The Do-Pro: A Simplistic Stereo-vision Camera

Timothy Do[1], Daniel Jilani[1], Zaya Lazar[1], Harrison Nguyen[1]

[1] Department of Electrical Engineering and Computer Science,
University of California, Irvine, Irvine, USA

***Abstract*—To make computer vision technology more accessible to small consumers such as scientists, hobbyists, and students, the Do-Pro is proposed as a simple and affordable depth perception camera. Compared to state-of-the-art LIDAR systems that cost tens of thousands of dollars with few examples of mass manufacturing, the Do-Pro boasts a $320 price tag made up of exclusively off-the-shelf components. The Do-Pro can accurately perceive depth from 1-3 feet with approximately 1.5 inches (6%) error, sufficient for ADAS and 3D Scanning applications.**

***Index Terms*—computer vision, embedded systems, image processing, internet-of-things, stereo-vision.**

I. Introduction

STEREO vision cameras are a type of specialized camera that observe a scene from at least two perspectives to extract the depth in a scene. They are popularly employed in problems involving object tracking/detection and 3D reconstruction. Islam et al. [1] developed a stereo vision system for modeling human movement in a 3D-space set-up with two aligned and parallel Go-Pro cameras separated by 1 meter. Wong et al. [2] investigated a navigation aid system using stereo vision that enabled blind participants to understand the distance, size, and movement of obstacles. Achmed et al. [3] proposed a novel stereo-matching algorithm and demonstrated the capability of stereo vision to reproduce 3D scenes using point clouds. These publications show that stereo vision can be a versatile tool for scientists and researchers, but often require complex, technical, problem-oriented configurations that limit the accessibility of these technologies outside of the computer vision community. The Do-Pro is designed as an inexpensive and generic stereo-vision camera to expand the accessibility of this technology.

A. Stereo-Vision

A typical stereo-vision setup (Fig. 1A) consists of two identical cameras on the same longitudinal (z) and vertical (y) axis separated by a set lateral distance. The concept of stereo-vision is to capture a scene from different lateral perspectives, then compute the displacement of each point between the two scenes (Fig. 1B) known as *disparity* [4]. Depth can then be calculated with optical geometry and intrinsic properties of the camera model (Fig. 1C). From a geometric point in the scene, the governing equation to calculate depth is

$$Z = \frac{bf}{x'_L - x'_R} \tag{1}$$

where Z is the perceived depth, b is the baseline lateral distance between the two cameras, f is the intrinsic focal length of the camera, and x'_l and x'_r are the horizontal pixel locations from the left and right images respectively. To generate a depth map, disparity is calculated at all pixel locations shared by the two scenes and then the equation is applied, given that b and f are constants.

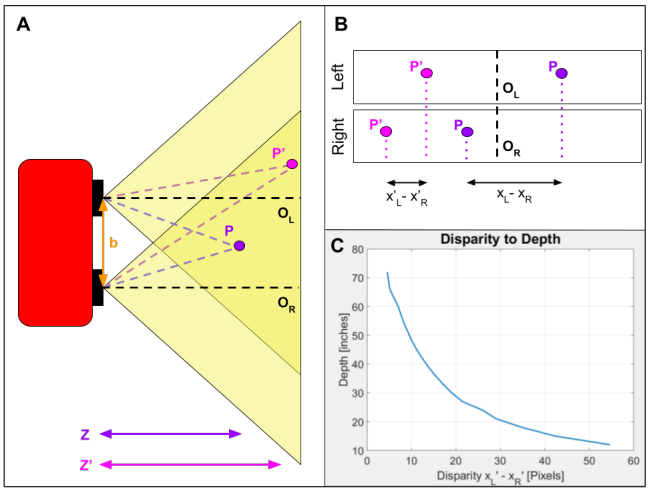

Fig. 1: The Do-Pro Stereo-Vision Concept. (A) Camera capturing scene points P and P' located at depths z and z'. (B) Disparity of point P and P' in the left and right images. (C) Mapping disparity to depth.

