CSC2529 Project Report: Spiking neural networks for depth estimation

Martin D. Pham

December 9, 2022

Abstract

We apply different neuronal models of membrane potential voltage to the residual bottleneck block of a U-Net SNN for depth estimation.

1 Introduction

An important sensing feature of many cognitive systems is vision. Artificial intelligence systems taking inspiration from natural vision systems obtain state-of-the-art performance (e.g. depth estimation [PTB+22]) but lack biological plausibility: biological vision systems involve a heterarchy of physical, chemical, protein and genetic regulatory dynamics obviously unaccounted for in a frame-based model of vision. That is, they are not simulations of an *in vivo* biological process of vision perception. Spiking neural networks (SNN) are an area of research addressing this biological fidelity seeking more accurate models of neurons and action potential [HPGBF18, NCCC22]. SNNs model the dynamics of biological neural networks communicating action potentials via synapses as differential equations representing membrane voltage potential. Neuronal models can be categorized in two ways: pointwise (where the dendrites, somas and axons are treated as occupying the same point in space) or compartmental (where dendrites, somas, axons are treated separately and coupled via state variables in the neuronal system of equations). The connection of these neurons via synaptic models form a network topology. Different topologies and synaptic weights produce different neural representations and processing of an input field over time (e.g. light for vision systems, chemical neuroreceptors for olfactory, sound for auditory). SNNs differ from typical artificial neural networks (e.g. multilayer perceptrons, convolutional neural networks) because there is an assumed temporal dimension to neuronal dynamics [DF21]. They provide a membrane potential account of neural circuitry and therefore may be used to test theories of embedded cognition (i.e. cognition emerges from the interplay of an embrained body and embedded in an environment) such as neuroconstructivism [WMJ+07]. However, the relationship between structure (i.e. topologies) and function (i.e. neural activity) is not well understood in biological brains. The purpose of this paper is to apply a SNN model to stereo depth estimation.

2 Related work

For vision, we may consider a simplified version of the human visual pathway: optical lens controlled by ciliary muscles to photosensitive retina neurons (where light is encoded into neuronal spikes) down optic nerve across chiasmus through lateral geniculate nucleus into striate cortex, as well as feedback throughout. [ZDL+20] presents a retina-like spiking neural network for image reconstruction. [RD20] makes use of known areas of the brain along the human vision pathway to construct a topologically analogous SNN network that models spatial and visual mental imagery. [BSGB+17] similarly organizes its neural architecture based on known anatomical structure, but specifically details the lateral geniculate nucleus [GKRL17], a subregion of the thalamus involved in vision, treating other neural populations as retinal neurons or interneurons. Such large scale network models may be useful for ablation experimentation where lesions (e.g. reducing number of neurons in a population, inhibiting synaptic dynamics, etc.) at particular points along the pathway are known to produce particular patterned artifacts in the reconstructed image, and therefore may be compared to empirical evidence. This may be seen as a perturbation of model parameters to observe change in model behaviour (e.g.

how is the reconstructed perceived visual field affected by reducing the number of neurons within a population group along the visual pathway? Or, how is the performance of a task affected by changing the neuronal model?) [CDGM22]. Mesoscopic-level perturbations include lesioning areas (represented by anatomical subnetworks/modules) such as the retina (i.e. some percentage of either cones and/or rods are dropped/inhibited) to observe the degraded performance of image reconstruction/task. A lower-level perturbation of research interest is the neuronal model, as many large-scale simulations make use of simplified/reduced models in order to be computationally tractable.

