Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2019-00

M.Sc. Thesis

Neuromorphic Retina Design to encode LIDAR based Scene Dynamics

Rahul Rakeshkumar Vyas - 4747798

Faculty of Electrical Engineering, Mathematics and Computer Science **Delft University of Technology**

Neuromorphic Retina Design to encode LIDAR based Scene Dynamics

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Computer Engineering

by

Rahul Rakeshkumar Vyas - 4747798
born in Jodhpur, India

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Neuromorphic Retina Design to encode LIDAR based Scene Dynamics"** by **Rahul Rakeshkumar Vyas - 4747798** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 29-November-2019

Chairman:

Dr. Rene van Leuken

Advisors:

Dr.Sumeet Kumar

Dr. Amir Zjajo

Committee Members:

Dr. Stephan Wong

Abstract

Autonomous vehicle (AV technology) relies heavily on vision based applications like object recognition, obstacle/collision avoidance etc. In order to achieve this, understanding and estimating the dynamics in the environment is extremely important. LIDARs are proven to detect both shape as well as the speed/movement of the objects in the scene but one of the biggest challenges faced in adapting LIDAR technology is the huge amount of data it produces and the way it is processed. Most of this data is redundant static information which results in wastage of system memory, computational resources, power and time. Inspired from biological retina, first Neuromorphic-Retina for LIDAR is proposed that is able to extract and encode movement happening at particular distance, particular angle and with particular velocity from raw LIDAR temporal pulses into unique spike sequences so that the information about the dynamic environment can be efficiently classified and processed by event based and low powered Neuromorphic processing unit. The system is designed in such a way that it avoids consumption of large amount of computational resources and system memory. Simulation results show that the Retina is able to filter out redundant static information from the LIDAR data stream thereby reducing data throughput of around 50 - 70 % with 5 - 22 % spatial quality loss (based on scenario) as well as remove noise caused due to luminous reflections. This has tremendous impact on system latency and power consumption due to drop in memory accesses.

Acknowledgments

I would like to thank my advisors Dr. Rene van Leuken , Dr.Sumeet Kumar
and Dr. Amir Zjajo for their assistance during the writing of this thesis. Without them, this would not have been possible. I would like to thank my colleague Johan Mes for providing the simulation tool for the neuromorphic machine. I would like to express my gratitudes to my family. Especially to my parents who have always supported me in my life and Apoorva for supporting me throughout the process.

Rahul Rakeshkumar Vyas - 4747798
Delft, The Netherlands
29-November-2019

Contents

List of Figures

List of Tables

1 Introduction

This chapter gives an overview of the thesis, defines the overall project, what is the main problem and how the proposed design can solve it.

1.1 Outline

Autonomous vehicle technology is the new buzz word in modern AI era. More and more companies are investing heavily into self driving car technology. The applications like automatic cruise control, automatic lane changing and maneuvering, collision avoidance, pedestrian/vehicle detection etc not only makes life of the driver easy but also increase road safety for both driver as well as pedestrians and other surrounding vehicles [9].

Centre of these applications is the AI driven vision technology which relay heavily on understanding and estimating the dynamics of the environment. Hence, capturing and interpreting this information is extremely important. Sensor fusion from various sensors like LIDAR, RADAR and camera have proved to be quite beneficial for safe and efficient AI based processing. The report briefly describes each and every sensor in detail and how their fusion affects AI based decision making in the vehicle.

LIDARs are becoming popular in modern navigation and geolocalization based applications. LIDAR provides both shape as well as motion information to the user which can be harnessed by AI for vision based applications like obstacle avoidance, emergency braking, efficient real-time navigation, cruise control etc. The report describes state of the art LIDAR sensors, their properties, why they are important and their drawbacks. One of the major drawbacks targeted in this thesis is the enormous amount of data that the LIDAR sensor produces and how it can be addressed.

Motion perception is a biological activity for understanding direction and speed of the elements occurring in the view [10]. Human retina have mainly two types of neurons which are sensitive to two different applications. First, Motion Detection and second, Directional Selectivity. Retina neurons pre-process the visual information before transmitting it to the visual cortex. There have been attempts to mimic the neural activity occurring in retina.

There are two major areas where the progress have been made and which are the main focus of the thesis. First, neuromorphic processing which is based on the concept of event based processing like human brain and second, data acquisition process like artificial retinas [11] that detects changes in the scene and asynchronously transmits them.

The event based processing is extremely low powered while artificial retina removes huge amount of redundant information. Therefore, less throughput means lower system memory accesses and storage , less computational requirements thereby reducing overall power consumption and systems latency.

1.2 Problem statement

Biggest challenges faced in adapting LIDAR technology is the huge amount of data it produces and the way it is processed. Most of this data is redundant static information which results in wastage of system memory, computational resources, power and time. To address this issue three major questions are answered.

1. How to extract relevant information from the LIDAR data stream and filter it?
2. How to make this technology computationally efficient without consuming large amount of memory and computational resources?
3. How can this information be encoded for efficient use in vision applications?

1.3 Goal

The main goal of the thesis is to design a Neuromorphic-Retina to directly pre-process raw LIDAR data at sensor level in order to extract, encode and transmit motion, distance and directional selective information from the raw LIDAR data stream without consuming excessive memory and computational resources.

1.4 Solution

Inspired from biological retina, first Neuromorphic-Retina for LIDAR is proposed that is able to extract and encode movement happening at particular distance, particular angle and with particular velocity from raw LIDAR temporal pulses into unique spike sequences so that the information about the dynamic environment can be efficiently classified and processed by event based and low powered Neuromorphic processing unit.

1.5 Approach

The most important information in LIDAR data is the movement information at particular distance and angle. Hence, these features were targeted in this thesis. An event based Spiking neural network was designed to detect motion and encode it into unique spike train representing motion at particular distance, angle and speed. In order to qualitatively and quantitatively analyse and compare the results same functional module was designed using DSP and RAM based mechanism. The final step was to simulate, visualize and verify the results.

1.6 Contribution

There are two major contributions that this thesis made: -

1. The proposed LIDAR pre-processing model is first Neuromorphic Retinal that cognitively detects motion in a LIDAR scene and filters out the redundant information without using large amount of system memory and computational resources.
2. This technique also removes static noise without using heavy signal processing operations.
3. The motion information is further utilized to extract and encode events happening at particular distance, speed and angle into unique temporal pulses so as to be processed on event based low powered Neuromorphic computing units for efficient vision based classifications and optical flow predictions.

1.7 Thesis overview

The report is subdivided into 3 major Parts: -

1. Part I: Literature study
2. Part II: Methodology, construction and results
3. Part III: Conclusion and future work

Part I is further divided into 3 chapters: -

1. Chapter II: Autonomous driving technology and sensors
2. Chapter III: Vision
3. Chapter IV: Neuromorphic computing and SNN simulator

Chapter I tries to answer the following questions: -

1. What is autonomous driving technology?
2. Which sensors are involved?
3. What is the importance of LIDAR in autonomous driving technology?
4. What are the available LIDARs technologies and products in the market?
5. Importance of sensor fusion?

Chapter II tries to answer the following questions: -

1. Importance of Artificial intelligence for autonomous driving technology?
2. What are various vision based applications for autonomous vehicles?
3. What is the Importance of analyzing dynamics of the scene?

4. Why LIDARS are effective than Cameras/RADAR for Dynamics of moving objects?
5. What are the Available/Possible Deep learning methodologies/techniques CNN/SNN?
6. How SNN is better than CNN?
7. What is the Scope of the Project w.r.t to SNN?
8. Why do we need to reduce the data for vision applications?
9. What is the concept of artificial retina what are its benefits and how can it be adapted in the project?

Chapter III: tries to answer the following questions: -

1. What is neuromorphic computing?
2. Which simulator is chosen for the project and what is its architecture?

Part II is further divided into 2 chapters: -

1. Chapter V: Methodology
2. Chapter VI: Results and discussion

Chapter V: This chapter describes in detail the process, system as well as the simulation architecture followed, designed and developed during the thesis. This chapter is further subdivided into subsections:-

1. Understanding dynamics of the scene by experimenting with event information extraction in camera data.
2. LIDAR data processing and visualization
3. SNN architecture design for motion detection
4. CODEC (Encoder/Decoder) design for the proposed SNN
5. Experiments conducted

Chapter VI: This chapter describes the results and provides analysis on them. This chapter is further divided into subsections: -

1. Single pixel based simulations and result analysis
2. Multiple parallel pixel based simulations and result analysis
3. implementing the SNN on LIDAR data, visualization of the output and its detailed analysis
4. Section wise neighbor data processing
5. Comparison between SNN based approach and signal processing based approach

Part III discusses conclusion and future work.

Part I
Literature Study

Autonomous vehicle technology and sensors

2

Autonomous vehicles (AV) are key to the next generation of transportation. Automatic driving assistance system (ADAS) gives cognitive ability of vision and thinking to the vehicles.

How much autonomous is the vehicle is characterized in 5 levels [12] :-

1. Level 0: Human controls direction, acceleration and navigation
2. Level 1: Adaptive cruise control assistance with human control
3. Level 2: Partial automation under human monitoring
4. Level 3: Safety critical system are fully controlled by ADAS and human intervention allowed
5. Level 4: Vehicle handles safety as well as speed, steering and navigation without human monitoring. All the driving conditions are not met
6. Level 5: Full automation, driver less vehicle

2.1 Sensors

Sensors are the eyes and ears for the autonomous cars. In order to reach full automation of level 4 and 5 the ADAS operates on data from various sensors like LIDAR, RADAR, SONAR, Camera, GPS etc. This fusion helps vehicle to better understand the environment around it.

Each sensor has its own benefits/drawbacks and processing requirements. Hence, fusing data from multiple sensors not only produces better results but also require heavy processing involving complex algorithms. The following section discusses the most important vision based sensors available today namely RADAR, LIDAR and camera.

2.1.1 RADAR (Radio Detection and Ranging)

RADAR works on the principle of doppler effect to estimate the velocity and distance of the moving object. RADARs are good in detecting movement rather than shape of the object. Hence, they are used in many ADAS applications ranging from automatic cruise control, collision avoidance to critical emergency braking systems.

There are 3 types of RADARs:-

1. Short range (SRR) - 0 to 20m
2. Medium Range (MRR) - 0 to 100m
3. Long range (LRR) - 0 to 250m

SRRs have high field of view and are mainly used for proximity detection applications like parking assistance. MRR and LRRs are mainly used in emergency braking systems and cruise control systems.

RADARs are used by many vision applications to enhance vehicle navigational capabilities and safety. RADAR works only when something moves in the scene. Hence, it produces way less data than its optical counterparts.

Main working principle of RADAR is as follows. The sensor has lanes which detects the reflected signals. These signals are transmitted serially to the processing unit. This data is processed to extract:-

1. Doppler velocity
2. Object's coordinate
3. Object's azimuth angle
4. Radial distance w.r.t to the source

From this information, an occupancy grid is created in order to apply vision based algorithms.

2.1.2 Image sensors

Cameras are termed as the eyes of autonomous vehicles and one of the most important sensors used for accurately detecting shapes and sizes of various objects in the scene.

Camera is usually based on CMOS image sensors that detects light and produces RGB data stream. For achieving high vehicle autonomy, both 360 degree and rear cameras are used.

Cars use 2D or 3D camera based in the cost-accuracy trade-offs as 3D camera use more number of cameras and hence are more accurate and precise. Therefore,they are much more suitable to achieve 4 to 5 level of autonomy. 2D cameras are used in augmented reality fashion to provide the driver extra information like vehicle proximity, lane enhancement etc.

State of the art 360 degree and rear camera image sensors produce 40 to 60 frames/second with 24 bit depth. With multiple fusion of these cameras tremendous amount of data is produced. To process this data huge hardware resources including processors and accelerators are used.

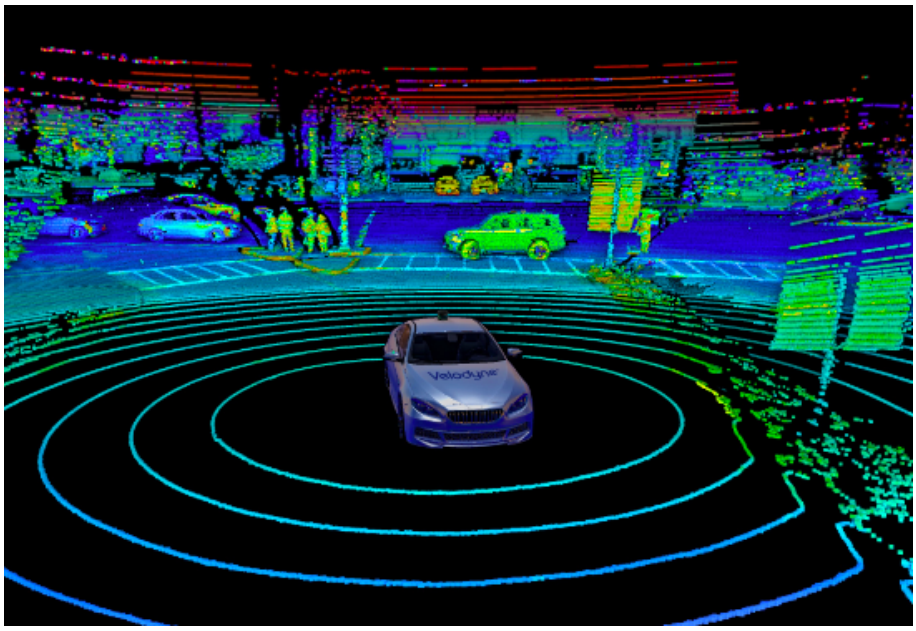


Figure 2.1: Scene visualization through rotating Velodyne LIDAR [1]

Forward facing cameras are used to detect objects upto 250 yards (medium and high range). These cameras are used in vision applications like pedestrian/vehicle detection, traffic signal detection etc.

2.1.3 LiDAR (Light Detection and Ranging)

LIDAR uses laser to estimate the distance of objects. It emits continuous laser pulses in order to detect time of flight (TOF) of reflected laser pulses. This information is further processed to calculate the distance [13]. Using this information, 3 dimensional point cloud is created which replicates the environment around the source (in our case Autonomous Vehicles).

The point cloud information is further exploited to determine shape as well as velocity of the object. LIDAR offers various applications to the AV like obstacle avoidance, realtime navigation and cruise control.

LIDARS available in market offers a range of about 250m and resolution of about 0.02m. Apart fro the autonomous vehicles, LIDARs are also used to map various geographical terrains when mounted on drones/aircrafts. Using GPS locations the LIDAR accurately creates 3D terrain maps. Hence, LIDARs are also extensively used in geo-localization based application [9].

2.2 Importance of LiDAR for Autonomous Driving Technology

LIDAR offers two major benefits, first, it captures the shape of the object like camera and second, it records the motion of the object like RADAR. Utilizing these two functionalities, the vision based algorithms can precisely mimic the dynamic environment surrounding the vehicle.

This increases vehicle safety and its navigational efficiency. Hence, because of LIDARs, vehicle can become less prone to accidents due to sudden change in the vehicle's environment [14].

2.3 Available LiDAR technology and products in the market

In order to decide which LIDAR has state of the art features as well as future research growth possibility, we looked into various commercially available LIDARs.

LIDARs are mainly classified into long and short range. Long range LIDAR can detect distances upto 250m while short range can detect upto 50m.

Flash LIDARs [2] are short ranged, they illuminate the entire region at a time and detects distances based on the light intensity detected by the pixels. This gives them high field of view. Scanning LIDARs are long ranged as they focus on small regions unlike Flash LIDARs, they transmit beam that hits a section of the target. The reflection is captured by a single pixel which gives better resolution and quality over long distances. The laser beam is then rotated in such a way that the entire scene is covered by all the pixels [2].

Hence, scanning LIDARs are preferred choice for applications like pedestrian/vehicle detection/avoidance.

The most famous scanning LIDARs today [2] have rotating motion in order to get full 360 degree view. This scanner scans the scene by transmitting beam and receiving line by line reflections at constant angular velocity.

Now, the problem with these scanning LIDARs is the mechanical part. Its huge and adds on the weight to the system and hence have huge energy requirements.

Therefore the industry is now looking towards special type of LIDARs without such huge mechanical moving components called the solid state LIDARs. There are two major state of the art solid state LIDARs. First, Optical phase array (OPA) and second Microelectromechanical mirror system (MEMS) [9].

Both of them are affordable i.e their cost is less than 175 euros. OPA scans at 100 kHz using micro structured waveguides to guide the beam. The problem is that its still not been able to work for long ranges i.e.distances beyond 200m. MEMS based

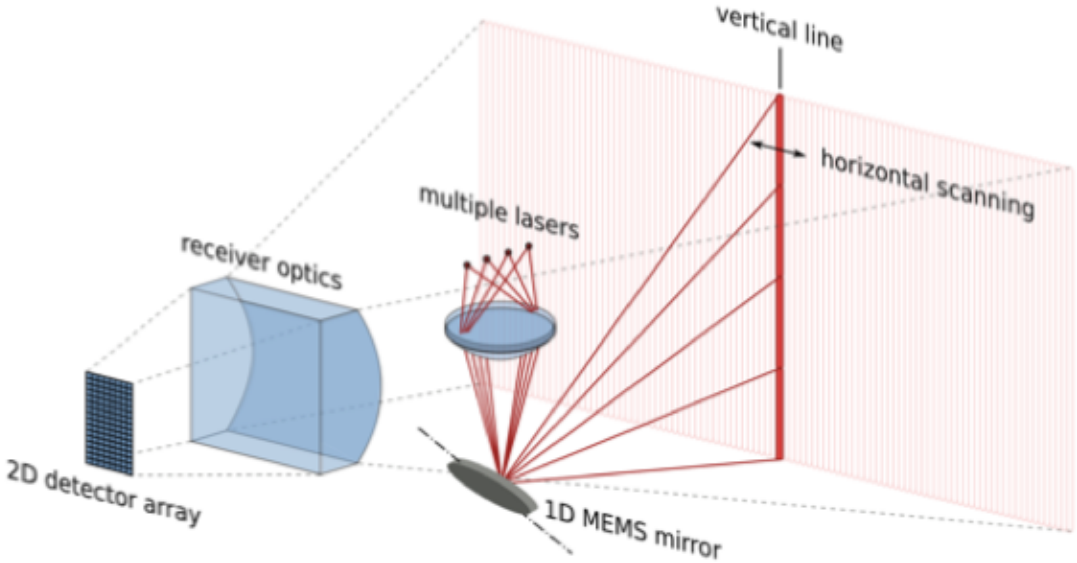


Figure 2.2: Infineon 1D MEMS LIDAR [2]

LIDARs on the other hand uses micro moving mirrors for scanning purposes, they are proven to be more robust than OPA and offer upto 250m range [2].

MEMS based LIDARs can scan either on one dimension or two dimensions depending on the requirement. One dimensional scanning offers higher frequencies than two dimensional counterparts. The micro-mirrors rotate in either resonant or non resonant fashion. Resonant mirror movements are preferred because they are more robust to external vibrations.

Hence, a one dimensional resonant MEMS based LIDAR was chosen to perform experiments for this thesis.

2.3.1 LIDAR technology used in the thesis

As described in the previous section, 1D MEMS based LIDAR is the best state of the art LIDAR available to conduct research. Hence, Infineon 1D MEMS Lidar [2] is used in the thesis.