B. Stereo Matching Algorithms

To compute depth, the pixel disparity between the images is computed using a stereo-matching algorithm which forms disparity maps. Rule-based stereo matching algorithms can be split into two types, namely local and global matching. Local matching algorithms are window-based techniques that minimize the cost function of disparity in order to compute the disparity map [5]. Global matching algorithms attempt to find a function of disparity that minimizes the energy of the agreement between two images globally and applies a regularization term to smooth the disparity map [5]. Overall, global matching methods provide higher accuracy than local matching methods but require more complex implementations, higher run times, and computational intensity [6]. Both local and global block-matching methods were implemented into the camera. Still, the best overall method for stereo-matching was a hybrid method implemented by OpenCV called the "Semi-Global Block Matching algorithm".

II. Materials & Methodology

A. System & Parts Overview

The software and hardware systems work in conjunction with each other to achieve the functionalities of the Do-Pro. As shown in Figure 2, the hardware system of the Do-Pro compacts a set of embedded webcams to take sets of images with a user interface. The Python-based platform computes the depth map from the image pair and renders various visualizations that can be saved to a gallery for later viewing. The platform is hosted on a git repository [7] to employ version control across multiple operating systems.

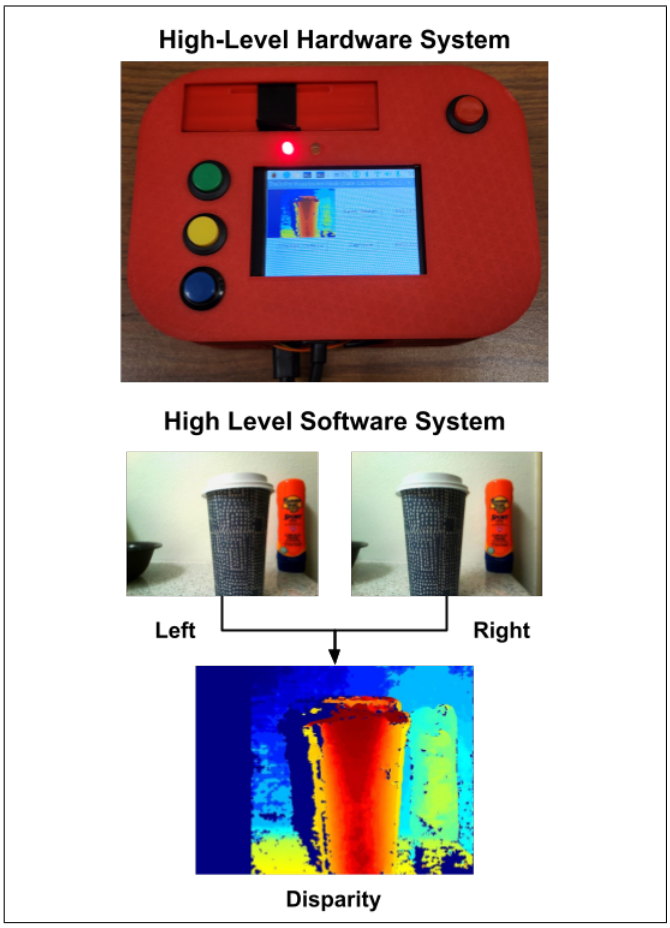

Fig. 2: High Level Project Overview

After finalizing the high-level system requirements, the hardware system is assembled with off-the-shelf components, totaling to an amount of approximately $320 (MSRP; $425 with markups). The components were chosen based on cost, availability, and quality. The most integral components of the Do-Pro are:

- **Raspberry Pi 4 (4GB)** - Control and computing unit for basic camera functionality and image processing
- **UTRONICS Touchscreen Display** - Graphical interface with user input control and image viewing
- **Arducam 8 MP Embedded USB Camera (x2)** - Dual lens and sensor used to capture two images simultaneously at 1920x1080p resolution
- **18650 Battery Cell** - Rechargeable system power supply

B. Hardware

The main hardware deliverables for the Do-Pro project are the 3D-printed enclosure and the custom PCB that connects the stereo-vision system. The enclosure is designed to secure all components into a compact form factor.

1) Custom PCB: The custom PCB is designed in Altium Designer with RoHS-compliant components. The schematic shown in Figure 3 depicts the sub modules and connections that are implemented in the PCB. The PCB connects power to all components, interfaces the touch screen and Raspberry Pi, implements low-battery monitoring, and most importantly minimizes the space required to perform these operations.

