Lite [1].

A significant reason for the success of the project was the availability of open-source monocular depth estimation models. One such model was the recently published inverse-depth models developed by [7]. The pretrained MiDaS v2.1 was designed to meet the industry need for a model robust enough to achieve high accuracy across datasets. Additional optimizations introduced in MiDaS v2.1 Small version allowed for a reduction in inference time on mobile devices.

The Tensorflow Lite version of the small model was implemented in the prototyping phases of the project. Inference time for this version at varying levels of thread count are provided in Table 2. Inference times approached 1 FPS but were deemed insufficient for the intended product use case. The addition of the XNNPACK delegate brought frame rate up to 2 FPS but was still not ideal. The final version of the model implemented is the quantized version on the Alibaba MNN Framework [9] at four threads for a framerate of 3 FPS. The reduction in inference accuracy did not pose an issue as the result is ultimately resized to a 3 × 3 grid.

	TF Lite[1]	**TF Lite w/ XNNPACK**	**MNN**[2]
Floating Point Weights	T_1 = 1,906 T_2 = 1,213 T_4 = 948 T_8 = 1,619	1,170 655 511 1,136	744 445 398 578
Int Quantized Weights[3,4,5]	1,111 670 468 1,249	1,106 682 467 1,179	901 486 <u>302</u> 454

Maximum inference times (ms) for various thread counts T_i. All benchmarks set to 5 warm up runs, 50 runs. Tests selected randomly from 24 conditions. Tests conducted in open air, no fan, ensuring VideoCore temperature of CM4 below 70.0 ℃ before proceeding.

[1] Results from TF Lite linux arm benchmark tool [5]

[2] Results from MNN model benchmark tool [7]

[3] Weights quantized using RedWeb V1 dataset [4]

[4] INT8 quantized weights for TF Lite from default optimization tool (See **[7]**)

[5] UINT8 quantized weights for MNN from the quantization tool (See [9]). KL divergence for features, max absolute value for weights

Table II. Framework and weight quantization comparison at varying thread counts.

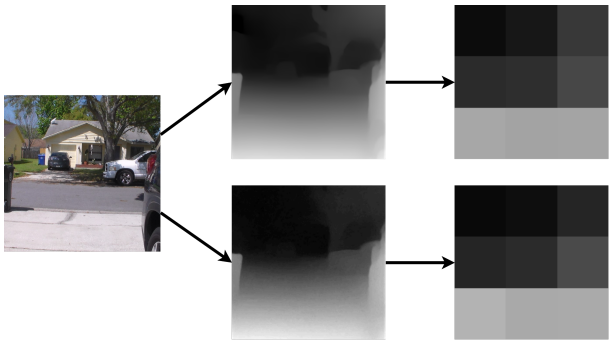

Fig. 5. FLOAT32 (top) versus UINT8 (bottom) versions of MiDaS v2.1 Small converted directly from TFLite to MNN, bilinear downsized.

Additional filtering was required to convert the 256 × 256 depth map into a 3 × 3 grid of motor responses. The filtering algorithm was designed to account for the nonlinear relationship between estimated and real depth, the gradient introduced by the floor (e.g. the floor in front of the user registering as a near-object), and instability in the inference caused by using images rather than video (See [3]). The filtering algorithm is as follows:

1. Subtract a linear box filter from the depth map
2. Clip the result to fit between two values of estimated depth
3. Discretize the resulting depths to be integers on [0, N) where N is the desired number of feedback levels including zero
4. Maxpool with an 85 × 85 kernel
5. Compare the result with the previous 3 × 3 grid. Only update an entry if the entry increases, or decreases to zero.