The working architecture of LIDAR is described in figure 2.2. A laser beam pulse with four parallel edges hits the rotating MEMS mirror to produce a beam line scanner. The vertically rotating mirror also moves linearly in horizontal axis to produce a two dimensional scan [2]. The time between transmission and reception of laser pulse called Time of Flight (TOF) is recorded by each pixel. This information is passed through an FPGA that calculates real time 12 bit raw TOF values. These values are passed through a controller ASIC that calculates the distance from raw TOF values based on

the equation 2.1.

$$Dis = \frac{V_{Pulse}\Delta T}{2} \tag{2.1}$$

Where Dis is the Distance (m) of the object from LIDAR, V_{Pulse} is the velocity of the laser pulse (m/s) and ΔT is the raw TOF value.

The receiver hardware consists of photodiodes to detect light, transimpedance amplifiers to detect noises produced due to photodiodes and analog to digital converters for transmitting raw TOF data.

The infineon 1D MEMS LIDAR has range upto 250m with 0.12m resolution, 25 to 180 frames/sec frame rate and 120x32 pixels (HxV) field of view (FOV) [2].

2.4 Importance of LIDAR data reduction/filtering

The average data throughput from LIDAR is usually in Gb/sec. 1D MEMS LIDAR output data depends on various factor as described by the following equation:-

$$Data = Pulse_{RF} \times V_{Px} \times ADC_{Res} \times ADC_{Rate} \tag{2.2}$$

Where Data is the raw Time Of Flight, $Pulse_{RF}$ is the pulse refresh rate, V_{Px} is the number of vertical pixels and ADC_{Res} and ADC_{Rate} are the Analog to digital converter resolution and rate respectively.

We consider infineon 1D MEMS LIDAR [2] for our analysis. This LIDAR uses 12-bit ADC resolution with with 1.5 GHz of sampling rate and pulse repetition rate of 100 KHz. Each LIDAR frame is 120x32 pixels.

Putting these values in the equation 2.2 we can see that several Gbs of data is produced per second which has to be further processed to construct 3D point cloud with real time constraints. This huge amount of data therefore requires filtering and/or compression. There are many techniques like histogramming that attack 3D point cloud but filtering at sensor level itself is usually ignored [15].

In this thesis we directly target the raw TOF pulses at the sensor level so as to remove the unwanted information cognitively before transmitting them to the system memory.

2.5 Sensor fusion

Sensor fusion is one of the most important step in increasing the intelligence of autonomous vehicles. This section gives an overview of the benefits and applications of sensor fusion w.r.t autonomous vehicles.

A collective set of different sensors acts as eyes and ears to the autonomous vehicles. Fusing data from different sensors means that AI running in the ADAS can better understand the scene and can take decisions more accurately and with high precision. This is possible because the decisions made by AI are guided by the data fed into it.

This also means that if sensors are not doing their job properly then the AI will be bound to take wrong decisions. So, for AI to take better decisions, fusion helps in major ways:-

First, it introduces redundancy in the system to increase robustness. For example, RADAR-LIDAR pair can be used to accurately predict velocity of the moving object while LIDAR-Camera pair can be used to accurately determine the shape and size of the object

Secondly, one sensor can overpower the short comings of the other. For example, camera has high image resolution and can very accurately shoe the colored environment but is affected by harsh conditions like sunlight, darkness, fog, rain, hail etc. RADAR on the other hand has low image quality but can work without being affected by too much or too less light and can penetrate through rain, fog, hail etc. Hence combination of sensors provide a better picture of the environment to the AI so as to take more safe and accurate realtime decisions.

One of the major disadvantages of sensor fusion is the huge amount of data it produces. Hence, processing of fusion data usually requires huge amount of computational resources, power and memories. Thus, there is a need to pre-process the sensor data such that only key information is passed to the fusion stage.

Focusing majorly on LIDAR this thesis demonstrates a way to cognitively pre-process LIDAR data such that irrelevent redundant information is removed at the sensor raw data processing stage itself.

2.6 Conclusion

This chapter answered the following questions - What is autonomous driving technology, Which sensors are involved, What is the importance of LIDAR in autonomous driving technology, What are the available LIDARs technologies and products in the market, Why is it important to filter/Compress LIDAR data and what is the Importance of sensor fusion?

Hence based on the information provided three main conclusions are drawn :-

1. Vision is one of the most important aspect for autonomous velocities

2. As LIDAR provides both shape as well as velocity information, it plays a vital role for vision based applications.

3. 1D MEMS based resonant solid Lidar is state of the art LIDAR available in the market.

4. There is a need for data reduction in LIDAR throughput at sensor itself.

3 Vision

This chapter focuses majorly on AI and the technologies used in the automotive industries to provide vision based solutions to the autonomous vehicles.

We first start by looking into various deep learning strategies. There has been tremendous growth in different deep learning mechanisms mainly due to increase in computing capabilities , increase in access to huge amount of data and advancement in machine learning algorithms. These techniques are used in various applications from natural language processing, voice recognition to vision based image segmentation, object detection etc.

With the amalgamation of various sensors and deep learning techniques the AI is able to see, listen as well as think the way humans do while driving. Thus achieving human like cognitive abilities required for driving . The vision system interprets sensor information to estimate how the road, lanes and nearby vehicles look, what is the vehicle's proximity with other vehicles and what to do when an object like car or pedestrian comes dangerously close.

Learning strategies used in deep learning are mainly three types :-

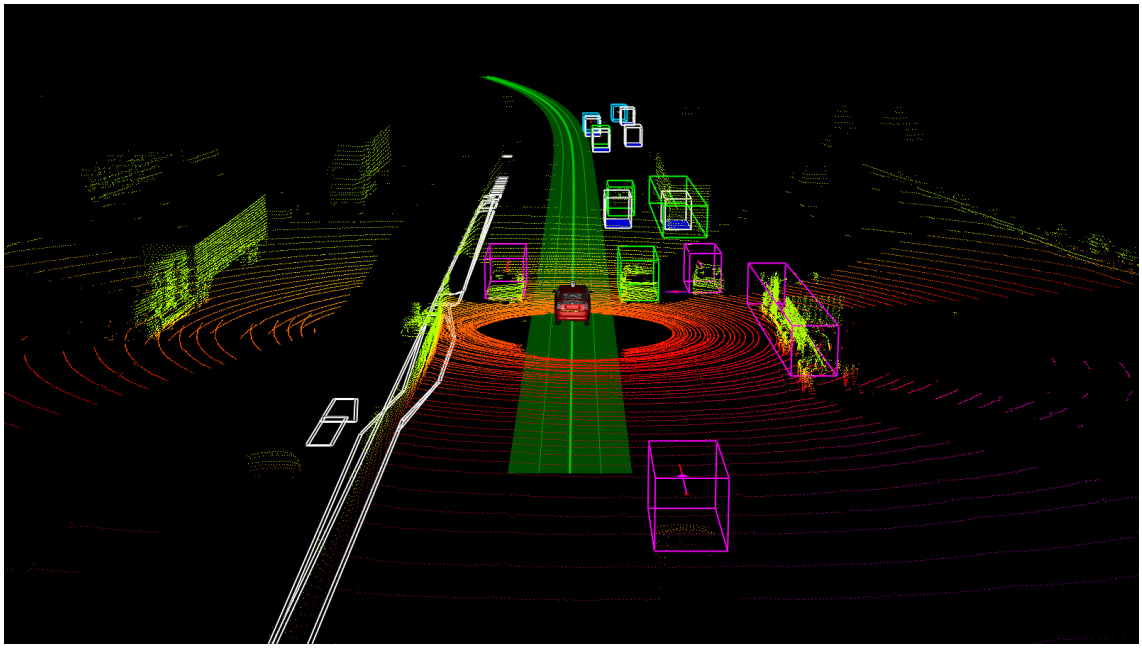
1. Supervised learning: that uses labelled dataset in order to train the deep learning models so as to achieve high accuracy.
2. Unsupervised learning: tries to classify the data based on some common patterns found in the dataset. This technique is usually used when labelled dataset in not available.
3. Semisupervised learning: is applied on unlabelled dataset using only small amount of available labelled dataset.

General step used in machine learning are feature extraction, training the classifier by twitching the parameters to decrease resulting error and finally testing the trained model on the available dataset. The figure 3.1 below shows different vision applications used in autonomous vehicles.

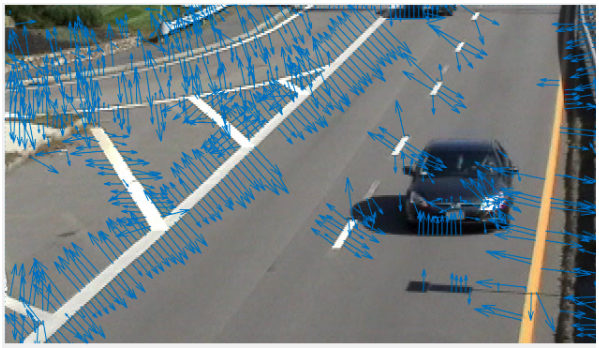
Further sections will describe basic components and various deep learning techniques as well as discuss why understanding dynamics of the scene is important and how can it be used for vision based applications.

3.1 Dynamics of the scene analysis

For autonomous vehicles understanding/estimating the dynamic changes occurring in the environment is of utmost importance. Recognising changes and predicting the



(a) lidar based object tracking [16]



(b) optical flow prediction [17]

(c) Semantic segmentation [18]

Figure 3.1: Different vision based applications used by autonomous vehicles

next movement ensures efficient and accurate safety and path planning. For example for tracking vehicle or car, the system first needs to identify the moving objects in the scenes, then classify them and then predict their behavior, using this information AI running on the ADAS can take calculative decisions to either change direction, speed or apply brake.

Majority of the safety critical applications like obstacle avoidance, pedestrian/vehicle tracking, emergency braking, critical path navigation etc requires the knowledge of fine changes happening around the vehicle.

3.1.1 Importance of LIDARs for dynamics of the scene analysis

Camera data is usually used in optical flow predictions but requires heavy post processing and is not extremely robust due to limitations in acquiring distance information. RADARs are able to provide the movement information in terms of doppler-shift but lacks geometric information. LIDARs on the other hand provides data which has both the geometrical as well as movement information.

Hence LIDAR is the key technology used for dynamics analysis and thus is used in this thesis.

3.2 Available/Possible Deep learning methodologies/techniques

3.2.1 ANN: Artificial Neural Networks

ANN tries to replicate the original neural network of human body. An ANN neuron processes input information scaled by a configurable weight, these weighted input to neuron connections are called synapse. A neuron can be connected to many synapses to perform accumulative operations. Hence a neuron usually performs a multiply-accumulate based linear functions

Although, these networks are not biologically plausible but are still used in wide array of learning based applications.

3.2.1.1 Basic components of artificial neural networks

As discussed in the previous section, the network consists of both linear as well as non linear processing elements. (Figure 3.2) The linear processing neuron which performs multiply-accumulate operations on the input synapse is governed by the following equation 3.1 :-

$$B_j = f(\sum_{i=1}^{n} w_{i.j}.A_i) \tag{3.1}$$

Where A_i is the i^{th} input value, B_j is the processed output and w_{ij} is the weight of the synaptic connection between i^{th} and the j^{th} neuron.

Non linear processing elements called the activation functions like sigmoid, Rectifier-Linear-Unit etc are used to introduce non linearity into the network. The principle on which they are activated is based on a condition. When the input synapse crosses a defined threshold value then the function activates.

Figure 3.2 represents two major aspects of ANN. First, connectivity: the connectivity between two layers can be fully connected (when each neuron of layer one is connected to each neuron of layer two) or sparsely connected.
Second, the network style: Networks can be feed-forward (where the processing occurs from one layer to another without any feedback loop involved) or Recurrent (where

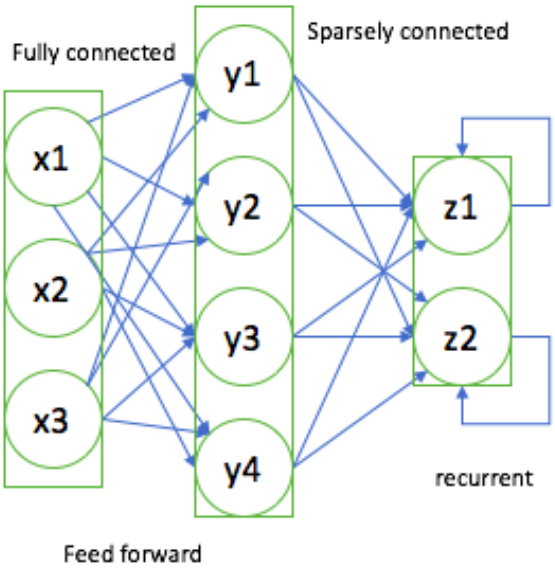


Figure 3.2: Schematic of a basic artificial neural network

feedback loops are involved to find temporal patterns)
Third, Hidden layers are used to make the model more efficient because extra layers of learning are introduced.

3.2.2 CNN: Convolutional Neural Networks

CNN is a type of ANN and is one of the most famous deep learning strategies used today for various applications using textual, visual and auditory data. They are most widely used in vision based applications.

One of the major advantages of CNN is that CNN usually many deep layers including regular stages of pooling layers that reduces the net throughput for the subsequent layers as shown in the figure 3.3. This facility has huge impact of net computation and power consumption.

The basic flow for CNN architecture is shown below:-

1. Convolutional layers: These layers try to extract different features from the input data so that the classifier can more efficiently and accurately classify the data.
2. The non linear activation layer is introduced after every convolutional layer to introduce non linearity into the network.
3. The pooling layer is introduced to compress the convoluted non linear data.
4. These three stages are repeated and the network is made deeper based on the application. Finally a fully connected output classifier produces the corresponding probability distribution for the output classes.

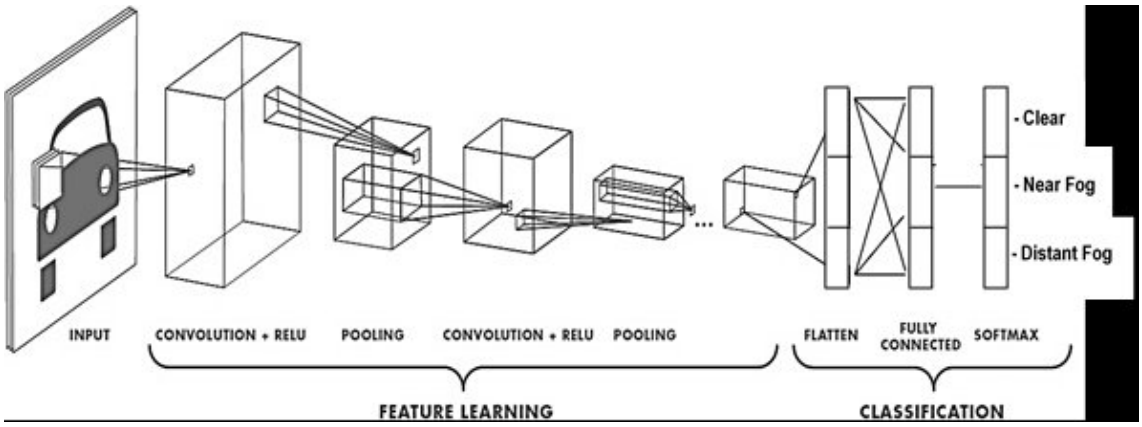


Figure 3.3: CNN data flow [3]

3.2.3 SNN: Spiking Neural Networks

Spiking neural network are closest to the biological neural networks. They work on temporal events/spikes unlike the ANNs. The fundamental concept behind SNN is that the network is sensitive to certain temporal patterns i.e it can detect temporal patterns from the dataset.

The membrane potential is affected by synaptic strength as well as time difference between the spikes. Hence, using this property a neuron is sensitive to temporal patterns and thus can be trained to identify them.

SNN usually depends on the way an input is encoded into spike based temporal pattern. Hence, choosing correct encoding scheme to train SNN is really important. In theory SNNs are known for power efficient processing because they process events rather than absolute data which saves lot of power. The detailed description of spiking neural network is described in chapter 4.
The upcoming section describes how much power is consumed by ANNs and their hardware requirements.

3.3 Hardware Requirement/ Computational requirement of ANN

This section is adapted by the studies conducted by [8]. For highly accurate image classification a frame size of atleast 1280x1280 pixels is required (1.5 Mpixels). 32 bit floating point are required for weight update and mac operations. The results displayed in table 3.1 are based on two most famous ANNs used in computer vision industry today namely Alexnet and VGGNET-16.

As the image resolution and network depth increases the overall power consumption as well as computational requirements will increase.

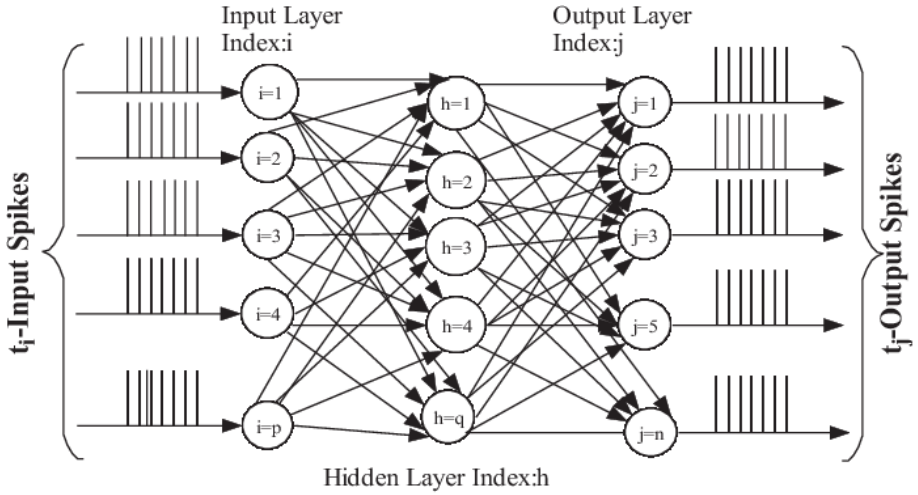


Figure 3.4: Spiking neural network [4]

Table 3.1: Compute requirements 28nm [8]

Metrics	AlexNet	VGGNet-16
Total operations	25.5J	505G
Parameters(weights)	1.6J	3.3.9J
Memory access	77G	1515J
SRAM access energy	11.5mJ	227.2mJ
DRAM access energy	862.4mJ	16968mJ
Total energy	873.95mJ	17195mJ
Power	4.3W	85.9W

Therefore, to process this huge amount of data hardware accelerators like FPGAs and GPUs are needed. Heterogeneous processing platforms consisting of combination of different processing platforms like neuromorphic computers are also used (described in chpter 4)

3.4 SNN over ANN

The paper [19] quantitatively describes the memory, energy and computational requirements of CNN based on profiling on certain CNN benchmarks. The results clearly indicate the requirement for massive optimization in terms of algorithm and hardware resources in order to run vision based applications efficiently on the autonomous vehicle computing unit in real time without hampering other processes.

SNN as described in previous section are more closer to actual human neural system. Hence, they have higher chances of gaining human like cognitive abilities than their artificial neural counterparts like ANNs. As SNNs process information in

temporal format it is proven that they are much more computationally efficient than ANNs.
Unlike ANN where a neuron has to wait for all the pre-synaptic updates from the previous layers a spiking neuron works in asynchronous fashion where the information is processed whenever a pre-synaptic spike is issued. This decreases overall system processing latency

Therefore, SNNs are preferred over ANNs because they are more computationally efficient and less power hungry. Hence, in conclusion SNNs are better choice than ANNs for vision based deep learning applications due to above reasons.

3.5 SNN challenges

Due to temporal and asynchronous pre-processing aspect, designing machine learning algorithms is also tricky. Hence, still a lot of work is needed in that direction.