Artificial photosensitive spiking retinal networks relate to a new kind of camera sensor called event-based cameras or dynamic vision sensors (DVS). See [GDO$^+$22] for an extensive review of event-based vision problems and algorithms. A review of event-based camera and spike-aware algorithms for depth estimation can be found in [FLB22]. Event cameras code visual information as discrete events (e.g. changes in light intensity) concurrently over a field of independent photosensitive neuronal models and require spiking algorithms to reconstruct and process the image. For this reason, SNNs are comportable with these event-based retinomorphic sensors that more closely capture the photosensitive action potential dynamics of biological vision systems. The exploitation of event-based representation is demonstrated in [RCI21] where two streams of event camera data are passed into an SNN architecture composed of two cooperative populations (one for coincidence, one for disparity), producing instantaneous stereo depth perception with real-world stimuli. The work of [OIBI17] similarly uses two spiking neuron populations connected to two neuromorphic cameras for solving the stereo correspondence problem. However, hardware for event-based cameras can be prohibitively expensive and so algorithms trained on preexisting datasets with estimated ground truths is an alternative. One such example is the Multivehicle Stereo Event Camera (MVSEC) dataset [ZT$^+$18] which provides event-based data for 3D perception tasks. The work of [RCCM21], called StereoSpike, makes use of this dataset to do depth estimation using SNN-type neural processing in a UNet-like encoder-decoder architecture. For spiking data and networks, rather than a single pass of information as in a convolution, the network is always ”on” and processing changes to the current neural representation as opposed to recording raw values in conventional camera systems. Further challenges arise when attempting to use conventional image datasets such as the 2021 Middlebury Stereo Dataset [SHK$^+$14] because an encoding scheme is required to convert input data into the spike domain. A comparative review of encoding schemes for spiking neural networks can be found at [GFES21].

3 Theory

We extend the work of StereoSpike [RCCM21] by changing the spiking neuronal model at the residual bottleneck. This is of interest because there is a trade-off between fidelity to explainable reproduction of biological features versus computational tractability of simulations [Izh04]. We describe the StereoSpike depth estimation for the MVSEC dataset [ZT$^+$18] as follows.

Two event camera sensors with channel dimensions $(H, W) = (346, 260)$. For each sensor there are two channels: one for positive log changes in intensity, another for negative. The input $f \in \mathbb{R}^{4\times H\times W}$ is downsampled by five blocks, passed into two residual blocks (the U-net bottleneck, two blocks of size $4 \times H2^{-5} \times W2^{-5}$), and upsampled back. Convolutions were done by 2-strided 7-wide kernels. The depth map is produced by connecting the upsampling layers from different scales to interneurons and training a weight tensor to produce a single image of size (H, W) that estimates depth-per-pixel. Learning was done using surrogate gradient descent where *arctan* was chosen for its smooth approximation to the Heaviside activation function of spiking neural networks. A combination of regression loss and a smoothness regularizer are used during learning. For the outputted estimated depth map u and residual $R(u) := u - u_{groundtruth}$, consider the losses:

$$L_{regression} = \frac{1}{n}(\sum_u (R(u))^2 - \frac{1}{n^2}(\sum_u R(u))^2)$$

$$L_{smooth} = \frac{1}{n}\sum_u (|\nabla_x R_x(u)| + |\nabla_y R_y(u)|)$$

$$L_{total} = L_{regression} + \lambda L_{smooth} \text{ where } \lambda = 0.5$$

We consider four spiking neuronal models implemented by the SNN package SpikingJelly [FCD+20]: integrate-and-fire, parametric leaky integrate-and-fire, quadratic integrate-and-fire, exponential integrate and fire.

Integrate-and-fire (IF), an ideal integrator:

$$V[t] = V[t-1] + X[t]$$

$$\text{if } V > V_{threshold}, \text{ then } V \leftarrow V_{reset}$$

$$V_{reset} := \text{reset voltage after spike}$$

$$X[t] := \text{(integrated) input at time } t$$

$$V_{threshold} := \text{voltage spike threshold}$$

Parametric leaky integrate-and-fire (PLIF), a:

