A Complete Neuromorphic Solution to Outdoor Navigation and Path Planning

Tiffany Hwu*†, Jeffrey Krichmar*‡, and Xinyun Zou‡
*Department of Cognitive Sciences, University of California, Irvine
Irvine, California, USA, 92697
†Northrop Grumman Corporation
Redondo Beach, California, USA, 90278
‡Department of Computer Science, University of California, Irvine
Irvine, California, USA, 92697
Emails: hwut@uci.edu, jkrichma@uci.edu, xinyunz5@uci.edu

***Abstract*—Recent developments in neuromorphic engineering have enabled low-powered processing and sensing in robotics, leading to more efficient brain-like computation for many robotic tasks such as motion planning and navigation. However, present experiments in neuromorphic robotic systems have mostly been performed under controlled indoor settings, often with unlimited power supply. While this may be suitable for many applications, these algorithms often fail in outdoor dynamic environments that could benefit the most from the low size, weight, and power of neuromorphic devices. We present the current challenges of outdoor robotics, how current neuromorphic solutions can address these problems, our current approaches to the task, and what further needs to be achieved to create a complete neuromorphic solution to outdoor navigation and path planning.**

I. Introduction

Robotics in outdoor environments presents a complex set of challenges unseen in more controlled indoor settings. Aside from the potential damage of electronic components in harsh weather conditions, the dynamic nature of outdoor environments produces unpredictable sensory input. Changes in lighting add unpredictability to computer vision, and different terrains cause problems for path planning algorithms that assume a uniform space. Furthermore, the lack of a continuous power source limits the time that a robot can be in operation, requiring careful consideration of the power consumption of sensors, actuators, and computational procedures. Neuromorphic systems address the challenge of limited power supply and operation over long durations. Mimicking the energy efficiency of the brain, neuromorphic hardware uses a massively parallel, event-driven architecture that performs computations at magnitudes lower power than the traditional Von Neumann architecture [1], [2]. This is useful for outdoor navigation tasks where energy efficiency can extend the operation time and increase the survivability of robots in resource-scarce areas. For example, conventional path planning algorithms such as A star and Dijkstra's algorithm can be computationally expensive and would benefit from a neuromorphic solution [3], [4]. Existing neuromorphic approaches include a path planner inspired by place cell activation implemented on spiking VLSI neurons [5] and a cost-encoding path planner implemented on a field programmable gate array [6].

Despite the progress in developing navigational strategies for robots in the neuromorphic domain, little work has been done to show that these techniques succeed on embedded systems in unpredictable environments. In order to show the advantages of neuromorphic engineering in robotics, it is necessary to test such applications in true power-limited, dynamic environments. For any autonomous navigation system, the agent must have long-term strategies such as path planning, as well as short-term reactive strategies for obstacle avoidance and road following. In this paper, we describe our current advances in developing an integrated neuromorphic system for performing all aspects of outdoor navigation.

II. Methods

A. Android Smartphone Solution

1) Android-Based Robotics Platform: In creating an integrated neuromorphic system, we needed to create an experimental setup in which long term and reactive motion planning could be isolated and swapped for neuromorphic or traditional implementations. The Android-Based Robotics Platform was created at the University of California, Irvine, consisting of commercial off-the-shelf components [7]. A durable ground robot controlled by an Android smartphone (Figure 1) was ideal for an outdoor neuromorphic system of navigation, as it can run simulations of neuromorphic algorithms and has all of the necessary hardware for communication and localization. Instructions for building the robot can be found at http://www.socsci.uci.edu/∼jkrichma/ABR/. With this platform we were able to test the robot in Aldrich Park, a 19-acre botanical garden at the University of California, Irvine, which contained roads of different widths, terrains, and inclinations.