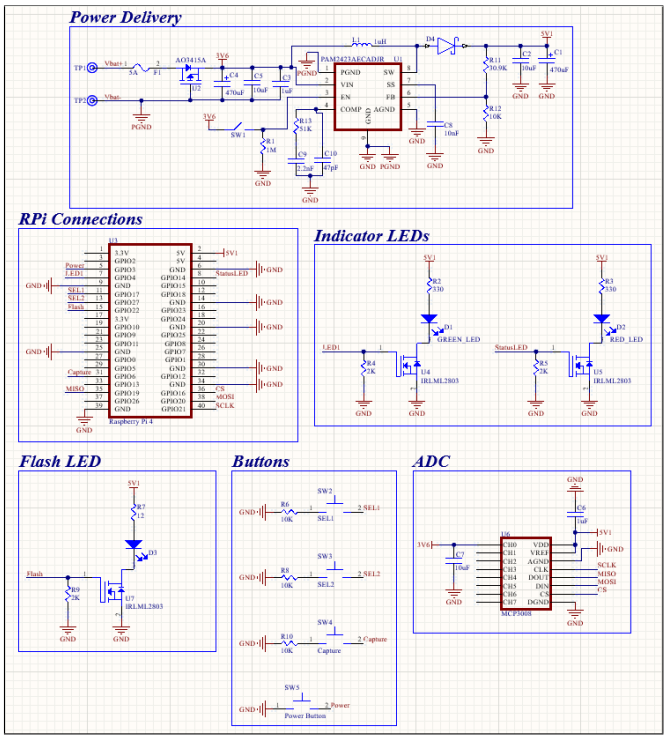

Fig. 3: Custom PCB Schematic

One notable feature of our PCB is that it utilizes a switched-mode boost converter for power delivery. This topology provides more power efficiency for converting 3.6V to 5.1V (90-95%) compared to linear regulators (60-70%) [8]. Additionally, the Raspberry Pi does not switch any load, including LEDs. We followed electronics best practices by switching loads via MOSFETS instead of GPIO pins. The custom PCB is a two-layer board, and the team paid special attention to the data sheet layout recommendations for implementation.

2) Enclosure: The enclosure is designed in Auto-CAD and 3D-printed using the Creality Ender 3 Pro. The design philosophy was to secure all the components through easy assembly but also provide ease of access to the electronics for troubleshooting and/or upgrading. The design focuses on optimizing the human interface for comfort and ease of use.

The material used in the printing process was polyethylene terephthalate glycol (PETG) because of its impact durability and heat resistance. It is also a cost-effective material for

prototyping and does not have any special requirements during the printing process (e.g., heated enclosure or fume extractor).

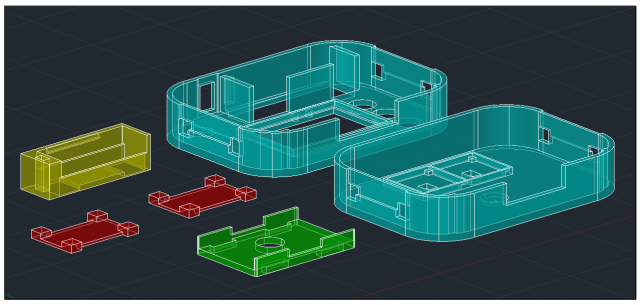

Fig. 4: Camera Housing CAD Model

3) Camera Calibration: To ensure that the disparity is properly computed, the position of the cameras must be along the same vertical (y) and longitudinal (z) axis. In regards to equation (1), b and f must stay constant. The depth equation is rearranged to solve for bf.

$$bf = Z(x_l' - x_r') \tag{2}$$

With our stereo system, a uniform checkerboard pattern is placed in front of the cameras (see Fig. 5). A checkerboard pattern exhibits distinct features in the edges of the grid to match points of correspondence. The cameras are fixed at a stationary position and take a sequence of 50 image pairs, where the checkerboard pattern is oriented and translated over different parts of the frame.

Afterwards, the image pairs are passed through MATLAB's stereo camera calibrator app [9], estimating the intrinsic parameters of our stereo-system for remapping points that violate equation (2). The intrinsic parameters are included in our pipeline to rectify the image before the disparity is computed.