$$V[t] = V[t-1] + \frac{1}{\tau}\big(X[t] - (V[t-1] - V_{reset})\big)$$

$$\text{if } V > V_{threshold}, \text{ then } V \leftarrow V_{reset}$$

$$V_{reset} := \text{reset voltage after spike}$$

$$X[t] := \text{(integrated) input at time } t$$

$$V_{threshold} := \text{voltage spike threshold}$$

$$\frac{1}{\tau} := Sigmoid(w), w \text{ a learned parameter}$$

Quadratic integrate-and-fire (QIF):

$$V[t] = V[t-1] + \frac{1}{\tau}\big(X[t] + a_0(V[t-1] - V_{rest})(V[t-1] - V_c)\big)$$

$$\text{if } V > V_{threshold}, \text{ then } V \leftarrow V_{reset}$$

$$V_{rest} := \text{resting potential of membrane}$$

$$\tau := \text{membrane time constant}$$

$$V_{threshold} := \text{neuron threshold voltage}$$

$$0 < a_0 := \text{quadratic term parameter}$$

$$V_{reset} := \text{neuron reset voltage}$$

$$X[t] := \text{(integrated) input at time } t$$

$$V_c := \text{critical voltage threshold by short current pulse}$$

Exponential integrate-and-fire (EIF):

$$V[t] = V[t-1] + \frac{1}{\tau}\big(X[t] - (V[t-1] - V_{rest}) + \Delta_T exp(\frac{V[t-1] - \Theta_{rh}}{\Delta_T})\big)$$

$$\text{if } V > V_{threshold}, \text{ then } V \leftarrow V_{reset}$$

$$V_{rest} := \text{resting potential of membrane}$$

$$V_{reset} := \text{reset voltage after spike}$$

$$X[t] := \text{(integrated) input at time } t$$

$$V_{threshold} := \text{voltage spike threshold}$$

$$\tau := \text{membrane time constant}$$

$$\Delta_T := \text{sharpness parameter for exponential}$$

$$\Theta_{rh} := \text{rheobase (membrane potential excitability) parameter}$$

Neuronal model	Average epoch time (s)
IF	179.7935849768775
PLIF	197.56847542354038
QIF	181.4140674148287
EIF	183.87649503435406

Table 1: Average epoch training time over 70 epochs for each neuronal model

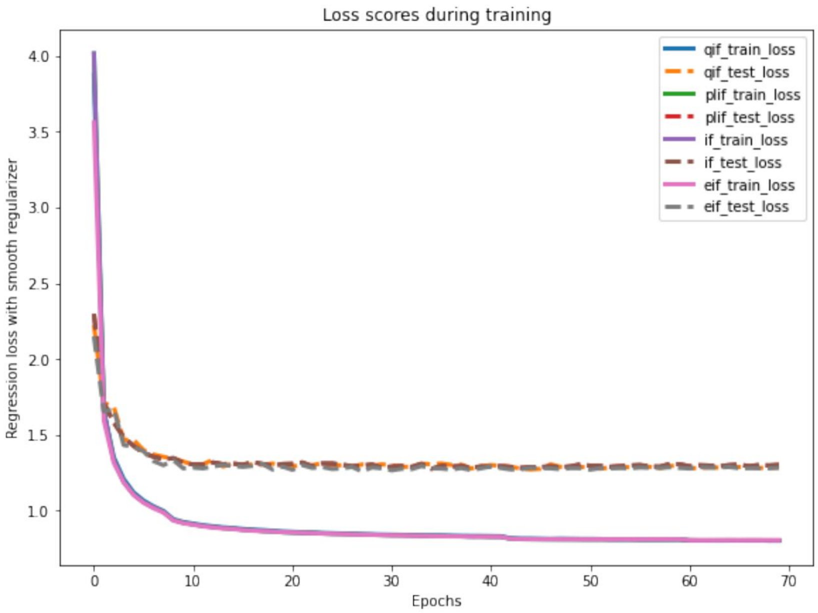

Figure 1: Losses were computed for each epoch and compared between training/testing sets

4 Results, Analysis and Evaluation

Code was forked from [RCCM21] and is available at `https://github.com/mdpham/StereoSpike`. The file of interest for this report is the *network/blocks.py* file where different neuronal models are chosen for the *SEWResBlock* class. The script *train.py* runs the learning algorithm and logs into the *results* folder for post-processing. All experiments were run on a 32GiB Memory, AMD Ryzen 9 3900x 12-core processor x 24 Ubuntu 20.04.3 LTS machine equipped with an NVIDIA GeForce RTX 2080 Ti. Preliminary experiments on the first generation Mac Mini M1 took approximately 5 hours for each epoch.