Second major challenge is temporal encoding. Temporal encoding is required to represent every data point or features uniquely as a spike sequence. The project focuses more on pre-processing using spiking neural network in order to extract event based features and encode them into unique spike trains so as to be processed efficiently on a neuromorphic processor.

3.6 Concept of Artificial/Silicon Retina

In human visual system, Retina preprocesses data so as to extract important information from the scene and transmit only events to the visual cortex in brain. It doesnt analyze the scene frame by frame but rather perceives changes in asynchronous manner unlike modern day cameras [20].

Motion perception is a biological activity for understanding direction and speed of the elements occurring in the view [10]. Human retina have mainly two types of neurons which are sensitive to two different applications. First, "Motion Detection" and second, "Directional Selectivity". There have been attempts to mimic the neural activity occurring in retina.

Artificial retinas have been developed [11] that detects luminous changes in the scene and asynchronously transmits them. This removes huge amount of redundant information compared to frame based approach. Further, less throughput means lower system memory accesses and storage , less computational requirements thereby reducing overall power consumption and systems latency.

As this process is asynchronous, extremely minute changes are reported by each

pixel thereby increasing the temporal image resolution.

Dynamic vision sensors (DVS) (state of the art artificial retina) [21] are event based devices which can detect luminous changes in the scene. They use analog circuits to detect logarithmic changes in the light intensity.

Each DVS pixel generates ON/OFF flag indicating positive/negative event detected by the pixel and transmit the data on an AER bus [22]. This enables the system to store and process only relevant data/changes.

Therefore, DVS proposes several benefits as compared to conventional frame based cameras. First, Ultra low latency event data. Second, High dynamic range and third long system battery life [23].

3.6.1 Address Event Register protocol

AER is a communication protocol which is widely used in event based neuromorphic computing [22]. In this protocol any event occurring at specific address is transferred over the bus enabling event based processors to process information synchronously. The basic signals involved in AER protocol are shown below in figure 3.5

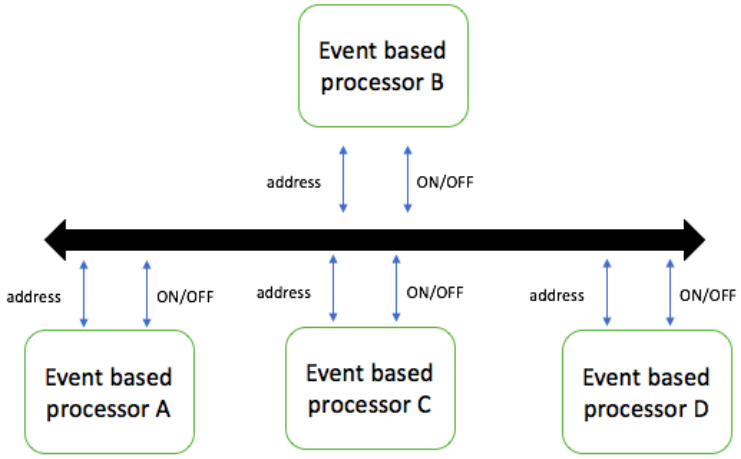


Figure 3.5: AER based event processing

Different neuromorphic or event based computing units can be connected via this bus (figure 3.5) in order to scale the system and process the events concurrently.

3.7 Conclusion

This chapter answered the following questions :- What is the Importance of Artificial intelligence for autonomous driving technology? What are various vision based applications for autonomous vehicles? What is the Importance of analyzing

dynamics of the scene? Why LIDARS are more effective than Cameras/RADAR for Dynamics of moving objects? What are the Available/Possible Deep learning methodologies/techniques CNN/SNN? How SNN is better than CNN? What are the drawbacks of SNN and What is the concept of artificial retina and what are its benefits?

Based on the above discussions we can draw the following conclusions: -

1. Understanding dynamics of the scene is of utmost importance for designing highly efficient and safe AI for autonomous vehicles.
2. SNN is the best choice for achieving high computational as well as power efficiency.
3. DVS is able to detect motion in the scene and hence can be used for directional selectivity but accurate information about speed is still missing from the data. Hence, there is a need to fill this void. There is a need to come up with a device that not only detects motion but also gives speed information so as to complete construction of an actual artificial retina.

Based on the chapters 2 and 3 two major concerns can be reported

1. LIDAR is one one the best sources to extract shape, speed and hence directional selectivity from the scene. Hence, to fill the void available in state of the art DVS LIDARs can be used. Therefore, in this thesis we designed first of a kind LIDAR based silicon retina that is able detect motion, shape as well as directional selectivity of the elements in the scene.
2. Form chapter 1 we concluded that there is a need for throughput reduction of LIDAR data. This demand is also achieved by the proposed silicon retina.

4 Neuromorphic computing

This chapter Describes the basics of neuromorphic hardware, biological neural networks as well as artificial models used to mimimic the these networks

4.1 Neuromorphic Hardware

Neuromorphic machines mainly consists of 3 parts namely neuron, synapse and connectivity. The upcoming sections will briefly describe each of these components in detail with their biological motivations.

4.1.1 Neuron

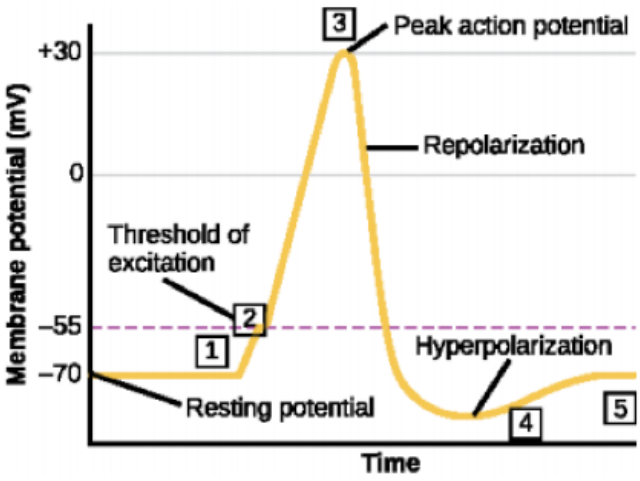


Figure 4.1: Neuron membrane potential vs time plot over single synaptic stimulus [5]

Figure 4.2 shows the basic structure of a neuron. It is fundamental element of nervous system. Humans gain their cognitive abilities by training these neural connections. The information in a neuron is transferred by electro-chemical reactions via electrical impulses (ion transfers) [5].

Each neuron has a nucleus inside a body called soma. The input stimulus enters neurons through dendrites. Based on certain chemical reactions the neuron spikes an output which is transferred to the other neurons via axon. Each neuron is activated from the impulses coming from dendrites. This current can either be positive (excitatory) or negative (inhibitory) which induces electric potential across the neural

membrane. If this potential exceeds a specific threshold then the axon of the neuron spikes or produces an electrical impulse.

After producing the axonal pulse, the neuron enters a refractory period where the membrane potential drops below the resting potential. The mathematical models for neurons are described in section 4.2

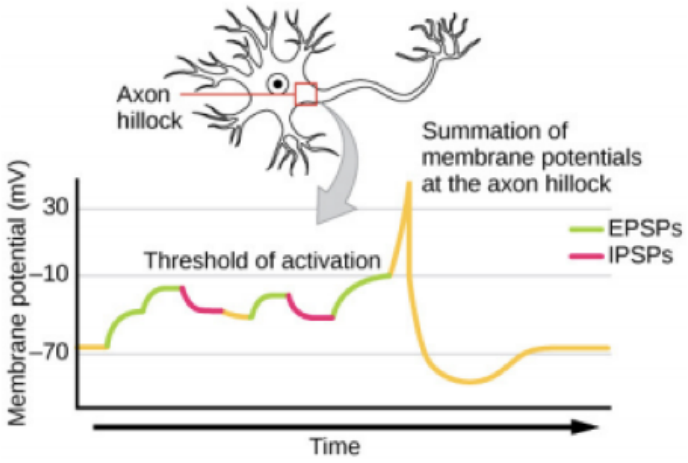


Figure 4.2: Neuron membrane potential vs time plot over multiple synaptic stimulus [5]

4.1.2 Synapse

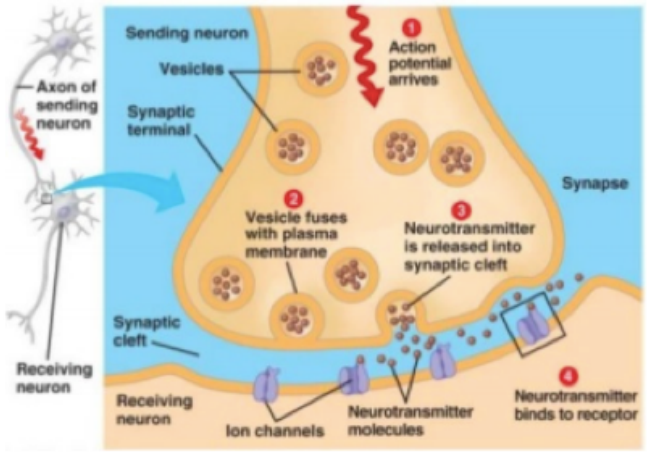


Figure 4.3: Biological Synapse [5]

Synapse is a non connected region between axon of one neuron and dendrite of another. The gap prevents short circuit between neurons. Neurotransmitters are

released when an axon spikes. They cross the synapse and induce current through the dendrite of another neuron by releasing ions.(Figure 4.3).

Hence, ion exchange occurs across the synapse which leads to information transfer across neurons. Each synapse has a weight, synaptic plasticity is associated with weight adaptivity of synapse. this is the main reason behind neural learning.

Hence in a neuromorphic processing hardware synapse a connection with access to the weight memory (usually stored in memristors). Based on the learning rules these weights are updated.

4.1.3 Axon Delay

Axon delay is referred to the delay in conducting an activation spike from one neuron to another depending on the neural distances and conductivity. When a neuron excites and produces a spike, the dendrite of other neuron doesn't activates immediately (like electric circuits). There are many techniques to implement axon delays [24][25][26]. This section will mainly focus on the state of art technique shown in [7].

The basic principle described in [7] is that a programmable delay can be produced by passing a user defined voltage and a time varying voltage produced by a ramp generator through a comparator. The comparator will spike when the ramp signal exceeds the stored voltage. This produces a delayed spike. There are three main components to the delay circuit - Ramp generator, delay storage cell and spike generator. The circuit is described in detail in appendix 8.

4.2 Neuron mathematical Model

Spiking neuron models try to represent the biological neural functionality by mathematical functions. Hodgin and Huxley (1952) [27] model is closest to the actual biological neuron but is too complex to model an entire network. The next section describes detail model of a Leaky Integrate and Fire (LIF) neural model which is plausible to implement electrically.

4.2.0.1 Leaky Integrate-and-Fire model

This model is the most plausible model to emulate on actual hardware. It is a simple RC circuit with a comparator. The information entering LIF's synapse is in form of temporally encoded spikes. An input current $i(t)$ generated from the incoming spikes passes through the RC circuit and stores the "membrane" voltage (V_m)across the capacitor. Based on the RC time constant τ the charging as well as the discharging time of the capacitor is decided and hence the increase/decrease on the V_m.

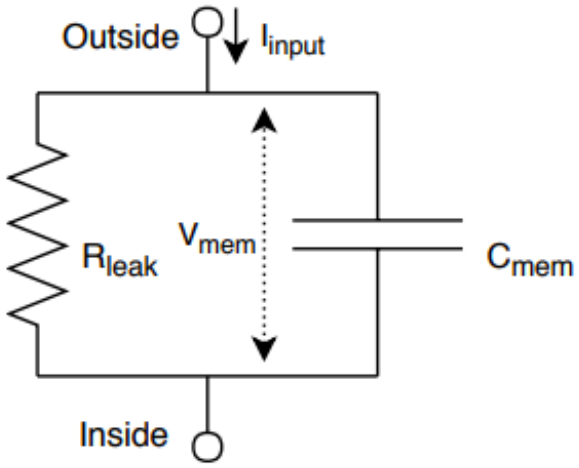


Figure 4.4: LIF circuit diagram

At a certain time if V_m is greater than the user defined V_{th} then the comparator will issue a spike. The behavior of V_m over time is shown below as differential equation:-

$$CdV_m(t)/dt = V_r(t) + Ri(t) \tag{4.1}$$

Where $V_r(t)$ is the resting potential of membrane once the LIF issues a spike. The current $i(t)$ generated by incoming spike train can be approximated by the equation:-

$$i(t) = w * [\sum_i \Delta(t - t^i) * (e^{-\frac{t}{\tau}})] \tag{4.2}$$

Where $\Delta(t - t^i)$ is an impulse train from i^{th} synapse at time t.

4.3 Information representation in SNN: Spike Encoding

Like brain, representing information about space and time in terms of events/impulses/spikes is key to the information processing in spiking neural networks. There are two major types of spike based information representation schemes:- Rate encoding and pulse encoding [28].

4.3.1 Rate Encoding

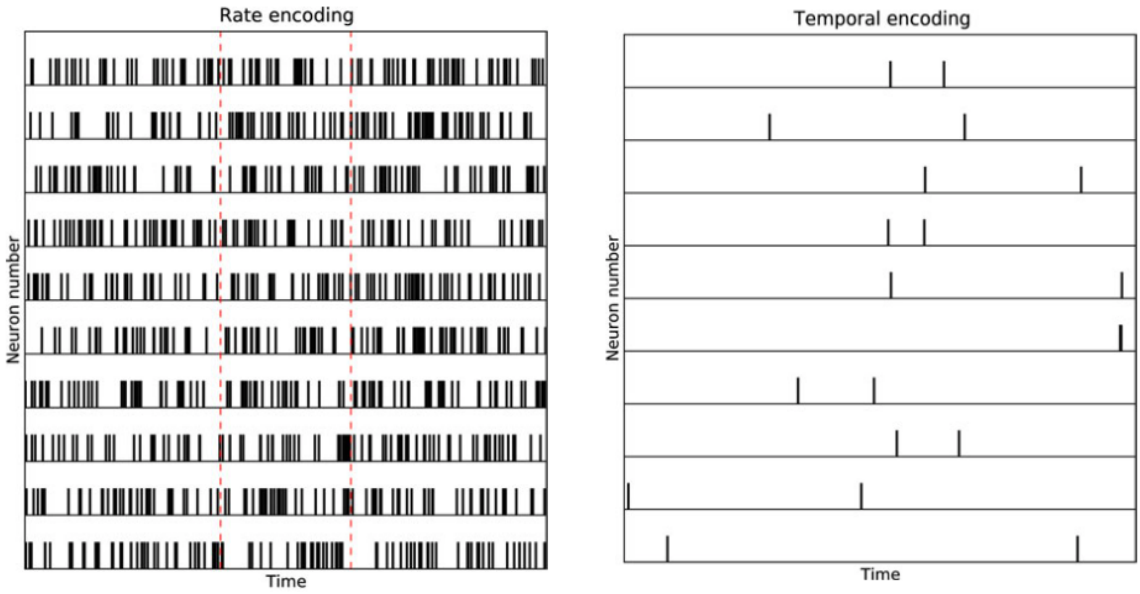
Rate encoding refers to representing information in terms of average number of spikes over a time frame. Hence, the information is encoded in frequency of spike train.

$$r = N_{spikes}/T \tag{4.3}$$

where r is the rate, N_{spikes} is the number of spike over time T. This encoding is majorly used to represent information/stimuli which can be affected by noise. As we average the number of spikes over time we tend to remove the effects of noise on the input signal.

4.3.2 Temporal Encoding

This encoding scheme is time based rather than frequency base. Here, the exact temporal position of the spike matters. Hence, this technique is usually used to represent the classified information in final stages on the neural network.



(a) Rate Encoding [28]

(b) Temporal encoding [28]

Figure 4.5: Encoding Schemes

4.4 Conclusion

Based on the discussing in the current chapter we can draw following conclusions:-

1. Neuromorphic computing is low powered temporal event based asynchronous processing and hence is suitable for AI based vision applications for autonomous vehicles.

2. There is a need to encode relevant information in temporal binary spike data. For LIDAR based silicon retina major information is shape, velocity and directional selectivity(event per angle). Encoding should also be robust to noise.
   Hence, keeping these factors in mind the proposed neuromorphic retina encodes shape (TOF), speed as well as directional selectivity information into spike trains with rate encoding scheme.

Part II

Methodology, construction and results

Methodology and system design

5

5.1 Overview

This chapter describes in detail the process, system as well as the simulation architecture followed, designed and developed during the thesis.

Based on the literature review, we came to the following conclusions essential for the system design. :-

1. Event analysis/dynamics of the scene analysis is the most important task for LIDAR.
2. SNN is the most favorable approach for low powered and computational resource efficient design for dynamics of the scene analysis.
3. There is a need for data reduction at sensor level itself in LIDAR.
4. There is a need for event based feature extraction for efficient SNN based classification.

Based on the conclusions made above, a neuromorphic retina is designed for LIDAR in order to tackle above requirements. The SNN based retina for LIDAR firstly, reduces the overall LIDAR data throughput and secondly provide motion based features like velocity and directional selectivity for efficient SNN based classification. In order to verify and visualize the proposed SNN design a Matlab based module is developed that produces reference data to verify the quality of the output of SNN based design and visualize it.

Upcoming sections are categorized as following :-

1. System design, simulation and verification
2. DSP based event detector design
3. SNN based Silicon Retina Design for LIDAR
4. Verification and quality analysis strategy.

5.2 Concept

The concept of event based LIDAR preprocessing or neuromorphic preprocessing has been adapted from the neural processing by retinal neurons. Retinal neurons extract event based motion information from the scene captured by luminous sensors in eye.

These features (spikes) are then transferred to visual cortex in brain to process the information efficiently. Similarly, motion or event based pre-processing at LIDAR sensor level itself reduces the overall throughput, power consumption and helps the further classification stages perform operations efficiently.

Hence, as shown in the figure 5.1, the proposed neuromorphic retina processes raw LIDAR TOF data with spiking neurons and extracts important motion based features in order to filter out redundant static information as well as assist future classification stages in order to perform more efficient vision based operations/applications as well as reduce over all system throughput and power consumption.

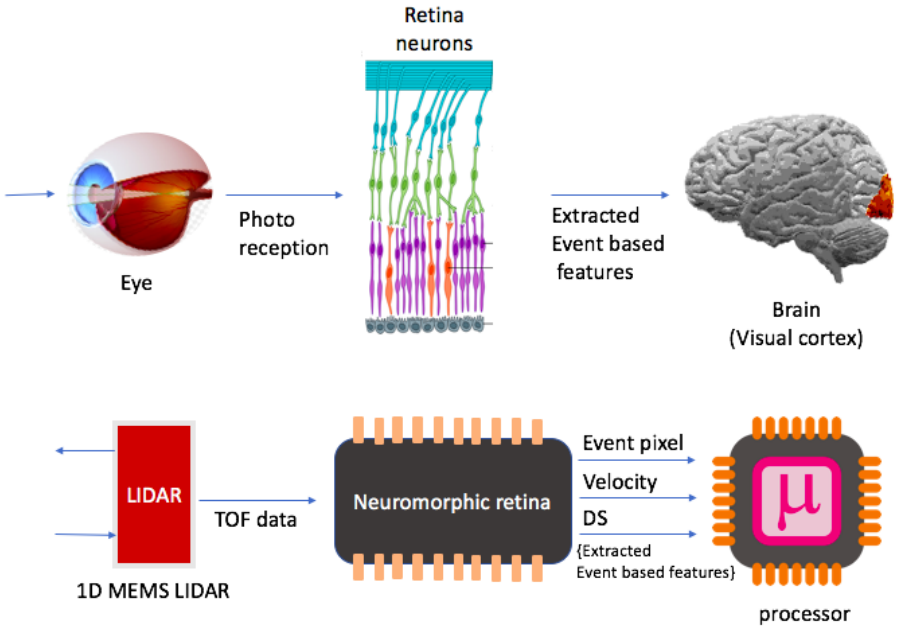


Figure 5.1: Concept of Artificial Retina

5.3 System Design and verification approach

Thesis is divided into three major modules (Figure 5.2).