2) Spiking Wavefront Propagation and Robot Implementation: We created an algorithm for path planning that could be run on neuromorphic hardware [8]. We based our approach on traditional wavefront path planning methods [9], which are ideal due to their parallel and distributed nature. Similar to other neuromorphic wavefront propagation techniques [5], [6], we connected spiking neurons in a topographical map corresponding to locations in 2D space. An efficient path between a start and goal location could be obtained by initiating a spike in the neuron corresponding to the start location, and recording the spike times in an Address Event Representation (AER) table as the neural activity propagated

Fig. 1. Android-Based Robotics Platform. The robot runs on a Dagu Wild Thumper 6-wheel-drive all-terrain chassis, with an SPT 200 pan and tilt to hold the Samsung Galaxy S5 smartphone and control the view of the phone camera. Front-facing MaxBotix LV-MaxSonars can detect obstacles. An ION Motion motor controller and IOIO-OTG microcontroller are housed in the back of the robot. Computing is handled by the Android phone, which accesses the sensors and actuators through a Bluetooth connection with the IOIO-OTG.

to the destination. We could then trace the shortest route by reviewing the spike times of the neurons and determining which sequence of spikes arrived at the destination first. Our novel addition to the wavefront propagation algorithm was the use of adjustable axonal delays to encode the costs traversing through the environment. Inspired by experience-dependent white matter plasticity in the brain [10], the model was able to sample the costs of traversing different parts of the map, incrementally increasing or decreasing the axonal delays accordingly. This is useful for an outdoor environment, as the robot must consider costs such as obstacles, rugged terrain and inclines. The temporal coding of the costs in spike timing allows for a very efficient representation that can take advantage of neuromorphic hardware.

We tested the algorithm on our Android-Based Robotics Platform at Aldrich Park, by running the spiking wavefront propagation algorithm on the smartphone and creating cost maps corresponding to real terrain costs in the park [11]. There were low costs for smooth asphalt, moderate costs for grass, and high costs for obstacles and boundaries. Paths generated by our algorithm were comparable to the non-neuromorphic A star method in terms of generating the shortest and least costly paths. Using the compass and GPS sensors on the phone, the robot then navigated along the calculated path. We then compared the trajectories of the planned route and actual route as recorded by GPS logs on the phone, and found that the robot was indeed able to navigate along the assigned path. Thus, we were able to demonstrate the viability of our neuromorphic path planner in a real outdoor environment, using an easily accessible and affordable robotic platform.

3) Road Following Algorithm: In our robot experiments, the GPS resolution was insufficient to reliably place the robot on the paved paths in Aldrich Park. Therefore, we developed a computer vision road following algorithm for when the robot expected to be on smooth pavement. This required strategies for finding the road from an off-road location and vice versa. From our examination of the literature, we found a lack of techniques to achieve this. Existing work on road tracking usually assumes that the vehicle or the robot is already on the road and that the road has a regular shape with obvious borders so that its two sides can be approximated by a linear lane model [12]. The performance of those algorithms usually relies on obvious color distinction between the road and its surrounding environment and a high-resolution camera which a normal smartphone is not equipped with to generate video frames of appropriate contrast and exposure [13]. Their road tracking results can be negatively affected by shadows or obstacles on the road [14]. Therefore, we developed a road following algorithm (Figure 2) using the image processing class provided by OpenCV [15]:

1) Starting with a camera frame from the phone, we used GaussianBlur to remove noise from the image and converted the image from RGB to grayscale.
2) We applied Eq. (1) [16] to adjust contrast and brightness of the grayscale source image pixels $f(i,j)$:

$$g(i,j) = \alpha \cdot f(i,j) + \beta \tag{1}$$

where $g(i,j)$ is the output image pixel, and the parameters $\alpha > 0$ and β are the gain and bias [16].
3) We used Sobel approximation to estimate the gradient and find edges in the source image.
4) We thresholded pixels ranging from 0 to 255 to binary values. The output image pixels for the road were desired to have the value of 0.
5) We used dilation to expand gaps in the non-road regions with an elliptical kernel. Then each path pixel was chosen at the middle of each row's largest black segment of pixels to make sure the robot move along the road center.

