Robotic Platform for Testing a Simple Stereopsis Network

Shamil S. Patel[1](✉), Jenny C. A. Read[2], Vivek Nityananda[2], and Nicholas S. Szczecinski[1]

[1] Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV 26505, USA
ssp0006@mix.wvu.edu

[2] Faculty of Medical Sciences, Biosciences Institute, Newcastle University, Newcastle Upon Tyne NE2 4HH, UK

Abstract. To better understand how insects use stereopsis to discern the proximity of objects, we previously developed a stereopsis algorithm that was tested in simulation. In the present study, we have begun to implement this algorithm as part of a robot controller. This includes building a small mobile robot, characterizing its cameras, and building a test arena. Implementing the stereopsis algorithm on a robot will test whether the algorithm performs as intended in real-world scenarios and will enable a robot to determine the proximity of objects in its field of view with a low power algorithm.

Keywords: Stereopsis · Stereoscopic vision · Depth cues · Binocular vision · Robotics · Insect vision

1 Introduction

The computational demands of performing stereopsis for mobile robots are very high and may require an impractical amount of computing power. Despite this challenge, vision is a key functionality for many unmanned mobile robots [1, 9]. Robotic vision necessitates real-time operation, meaning that outputs such as object detection, segmentation, and 3D reconstruction must be performed in real-time or faster [10]. One way that engineers try to cope with the complexities of these applications is to train deep neural networks that can replicate these operations, then deploy them onboard robots [1]. However, deploying such networks may require specialized hardware such as large CPUs, GPUs, and batteries that may require space and energy that mobile robots cannot provide [4]. Such limitations require that performance be sacrificed for mobile operation, often resulting in less refined and capable robots [2].

Insects can be used as inspiration for solutions that address these challenges. Insect visual systems possess high temporal resolution while only using low spatial resolution

This work was supported by NSF DBI 2015317 as part of the NSF/CIHR/DFG/FRQ/ UKRI-MRC Next Generation Networks for Neuroscience Program.

A. Hunt et al. (Eds.): Living Machines 2022, LNAI 13548, pp. 195–198, 2022.
https://doi.org/10.1007/978-3-031-20470-8_20

allowing for the rapid detection of colors, polarized light, and geometric patterns [7]. Some insects, such as the praying mantis, have eyes with overlapping fields of view, which has been shown to underlie stereopsis to discern distance of objects from the insect [3]. Furthermore, due to their low spatial resolution and the hardiness of insects, their visual systems are more experimentally accessible than those of other living beings such as mammals.

Modeling of insect vision has been conducted to emulate the neural capabilities of insects [7]. The success of robots such as the Beebot Quadcopter has shown that insect-inspired visual processing is a promising approach for real-time visual processing [8]. However, processing may be too computationally demanding to be performed online, requiring the robot to send video to an offline server, process the data, then send the result back to the robot, which is susceptible to network interruptions.

To move toward a computationally inexpensive way to locate the closest object to a robot using stereopsis, we have begun to embody a model of insect stereopsis in a mobile robot. We designed a robot and a testing arena and show that our hardware measures disparity between its eyes. Our approach differs from previous studies in that it utilizes a simple correspondence-free algorithm which doesn't need to explicitly match pixels between cameras but rather has as "intuitive" understanding of the distance of objects [5]. We discuss how the algorithm will be implemented onboard the robot in the future.

2 Methods

2.1 Robot and Cameras

We built a small mobile robot with which to test the stereopsis algorithm (Fig. 1A). The robot is a Zumo chassis with a Raspberry Pi 3B and two Pixy2 cameras [6] (Fig. 1B) mounted on top. These components are attached via a custom 3D-printed body. Each Pixy2 has wide-angle lens with 87° field of vision and an integrated filtering system that locates specific color hues within the field of view. The Pixy2s are mounted on the robot 3.825 cm apart and 10.9 cm above the arena floor, facing straight forward, resulting in 87° of overlap between each camera's field of view. The Raspberry Pi serves as the "brain" of the robot by receiving visual input and directing behavior by issuing motor commands to the Zumo's Arduino microcontroller. To simplify validation, we constructed a controlled visual environment (CVE) with a uniform background (Fig. 1C). The usable volume of the CVE is 87.6 cm $\times$ 87.6 cm $\times$ 30.5 cm constructed using a series ten 80/20 framework and 6.35 mm-thick white acrylic sheets. The floor of the CVE has a black border approximately three centimeters from the walls, which the robot detects to avoid collision with the walls.