1. DSP based event detector for LIDAR module
2. SNN based retina design
3. LIDAR visualization and SNN verification module.

DSP based event detector (described in detail in section 5.4) is a Matlab based module. The module performs following operations. First, it converts raw .ild LIDAR packet into TOF matrix. Second, it detects changes in the TOF values from consecutive LIDAR frames and third, it classifies the TOF matrix into dynamic and stationary objects.

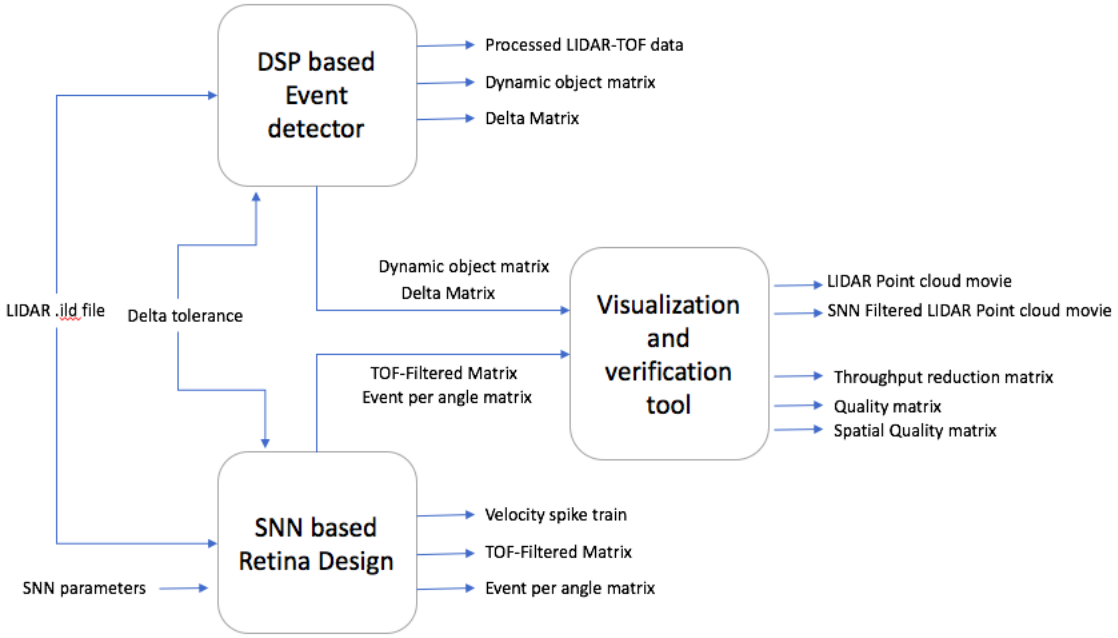


Figure 5.2: System Design and verification approach pipeline

SNN based retina module is designed and simulated in Matlab. The module performs following operations. First, it encodes the raw LIDAR TOF data into spikes. Second, it performs motion detection operation using spiking neural network and third, it produces event based filtered TOF spike matrix, velocity spike matrix and event per angle matrix.

The visualization and verification module performs following operations. First, it converts TOF LIDAR data into 3d point cloud movie for visualization. Secondly, it produces 3 major verification matrices :-

1. Throughput reduction matrix: this matrix describes the amount of data reduced after applying SNN based data reduction technique.

2. Quality matrix describes the amount of pixels of moving objects lost due to SNN based data reduction technique.

3. Spatial Quality matrix describes the quality of moving objects affected due to SNN based data reduction with respect to distance of the object from the sensor.

Hence, the final deliverable of the thesis are the following:-

1. LIDAR 3D point clouds representing actual as well as filtered data.

2. Silicon retina outputs:- event based TOF spike train, velocity per pixel spike train and event per angle matrix.

3. Verification and quality matrix.

5.4 DSP based motion detector and scene segmentation

As discussed in the previous sections, a DSP based model is designed as a reference design for SNN based retina model. This section gives detailed description of the system architecture as well as the algorithms involved and resource utilization.

The module is further subdivided into three submodules or processes (As shown in Figure: 5.3)

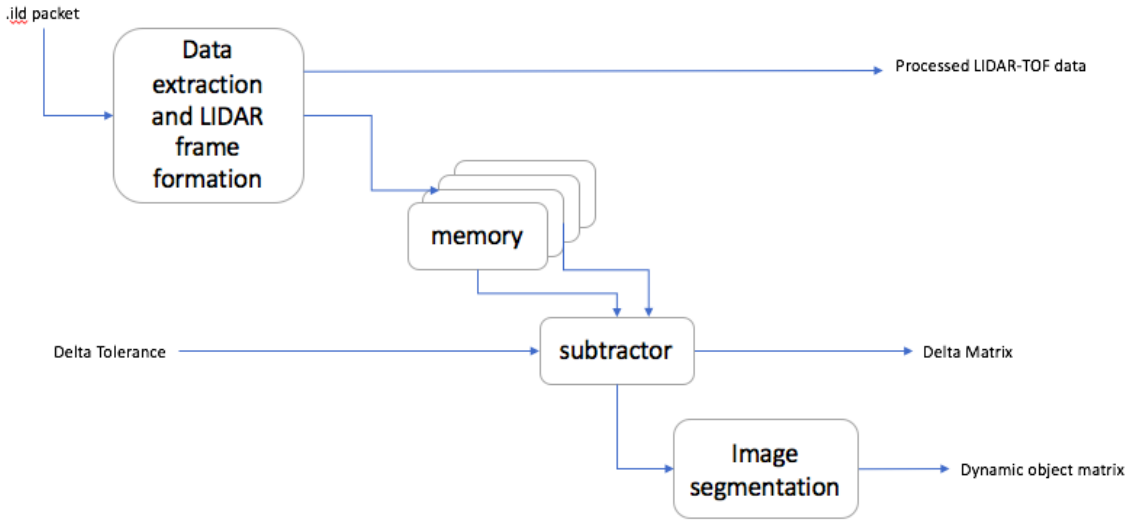


Figure 5.3: Signal processing schematics for event detection in pixels

1. LIDAR data extractor: This module extracts TOF values from the 1D MEMS based LIDAR packet (.ild files). These TOF pixels are further converted to frames for further processing and visualization.

2. Filter module: This module finds difference between the TOF values of consecutive frames. If the difference is not able to exceed the user defined Δ^T then that pixel is removed from the frame. Thereby filtering out those pixels which don't report any significant change. (Figure 5.4)

3. Image segmentation: This module utilizes the TOF difference information front he previous module to segment the scene into moving and static objects based on "motion based scene segmentation" technique. This "moving objects" matrix is further utilized as reference for checking the data quality produced by the SNN based retina. (Figure 5.4)

5.4.1 Implemetation on Raw LIDAR Data

LIDAR data used for experimentation was procured from Infineon tech. (1D MEMS). Three major matrices are generated from the DSP module:-

1. Processed LIDAR frames matrix

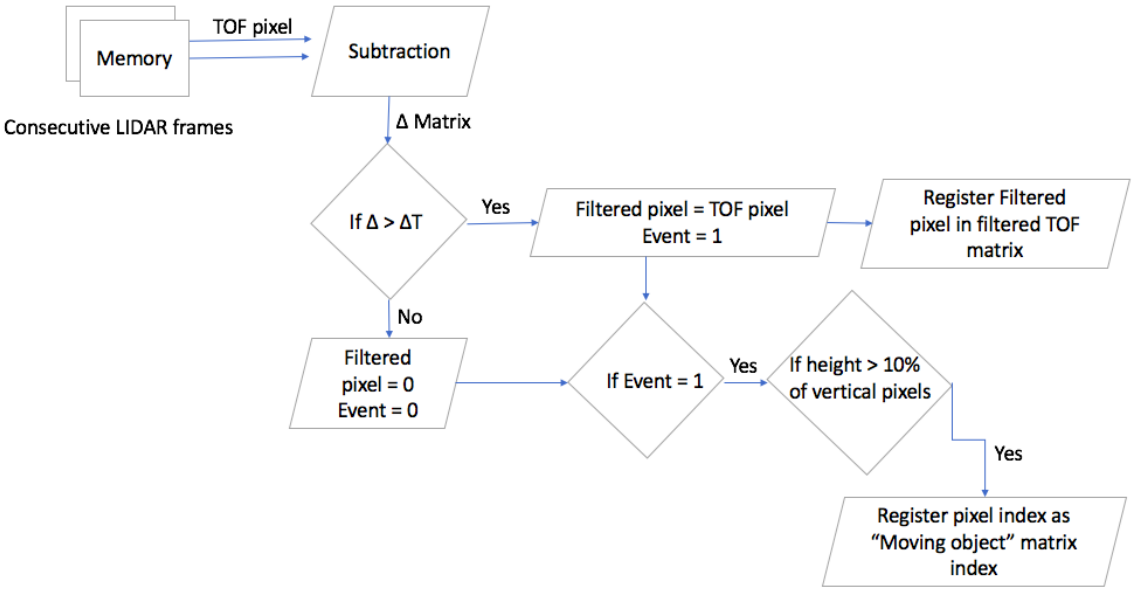


Figure 5.4: Flow chart representing algorithm used for LIDAR TOF data filtering and scene segmentation

2. Filtered LIDAR Frames matrix

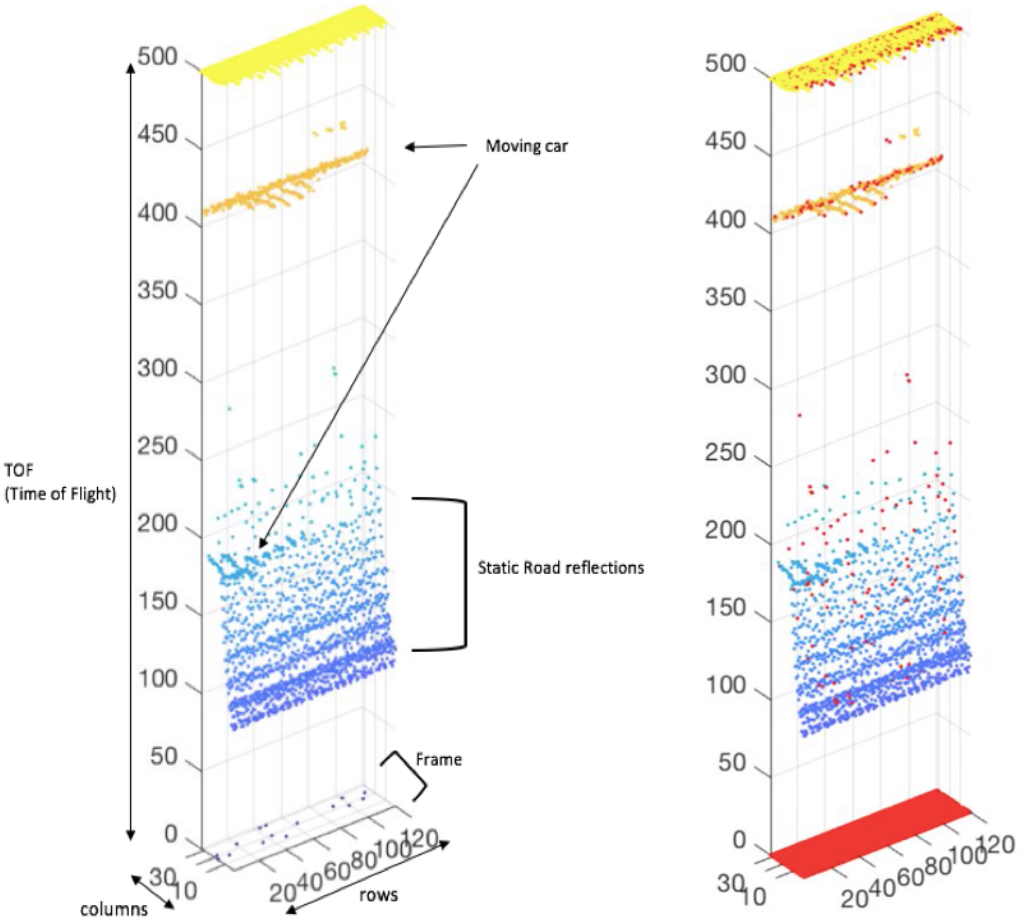
3. "Moving object" matrix

A visualization module is developed to convert these matrices into 3D point cloud. The following figures gives glimpses of the output of visualization module when implemented on the DSP data.

Figure 5.5 shows a scenario where LIDAR mounted on a moving car. The first part shows original LIDAR TOF 3d point cloud while the second part shows filtered data. Third part shows the ground truth captured from a camera mounted on the car. Its visually evident from figure 5.5b that the DSP based algorithm is able to capture the information of moving objects (highlighted in red). This information is used as a basis for classifying the scene into the "moving object" and "static information" shown in Figure 5.7 using standard motion based scene segmentation technique. Figure 5.6 shows the implementation of a scenario where the LIDAR is static and a person is moving away from it.

5.4.2 Resource utilization

The algorithm was synthesised in VHDL to analyse the resource utilized by the system. The system majorly consumes per pixel one 12 bit SRAM, one 12-bit subtractor, one 12-bit comparator and one 12 bit multiplexer. when extrapolated to an entire LIDAR frame (120x32) (12 bits per pixel), then the system utilizes 46 Kb of RAM and 3840 12-bit subtractors, comparators and multiplexors each.

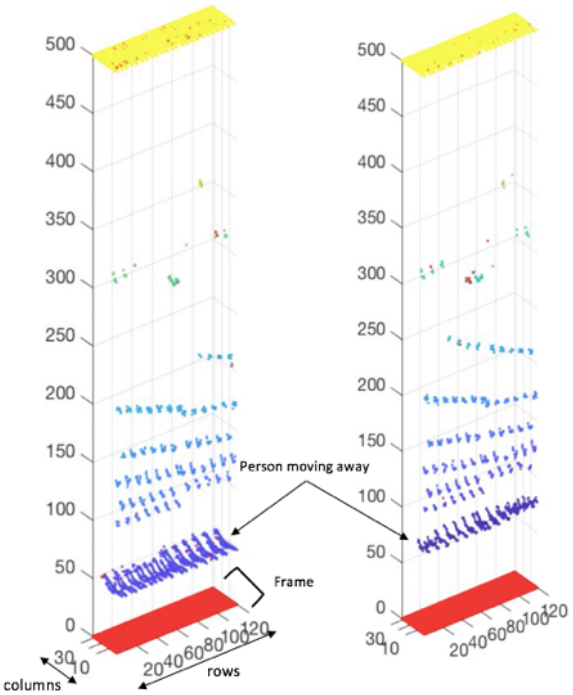


(a) shows the visualization of the scene in 3d point cloud environment

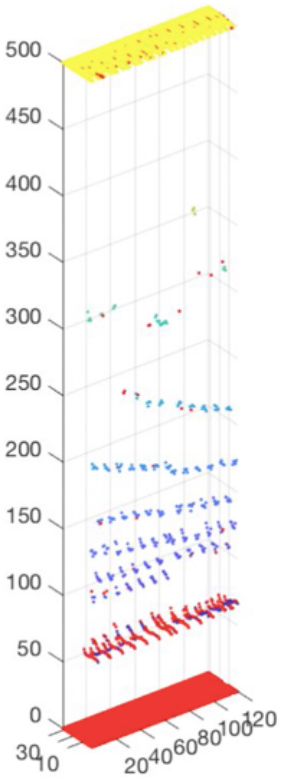
(b) shows the same data highlighting (as red) the pixels having significant change in displacement of frames

(c) shows the ground truth as captured by a camera

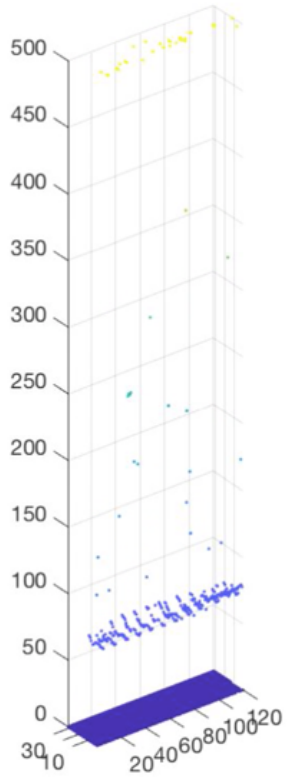
Figure 5.5: Scenario: LIDAR mounted on a moving car, LIDAR point cloud representation



(a) shows the visualization of the scene in 3d point cloud environment where a person is moving away from the car



(b) highlighted as red, the pixels having significant change in displacement of frames



(c) shows the filtered output based on events shown in 5.6b

Figure 5.6: Scenario: LIDAR mounted on a stationery car and a person is moving away from the sensor, LIDAR point cloud representation

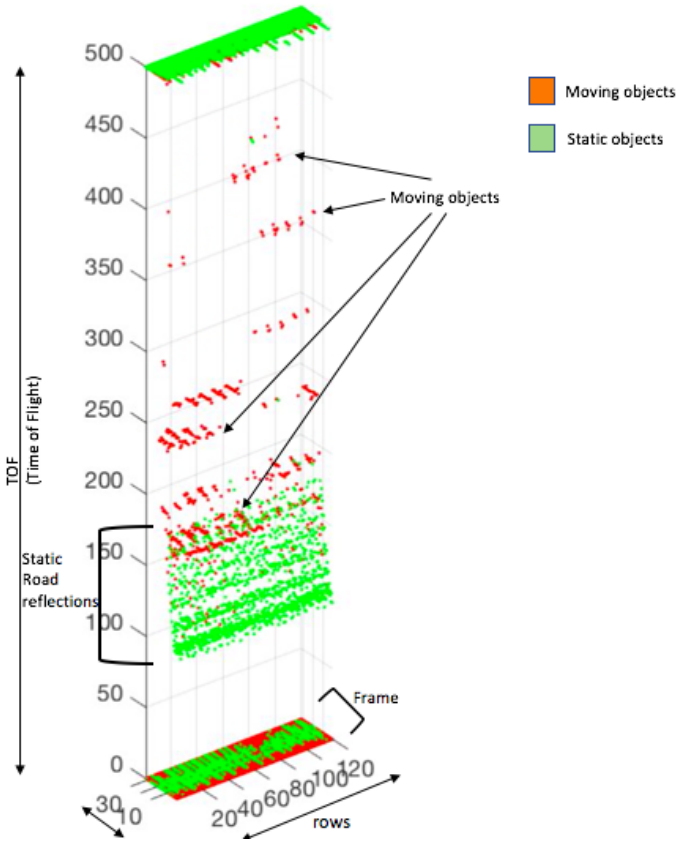


Figure 5.7: LIDAR scene segmentation based on dynamics of the environment

5.4.3 Conclusion

As discussed in this section, the DSP based event detector module is able to segment the LIDAR data into dynamic and static objects. This information is further used as reference for SNN based retina for verification purposes. This module also demonstrates that huge throughput can be reduced by "event" based filtering in LIDAR data, but this comes with a cost of utilizing large amount of SRAM which further affects overall system power consumption due to memory accesses.

Hence, There is a need to implement this algorithm without using large memory and computational resources. This void is filled by SNN based retina design explained in the next section.

5.5 SNN based Silicon Retina Design for LIDAR

As discussed in the previous section a different approach was needed in order to reduce overall memory and computational resource consumption. Hence, there is a need to design a system that detects events as described in the section 5.4 but without consuming large memory elements.

