Virtual Neurorobotics Practical Report Group 5

Sebastian Monka
Matriculation number:

Fabian Peller
Matriculation number

Veith Röthlingshöfer
Matriculation number: 1764630
Informatics Master SPO2015

***Abstract*—In this report, we present our results on the three defined challenges: the perception challenge "Thimblerigger", the motion challenge "Lauron QWOP", and the locomotion challenge "Robotic tennis".**

I. Introduction

All experiments run inside the Neurorobotics Platform (NRP) [1]. The challenges should be solved by utilizing spiking neural networks that are built into the NRP. Simulations within the NRP are based on the Gazebo simulator [2].

II. Perception Challenge

A. Challenge definition

The perception challenge is titled "Thimblerigger". An iCUB robot starts in an empty world that contains three red cylinders. A cylinder represents a mug. One of these mugs contains a green ball underneath. The challenge can reveal which mug contains the ball on request. The task is to enable the robot to:

1) Find the mug that contains the ball
2) Track that mug while the mugs are being shuffled
3) Decide which mug contains the ball after shuffling

Fig. 1 displays the setup of the challenge.

In the following, we describe approaches to solve the challenge.

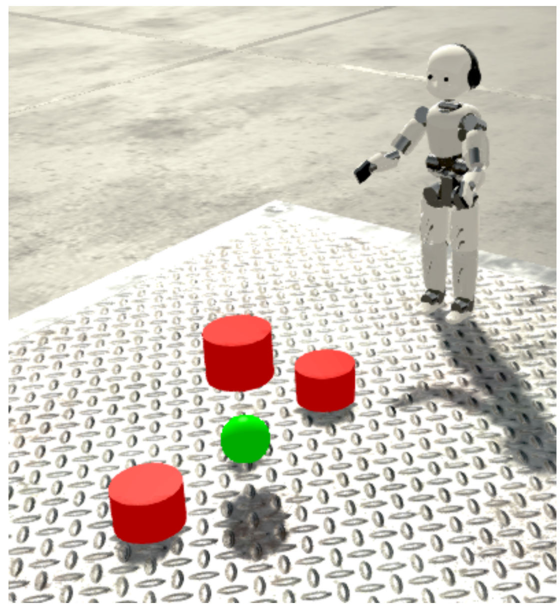


Fig. 1. The perception challenge environment. The second mug is lifted up to display that it contains the ball.

B. Working version

We solve the challenge via a distance-based approach. We move the robot to capture the scene from the side instead of from the front to avoid overlapping mugs in the camera image stream.

We employ four transfer functions to track the ball:

1) Track green: Maps the green channel of the input image stream to three neurons $green_i, i \in \{1, 2, 3\}$.
2) Find initial track: Notices once the correct mug is lifted at challenge start by observing the output spikes of $green_i, i \in \{1, 2, 3\}$.
3) Extract centers: Finds the positions of all three mugs in frame t by finding the center points of the three most prominent red contours in the input image stream. For this reason it is important that the mugs do not overlap themselves in the input image stream. The transfer function keeps track of the center point of the predicted mug in frame $t-1$. The prediction is updated based on the closest distance of the previous prediction to the current centers.
4) Predict: Maps the mug index predicted to contain the ball to one of three $estimate_i, i \in \{1, 2, 3\}$ neurons.

All transfer functions requiring image input from the robot camera rely on the stream from the "left eye" camera. Equations (1) to (10) formalize the tracking mechanism.