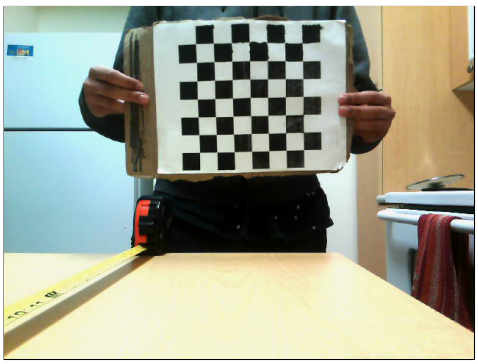

Fig. 5: Scene for MATLAB Checkerboard Calibration

4) Power Regression Model: A power regression model was created to estimate depth from raw disparity output. A flat object (e.g, a pillow) is placed in front of the cameras at a measured distance. Afterward, the disparity map is computed and the output at the center of the object is recorded. The process is repeated for increments of 3 inches from 1-3 feet. The data is finally passed through Microsoft Excel to create a power regression in the form:

$$Z = \alpha(x_l' - x_r')^{\beta} \tag{3}$$

C. Software

1) Algorithms: Several algorithms were implemented on the Do-Pro to perform stereo-matching such as the multi-block matching algorithm proposed by Change and Maruyama [10]. However, here only the best algorithm, semi-global block matching (SGBM) [11], will be explained. To begin, all stereo-matching algorithms require a cost function $C(\cdot)$ which inversely measures the similarity of two image points. By making disparity a parameter of the cost function, the corresponding disparity of each image pixel can be computed by minimizing the cost function. However, flukes can easily occur in the minimization of the cost function resulting in incorrect disparities hence penalty terms are added to increase the cost for differences in the disparity between neighboring pixels. By combining the cost function and penalty terms, the energy function is created:

$$\begin{aligned} E(D) = \sum_p [C(p, d_p) + \sum_{q \in N_p} (P_1 \mathbb{1}\{|d_p - d_q| = 1\} \\ + P_2 \mathbb{1}\{|d_p - d_q| > 1\})] \end{aligned} \tag{4}$$

where D is the disparity map, d_p and d_q are the disparities of pixel p and q, P_1 and P_2 are the penalty weightings, and $\mathbb{1}$ is the indicator function. By minimizing the energy function with respect to the disparity map, a smooth and accurate disparity map is computed. This algorithm is the semi-global matching (SGM) algorithm but can be extrapolated to the SGBM algorithm by extrapolating the equation to compare blocks of pixels instead of individual pixels.

2) User Interface: The Do-Pro incorporates both a touchscreen and a physical button interface. Various touchscreen interfaces (see Fig. 6) are designed in python Tkinter, allowing users to operate the Do-Pro using touchscreen tiles. On the upper left, users have the option to select between the various depth application modes of the Do-Pro. After selecting a mode, a visualization like the bottom will appear, with tiles to capture depth, save images, view gallery, and access settings. The settings, shown in the upper right, allow the user to adjust disparity, rectification, camera exposure, etc.

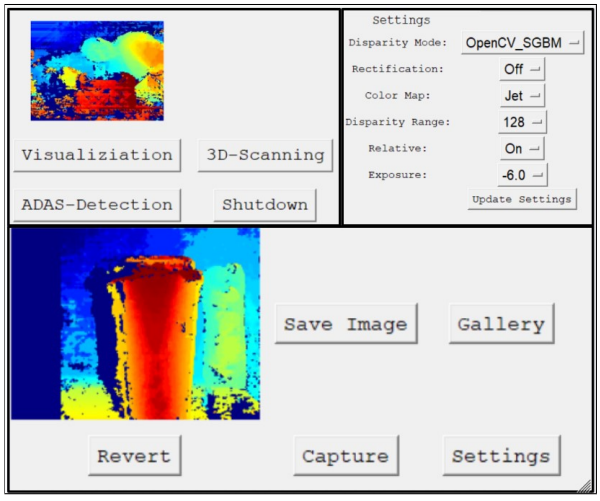

Fig. 6: Various Touchscreen User Interfaces.

The physical button layout is designed for users who want a more traditional photography experience. The inputs available are *Sel1* (toggling between cameras), *Sel2* (Settings), *Capture* (for capturing the disparity), and *Power* (On/Off). All inputs are connected to the Raspberry Pi in a pull-up configuration. The Raspberry Pi reads the button inputs via the *RPi.GPIO* Python library. The digital logic for the Do-Pro button inputs is mapped to functions of the touchscreen interface.