The following section presents detailed construction, I/O constraints and properties

of SNN based event detector system for LIDAR.

5.5.1 System Architecture

Figure 5.8 describes complete hardware architecture required for the SNN based silicon retina. As described in the previous sections, the proposed architecture directly accepts raw TOF pulses generated from the photodetector array of the LIDAR.

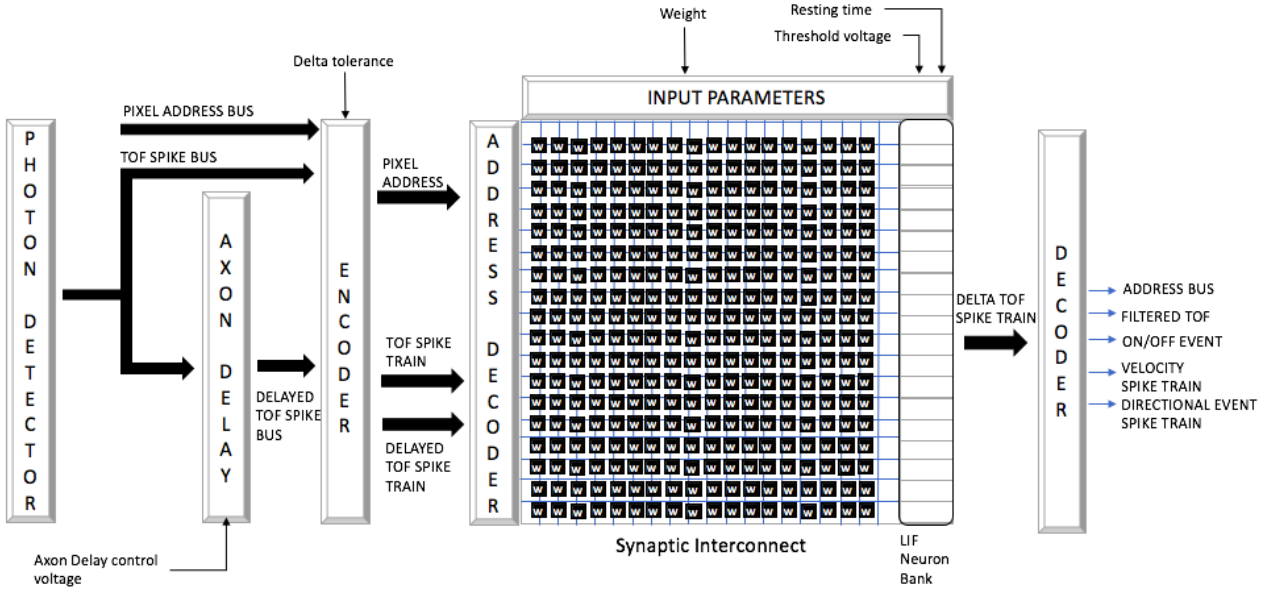


Figure 5.8: SNN based neuromorphic architecture

As described in [2] the sensor produces digital logic '1' when it receives a laser pulse. Hence, every pixel has two signals- event data line and corresponding address bus. This channel is routed to both spike encoder as well as the axon delay module.

The axon delay module is programmed to hold incoming spike and release it after a delay of 4 msec. This delay is equivalent to the frame rate of the LIDAR. The delayed spike is then routed to the encoder. At time t the encoder receives spike from the current frame as well as the previous frame. Hence, it receives the concurrent signals.

The encoder converts TOF spike into corresponding frequency based spike train (described in detail in section 5.5.4). The spike trains representing both current as well as previous frame's TOF values is transmitted in parallel to the input of the neuromorphic hardware.

The synaptic interconnect is the programmable connection between the neurons their inputs and outputs. This connection is established via a multiplier unit connected with weight memory (where corresponding weights are stored)

The spike inputs and corresponding pixel addresses enter the neuromorphic proces-

sor are connected to the LIF neurons via the synapses. The processed spike train is the decoded via decoder to generate the required output in AER format. A detailed description of each of the involved module is described in the upcoming sections.

5.5.2 System Design

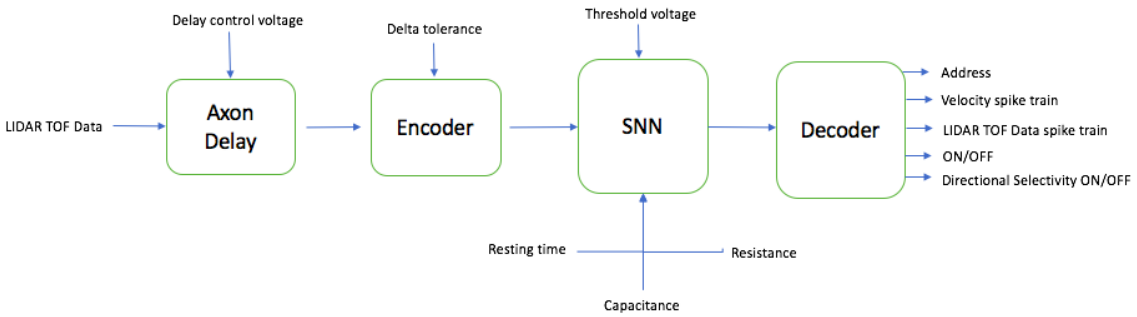


Figure 5.9: System design for SNN based LIDAR Data Pre-processing

The proposed system (Fig: 5.9) consists of mainly four modules:-

1. Axon Delay module (Mixed signal) : Described in section 4.1.3 is state of the art axon delay module that delays the incoming spike based on a programmable "voltage to time" converter.

2. Encoder (Digital): Described in section 5.5.4 is a standard "value to spike rate" converter that converts a digital value to corresponding spike train.

3. SNN based motion detection model (Analog/mixed signal): The neural network (described in section 5.5.5) processes spike trains based on LIF mathematical model (described in section 4.2.0.1) and detects temporal frequency changes.

4. Decoder (Digital): Described in section 5.5.6 works as a pulse/spike counter and is responsible for filtering out zero or low pulsed signals over a fixed period of time.

The system takes in raw TOF spikes directly from the photo-sensor of LIDAR and produces 4 major signals, Event ON/OFF signal, velocity spike train, filtered TOF spike train and directional selectivity on/OFF Boolean signal(per angle) per pixel address. Hence, this system is able to replicate human retina with additional feature of distance and displacement determination.

The Input output channels of the system as well as parametric constrains are mentioned below:
Input/Output channels :

1. Axon Delay (32x120 signal input channel): LIDAR TOF Data (Boolean)

2. Encoder (32x120 signal input channel): LIDAR TOF Data (Boolean)

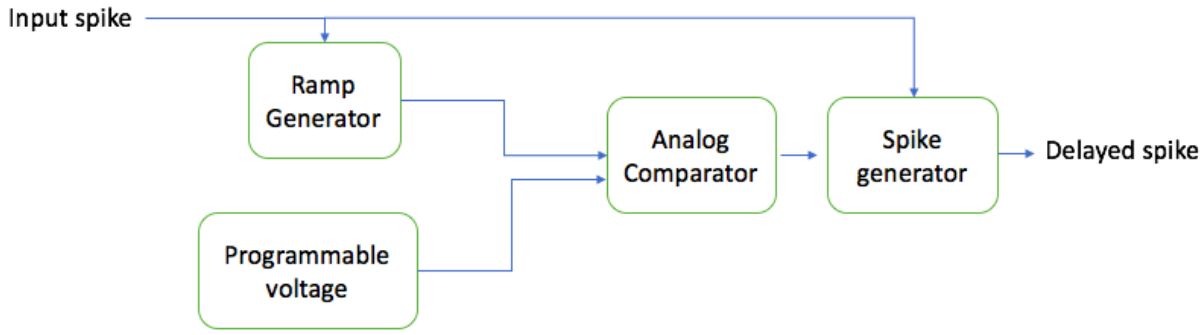
3. Decoder (4x32x120 signal output channel): Velocity spike train (Boolean), LIDAR TOF Data spike train, ON/OFF (Boolean- Event enable signals) and Directional selectivity ON/OFF (Boolean).

Input Parameters:

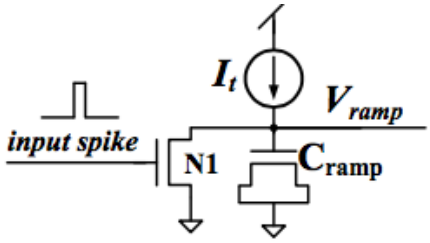
1. Axon Delay: Delay voltage (mV) - (16 bit integer)
2. Encoder: Delta tolerance (4 bit integer)
3. Spiking Neural network : Threshold Voltage (mV) - (16 bit integer), Resting time(ms) - (16 bit integer), Resistance (Mohm) - (16 bit integer), Capacitance (nF) - (16 bit integer)

5.5.3 Stage 1: Axon Delay

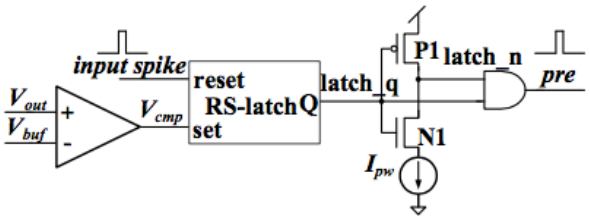
The design of axon delay for thesis is inspired from the design [7] described in section 4.1.3. This design can be used to provide a programmable delay between post-synapse of one neuron and pre-synapse of another. For the thesis we require a delay of 4 msec which is equivalent of the LIDAR frame rate (frame clock) of 25 frames/sec.



(a) Axon Delay block diagram



(b) Ramp generator Circuit [7]



(c) Spike generator Circuit [7]

Figure 5.10: Axon Delay design

In order to achieve this delay we chose to adapt the model shown in figure 5.10a. As soon as a spike is issued, the ramp generator which is a basic RC based integrator starts moving towards Vdd due to $I_t(200pA)$. This linear ramp up of voltage over time acts as delay for the circuit. This V_{Ramp} is compared with a programmable voltage

V_{buf} set by the user using an analog comparator shown in figure 5.10a. In order to achieve 4 msec delay the V_{buf} is set to $200mV$. Hence when the V_{Ramp} exceeds V_{buf} a spike/pulse is issued at the set pin of SR Latch shown in the spike generator Figure 5.10c.

Spike generator works like a normal pulse based SR latch. When a pre synaptic spike is issued, the reset pin of SR latch is set high for some time and then set low again with the drop of spike pulse. This stores a digital logic 1 in the latch. When another pulse is issued at set pin of the latch, the stored logic 1 is released in form of a spike. The pulse width of this spike is determined by a current starved inverter connected with an AND gate using programmable I_{pre} as shown in Figure 5.10c.

Therefore in order to summarize, when a pre-synaptic spike is issued, the ramp generator resets and starts rising the voltage while the SR latch in spike generator stores the spike information in form of logic 1. When the ramp voltage exceed the programmable voltage, the SR latch releases the delayed spike. Hence, after every LIDAR frame this axon delay is used to delay the TOF spike issued by the LIDAR photodetector.

5.5.3.1 Importance of Axon delay block

Axon delay acts as an analog memory element by delaying the information. One could argue why not use standard SRAM to store the information as done in the section 5.4. There are several benefits of using the axon delay mechanism over SRAM stated below:-

1. First of all it saves system memory space which can be used by other memory based operations in neuromorphic hardware like weight updates.
2. Overall system latency drops firstly, due to asynchronous way of processing for axon delay and secondly avoiding unnecessary time lost due to memory accesses.
3. Axon-delay delays/stores single pixel information in terms of spikes which makes system less resource, computation and power hungry compared to SRAM based design which stores information digitally in 12 SRAM cells per pixel.

Hence, asynchronous spike based axon delays are more suitable for neuromorphic computing than their SRAM based counterparts. The resource utilization analysis is covered in appendix 8.

5.5.4 Stage 2: Spike Encoder

Spike encoder (figure: 5.11) is a generic "Rate Encoder". Rate encoding is one of the most common encoding techniques used to represent data in spike rate form for spiking neural networks. The encoder not only feeds event/motion detector SNN but also represents the TOF (Time of flight) i.e. distance information to the subsequent classification and fusion stages.

The encoder (digital logic) was modelled in MATLAB as well as implemented in RTL so as study the resource utilization. The encoder converts input TOF spike to corresponding spike train. The relationship between spike train rate and the input value in shown below in equation 5.1.

$$F(Data) = Frequency = \Delta^T / Data \tag{5.1}$$

Where "Delta Tolerance" (Δ^T) is defined as the change required in the pixel value over time to be considered as an event, it is set by the user so as to make the system sensitive to a specific displacement in the scene represented by the the change in the Time of flight values.

This also enables the system to remove various types of thermal and luminous based noises affecting optical sensors of LIDAR.

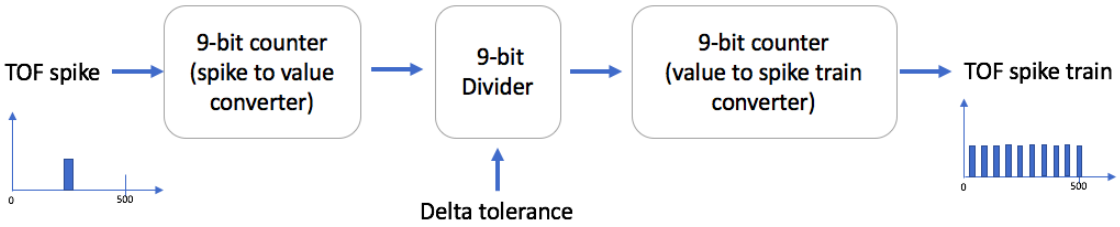


Figure 5.11: SNN encoder

The system flow of encoder is described below:-

1. Output of axon delay/photodetector is connected to the "Spike to value" converter of the encoder, this is a 9-bit counter counts upto 500 clock edges and registers a value corresponding to the number of clock edge where an input TOF spike is observed.
2. The registered value in previous stage is divided by Δ^T by a 9-bit Divider.
3. The output of divider is then fed to a "value to spike" converter which is a 9-bit counter that converts the input value into a voltage spike train with rate governed by the equation 5.1.

5.5.4.1 Current Equation

The currents applied as inputs to the spiking neurons of the proposed network 5.5.5.1 due to the encoder spike train is shown in the equation 5.2. Each pixel value k (TOF) is approximately converted into a constant input current (eq: 5.2) when the voltage spike train is applied across the LIF membrane based on eq: 5.3.

$$i(k) \propto \frac{\Delta^T}{k} \tag{5.2}$$

$$i(t) = w * [\sum_i \delta(t - t^i) * (e^{-\frac{t}{\tau}})] \tag{5.3}$$

Where $\delta(t - t^i)$ is an impulse train from i^{th} synapse at time t and τ is the time constant for the LIF RC circuit shown in figure **??** and w is the synaptic weight of the connection.

5.5.4.2 Output spike train frequency range

The input TOF values are in range [0 to 500] [0 to 50 m]. Hence based on the equation 5.1. (The range can be increased while decreasing the input sensitivity by diving the input TOF value with higher delta tolerance value). The output frequency range of the spike trains is [8 to 1000] Hz.

5.5.4.3 Encoder Resource Utilization

Based in the RTL implementation of the encoder, we concluded that three major modules are involved namely two 9-bit counters and one 9-bit divider per encoder. The proposed design requires two encoders per pixels. Hence, four 9-bit counters and 2 9-bit dividers are required per pixel. Hence, for the entire 120x32 pixel design we require 15360 9-bit counters and 7680 9-bit dividers.

5.5.5 Stage 3: SNN based event detection model design

This section provides a detailed description of the spiking neural network based event detector module as a major part of the neuromorphic retina activity. Like human retina that extracts visual information with respect to luminous changes in space and time, the proposed network is able to cognitively detect dynamic changes in the scene. The section is divide into three parts:-

1. Neuron model
2. Network model
3. Simulation model

5.5.5.1 Neuron model

Neuron used in the SNN model (section: 5.5.5) and the simulator is LIF based neuron (Figure: 5.12) as described in section 4.2.0.1. Certain properties of this Neuron like threshold voltage, RC-time constant as well as the resting time were exploited so as to achieve the task of finding temporal frequency changes in the incoming spikes.

Four major input parameters (Fig: 4.1) mentioned before are described below:-

1. Threshold Voltage (V_{Th}): When $V(t) > V_{Th}$ in equation 4.1 , a spike is issued and transmitted to the down-stream synapses.
2. RC-Time constant (τ): It controls the rate at which the capacitor charges and discharges in the LIF circuit.

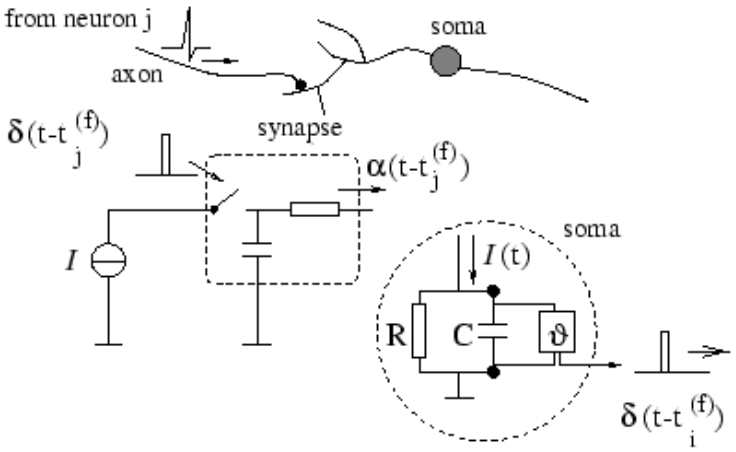


Figure 5.12: LIF schematics and circuit diagram [6]

3. Resting Time (T_{Rest}): Biological neurons enter a refractory period immediately after a spike is issued during which another spike cannot be issued. likewise T_{Rest} determines the time duration in which next spike can not be issued.

5.5.5.2 Effects of Varying Neuron Parameters

The rate of post synaptic spike train is dependent on the neural parameters mentioned in section 5.5.5.1. This section discusses the impact of each parameters on the post synaptic spike train.

1. Threshold Voltage (V_{Th}): Threshold voltage is responsible for setting the sensitivity of the temporal input synaptic spikes. That means if V_{Th} is low then the neuron will be sensitive to a larger pallet of input spike rates as compared to when it is set high because lower V_{Th} will enable the neuron to spike even if the input spike rate is low.

2. RC-Time constant (τ): τ is described as a product of Resistance and Capacitance of the RC circuit (Neuron membrane). It is the time required to charge a capacitor to 63 % of full charge or to discharge it to 37 % of its initial value.

   Hence, if τ is dropped then the‘ tendency of the neuron to hold the voltage drops. This implies that an input synaptic spike train with low frequency and power will not impact much on the post synaptic spike train because the neuron will discharge quickly.

   Therefore, in order to maintain the dynamic range of the input synaptic spike rates (representing input (TOF) values to the overall system) τ should not be too less. On the other hand the simulator does not allow high τ and produces delay in the system over time. Hence, a balanced value of Resistance R and capacitance C is selected based on several experimentation. R is fixed at 250 GΩ and C is fixed at 100 fF.

3. Resting Time (T_{Rest}): Lower (T_{Rest} is more sensitive to input spikes as the system is ready for another synaptic spike soon after a post synaptic spike. Hence, The dynamic range of post synaptic spike also increases. Therefore, The (T_{Rest} is fixed at a low value (but high enough for the proper functioning of the simulator) at 10 msec.