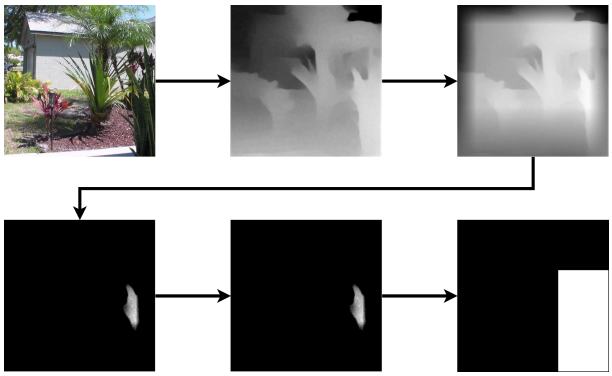

Fig. 6. Demonstration of additional processing to map

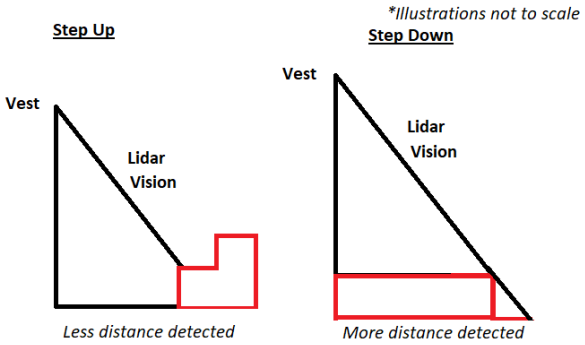

Fig. 7 LiDAR curb detection.

The initial step serves to decrease the effect of the floor, ceiling, and walls in narrow hallways. The second step restricts the range in order to ignore distant objects (e.g. greater than 6 m away) and to cap near-objects to a uniform maximum. The third and fourth steps convert the 256 × 256 floating point depth map to a 3 × 3 integer map. Maxpool is used for safety, in order to alert individuals to any potential obstacle detected by the camera. The final step reduces the instability in the output signal by assuming that obstacles persist even if the depth estimated by the model decreases. Signals must be zeroed before they are represented as such in the output.

Finally, the 3 × 3 grid is translated to the haptic feedback motors. The response is a simple linear response. Values in the grid are divided by N - 1 to form percentages, which are then multiplied by the maximum desired feedback intensity. Nonlinear approaches were considered, such as using some polynomial, logarithmic, or sinusoidal function. These response patterns were not implemented as these were deemed unintuitive, and would thereby negatively impact required training time.

B. LiDAR Detection

Two LiDARs communicate with the system via an I2C connection with the purpose of responding to what camera vision alone cannot detect.

The LiDAR located at the top of the device is angled toward the floor at 30 degrees from the normal plane of the vest. Changes in the distance value are interpreted as being either step-up, or step-down curbs. This angle allows the device to detect curbs 7 feet away assuming that it is 4 feet above the ground. A finite queue of distance values is maintained. The algorithm for detecting curbs on update of the queue is as follows:

1. Find the maximum and minimum of the queue, as well as the indices of both
2. Check if the range is bounded by some upper and lower threshold
3. Compare the indices of the maximum and minimum. If the maximum occurs before the minimum, the curb is step-down. Otherwise, step-up
4. Clear the queue if a curb was detected

The first step sets up the parameters necessary to serve the logic. The second step differentiates between noise in the signal and an actual curb while ensuring that obstacles such as barricades are not interpreted as traversable. This threshold was determined by comparing actual curb heights (Approx. 0.5 ft) and the standard deviation of the LiDAR depth (SD of 0.39 in. at 13 ft. away; [2]. The third step determines the direction of the curb, and the final step ensures that the system will no longer react to the same curb.

Once a curb is detected, haptic motors across the bottom row vibrate in a particular pattern. For step up curbs, the motor on the left fires at half of the maximum intensity, and moves toward the right. For step down curbs the response moves in the opposite direction.

The second LiDAR placed on the lower portion of the vest faces directly in front of the user and acts as an emergency detection in case an obstacle appears too quickly for the camera to detect (e.g. a bicyclist, pedestrian). Additionally, this real depth information is made available to the user via the motor centered on the chest to allow for navigation in tight spaces. Upon reaching the emergency threshold, all operations on the vest cease and the motors rapidly flash between 0 and the maximum intensity until they are clear of the obstacle.

C. Scheduling and Synchronization

There are two forms of synchronization implemented by the system: Polling and waiting. On the execution of a mode (e.g. the emergency detection and computer vision as a 'General Mode'), the system will execute the logic associated with it, and then poll a 'mode changed' flag. The mode will continue to execute until this flag is set by an interrupt (i.e. a button pressed by the user). Given that the flag is only polled after the execution of the mode, change requests may not be addressed immediately. However at worst case, the delay between pressing the button and the mode shifting is bounded by 400 ms unless the emergency signal is active. The alternative synchronization approach utilized by the low power mode is the wait. The thread executing the mode is suspended until the mode changed event is registered. Once a mode change event is registered, the system mode is shifted to the mode associated with the button and the flag is cleared.