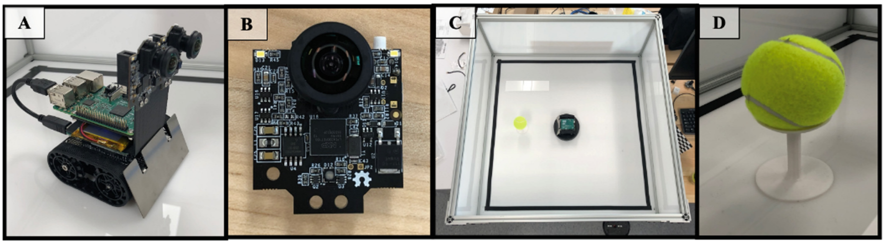

Fig. 1. A) Zumo Robot with attached Arduino R3, B) PixyCam2 with wide-angle lens, C) Controlled visual environment with Zumo Robot, D) "Prey" Targets

2.2 Neural Network

The visual data collected by each individual Pixy2 will be inputted into a neural network that determines the location of the closest landmark in the visual field [6] (Fig. 2A). The network has a winner-take-all (WTA) structure that maps the landmark coordinates from each eye to the azimuthal position of the closest landmark (Fig. 2B). The network's function is the result of its spatially-correlated synaptic weights (Fig. 2C), which in effect solve the correspondence problem [7].

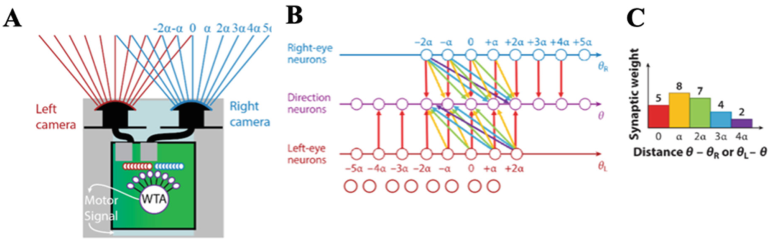

Fig. 2. A) Robotic platform with field of view broken into bins, B) Corresponding links between the two camera bins, C) Synaptic weights generated from links

3 Preliminary Results and Discussion

3.1 Camera Binocular Images

Figure 3 shows two images of variable height targets at varying distances taken from each of the robot's eyes. These images demonstrate that there is noticeable binocular disparity between the cameras, suggesting our camera configuration is sufficient for use with the stereopsis algorithm in question [7].

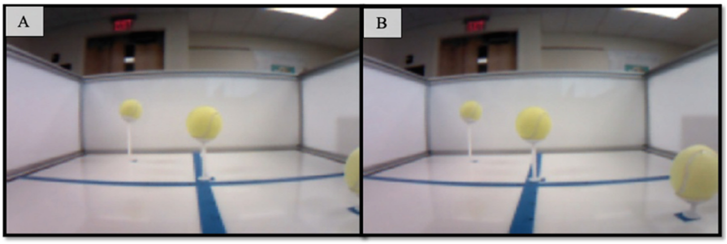

Fig. 3. A) Image captured by "left eye", B) Image captured by "right eye"

3.2 Future Work

With a robot and arena constructed and a control network that functions in simulation [7], our next step is to integrate the network with the robot and evaluate how well the network performs when confronted by real-world conditions, e.g., limited bandwidth and self-motion. The neural network's number of inputs and synaptic weights will be updated to reflect the robot's cameras' resolution, field of view, and interocular distance. The network will be implemented on the Raspberry Pi 3B and will receive input from the Pixy2 cameras. The output of the network will drive saccades of the body to orient the robot toward the closest "prey" target. In the future, we plan to test the "predator" robot's ability to track multiple moving "prey" robots in real-time using our stereopsis model.

References

1. Kakaletisis, E., et al.: Computer vision for autonomous UAV flight safety: an overview and a vision-based safe landing pipeline example. ACM Comput. Surv. (9), 1–37 (Dec 2021)
2. Knight, W.: Why robots and humans struggled with DARPA's challenge. MIT Technol. Rev. https://www.technologyreview.com/2015/06/09/110342/why-robots-and-humans-struggled-with-darpas-challenge/. 2 Apr 2020
3. Giurfa, M., Menzel, R.: Insect visual perception: complex abilities of simple nervous systems. Curr. Opin. Neurobiol. **7**(4), 505–513 (1997)
4. Martin, J.F., Chiang, L.: Low cost vision system for an educational platform in artificial intelligence and robotics. Comput. Appl. Eng. Educ. **10**(4), 238–248 (2002)
5. O'Keeffe, J., Yap, S.H., Llamas-Cornejo, I., Nityananda, V., Read, J.C.A.: A computational model of stereoscopic prey capture in praying mantises. PLoS Comput. Biol. **18**(5), e1009666 (2022). https://doi.org/10.1371/journal.pcbi.1009666
6. Pixy Is the Easiest Way to Add Vision to Your Robot!. PixyCam. https://pixycam.com/
7. Read, J.C.A.: Binocular vision and stereopsis across the animal kingdom. Annu. Rev. Vis. Sci. **7**(1), 389–415 (2021)
8. Sabo, C., et al.: A lightweight, inexpensive robotic system for insect vision. Arthropod Struct. Dev. **46**(5), 689–702 (2017)
9. Susnea, I., et al.: Machine vision for autonomous vehicles – potential and limitations. a literature review. Annals of the University Dunarea de Jos of Galati Fascicle III: Electrotechnics, Electronics, Automatic Control and Informatics **41**(2), 24–30 (2018)
10. Sünderhauf, N., et al.: The limits and potentials of deep learning for robotics. Int. J. Rob. Res. **37**(4–5), 405–420 (2018)