Hence, in the thesis R, C and T_{Rest} values are kept constant (as they are hardware parameters) while V_{Th} is kept as variable controlled by the user.

5.5.5.3 Network Design

Main task of the spiking neural network is to identify frequency changes in the incoming spike trains. These frequency changes represents change in TOF LIDAR values which indeed represents displacement of the target object.

Hence, The neurons in the network are sensitive to both positive as well as changes in the frequency of incoming spike train of a pixel. To achieve this the concept of axon delay from the human neural system has been incorporated in the design. Like in biologic neural networks, phase shifts in synaptic spike trains are introduced through axon delays, similarly axon delays (Section:4.1.3) are used in the proposed network to delay spikes from the LIDAR so that the neurons receive TOF spikes from current as well as the previous frame concurrently.

This enables the neurons to sense changes asynchronously. The design is inspired from the SNN explained in paper [29]. Hence, based on the axon delay mechanism a model is proposed to detect moving objects. Its structure is shown in Fig: 5.16.

The network is divided into three major phases:-

1. Axon delay
2. Positive/Negative change detector
3. Net change detector

In order to detect positive/negative change we have to target the current flowing across the neural membrane. If this current is high enough to raise a voltage surpassing the threshold voltage then we can say an information has been detected. Hence, we use the concept of inhibitory and excitatory synapse in this network. An inhibitory synapse produces negative current across the neural membrane while an excitatory synapse produces positive current.

In mathematical terms inhibitory synapse multiples the input current with a negative weight while excitatory synapse multiples the input with positive weight. Current across the neural membrane is directly proportional to the incoming spike rate represented by the equation 4.1

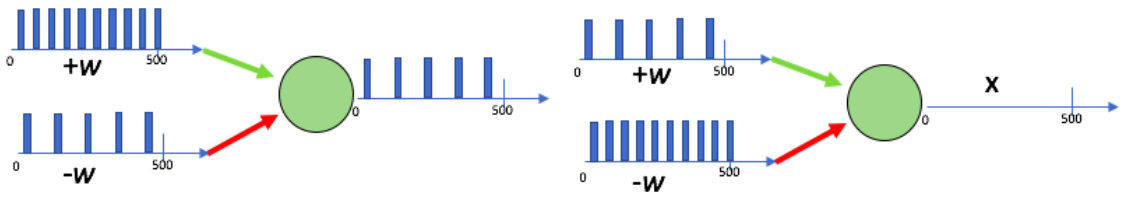


Figure 5.13: Neuron sensitive to positive change

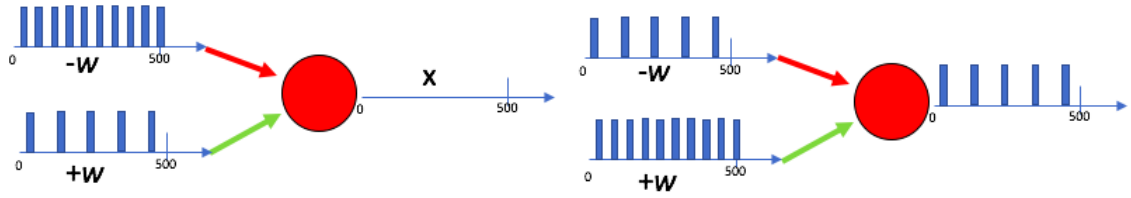


Figure 5.14: Neuron sensitive to negative change

Hence, using this information a neuron synapse interaction is shown in the figures 5.13 and 5.14. Suppose first synapse is the current pixel value while second synapse is the previous frame value. As shown in figure 5.13 a positive change is occurred when the current across first synapse is higher than the current across second synapse. First synapse is an excitatory synapse while the second one is inhibitory. Hence, if the synaptic current on the excitatory channel is higher than the inhibitory channel then it is more likely that the membrane voltage will surpass the threshold voltage to generate an output axon spike.

Similarly figure 5.14 demonstrates negative change where first synapse is inhibitory in nature while second synapse is excitatory. Hence, if first synapse has higher input current than second then no output spike is produced. An output spike is only produced when the current on second synapse is higher than the first one. Which means this neuron is sensitive to all the negative changes occurring.

In order to detect the net change we need a neuron that spikes when any one of the synapse spikes. This is called a "Winner takes all" type spiking neuron. In order to achieve this we increased the synaptic weight to a comparatively high value so that the neural membrane voltage surpasses the threshold voltage whenever any one of the synapse spikes as shown in figure 5.15.

The proposed neural network is a feed forward network of two neural layers as shown in figure 5.16. Current and delayed TOF spike trains enter the network from each pixel $P(x, y)$ concurrently to the first layer $SN_x(x, y)$.

As shown in equation 4.1 a constant current is generated due to the input synaptic spike train produced by the encoder. Let $I_x P(x, y, t - t_0)$ be the current from the input pixel $P(x, y)$. I_e and I_i represents excitatory and inhibitory synaptic

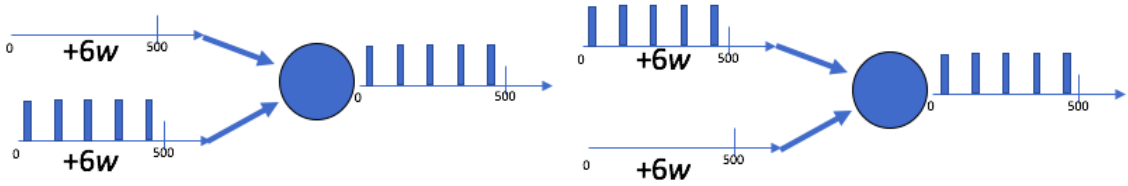


Figure 5.15: Neuron sensitive to both positive/negative change

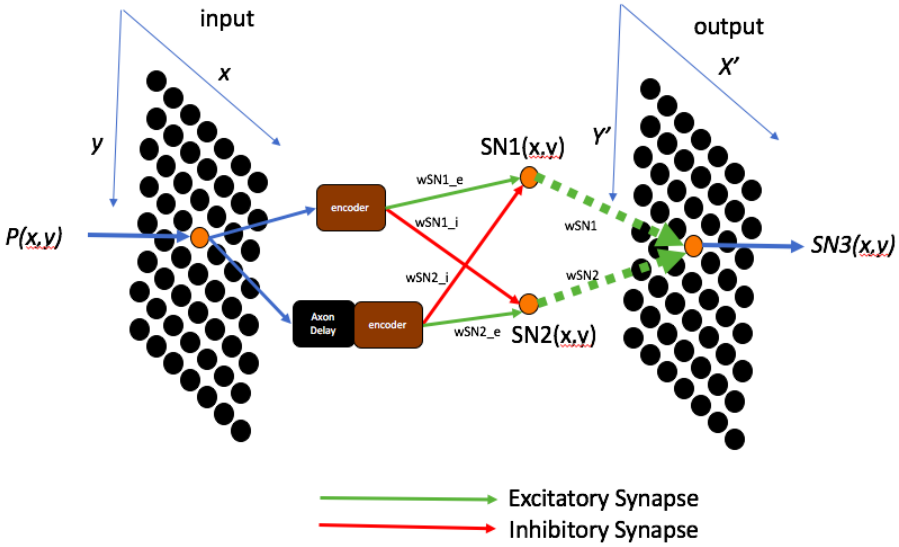


Figure 5.16: Spiking neural network model for motion detection

currents respectively. If $I_eP(x, y, t)$ is greater than $I_iP(x, y, t - f)$ then the neuron SN_1 will spike. if $I_eP(x, y, t - f)$ is greater than $I_iP(x, y, t)$ then SN_2 will spike. This enables the system to detect positive and the negative changes in the pixel $P(x, y)$.

The second layer $SN3$ contains one neuron per pixel. This neuron is "winner takes all" type neuron as described in previously. The synaptic connections $wSN1$ from neuron SN1(x,y) and $wSN2$ from neuron SN2(x,y) both have comparatively high weights to spike the neuron SN3 when either one of the neuron from previous layer spikes.

Therefore, temporal changes in pixel (TOF) values of LIDAR frames are reflected in the output neuron layer.

5.5.5.4 SNN Simulation Model Design

Based on the neural network design mentioned above, a feed forward neural network model was developed in MATLAB for simulating the network. In order to perform simulations, TOF data represented in int32 format was taken directly into the simulation

environment instead of TOF temporal spikes. Hence, the simulation encoder converts the TOF value directly into spike train with frequency governed by the equation 5.1. For efficient simulation, the delay block is placed after the encoder which instead of delaying a single spike delays the entire spike train.

Neuron A and B shown in figure 5.17 represents the current and the delayed inputs respectively. Neuron C spikes only when the frequency of the spike train at A is significantly (based on user controlled threshold) higher than the frequency of spike train at B. Neuron D spikes only when the frequency spike train of B is significantly higher than the frequency of spike train of A. Neuron E spikes when either of the C or D neuron spikes.

Hence, neurons C and D are activated by positive and negative change in the pixel value respectively while neuron E is activated when there is either positive or negative change occurring in the pixel value over time.

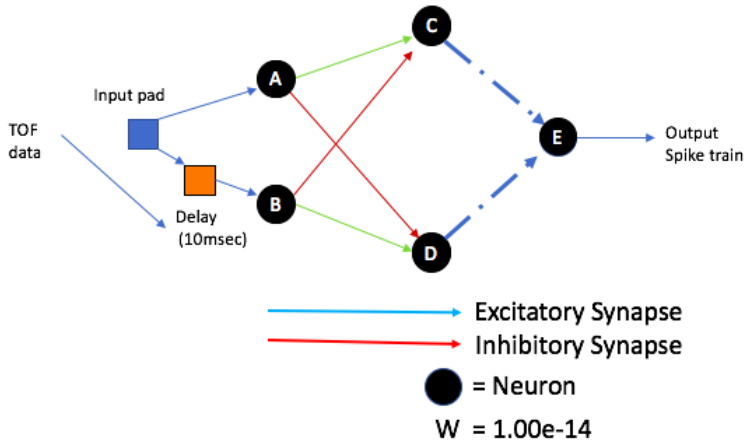


Figure 5.17: Spiking neural network model for motion detection

Constant synaptic weights are applied to the network. The excitatory connections between first and second layers W_{AC} and W_{BD} are assigned $+1.00e^{-14}$, inhibitory W_{AD} and W_{BC} are assigned $-1.00e^{-14}$ (W is the charge stored across the membrane capacitor in Coulomb - hence to analyse the network response due to change in C and V across the capacitor which are in the range of fF and mV respectively, we chose W to be $+1.00e^{-14}$). The neuron E in the last layer acts as "winner takes all" type spiking neuron i.e any synaptic spike will result in an instantaneous post synaptic spike.

Hence, to achieve this, the synaptic weights between second and third layers are set very high i.e six times the synaptic weights in other network layers ($6.00e^{-14}$) as shown in figure 5.17. The simulation results generated from the network are described in detail in Chapter 6.

5.5.5.5 SNN Resource utilization

The SNN architecture uses 3 spiking neurons per pixel which when extrapolated to the entire LIDAR frames constitutes to 11,520 (120x32x3) spiking neurons, 3840 axon delays and 23,040 synapses.

5.5.6 Stage 4: Spike decoder

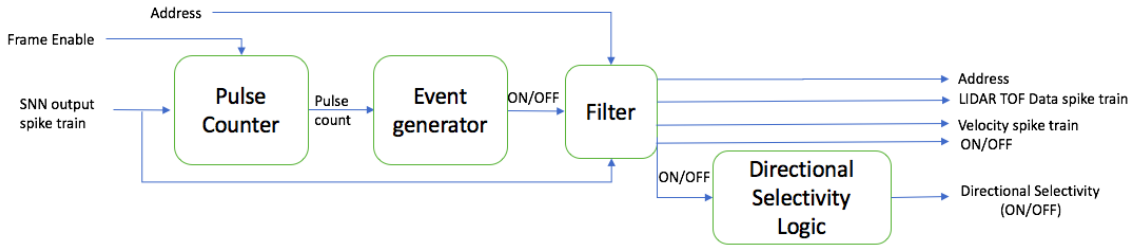


Figure 5.18: Spike Decoder schematics

Spike train generated from the neural network (mentioned in the previous section 5.5.5.4) represents the movement of an object recorded by the Time of flight values of LIDAR. If the rate of this spike-train is zero then there was no event. The rate also represents the displacement. Hence, the decoder determines the rate of incoming spike train and generates ON/OFF signals accordingly.

The decoder is subdivided into four parts. First, Pulse counter that counts the number of pulses occurring the spike train over specific time (in this case frame rate). Second, event generator, it generates ON/OFF signal based on the formula given below:-

$$f(v) = \begin{cases} ON, & \text{if } v_{spike} \neq 0 \\ OFF, & \text{if } v_{spike} = 0 \end{cases} \tag{5.4}$$

Where $f(v)$ is the multiplexer output and v is the SNN output spike train frequency or rate.

Third, filter that acts as a multiplexer so as to filter out pixel values based on the ON/OFF flag from the event generator. Hence, only those pixels where motion occurred are allowed to pass through the filter. This means that both TOF input spike trains as well as the SNN output displacement spike trains are filtered by the filter. Thus, the decoder generates the AER [22] based event information at every pixel address.

Fourth, Directional selectivity module, this module spikes when more than half of the vertical pixels encounter event. Hence, for every vertical column of a LIDAR frame (which represents each degree in 120 degree field of view) an ON/OFF signal is generated. This signal represents whether in a particular direction some motion has occurred.

Decoder produces majorly three types of signals/ features for further SNN based classification stages:-

1. Filtered TOF signal
2. Velocity/displacement signal
3. Directional selectivity signal selectivity

5.5.6.1 Displacement/velocity spike train

Based on the neuron parameters (section 5.5.5.2) the displacement information is represented in 8 different rates which are determined both by the current TOF values as well as the difference between two consecutive TOF values as shown in section **??**.

Frequency range of the velocity spike train is mainly influenced by the T_{Rest} parameter of the neuron. The relationship is described below:-

$$\upsilon_{Range} \propto \frac{1}{T_{Rest}} \tag{5.5}$$

where υ_{Range} is the frequency range of velocity spike train. Hence, lower the value of T_{Rest} higher will be the frequency range.

5.5.6.2 Directional Selectivity

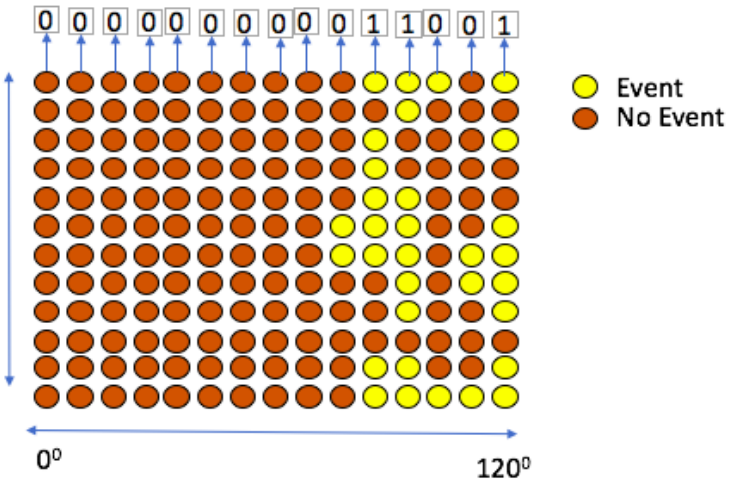


Figure 5.19: Directional selectivity functionality

Directional selectivity (DS) represents the DS cells in biological retina. These cells are fired when the retina captures motion in particular spatial direction. Similarly the decoder fires only when temporal motion is observed in a particular direction. As shown in figure 5.19, if the SNN senses motion in more than half of the pixels in a particular direction then the decoder cell representing that angle fires.

5.5.6.3 Decoder Resource utilization

Based on the RTL implementation of decoder, three major components are used per pixel namely one 9-bit counter, one comparator and one multiplexer. One 32 bit adder and a comparator is used for every 32 pixel column for directional selectivity. Hence for entire network, 3840 9-bit counters and multiplexers are used while 3960 comparators and 120 32-bit adders are used.

5.5.7 Net resource utilization of the neuromorphic retina

Proposed neuromorphic retina is a mixed signal processing unit. The following resources are used:-

1. Analog: In terms of transistor counts, approximately 195k to 337k transistors including LIF neurons as well as axon delays
2. Digital: This analysis includes both spike encoder as well as decoder. They require 19.2k 9-bit counters, 3840 multiplexers, 3960 comparators, 120 32-bit adders and 7680 9-bit dividers.

5.6 Conclusion

This chapter presented a detailed design of each and every module used during the thesis in order to design, analyze and verify the neuromorphic retina for LIDARs:-

1. DSP based "Motion based" scene segmentation module.
2. SNN based retina module including submodules namely axon delay, spike encoder, neural network and spike decoder.
3. LIDAR raw TOF data visualization and verification module.

Based on above discussions, certain conclusions are reported below:-

1. DSP based scene segmentation module produces two different matrices namely "objects in motion" and "static objects". These datasets are used for verification of the results generated by SNN based module.
2. SNN based module preprocesses the LIDAR data and extracts event based information at sensor stage itself. This feature extraction process is inspired from biological retina that extract useful motion information before sending it to visual cortex in brain.
3. The neuromorphic processing unit or retina extracts the following features from raw LIDAR data:-
   (a) Moving objects
   (b) Velocity of moving objects
   (c) Directional selectivity i.e. Event occurred per angle in the field of view.

These features are encoded in terms of spikes and can be further processed directly on a neuromorphic computer for further low powered LIDAR based classifications and optical flow predictions.

Various experimentation were conducted on the proposed SNN architecture. The results and the inferences are discussed in next chapter.

6 Experimentation and results

6.1 Overview

Previous chapter described detailed construction of neuromorphic retina and its verification algorithm. This chapter deals with simulations, results and inferences drawn from various experiments conducted to understand and verify the functionality and quality of proposed system. Simulations were conducted in the following manner: -

1. Scene Independent Simulations: Observed output based on manual input stimuli so as to understand the behavior of network due to varying input parameters.
2. Scene dependent Simulations: Observed the effect of applying proposed network on actual LIDAR frames to evaluate the effect of silicon retina on overall system.

Following matrices were analysed based on the experimentation conducted:-

1. Data Throughput
2. Power consumption
3. Data Quality
4. System robustness

6.2 Scene Independent Simulations

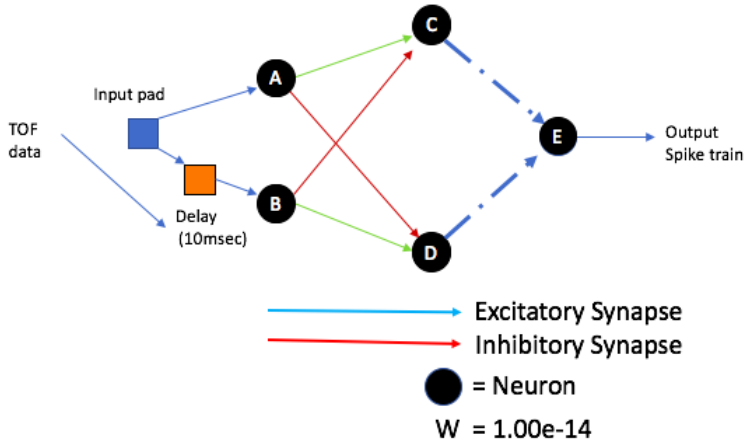


Figure 6.1: Neural network model showing connection parameters

Experiments were conducted over single pixel SNN architecture as shown in the figure 6.1

6.2.1 Simulations demonstrating Event detection

The weight, delay and plasticity of the neural connections were fixed as shown in the figure 6.1. The plasticity was fixed to zero because there was no learning involved in the process. The weights between N1 and N3 and N2 and N3 were forced to 6 times w ($w = 1.00e^{-14}$) so that any spike in either of the two neurons (N1/N2) spikes the N3 neuron and makes it winner takes all type LIF neuron.