By computing the difference between the mean x value of the detected path pixels and the center pixel value of the camera screen, the robot could determine whether it should move forward, left or right. The new road following algorithm adjusted the image contrast to highlight the road and removed shadows; thus, it may be applied to many outdoor road following cases. The robot would then switch between the original

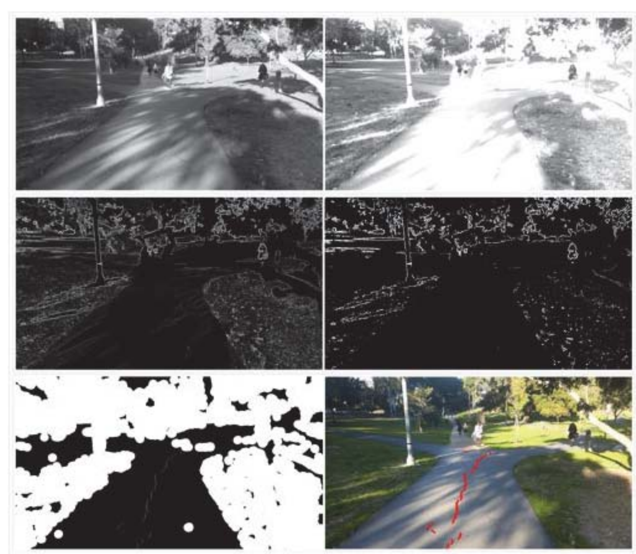

Fig. 2. Road following algorithm. Upper Left: Step 1. Gaussian blur and conversion from RGB to grayscale image. Upper right: Step 2. Contrast and brightness adjustment. Middle left: Step 3. Sobel gradient approximation. Middle right: Step 4. Binary thresholding. Lower left: Step 5. Dilation and path labeling. Lower right: Path display in red on the original test image.

navigation mode and the road following mode depending on whether its current destination along the route was on-road or off-road. In order to find the road from an off-road location, the robot would steer to match its compass direction with the direction of the destination until road pixels could be detected.

III. Results

We added our road following strategy to the spike-based path planning algorithm. We tested two routes, one requiring the robot to start off-road and end on-road, and another starting on-road and ending off-road. Figure 3 shows a sample run of the algorithm. We calculated the percentage of route length that the robot stayed on the path when it was supposed to from video footage and GPS logs. For the route starting off-road, the original implementation stayed on the road 51.2% of the time it was supposed to whereas the road-following implementation stayed on the road 63.2%. For the route starting on-road, the original implementation stayed for 41.2% while the road-following implementation stayed for 87.3%. We compared the Fréchet distance [17] between the actual and planned routes. For the route starting off-road, the distance of the trajectory was 15.8 m from the original implementation and 18.4 m from the road following strategy. For the route starting on-road, the distance was 13.9 m for the original and 15.5 m for road following. These initial metrics show that the robot was able to decrease physical costs of navigating difficult terrains by combining reactive planning with long-term path planning. We hope to perform further experiments to confirm the strengths of our road following approach.

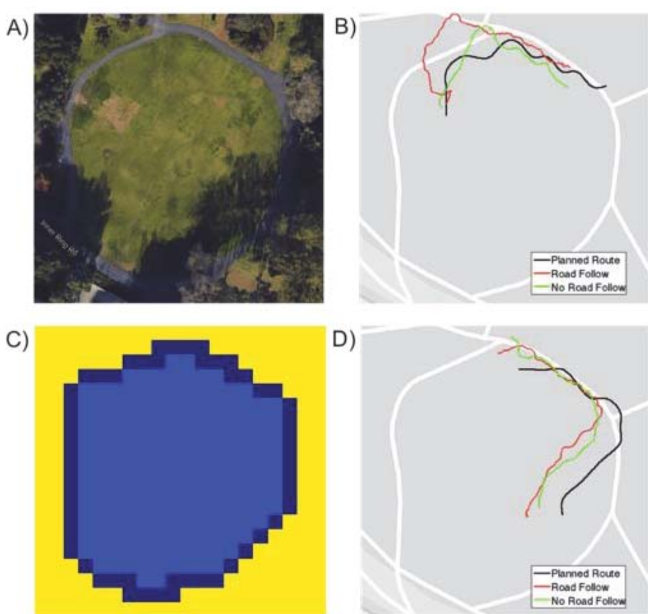

Fig. 3. A) Satellite image of section of Aldrich Park used for mapping. B) Route starting off-road and ending on-road. Black line indicates route assigned by path planner. Red line indicates trajectory taken by road following strategy. Green line is trajectory from original strategy. C) Cost map of area. The road was assigned with a low cost, the middle grass area a medium cost and outer areas high costs. D) Same as B but starting on-road and ending off-road.

