The Do-Pro: A Minimalistic Stereovision 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 improve the accessibility of stereo vision technologies outside of the stereo vision community, the Do-Pro is proposed as a general purpose stereo vision camera resembling common digital cameras. The project aims to minimize the size of the camera by designing a PCB to reduce connections between components, and to provide real-time imaging using a local block matching algorithm that has been optimized using vectorization. Overall the camera has been constructed and is functional, but the implemented stereo matching algorithm still has technical issues.**

***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 modelling human movement in 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 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 outside of the computer vision community. The Do-Pro project aims to design an inexpensive and generic stereo vision camera to expand the accessibility of this technology. To realize the aim of the project, the Do-Pro envisions the following design and application goals over two academic quarters:

- Designing the Do-Pro to be as compact as physically and financially feasible. (Fall 2022)
- Implementing real-time imaging methods for pictures and video. (Fall 2022/Winter 2023)
- Applying the Do-Pro to problems in object tracking and 3D reconstruction. (Winter 2023)

A. Stereo-Vision

A typical stereo-vision setup (see Fig. 1) 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,

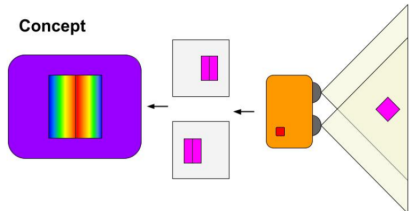

Fig. 1. The Do-Pro Stereo-Vision Concept

then compute the displacement in pixels between the two scenes known as *disparity* [4]. Depth can then be calculated with optical geometry and intrinsic properties of the camera model. From a geometric point in the scene, the governing equation to calculate depth from a stereo-vision system 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 equation is applied, given that b and f are constant in the system.

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]. Given these latter concerns we attempted to implement the algorithm described in Chang and Maruyama [7] which used multi-block matching to yield an average BAD2.0 score of 22.8 on the Middlebury data set. To optimize the process, the implemented algorithm's computational speed was increased by a factor of 10 through vectorization, which achieves stereo vision near real-time.

II. Materials & Methodology

A. System & Parts Overview

The high level 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 touchscreen interface. The sets of images is then feed to the software platform, hosted on the Raspberry Pi 4. The platform, programmed in Python, computes the disparity map and saves the result in a gallery for later viewing.

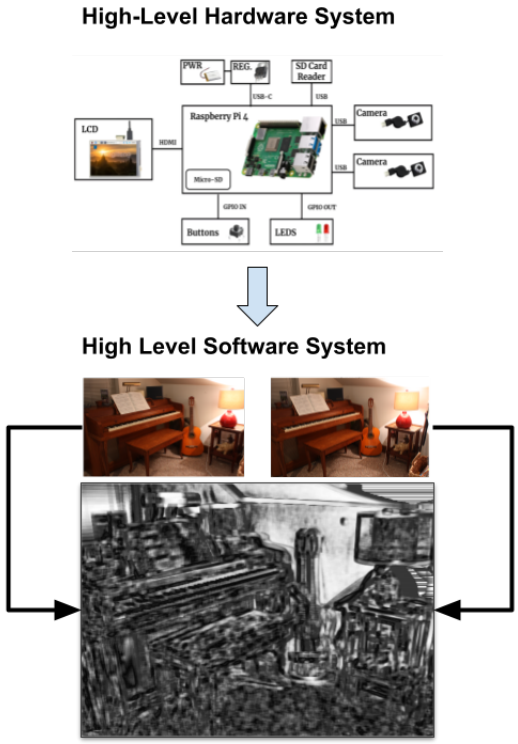

Fig. 2. High Level Project Overview

After finalizing the high-level system requirements, the hardware system is assembled with off-the shelf components, totaling to an amount of approximately $250. The components were chosen based on cost, availability, and quality. The most notable components of the Do-Pro are overviewed:

- **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
- **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. We use the Raspberry Pi 4 to host the platform for computation as well as the user interface. 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.