3) Applications: For visualization, the jet colormap is applied to the disparity, where close objects are red and farther objects are bluer. Aside from depth visualization through a heat map, The Do-Pro features pipelines for ADAS (Advanced Driving Assistance) and 3D Scanning.

For ADAS detection, the Do-Pro simultaneously detects close objects and alerts the user by flashing the LEDs rapidly. The Do-Pro computes the disparity colormap (size MxN) and applies K-Means clustering (with K=3) to distinguish regions that are 'far', 'close' and 'in-between'. Afterward, a relative color (R,G,B) & size threshold (larger than 25% of image) is applied to detect if objects are close to the user. A cluster C is classified as close if conditions (5) & (6) are met:

$$|255 - R| \geq 100 \wedge R \geq 1.3 * B \tag{5}$$

$$Size\{C\} \geq 0.25 * M * N \tag{6}$$

For 3D scanning, the Do-Pro first generates the depth map from the disparity using the power regression model described in equation (3). Afterward, a 3D point cloud is reconstructed by superimposing the depth dimension onto the original left image. Finally, the point cloud is voxel downsampled and passed onto Open3D to generate an STL Triangle Mesh via the alpha shape method [12]. Since the Raspberry Pi doesn't support OpenGL, the Do-Pro transfers the stereo-pairs of images via TCP sockets to a workstation to run Open3D's visualizer and convert it into an STL file for 3D printing.

III. Results

A. Hardware

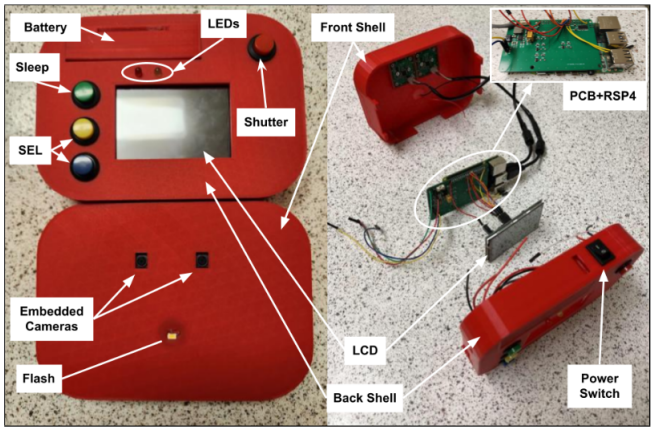

Fig. 7: Camera Exterior (Left), Interior Layers (Right)

To analyze the current design of the enclosure, the design, while larger than the team had expected, was good, but not without flaws. The enclosure was durable and the components met at the desired symmetric faces and planes. Though the design was intended for organized fitment, we realized in hindsight, that some wired connections required more leeway. The video feed between the LCD screen and the Raspberry Pi now protrude from the enclosure. We understand that the sizing error margins of the enclosure structure need to be increased, and some of the structure sections needed to be rearranged and modified overall to better fit the electrical hardware components. After taking various notes of this prototype, future development should presumably be smoother as core fitment issues have been determined. Upon camera assembly, we have opted for 2 fully assembled components Fig. 7). The front, in particular, was determined to be an outstanding part, fitting the lenses perfectly, making for easier camera calibrations. Additionally, a flash LED was added for improvements in picture taking quality. This LED provided consistent lighting which supported the software systems. Both the cameras and the LED were connected to the Raspberry Pi and are securely fitted to this front enclosure. On the other hand, the back enclosure held the user interactive features. This half includes a removable slot for power supply replacements, input control buttons, the LCD screen for UI optical viewing, and the power switch.

After finalizing the camera assembly, we tested its battery life. We installed a fully-charged (4.2V) lithium-ion battery with a capacity of 3200mAh and ran the OpenCV disparity calculations while monitoring the battery voltage. The camera lasted 77 minutes before reaching the cut-off voltage that we set in software (See Fig. 8). We chose 3.2V as the cut-off voltage for the lithium-ion cell as it is a conservative cut-off to prevent the battery from over discharging.