Table 1 shows runtimes for different neuronal models. Similar runtimes and performances are not surprising given that only a very small part of the model has been changed (the bottleneck neurons, with the fewest neurons of any blocks). It should be noted that the default neuronal model of PLIF used by [RCCM21] has a trained parameter that may contribute to the higher training time. More thorough profiling over statistically large samples is necessary in order to better gauge the effect of changing such a small number of neurons in the overall model.

Figures 1 and 2 show losses and depth errors (respectively) over the training phase of 70 epochs. The similar performance may be a result of all neuronal models being capable of representing the action potential spike trains which suggests that there is a biological complexity and computational tractability tradeoff to be made. If some set of neuronal models behaves similarly for identical input then it may be worth it to used the reduced models for improvements in speed at the cost of detail.

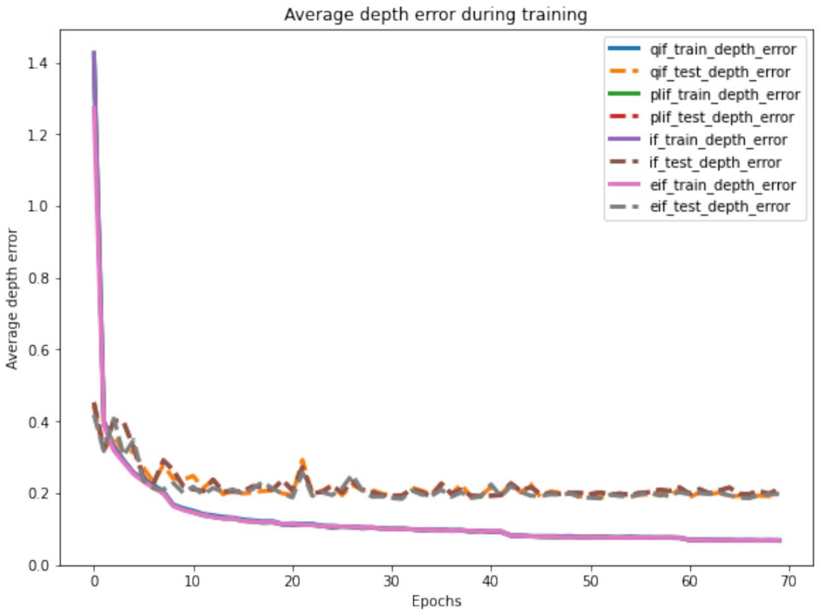

Figure 2: Pixelwise residuals between estimated depth map and Lidar groundtruth were computed for each epoch and compared between training/testing sets

5 Conclusion

We extended the work of [RCCM21] by testing more neurona l models on an end-to-end spike-coded depth estimation problem yielding similar performance and runtime. Further directions include changing different neuronal models for different blocks (e.g. replacing all neuronal models), applying a biologically plausible unsupervised learning rule such as spike-timing-dependent plasticity (a Hebbian learning rule) instead of surrogate gradient descent, and topologically modelling the network architecture more similarly to the human vision system. Some possibly interesting problems to expect for these directions include:

1. implementing the neuronal models above as differential equations in order to take advantage of the canonical neuronal model formulation of [Izh04] to investigate phase-space dynamics of each model as well applying different numerical methods (i.e. ODE integration) to simulate the voltage values,

2. comparing biologically plausible unsupervised learning rules against supervised deep learning rules,