In equation (1), we define the amplitude of the $green$ neurons responsible for tracking the green ball to be the mean value of the green channel g in their respective third at time t.

$$amplitude(green_{i_t}) = \begin{bmatrix} \frac{1}{93}\sum_{y=120}^{160}\sum_{x=85}^{137} g(x+y) \\ \frac{1}{93}\sum_{y=120}^{160}\sum_{x=138}^{191} g(x+y) \\ \frac{1}{92}\sum_{y=120}^{160}\sum_{x=192}^{240} g(x+y) \end{bmatrix}_{i_t} \tag{1}$$

Equations (2) through (5) describe how the center points of the mugs are extracted from the input image stream. Here, f is a manually tuned threshold function, r is the red channel of the input image, and b is a structuring element defined over B such that $(b \ominus f)$ implements morphological erosion. C is set of (x, y) coordinates of contour center points in the eroded image ordered by their x coordinate.

$$f(x) = \mathbb{1}\{x >= 150\} \tag{2}$$

$$(b \ominus f)(x) = \inf_{y \in B} [f(x+y) - b(y)] \tag{3}$$

$$C_{outlines} = \{c \in contours((b \ominus f)(x)) : area(c) > 50\} \tag{4}$$

$$C = \{center(c) : x \in C_{outlines}\}_{\leq(x,y)} \tag{5}$$

$$\tag{6}$$

Finally, E_t is the estimated mug index at time t. Time $t = 0$ is defined to be the moment a $green$ neuron spikes for the first time. The rate of the estimate neuron corresponding to the currently estimated index is set to 100.

$$E_0 = \arg\max_i green_{i_0} \tag{7}$$

$$E_t = \arg\min_i ||C_i - E_{t-1}||_2 \tag{8}$$

$$rate(estimate_{i, i \in \{1,2,3\}, i \neq E_t}) = 0 \tag{9}$$

$$rate(estimate_{E_t}) = 100 \tag{10}$$

Using this approach, the challenge is solved close to 100% of trials. It only fails if a glitch occurs inside the simulator during repositioning of the robot, causing it to fall over. This happens in about $\frac{1}{40}$ of trials and is presumably caused by a race condition inside Gazebo, although this could not be verified.

C. Other approaches

Two approaches that utilize the spiking neural network to a higher degree have been tested. However, none of them worked to our satisfaction for different reasons. The following paragraphs describe the approaches and why we believe they fail.

Neural retina: A task that is mainly solved using OpenCV [3] in the woking version is extraction of the center coordinates of the three mugs. We assume that it possible to solve this problem using a neural retina: A grid of n by m neurons that represents the input image. We map the red channel of the camera stream to the retina. The original resolution of the camera stream is 320 by 240 pixels, making a one-to-one mapping of pixel-to-neuron computationally challenging. However, the only requirement for the resolution of the retina is sufficient distinction between the different mugs. This is achievable with a resolution of 40 by 30 neurons, or $\frac{1}{8}$th of each side.

Using an *integrate and fire* model, we arange the neurons' position in the same grid using pyNN's [4] *Space* library and connect neighboring neurons with

1) a static synapse
2) a depressing Tsodyks Markram synapse

We theorize that this connection will focus the highest spiking rate at the center of each mug. To retrieve the current center

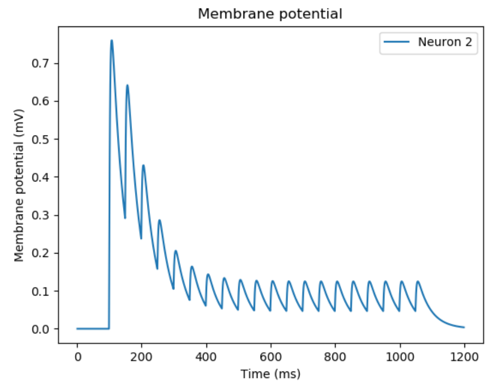


Fig. 2. The characteristic of a depressing Tsodyks Markram synapse. Source: [5]

points of the three mugs, the index of the three neurons with the hightest spiking rate needs to be found. This approach does not work due to two factors. Firstly, computational power: We run this experiment under a local installation of the NRP under Ubuntu 16.4.3 on a Intel Core i7-6700K CPU @ 4.00GHz x 8, 16 GB RAM and a Nvidia GeForce GTX 1070 8 GB VRAM graphics card, which manages to simulate about one second of time in five minutes of real time. This is infeasible to solve the challenge. Secondly, no wokring set of synapse and neuron parameters could be found using manual tuning or a random search. This manifests itself in the retina: If one neuron spikes, the whole net is overloaded immediately when using depressing Tsodyks Markram synapses. We assume that there is a set of parameters that works due to the fading spike characteristic of the depressing synapse (see Fig. 2).