Keeping network parameters namely $\Delta^T, V_{th}, T_{Rest}$ and τ constant to 4, 120mV, 10msec, 25msec respectively, the network was activated by a range of input stimuli so as to simulate its event detection property. (The network parameters were set based on multiple experimentation so as to come up with best results, the details about affect of these parameters on the network is described in detail in upcoming sections).

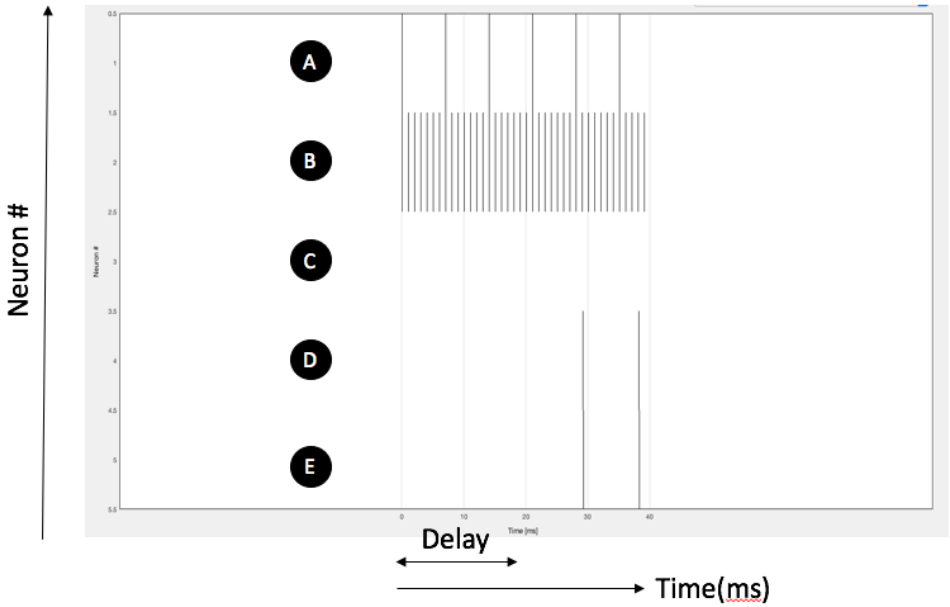


Figure 6.2: Wave form representing negative event

The wave forms shown above (figure: 6.2) represent post synaptic voltages of all the neurons in the network over time. Wave form represents output of all the neurons described in the network (figure: 6.1) A and B are inputs, C and D are hidden layer neuron responsible to positive and negative event detection respectively. E is the output "Winner takes all" neuron.

In order to detect positive event, a set of input TOF values was generated such that every successive input is lower than the previous value. (atleast by a margin of

Δ^T). The simulation results proved that the network is able to detect positive changes in the input data stream. Positive event is detected when there is a significant drop in the TOF value of the pixel. Lower the TOF value higher the frequency of its encoded spike train (eq: 5.1). Frequency of the spike train at input pad A is higher than the the frequency of the spike train at input pad B. Hence, the neuron C is activated while neuron D remains at rest. Neuron E activates instantaneously with the activation of neuron C. Therefore, the data collected at neuron C indicates occurrence of a positive event.

In order to detect negative event, a set of input TOF values was generated such that every successive input is higher than the previous value. (atleast by a margin of Δ^T). The simulation results proved that the network is able to detect negative changes in the input data stream. To demonstrate this behavior, screen shot of the system waveform is shown in figure: 6.2. negative event is detected when there is a significant raise in the TOF value of the pixel. higher the TOF value lower the frequency of its encoded spike train (eq: 5.1). Frequency of the spike train at input pad A is lower than the the frequency of the spike train at input pad B. Hence, the neuron D is activated while neuron C remains at rest. Neuron E activates instantaneously with the activation of neuron D. Therefore, the data collected at neuron D indicates occurrence of a negative event.

Hence, these demonstrations prove that the network is able to detect events in the successive LIDAR frames. These demonstrations were performed keeping various input parameters constant, upcoming section describes the effect of altering these parameters on the output spike train and what it means to the overall system.

A similar simulation with 3 parallel neurons is demonstrated below (figure: 6.3 and 6.4), with the network performing event detection across multiple parallel neurons with change in LIDAR frame. This time instead of providing 2 separate inputs we activated the network with single input data stream per pixel. A delay (Axon delay) was introduced so that the network could compare the successive data frames concurrently. The results demonstrate that pixels 1 and 2 are able to detect negative change while pixel 3 is able to detect positive change.

6.2.2 Network response analysis (Input stimuli)

As discussed in previous section, the proposed and demonstrated network is able to detect changes in the input TOF stimuli. When there is no change, the output spike rate is zero because the output neuron is not activated. If there is a significant change then the output neuron fires at specific rate. This section gives an analysis of the network response, its relationship with the input stimuli and its relevance in overall system design.

Network response $f(\upsilon)$ can be represented as a function of frequency or rate (υ) of the output spike train. In order to analyze the response, input TOF was varied in 5 different ranges (1-50), (50-100), (100-200), (200-300) and (300-500). These ranges also represent a range of distance of the target from the LIDAR sensor. Hence the

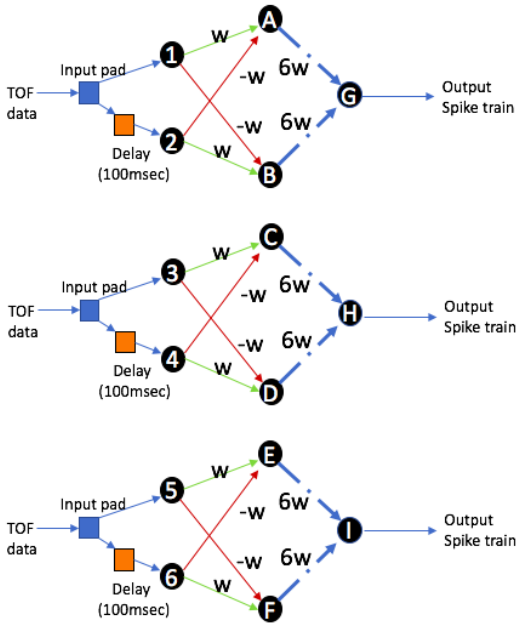


Figure 6.3: Neural network with 3 parallel sub-neural networks

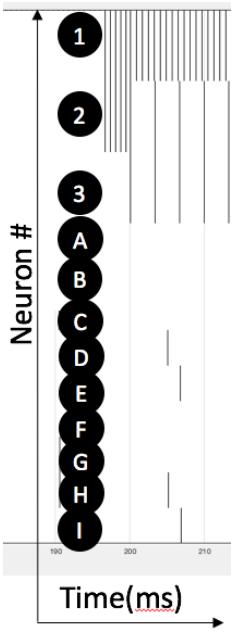


Figure 6.4: Simulation results for spiking with 3 parallel sub-neural networks

rate of the output spike train is observed at different input values as well as their differences at constant input parameters $\Delta^T = 4$ and 15, $V_{th} = 120mV$, $\tau = 25msec$ and $T_{rest} = 10msec$.

Output spike train also represents the displacement occurring at a specific distance at specific time. Hence by observing the network spike response we can extract the

velocity information of the objects moving at certain distance and angle.

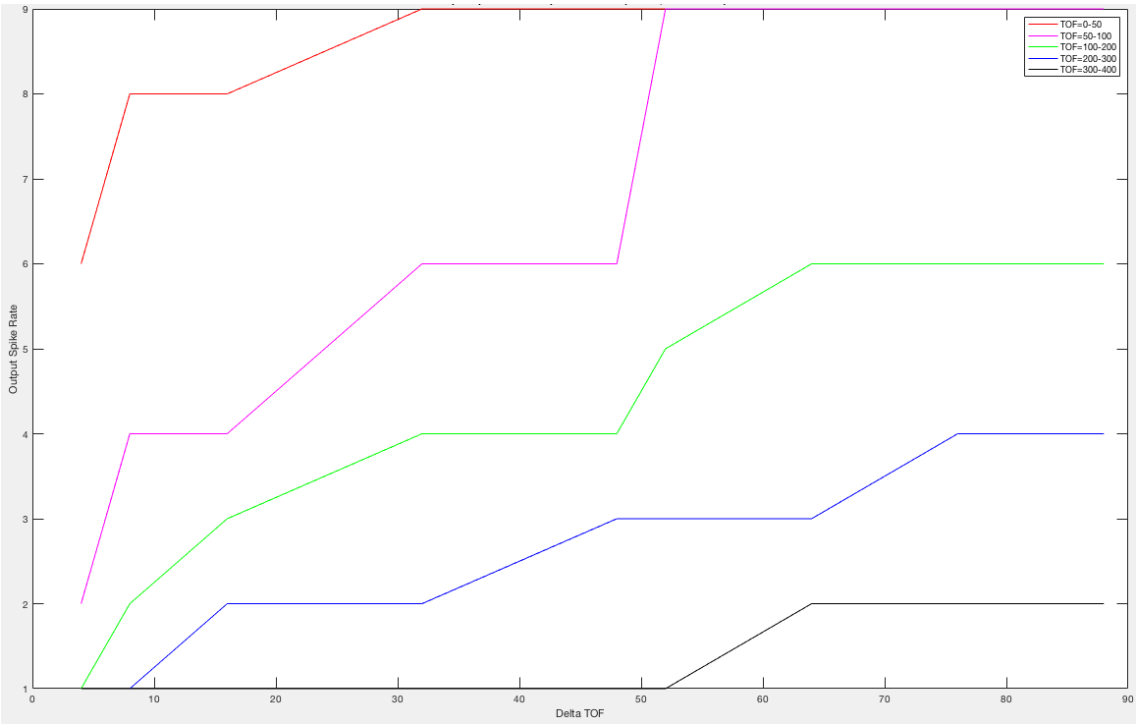


Figure 6.5: Network Response at $\Delta^T = 4$

6.2.2.1 Inference

Based on the plots shown in figure 6.5 and 6.6 we can infer following properties :-

1. Output spike rate increases as input Delta TOF (Δ_{TOF}) increases (Delta TOF is the difference between two consecutive TOF values of a pixel).
2. Output spike rate decreases as the input TOF value increases.

Hence, the output rate depends on both Δ_{TOF} as well as the successive TOF value of the inputs. This implies that a SNN classifier will require both Delta TOF as well as current TOF values to understand or extract the velocity information.

$$f(v) \propto \frac{f(\Delta_{TOF})}{f(TOF)} \tag{6.1}$$

6.2.3 Network response analysis (Input parameters)

Network response was observed by varying three major input parameters:-

1. Neural threshold Voltage (V_{th})
2. Neural Resting time (T_{rest})
3. Neural Time constant (τ)

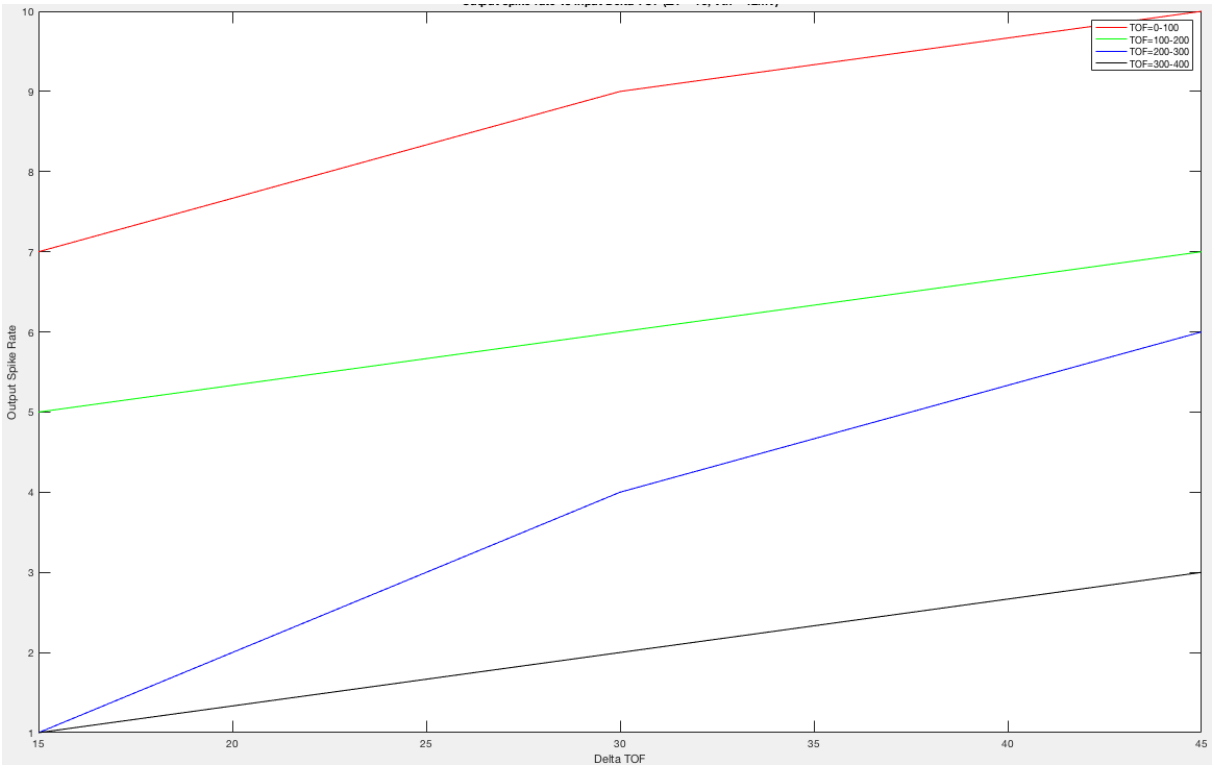


Figure 6.6: Network Response at $\Delta^T = 15$

6.2.3.1 Neural threshold Voltage (V_{th})

Keeping $\tau = 25msec$ and $T_{rest} = 10msec$, V_{th} was varied to study its effect on the network response at different Δ^T. Output response was observed based on input stimuli with V_{th} at 120 mV, 150 mV and 200 mV. Sensitivity is defined as minimum difference between consecutive pixel values that can be detected by the network. Sensitivity was observed for different input (TOF/Δ^T) values as shown in the fig: 6.7.

The simulation results show that the sensitivity increases with increase in input TOF values and increase in V_{th} as shown in eq: 6.2. Two major inferences are drawn from the experiments. First, keeping threshold voltage low (120mV) guarantees that the system will maintain constant sensitivity throughout the pallet of input values. This implies that once the delta tolerance is fixed by the user then an event will be reported only when the change in the value of pixel is more than the delta tolerance.

$$Sensitivity \propto f(V_{th}).f(TOF) \tag{6.2}$$

Second, keeping threshold significantly high enables the system to have a variable sensitivity. This means that the sensitivity will increase as the input TOF increases. This implies that as we go away from the sensor, the ability of the sensor to finely detect displacements decreases. Hence, the system cognitively fine tunes to the objects near the LIDAR source and course tunes to the objects far from the source. This property can be further exploited to decrease the sensor throughput by adjusting the sensitivity

as required by the user.

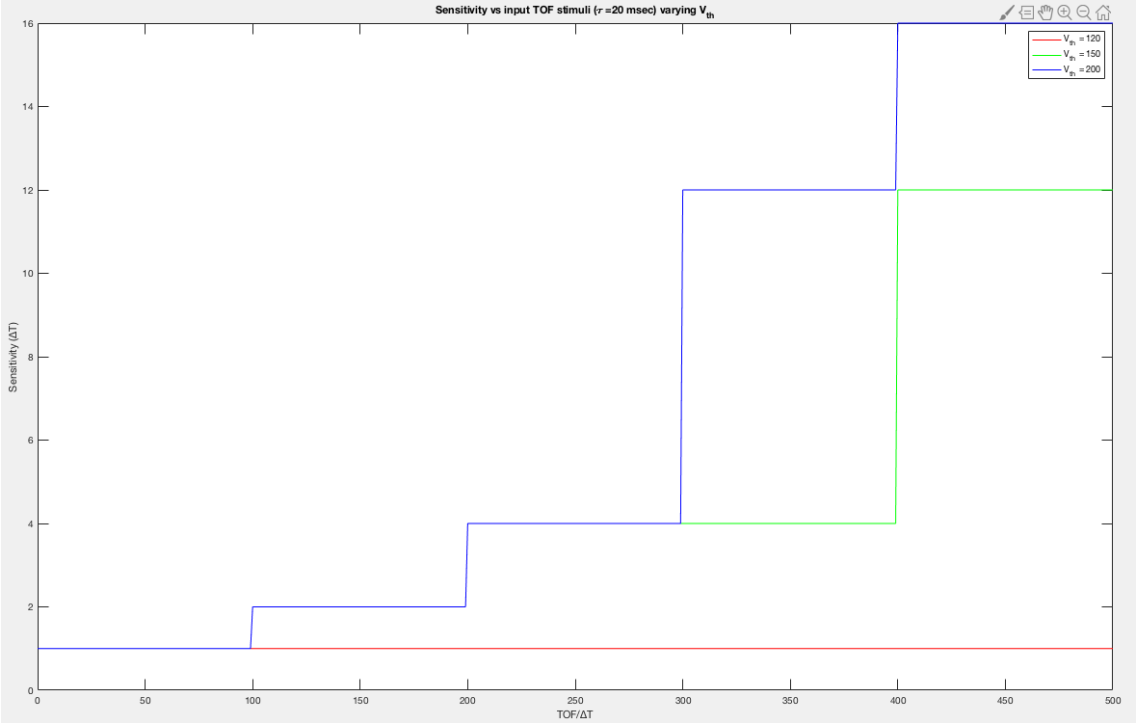


Figure 6.7: Sensitivity (Δ^T) vs Distance (TOF/Δ^T)

6.2.3.2 Neural Time constant (τ)

Keeping V_{th} = 120 mV and $T_{rest} = 10msec$, τ was varied to study its effect on the network response at different Δ^T. Output response was observed based on input stimuli with τ at 10 msec, 15 msec, 20 msec and 25msec. Sensitivity was observed for different input (TOF/Δ^T) values as shown in the fig: 6.8.

The simulation results show that the sensitivity increases with increase in input TOF values and decrease with τ as shown in eq: 6.3. Two major inferences are drawn from the experiments. First, keeping time constant high (25msec) guarantees that the system will maintain constant sensitivity throughout the pallet of input values. This implies that once the delta tolerance is fixed by the user then an event will be reported only when the change in the value of pixel is more than the delta tolerance.

$$Sensitivity \propto \frac{f(TOF)}{f(\tau)} \tag{6.3}$$

Second, keeping threshold significantly low enables the system to have a variable sensitivity. This means that the sensitivity will increase as the input TOF increases. This implies that as we go away from the sensor, the ability of the sensor to finely detect displacements decreases. Hence, the system cognitively fine tunes to the objects near the LIDAR source and course tunes to the objects far from the source. This property

can be further exploited to decrease the sensor throughput by adjusting the sensitivity as required by the user.