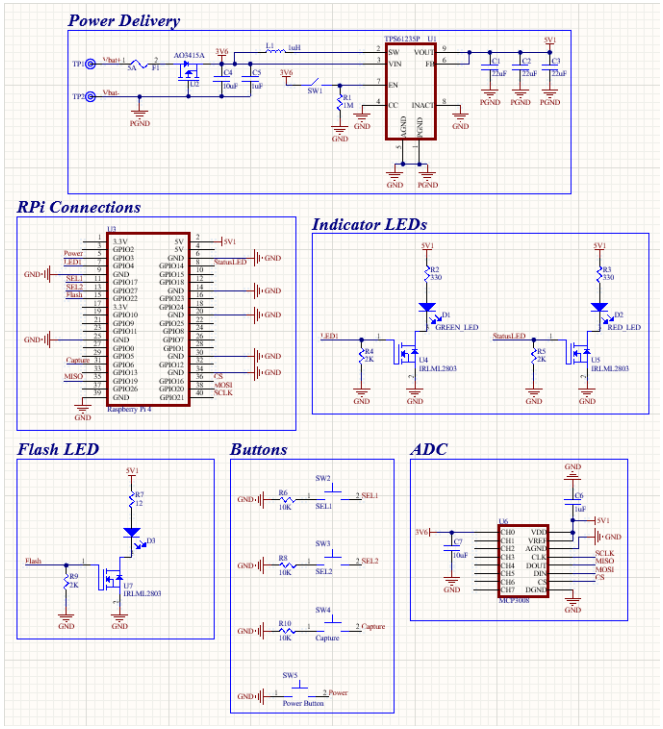

Fig. 3. Custom PCB Schematic

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). PETG was desirable due to its properties of 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).

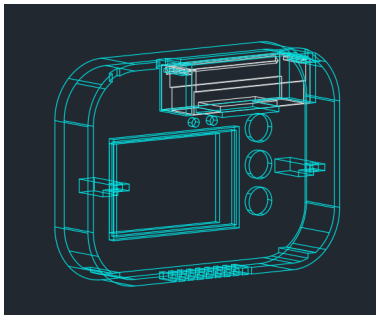

Fig. 4. The Do-Pro Stereo-Vision Concept

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 piece of paper with a dot in the middle is placed in front of the cameras (see Fig 5). The distance between the piece of paper and the lateral axis of the stereo system (Z) is arbitrarily set and recorded. The left and right images are then manually taken and analyzed in an image viewer called WebPlotDigitizer [9] to retrieve the sub-pixel coordinates of the center of the dot to compute disparity. bf is then computed using equation (2). The process is repeated with increasing Z to verify the intrinsic parameters.

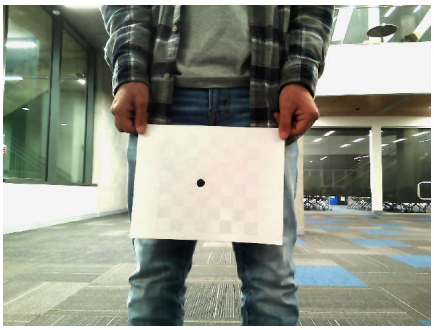

Fig. 5. Scene for Manual Camera Calibration

C. Software

1) Algorithms: The main software deliverable for the Do-Pro project is the stereo matching algorithm for calculating the disparity map of the scene and the computational optimization of the algorithm. Let the left and right images be defined as gray-scale pixel intensities *L(x, y)* and *R(x, y)* where the *x* and *y* correspond to the columns and rows of the image respectively. The matching cost *C(x, y, d)* of the disparity of the two images is calculated using a reference and sliding window *W*. The normalized cross correlation (NCC) matching cost function is described as follows:

$$C(x,y,d) = \frac{1}{\sigma_L(x,y)\sigma_R(x-d,y)} \sum_{x',y' \in W} (L(x',y') - \overline{L}(x,y))(R(x'-d,y') - \overline{R}(x-d,y)) \tag{3}$$

$$\sigma_L(x,y) = \sqrt{\sum_{x',y' \in W} (L(x',y') - \overline{L}(x,y))^2} \tag{4}$$

where W is the window for calculating the matching cost, and $\overline{L}$ and $\overline{R}$ are the averaged window of each respective image. $\sigma_R(x,y)$ is derived by replacing L with R in equation (4), and d is disparity. The cost is then aggregated using the block matching technique:

$$C_b(x,y,d) = \sum_{x',y' \in b(x,y)} C(x',y',d) \tag{5}$$

To improve the sensitivity of the block matching to sharp changes in the image, the multi-block matching method is applied:

$$C_B(x,y,d) = \prod_{b(x,y) \in B} C_b(x,y,d) \tag{6}$$

The block set B contains a square, horizontal rectangle, and vertical rectangle block. Finally, the disparity map of the image is derived by finding the disparity that maximizes the aggregated cost for each pixel:

$$D(x,y) = \arg\max_d C_B(x,y,d) \tag{7}$$

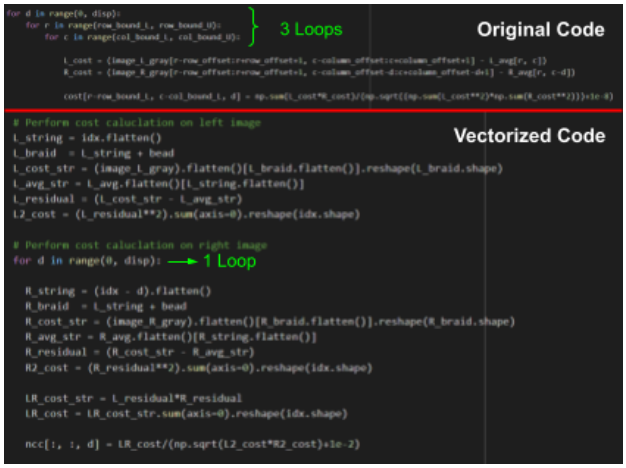

Fig. 6. Code comparison of the original and vectorized stereo matching codes

The algorithm was implemented using *numpy*. To speed up the computation of the stereo matching, the code is vectorized (Fig. 6). This reduces the cost matching for each pixel by instead grouping the pixel computations into a single operation.

2) User Interface: The Do-Pro incorporates both a touchscreen and physical button interface. The touchscreen interface (see Fig. 7) is designed in python Tkinter and shows the live frame of the camera (either raw or computed disparity). The user has multiple touchscreen buttons to interface with the Do-Pro, including viewing the other camera, changing settings, capturing disparity, saving the current image, and viewing those images in a gallery.

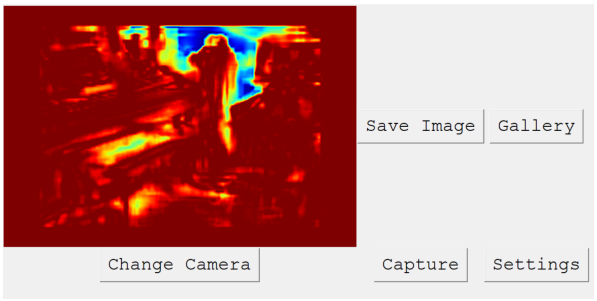

Fig. 7. Touchscreen User Interface with Python Tkinter

The physical button layout is designed for users who want a more traditional photography experience. The inputs available are *Sel1/Sel2* (for switching between the two cameras), *Capture* (for capturing the disparity), and *Power* (for powering on/off). All of these buttons are connected to the Raspberry Pi in a pull-down 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.

III. Preliminary Results

A. Hardware

The assembled prototype is pictured below.

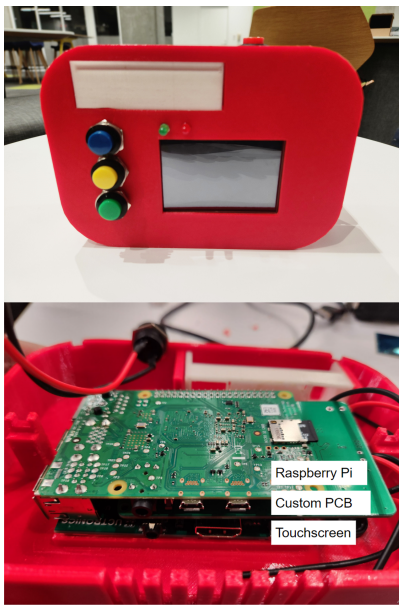

Fig. 8. 3D-Printed Enclosure (Top) with Layer Composition (Bottom) of The Do-Pro

To analyze the current design of the enclosure, the design, while larger than the team had expected, was good, but of course, not without flaws. The enclosure was durable and the components met at the desired symmetric faces and planes. Though the initial design was intended to be nearly perfect for less material consumption, we realized in hindsight, after trying to assemble the camera for the first time, there were clear complications. Some component holding mechanisms were too small and unstable, and various pieces did not align as intended, the locks in particular. We understood that the sizing error margins of the enclosure structure needed 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 first prototype, future development should presumably be smoother as new concept ideas have already been considered.

The front plate, in particular, was determined to be a very outstanding piece, fitting the lenses in perfectly, leading to easy camera functionality.

After assembling the prototype, we calibrated the stereo cameras. To that end, we took images (640x480 resolution) with each camera for each of the six Z-distances and calculated bf from equation (2). We varied Z from 25 to 200 inches. The tabulated results are in Table I.