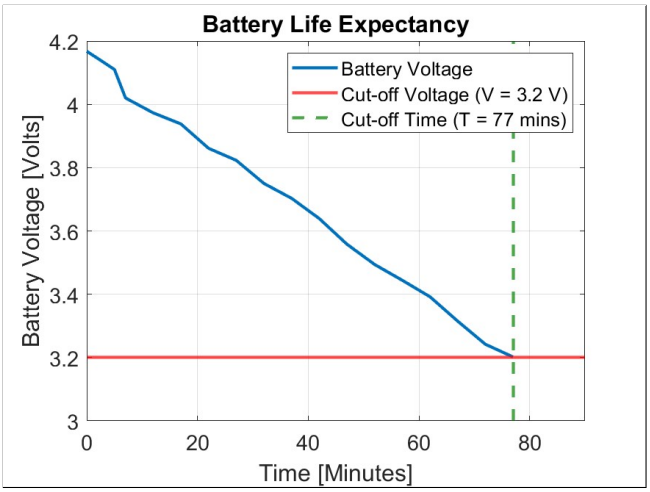

Fig. 8: Battery Voltage under Disparity Load (∼10W)

B. Software

Two metrics can be measured in the disparity algorithm of The Do-Pro: depth accuracy and relative depth perception. We also present the calculation of the power regression model over a feasible range for 3D scanning purposes and visually compared different pipelines for depth perception.

1) Power Regression Model: For the power regression model, the disparity to depth relationship falls closely to the governing stereo equation (1). We decided to use OpenCV's SGBM disparity algorithm to retrieve the power regression model. Notice that measured depth (Z_m) has an inverse relationship with disparity ($x_l' - x_r'$), as shown in Table I:

TABLE I: Measured Depth vs Computed Disparity

$x_l' - x_r'$ **(px)**	Z_m **(in.)**	$x_l' - x_r'$ **(px)**	Z_m **(in.)**
172.375	3	29.0625	21
93.4375	6	26	24
70.4375	9	21.4375	27
54.6875	12	19.0625	30
42.3125	15	17.125	33
34.9375	18	15.375	36

Based on the curve of best-fit (plotted in Fig. 9), the model presents a high correlation between depth and disparity, with an R^2 = 0.9919. From the power regression model, it was found that $\alpha = 616.33$, $\beta = 1.011$, & to fit into eq. (3), which becomes:

$$Z = 616.33(x_l' - x_r')^{-1.011} \tag{7}$$

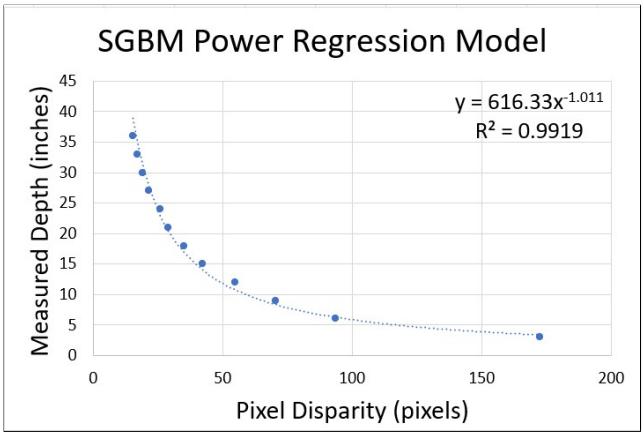

Fig. 9: Power Regression Fit Along Depth-Disparity

2) Depth Accuracy: For determining depth accuracy, we tested our power regression model with a different object (e.g, a mug) and measured the error of the perceived depth (Z_p) from 1-3 ft. The tabulated results are shown in Table II, with an average percent error of 6.2% (1.5 inches):

TABLE II: Measured Depth vs Perceived Depth

Z_m **(in.)**	Z_p **(in.)**	**Error (%)**
12	10.6932	10.89
18	17.909	0.506
24	23.512	2.033
30	31.406	4.687
36	40.608	12.8

3) Visualization Results: For an ideal scene where lighting across the scene is bright and uniform, the Do-Pro performs exceptionally well at computing depth. In Figure 10, the original image and its depth map display an extremely pleasing visual of depth. To achieve this, histogram equalization was applied to the disparity map to spread out the depth information and make it more apparent. In addition, the depth map and original image could be converted into a triangle mesh or applied to ADAS clustering pipeline for detection (with a 90% success rate).