It is essential that the services provided by the system run as quickly as possible for the safety of the end user. However, given the strict power consumption and UX requirements set by the designers, increasing the clock speed of the SoC was not an option. The largest determining factor of speed of the system was the computer vision inference time. Time permitting, additional multithreading may be added to the product to allow for the LiDARs to be polled more frequently than the base software can currently support.

D. User Interface

A total of four buttons were implemented in the design with simple but memorable features. The first button will act as a volume control for the speaker. At starting operation the volume is near max when continuously pressed will go to the minimum after reaching the maximum sound. The second button changes the mode of the device to General mode which would act as an operation used indoors where reflective tile may be common and curb detection is not needed. The third button changes the mode to Outdoor mode where LiDAR and camera work in conjunction to navigate the user. The last button places the device to sleep in cases of standing and talking to a friend, sitting down, or climbing stairs.

V. HARDWARE COMPONENTS

A. Motor Design

The design implements three input devices through an I2C bus connection. The inputs are connected to the significant PCB design and include a camera and two lidar. The output devices include a set of nine motors and a speaker that are all controlled through PWM. Each motor consists of a simple diode and capacitor on each pair of haptic motors designed to protect the CM4 as well as the motors themselves.

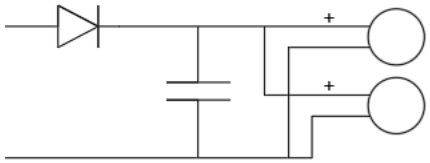

Fig. 8. Schematic of Haptic Motor Design

The amount of voltage used on each motor is 5V for powering the motor and PWM 3.3V to turn on the transistor which thereby allows the 5V to turn on the motor (load) and go to ground.

The speaker is connected to a pwm pin directly from the CM4 I/O unlike the motor PWM from the motor controller. The speaker has a load of 8 ohms with an intended input voltage of 3.3V; meaning the total expected wattage from the speaker is 1.36 Watts. From testing, the speaker is fairly loud and will not be an issue in terms of decibels.

B. Significant Design

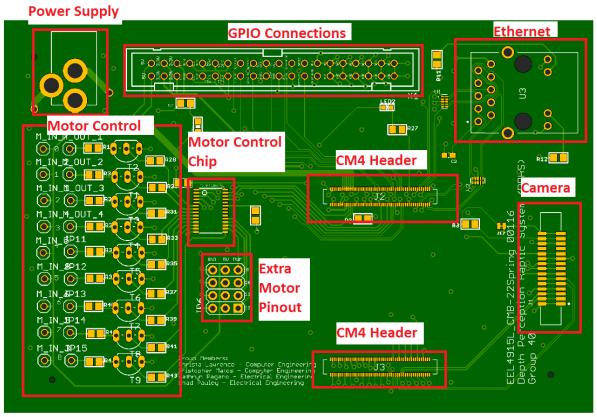

Fig. 9. Significant design PCB for DPHS I/O.

Overall design for this project involves the motor controller and CM4 I/O on a single PCB. The designed layout has a total of one possible SPI, four I2C, two PWM speaker outputs, two 5-volt outputs, four 3.3-volt outputs, and fourteen GNDs for the CM4 I/O. Traces and bias followed minimum specifications as noted by the PCB manufacturer JLCPCB.

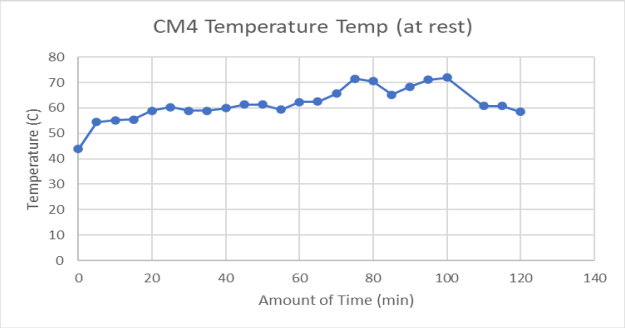

Fig. 10. Temperature of CM4 over time.

The PCB design is significantly cooler than the CM4 which is mounted and remains around room temperature.

Motor controller uses two significant components including a BJT and PWM control chip. The motor controller uses a total of nine sets of 2200 ohm resistors, 10 ohm resistors, and a 2N2222 BJT which acts as a gate between the pwm chip and each of the nine motors for a 5 volt on the collector end. Input for the motor controller uses an I2C bus and can output a total of 16 motors but for the purposes of this project only nine are used (with gate configuration) and four as backup on the board (without a gate configuration). Possible uses for the four backup PWM pins from the motor controller include adding more speakers, a fan to cool the CM4, making an LED indicator, and adding additional motors.