TABLE I
Intrinsic Parameter Calculations for Different Depths

Z **(in.)**	$x_l' - x_r'$ **(pixels)**	bf **(pixels-in)**
25	59.44	1486.07
50	29.72	1486.07
75	19.81	1486.07
100	13.87	1387
150	9.91	1486.07
200	5.94	1188.85

The intrinsic camera parameters bf stay relatively constant, with a standard deviation of approximately 120. The inverse relationship between the disparity and the perceived depth is linear up to around 100 inches, decreasing rapidly afterward. Perceiving depth from farther distances would be non-linear (i.e., similar disparities map to different distances for background objects).

B. Software

Two metrics can measured in the disparity algorithm of The Do-Pro: computation time and relative depth perception. We measured computation time for different block sizes for OpenCV's cost block matching and qualitatively compared our custom cost block algorithms to OpenCV's algorithm.

TABLE II
OpenCV Computation Time for Different Block Sizes

Block (pixels)	**Time (ms)**	**Block (pixels)**	**Time (ms)**
5x5	5.00	29x29	8.00
9x9	6.00	33x33	6.09
13x13	4.13	37x37	7.01
17x17	4.10	41x41	7.02
21x21	4.02	45x45	6.98
25x25	6.02	49x49	11.10

1) Computation Time: We applied the OpenCV block-matching algorithm to one of the calibration-set images at Z = 50 in. with 64 disparities, shown in Table II. There is a linear correlation that computation time increases proportionally to block size. The trade-off between small block sizes however is the presence of speckle noise. For a balance between optimization and depth resolution, we choose a block size of *13* for OpenCV block matching.

2) Relative Depth Perception: The input scene and the corresponding disparity maps are shown in Fig. 9 through Matplotlib's 'jet' colormap. The OpenCV block matching results is on the top right, NCC Cost Block Matching on the bottom left, and Multiblocking is on the bottom right.

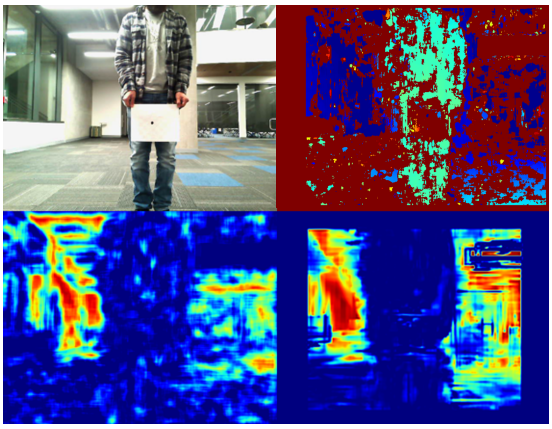

Fig. 9. Disparity Maps for the Different Algorithms (Input: Top Left, OpenCV: Top Right, NCC: Bottom Left, Multiblock: Bottom Right)

Visually, all of the disparity maps are able to relatively distinguish the subject holding the paper and the background. The subject is highlighted in a shade of blue, while the rest of the background is a shade of red. The custom NCC and Multi-block algorithms had more of the background blended it than in OpenCV. In addition to that, OpenCV block-matching has different levels of depth compared to the custom block-matching algorithms seen by the different hues of blue.

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 disparity computation.

V. Conclusion

In this paper, The Do-Pro stereo vision camera has been physically assembled. It has a working touchscreen, user interface, and an embedded stereo system all compacted into a 3D enclosure to capture disparity maps with relative depth perception. Improvements still need to be made in perceiving exact quantitative depth from disparity maps, further optimizing enclosure dimensions and soldering incoming components onto our custom PCB.

Future plans for this device include developing application pipelines for object tracking and 3D-scanning. Specifically, we plan to perform zebra fish movement tracking and model generation for 3D-printing during the Winter 2023 quarter.

Acknowledgement

The Do-Pro team would like to thank all of its team members (Timothy, Daniel, Zaya, Harrison) for dedicating time to the project for Fall Quarter 2022. The authors would also like to thank our advisor Dr. Glenn Healey, Dr. Stuart Kleinfelder, the teaching assistants, and students of EECS159A 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 thank our group members Timothy and Zaya for providing 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] 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.

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

[9] A. Rohatgi, "Webplotdigitizer: Version 4.6," 2022. [Online]. Available: https://automeris.io/WebPlotDigitizer