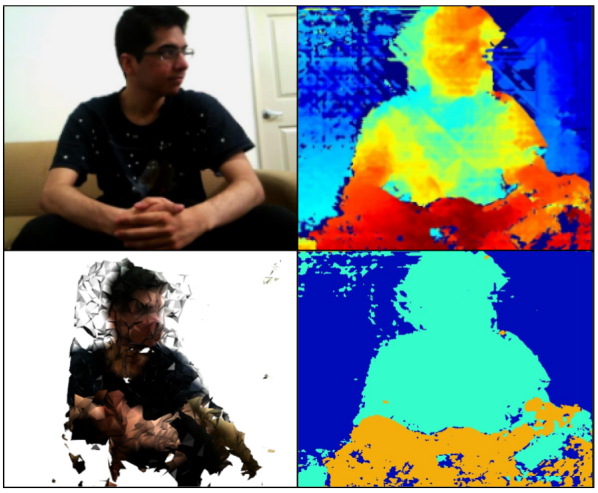

Fig. 10: Input and Outputs For Different Applications: Input (Top Left), Disparity Map (Top Right), 3D Triangle Mesh (Bottom Left), ADAS Clustering (Bottom Right)

IV. Summary

The Do-Pro is a portable stereo-vision system that utilizes interdisciplinary skill sets for its hardware and software components. The 3D enclosure is mechanically designed and printed to optimize dimensions when fitting all the hardware. The custom PCB is designed and fabricated to safely provide battery power to all the hardware. When assembled, the cameras are calibrated to extract intrinsic parameters for depth perception. Finally, the software platform is pushed to the Raspberry Pi to take images for different applications of depth perception.

V. Conclusion

In this paper, The Do-Pro stereo vision camera has been physically assembled and functions similarly to commercial digital cameras. The Do-Pro accurately computes depth and demonstrates the capability to be applied to real-world problems. The low-cost and soft learning curve of the Do-Pro makes it a promising innovation for accessible computer vision technologies.

Acknowledgement

The Do-Pro team would like to thank all of its team members (Timothy, Daniel, Zaya, and Harrison) for dedicating their time to completing the project. The authors would also like to thank our advisor Dr. Glenn Healey, Dr. Stuart Kleinfelder, the teaching assistants, and students of EECS159B for providing feedback and guidance to improve the quality of the project. The Do-Pro project's funding was generously provided by Dr. Kleinfelder. We personally thank Timothy and Zaya for providing the facility of their 3D printers to print the enclosure.

REFERENCES

[1] A. Islam, M. Asikuzzaman, M. O. Khyam, M. Noor-A-Rahim, and M. R. Pickering, "Stereo vision-based 3d positioning and tracking," *IEEE Access*, vol. 8, pp. 138 771–138 787, 2020.

[2] F. Wong, R. Nagarajan, and S. Yaacob, "Application of stereovision in a navigation aid for blind people," in *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, vol. 2, 2003, pp. 734–737 vol.2.

[3] M. H. Achmad, W. S. Findari, N. Q. Ann, D. Pebrianti, and M. R. Daud, "Stereo camera — based 3d object reconstruction utilizing semi-global matching algorithm," in *2016 2nd International Conference on Science and Technology-Computer (ICST)*, 2016, pp. 194–199.

[4] B. Horn, B. Klaus, and P. Horn, *Robot vision*. MIT press, 1986.

[5] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, 2001, pp. 131–140.

[6] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[7] T. Do, D. Jilani, Z. Lazar, and H. Nguyen, ""the do-pro"," 2023. [Online]. Available: https://github.com/dotimothy/TheDoPro

[8] K.-H. Chen, *Design of Switching Power Regulators*, 2016, pp. 122–169.

[9] MathWorks, "Stereo camera calibrator." [Online]. Available: https://www.mathworks.com/help/vision/ref/stereocameracalibrator-app.html

[10] Q. Chang and T. Maruyama, "Real-time stereo vision system: A multi-block matching on gpu," *IEEE Access*, vol. 6, pp. 42 030–42 046, 2018.

[11] R. Mahieu and L. Michael, "Real-time semi-global matching using cuda implementation," 2016. [Online]. Available: https://web.stanford.edu/class/cs231a/prev_projects_2016/semi_global_cs231.pdf

[12] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.