To increase simulation speed, we also test this setup without connecting the neurons to each other. This significantly improves the frame rate. However, the center points can no longer be found via the retina. We argue that it is not important to find the exact center points, as long as any point on each mug is found. Due to the implementation of $\arg\max$ in the mathematics library *NumPy* [6], this approach works as long the three mugs are level and the lighing of the scene is in such a way that there is a brightest spot on every mug in the red channel. When this is the case, we are able to identify the location of the three mugs consistently. However, once a mug moves and one is above or below the others in the robot's view, we lose track as neurons belonging to the same mug become "the most active" due to the evaluation order of $\arg\max$, which chooses the first instance of the maximal value if there are more than one. To aleviate this problem, we try a neural downsampling technique. An additional layer of neurons half the size of the previous retina layer on every side is added, and $4x4$ neuron patches are connected to one neuron in the next layer. This can be interpreted as a $2x2$ convolution with stride 2 over the retina. By downsampling with n such layers until every mug corresponds to a single neuron, it possible to find a unique position for each mug.

This approach is also limited by computational power, and by the fact that at this point the resolution of the center point coordinates is too small to reliably track them using the distance based approach described in section II-B. See Fig. 3 for a visual description of this process. It could be possible to circumvent this issue by placing the robot further away from the mugs inside the simulation, until the size of one mug corresponds to one neuron in the single-layer retina. Due to time constraints, we were not able to evaluate this hypothesis further.

Neural norm:

III. Motion Challenge

IV. Locomotion Challenge

V. Conclusion

The conclusion goes here.

References

[1] "Hbp neurorobotics platform," https://neurorobotics.net/, accessed: 2018-02-11.

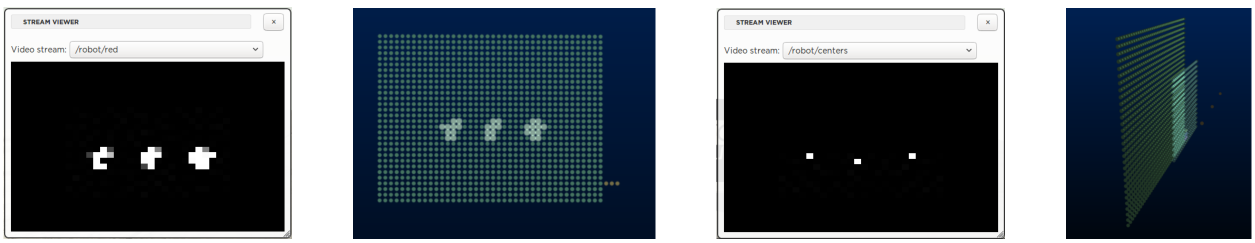
[2] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2149–2154.

[3] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[4] A. Davison, D. Brderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, p. 11, 2009. [Online]. Available: https://www.frontiersin.org/article/10.3389/neuro.11.011.2008

[5] "Nest simulator," http://www.nest-simulator.org/, accessed: 2018-02-15.

[6] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engg.*, vol. 13, no. 2, pp. 22–30, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1109/MCSE.2011.37



(a) Input layer to the neural retina: (thresholded) red color channel of the camera stream.

(b) Mugs being detected on the neural retina.

(c) Extracted center points from the neural retina. It works in this case due to ideal lighing conditions and the mugs being rooughly level.

(d) Downscaling the neural retina by connecting patches of 4 neurons to the next layer.

Fig. 3. Neural retina results