Device	Maximum Voltage Rating	Maximum Current Rating
LiDAR	5.25V	85mA
Compute Module	5V	3A
Motors	3V	150mA
Speakers	5V	.625A

Table 3. Device power ratings.

VI. HARDWARE COMPONENTS

Two significant lessons were learned by the group by the end of the project. The first is the difficulty in having CpE and EE place an order of priority and deliverance when placing the project together. Each separate part of the project was simple; however when everything came together, what needed to be tested before going on the vest and what needed to be on the vest before being tested created many delays in the development process. Learning from this experience, the team gained insight that using spreadsheets data and checking each other's work prevented many mishaps that would have typically resulted in the project becoming a failure.

The second lesson learned by the group was to always have a backup component for every part. During the testing phase of the project the CM4 burnt out when wires namely for the motors were pulled out while the device was still active. The team ultimately paid three times the original price in order to procure another one within sufficient time needed for the project.

VI. CONCLUSION

Although the group faced many challenges in the software and hardware aspects as well as integration; the project proved an overall success with a working system that covered many of our initial specification requirements.

VII. REFERENCES

[1] Allan, "Benchmarking TensorFlow Lite on the New Raspberry Pi 4, Model B," 2019. https://www.hackster.io/news/benchmarking-tensorflow-lite-on-the-new-raspberry-pi-4-model-b-3fd859d05b98

[2] Benewake, "Product Manual of TFmini-S," Benewake Co., 2018, .https://cdn.sparkfun.com/assets/8/a/f/a/c/16977-TFMini-S_-_Micro_LiDAR_Module-Product_Manual.pdf

[3] Ignatov, Andrey, et al. "Fast and Accurate Single-Image Depth Estimation on Mobile Devices, Mobile AI 2021 Challenge: Report." ArXiv, 17 May 2021, https://arxiv.org/pdf/2105.08630.pdf.

[4] JLCPCB, "Capabilities," JLCPCB, 2022 https://jlcpcb.com/capabilities/Capabilities

[5] K. Xian, C. Shen, Z. Cao, "The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)," 2018. https://sites.google.com/site/redwebcvpr18/

[6] M. Abadi, A. Agarwal, P. Barham, “TensorFlow, Large-scale machine learning on heterogeneous systems,” 2021. https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/tools/benchmark#tflite-model-benchmark-tool-with-c-binary

[7] R. Lasinger, K. Lasinger, D. Hafner, K. Schindler, and Vladlen Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer,” 2020 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

[8] Tensor Flow, “Post-training quantization,” 2021, https://www.tensorflow.org/lite/performance/post_training_quantization

[9] X. and W. Jiang, Huan, Chen, Yiliu and Wu, “MNN: A Universal and Efficient Inference Engine,” 2020, https://github.com/alibaba/MNN/tree/1.2.7/benchmark

[10] Zhang, Ziyu, et al. “A Simple Baseline for Fast and Accurate Depth Estimation on Mobile Devices.” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, https://openaccess.thecvf.com/content/CVPR2021W/MAI/papers/Zhang_A_Simple_Baseline_for_Fast_and_Accurate_Depth_Estimation_on_CVPRW_2021_paper.pdf.

VIII. BIOGRAPHY

Kathryn Pagano is currently a senior at the University of Central Florida graduating May of 2022 with a Bachelors of Science in Electrical Engineering. She is currently an Intern with POWER Engineers, and intends to continue working with POWER as an Engineer 1 after graduation.

Chad Pauley is an EE major at the University of Central Florida studying under a bachelor’s degree. He is on the general track and worked with the UCF Autonomous Robotics club for two years and held a position as an officer.

Christa Lawrence is currently a senior in Computer Engineering at the University of Central Florida, with a minor in Mathematics.

Cristopher Matos is a senior student majoring in Computer Engineering, Psychology, and minoring in Mathematics. He is interested in psychophysiology, distributed cognition, artificial intelligence, and control theory. His work experience extends from egocentric computer vision, to personality research, virtual reality, and front and back-end web application development.

IX. SPECIAL REGARDS

Special thank you to ‘Quality Manufacturing Services’ located in Orlando for soldering many parts free of charge for UCF students.