IV. Future Developments

In future iterations of this work, the robot will learn costs from the environment using information gathered from its sensors. Rather than using an a priori map of the environment, it would learn the environmental costs through active sensing to discover which areas could be potentially damaging or particularly efficient to get to the destination. We further hope to convert this approach to run mainly on neuromorphic hardware, bringing us closer to an energy-efficient solution for outdoor navigation.

A. An Integrated Neuromorphic Hardware Solution

An integrated solution for navigation requires neuromorphic implementations of path planning and reactionary motion planning such as obstacle avoidance and path following. Many of these projects are already underway, and could replace some of the non-neuromorphic aspects of our Android smartphone approach. For path planning, our spiking wavefront path planner has been implemented on the IBM TrueNorth chip [18], which could be integrated with our robotics platform to encode the costs of a map in real time. The complete spike wavefront and path readout are performed on the hardware itself, using time buffers to delay spikes according to costs. The algorithm could also be implemented on hardware that directly supports axonal delays [19].

The IBM TrueNorth has also been used for reactionary motion planning. The IBM NS1e board, which contains a TrueNorth chip of 4096 cores [20], can be powered by an external battery and is compact enough to fit on our current robotic platform. We recently built a communication pipeline between the NS1e and the Android smartphone using a Wi-Fi connection and mobile hotspot [21]. As proof-of-concept, we physically attached the NS1e board to the Android-Based Robotics Platform and powered it with the same 7.2V NiMH battery used to power the robot. We next trained the system to perform path following with neuromorphic vision processing. This was done by initially driving the robot by hand along a steep mountain trail while saving the video frames and corresponding commands, then using this dataset to train an Energy-efficient deep network (Eedn) [22], and transferring the connection weights to the TrueNorth. Then, the network was run in real time as the smartphone transformed current video frames into spiking format, sending the data through the Wi-Fi connection. The output spikes of the TrueNorth were then sent back to the phone, which used this information to determine whether to steer the robot left, right, or forward (Figure 4A). The trained network correctly classified 90% of its test data, and the robot was able to autonomously follow the steep mountain road (Figure 4B). For even more efficiency of processing, the smartphone camera could be replaced with a dynamic vision sensor mounted on the robot [23]. This way, the image data would already be in an efficient spiking format and save the energy of running a traditional frame-based camera.

As both path planning and road following have been implemented on the IBM TrueNorth, we envision a complete neuromorphic navigational system in the not too distant future. As the path planning algorithm simulates roughly one neuron per 10 m^2 in our experiments, it takes only a small number of neurons on the TrueNorth to work on a large map. Running a deep convolutional network for road following or obstacle avoidance could therefore be performed simultaneously on the unused portion of the TrueNorth, with both processes

Fig. 4. A) The IBM TrueNorth was trained to classify frames into three classes of steering commands, left, right, and forward. B) The neuromorphic road following system traveled autonomously on this narrow mountain path.

communicating through Wi-Fi with the Android phone. We would therefore have all the computationally intensive tasks of outdoor navigation handled by the energy-efficient hardware.

V. Conclusion

The challenge of robotics today lies in the development of algorithms that work in unpredictable environments such as an outdoor setting. We have shown a feasible method of offloading the computation necessary for navigation onto energy-efficient neuromorphic hardware. Our Android-Based Robotics Platform allows mapping, localization, path planning, and reactive controls to be implemented with traditional or neuromorphic methods. With recent developments in neuromorphic implementations, soon we may be able to replace all navigational functions on the Android-Based Robotics Platform to form a complete neuromorphic system for outdoor navigation and path planning. This would lead to practical advantages in the real world. For instance, the low size and energy efficiency of neuromorphic hardware allow for teams of small nimble robots that can explore unknown territory for long amounts of time. With the smartphone interface, the robots may communicate through an ad-hoc network to coordinate the search and rescue of survivors in disaster recovery.

Acknowledgment

Supported by the National Science Foundation Award #1302125 and Northrop Grumman Corporation. We also would like to acknowledge the Telluride Neuromorphic Cognition Engineering Workshop, The Institute of Neuromorphic Engineering, and their National Science Foundation, DoD and Industrial Sponsors.