3. extending to non-event camera based data and exploring the effect of encoding scheme on performance,

4. forming an anatomy-function model of human vision informed with where activity takes place in the brain during depth estimation tasks (this is limited by neurophysiological and cognitive science understandings in the literature, e.g. there are many different kinds of neuron in the brain that may require their own neuronal model),

5. implementing the depth estimation SNN onto a neuromorphic chip in order to improve on size, weight and power constraints,

References

[BSGB$^+$17] Basabdatta Bhattacharya, Teresa Serrano-Gotarredona, Lorinc Balassa, Akash Bhattacharya, Alan Stokes, Andrew Rowley, Indar Sugiarto, and Steve Furber. A spiking neural network model of the lateral geniculate nucleus on the spinnaker machine. *Frontiers in Neuroscience*, 11:454, 08 2017.

[CDGM22] Nuno Calaim, Florian A Dehmelt, Pedro J Gonçalves, and Christian K Machens. The geometry of robustness in spiking neural networks. *Elife*, 11:e73276, 2022.

[DF21] Simon Davidson and Steve B Furber. Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, 15:651141, 2021.

[FCD$^+$20] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Timothe Masquelier, Yonghong Tian, and other contributors. Spikingjelly. https://github.com/fangwei123456/spikingjelly, 2020.

[FLB22] Justas Furmonas, John Liobe, and Vaidotas Barzdenas. Analytical review of event-based camera depth estimation methods and systems. *Sensors*, 22(3), 2022.

[GDO$^+$22] Guillermo Gallego, Tobi Delbrck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jrg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022.

[GFES21] Wenzhe Guo, Mohammed E Fouda, Ahmed M Eltawil, and Khaled Nabil Salama. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience*, 15:638474, 2021.

[GKRL17] Masoud Ghodrati, Seyed-Mahdi Khaligh-Razavi, and Sidney R. Lehky. Towards building a more complex view of the lateral geniculate nucleus: Recent advances in understanding its role. *Progress in Neurobiology*, 156:214–255, 2017.

[HPGBF18] Michael Hopkins, Garibaldi Pineda-Garcia, Petruţ A Bogdan, and Steve B Furber. Spiking neural networks for computer vision. *Interface Focus*, 8(4):20180007, 2018.

[Izh04] Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

[NCCC22] João D Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S Cardoso. Spiking neural networks: A survey. *IEEE Access*, 10:60738–60764, 2022.

[OIBI17] Marc Osswald, Sio-Hoi Ieng, Ryad Benosman, and Giacomo Indiveri. A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Scientific reports*, 7(1):1–12, 2017.

[PTB$^+$22] Matteo Poggi, Fabio Tosi, Konstantinos Batsos, Philippos Mordohai, and Stefano Mattoccia. On the synergies between machine learning and binocular stereo for depth estimation from images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5314–5334, 2022.

[RCCM21] Ulysse Rançon, Javier Cuadrado-Anibarro, Benoit R. Cottereau, and Timothée Masquelier. Stereospike: Depth learning with a spiking neural network. *CoRR*, abs/2109.13751, 2021.

[RCI21] Nicoletta Risi, Enrico Calabrese, and Giacomo Indiveri. Instantaneous stereo depth estimation of real-world stimuli with a neuromorphic stereo-vision setup. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[RD20] Sean Riley and Jim Davies. A spiking neural network model of spatial and visual mental imagery. *Cognitive Neurodynamics*, 14:239–251, 04 2020.

[SHK$^+$14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[WMJ$^+$07] Gert Westermann, Denis Mareschal, Mark H Johnson, Sylvain Sirois, Michael W Spratling, and Michael SC Thomas. Neuroconstructivism. *Developmental science*, 10(1):75–83, 2007.

[ZDL$^+$20] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via spiking neural model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1446, 2020.

[ZT$^+$18] Alex Zihao Zhu, Dinesh Thakur, Tolga zaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039, 2018.