References

[1] C. Mead, “Neuromorphic electronic systems,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[2] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, “Neuromorphic silicon neuron circuits,” *Frontiers in neuroscience*, vol. 5, p. 73, 2011.

[3] S. M. La Valle, “Motion planning,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 108–118, 2011.

[4] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando, “Path planning in construction sites: performance evaluation of the dijkstra, a, and ga search algorithms,” *Advanced Engineering Informatics*, vol. 16, no. 4, pp. 291–303, 2002.

[5] S. Koul and T. K. Horiuchi, “Path planning by spike propagation,” in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[6] S. Koziol, S. Brink, and J. Hasler, “A neuromorphic approach to path planning using a reconfigurable neuron array ic,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2724–2737, 2014.

[7] N. Oros and J. L. Krichmar, “Smartphone based robotics: Powerful, flexible and inexpensive robots for hobbyists, educators, students and researchers,” Center for Embedded Computer Systems, University of California, Irvine, Irvine, California, Tech. Rep. 13-16, 2013.

[8] J. L. Krichmar, “Path planning using a spiking neuron algorithm with axonal delays,” in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 1219–1226.

[9] M. Soulignac, “Feasible and optimal path planning in strong current fields,” *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 89–98, 2011.

[10] R. D. Fields, “A new mechanism of nervous system plasticity: activity-dependent myelination,” *Nature Reviews Neuroscience*, vol. 16, no. 12, pp. 756–767, 2015.

[11] T. Hwu, A. Y. Wang, N. Oros, and J. L. Krichmar, “Adaptive robot path planning using a spiking neuron algorithm with axonal delays,” *IEEE Transactions on Cognitive and Developmental Systems*, 2017.

[12] H. Kong, J.-Y. Audibert, and J. Ponce, “General road detection from a single image,” *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[13] V. Marion, O. Lecointe, C. Lewandowski, J. G. Morillon, R. Aufrere, B. Marcotegui, R. Chapuis, and S. Beucher, “Robust perception algorithms for road and track autonomous following,” in *Proc. SPIE*, vol. 5422, Unmanned Ground Vehicle Technology VI, 2004, pp. 55–66.

[14] D. Lieb, A. Lookingbill, and S. Thrun, “Adaptive road following using self-supervised learning and reverse optical flow,” in *Proceedings of Robotics: Science and Systems*, 2005.

[15] G. Bradski, *Dr. Dobb’s Journal of Software Tools*.

[16] OpenCV, “Changing the contrast and brightness of an image,” http://docs.opencv.org/2.4/doc/tutorials/core/basic_linear_transform/basic_linear_transform.html, 2016, [Online; accessed 11-Dec-2016].

[17] H. Alt and M. Godau, “Computing the fréchet distance between two polygonal curves,” *International Journal of Computational Geometry & Applications*, vol. 5, no. 01n02, pp. 75–91, 1995.

[18] K. D. Fischl, K. Fair, W.-Y. Tsai, J. Sampson, and A. Andreou, “Path planning on the truenorth neurosynaptic system,” in *2017 IEEE International Symposium on Circuits and Systems*. IEEE, 2017.

[19] J. M. Cruz-Albrecht, M. W. Yung, and N. Srinivasa, “Energy-efficient neuron, synapse and stdp integrated circuits,” *IEEE transactions on biomedical circuits and systems*, vol. 6, no. 3, pp. 246–256, 2012.

[20] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[21] T. Hwu, J. Isbell, N. Oros, and J. L. Krichmar, “A self-driving robot using deep convolutional neural networks on neuromorphic hardware,” in *2017 International Joint Conference on Neural Networks*. IEEE, 2017.

[22] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M. D. Flickner, and D. S. Modha, “Convolutional networks for fast, energy-efficient neuromorphic computing,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 41, pp. 11 441–11 446, 2016.

[23] J. Conradt, F. Galluppi, and T. C. Stewart, “Trainable sensorimotor mapping in a neuromorphic robot,” *Robotics and Autonomous Systems*, vol. 71, pp. 60–68, 2015.