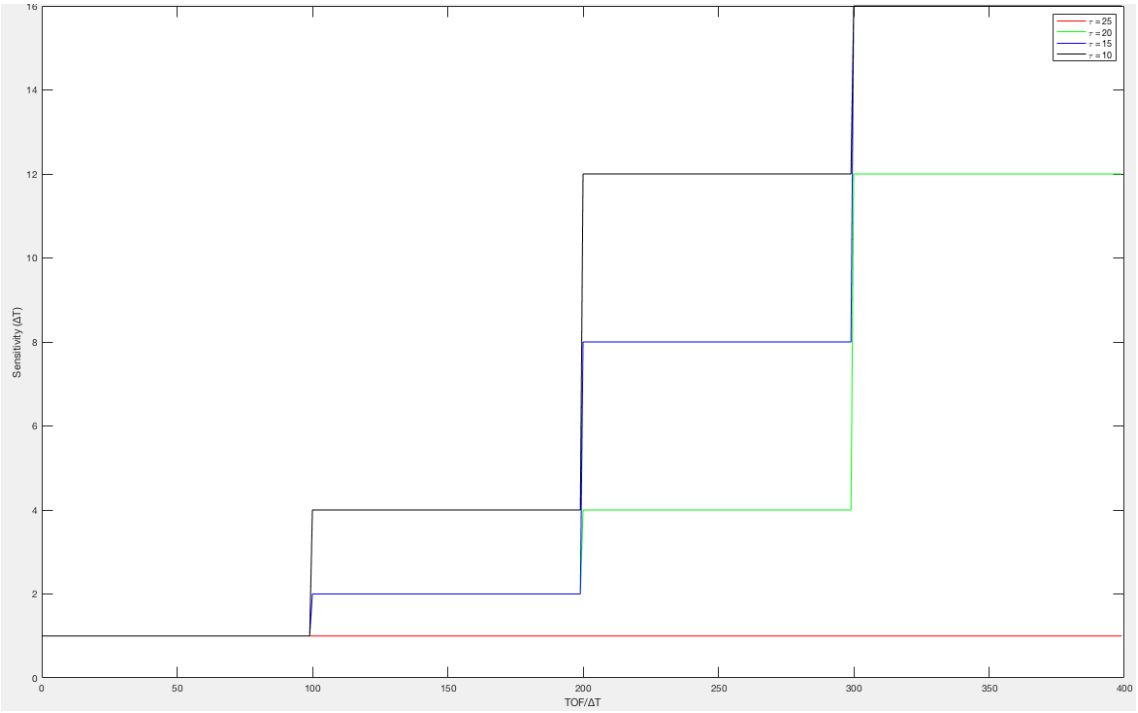


Figure 6.8: Sensitivity (Δ^T) vs Distance (TOF/Δ^T)

6.2.3.3 Neural resting time (T_{rest})

Keeping V_{th} = 120 mV and $\tau = 25msec$, T_{rest} was varied to study its effect on the network response at different Δ^T. Output response was observed based on input stimuli with T_{rest} at 5 msec, 10 msec, 15 msec and 20 msec. Output frequency range or bandwidth was observed for different input (TOF/Δ^T) values as shown in the fig: 6.9.

$$Bandwidth_{Spike} \propto \frac{1}{f(T_{rest})} \tag{6.4}$$

The simulation results show that the bandwidth increases with decrease in T_{rest} as shown in eq: 6.3. From the result shown below we can infer that as T_{rest} decreases the output spike bandwidth increases. Hence, the displacement can be represented more accurately as we decrease T_{rest}.

6.2.3.4 Inference

Hence, from the discussions made in this section we can conclude that the network response is a function of Δ_{TOF}, TOF, V_{th}, T_{rest} and τ.

$$f(\upsilon) = f(\Delta_{TOF}, TOF, V_{th}, T_{rest}, \tau) \tag{6.5}$$

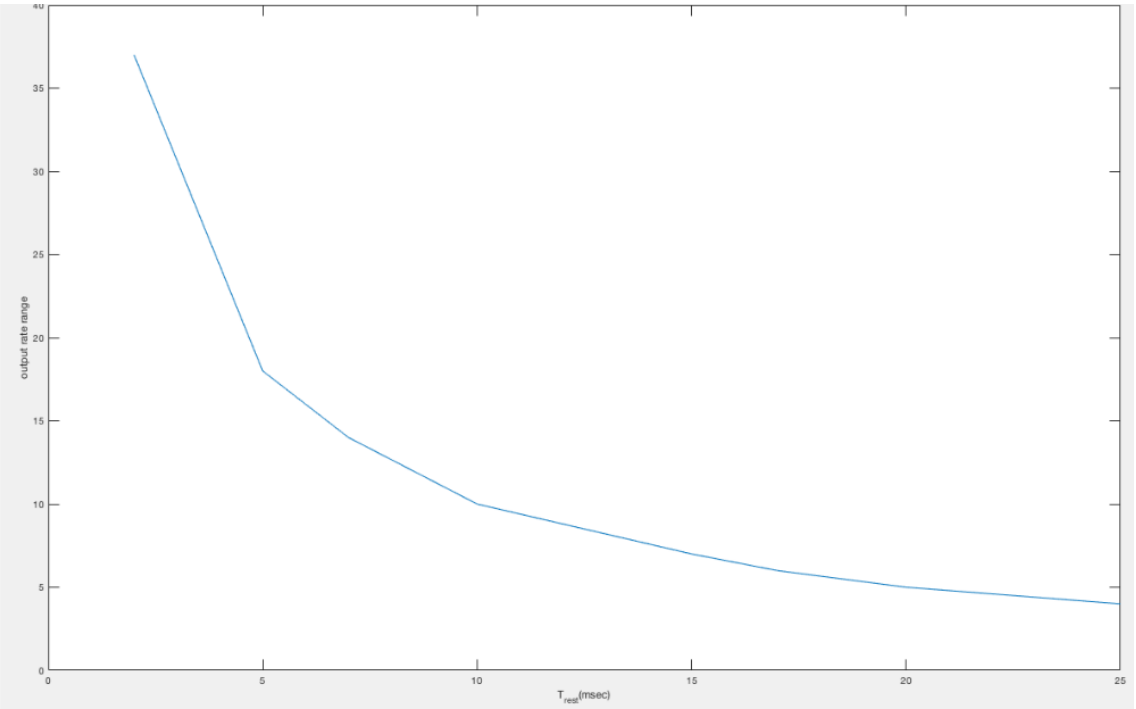


Figure 6.9: Bandwidth vs T_{rest}

6.3 Scene Dependent Simulations

This section demonstrates the properties, functionalities as well as robustness of the proposed neuromorphic retina. The system is applied across actual raw LIDAR TOF data path. The dataset describing two different scenarios with 5 sets each was acquired from Infineon Technologies BV, Netherlands.

6.3.1 Neuromorphic Retina functional simulations

As described in previous chapter, proposed Neuromorphic Retina is able to detect events at particular distance and particular direction. Two major functionalities mentioned below will be demonstrated in the upcoming sections.

1. Event based data filtering
2. Directional selectivity

6.3.1.1 Event based data filtering

To demonstrate filtering property of the system, the LIDAR frames were passed through 120x32 pixel parallel neural network with constant input parameters, $\tau = 25msec$, $T_{rest} = 10msec$, $\Delta^T = 4$ with varying V_{th} at 120 mV and 200 mV. As discussed in the previous section low V_{th} has constant sensitivity while high V_{th} has higher sensi-

tivity. Hence, we are able to demonstrate the network response at variable sensitivities.

The decoded output matrices are then imported to the custom developed Matlab based LIDAR point cloud visualization tool to generate video files, snippet of which is displayed below. Two major scenarios were captured by the sensor. First scenario: LIDAR mounted on a moving car and second scenario: LIDAR mounted on a non moving car and a single person is moving away from the sensor.

Simulations show that the retina is able to detect events in the scene and filter out the redundant static information. Figure 6.11 and 7.2 clearly demonstrates the reduction of static data due to the retina. There is even more data reduction in part c of figure 6.11 due to coarse tuning on the moving objects at far distances.

The quantitative analysis of net throughput reduction is presented in the next section. These demonstrations show that the retina is able to detect events according to programmable sensitivity and report only important dynamic LIDAR information to the SNN based classifier and the sensor fusion processing unit.

This functionality resembles the event based processing property of the biological vision system were retina pre-processes the information acquired from visual sensors in eye to extract event based important information and transmit only that information to the visual cortex in brain.

6.3.1.2 Directional Selectivity

Directional selectivity is a property of biological retina where certain group of cells are sensitive to certain direction. Incorporating this property to LIDAR means the Neuromorphic retina should detect events happening at certain direction, hence the system should be sensitive to point out events per direction.

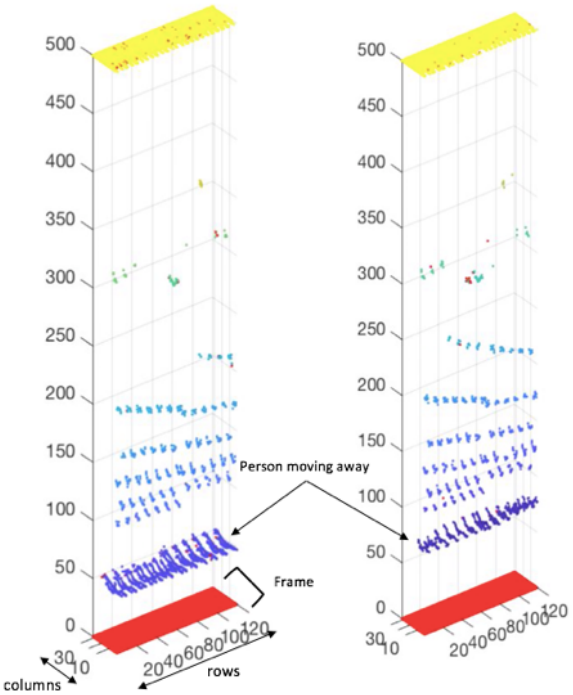
To simulate this property the directional selectivity matrix form the network is imported to the visualization tool so as to compare the events/angle Boolean information with actual LIDAR 3d point cloud. Figure 6.12 demonstrates the functional property. Left side of the image shows the actual LIDAR 3d point cloud while right side shows the 120x32 pixel LIDAR frame where any event occurring per angle is displayed in black.

Hence, due to this property we are able to detect event happening not only at particular distance but also at particular angle as well. This has several benefits for section based processing where other sensors of the fusion network can focus or fine tune in particular direction.

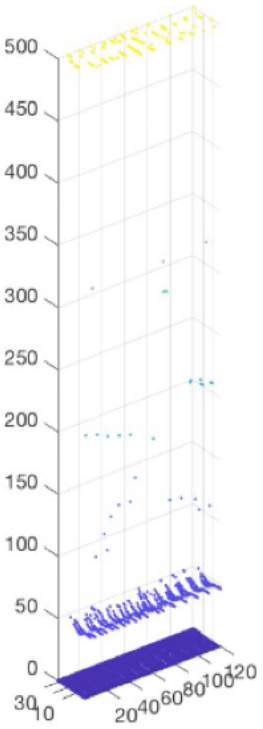
6.3.2 Neuromorphic Retina properties

This section gives qualitative analysis of neuromporphic retinal properties like:-

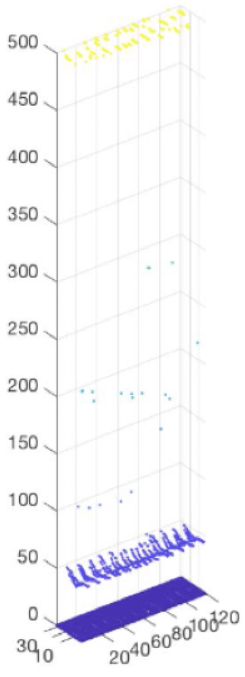
1. Throughput reduction



(a) shows the visualization of the scene in 3d point cloud environment where a person is moving away from the car



(b) shows the filtered output based constant delta tolerance



(c) shows the filtered output based variable delta tolerance

Figure 6.10: Scenario: LIDAR mounted on a stationery car, Visualization from. SNN decoder output

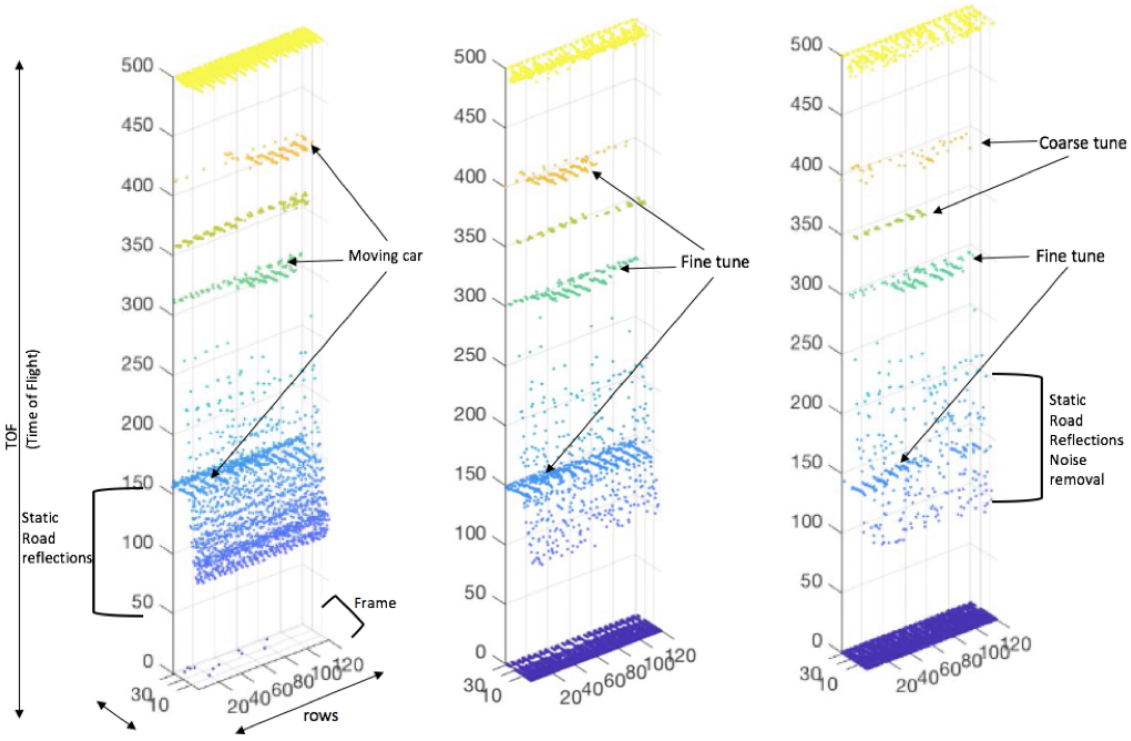


Figure 6.11: Visualization from SNN decoder output Scenario: LIDAR mounted on a moving car, first picture shows the visualization of the scene in 3d point cloud environment. second picture shows the filtered output based constant sensitivity, third picture shows the filtered output based variable sensitivity

2. Power analysis
3. Data quality analysis
4. System robustness

6.3.2.1 Throughput reduction

The table 6.1 shown below describes the reduction in the net throughput in each of the experiments when passed through the proposed neuromorphic retina. The results show that more than 50 - 60 % of the redundant data is removed from the LIDAR source when $Vth = 120mV$ while more than 60 - 70 % of data is reduced when $Vth = 200mV$. As described in previous sections, as Vth increases the sensitivity also increases. This results in coarse tuning of moving objects at far distances. Therefore, the throughput reduction is even more than low Vth based simulations.

6.3.2.2 Power reduction

In order to quantise the effect of reduction in throughput on original LiDAR data flow power consumption, energy consumed per bit of DRAM memory access was taken into account. in a data processing environment majority of energy is consumed in memory

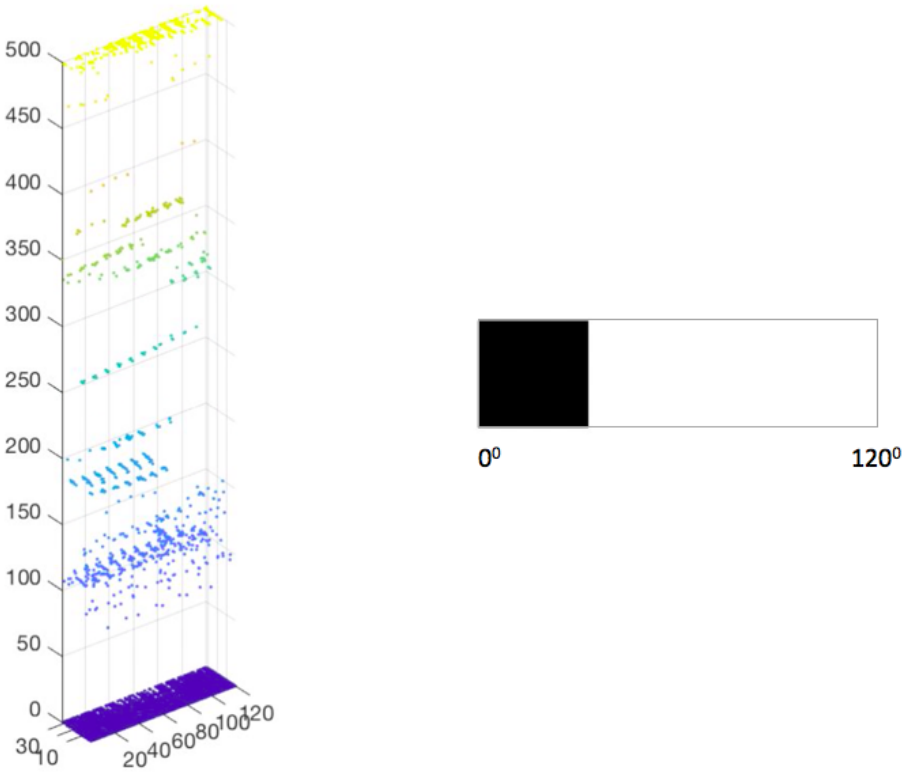


Figure 6.12: Directional selectivity Demonstration

Experiment	Data reduction (%) $Vth = 120mV$	Data reduction(%) $Vth = 200mV$
Exp Car moving 1	55	70
Exp Car moving 2	52	73
Exp Car moving 3	54	71
Exp Car moving 4	55	73
Exp Car moving 5	52	70
Exp pedestrian moving 1	59	63
Exp pedestrian moving 2	60	62
Exp pedestrian moving 3	58	65
Exp pedestrian moving 4	57	60
Exp pedestrian moving 5	61	65

Table 6.1: LiDAR Data reduction table

accesses [30]. The Energy consumed per bit per accesss of generic DRAM is around 20pJ [30]. Hence, for a 1D MEMS based LiDAR used in the thesis, power consumed per frame transfer is around $120 * 32 * 10 * 20 = 768000pJ$ or $768nJ$. Thus based on the results from section 6.3.2.1, this system saves around $384nJ$ to $460nJ$ of energy per frame.

Due to the AER system, only those pixels are updated in memory which are labelled with an event flag (ON/OFF signal). Hence, the proposed AER based neuromorphic

system reduces data throughput entering further LiDAR/Fusion signal processing vision units and thereby saving tremendous amount of system memory and power.

6.3.2.3 Data Quality

This matrices used to check how much quality is retained/lost when event based filtering operation is performed. This means that how much important information is also lost wile the filtering operation. To analyze this matrix, we use the motion based DSP scene segmentation technique described in previous chapter. This operations provides a matrix containing all the pixels representing the moving objects in the scene. Hence, the output from retina is compared with the moving objects DSP matrix so as to compare how much quality data has been lost. Table 6.2 describes the results.

Results show that with lower V_{th} the quality loss is less than the quality loss at higher V_{th}. When the sensor is in dynamic environment based on the collected data sets then the system loses about 10 - 25 % quality based on the sensitivity while it loses 5-7 % quality when movement is captured in a static environment.

Spatial Quality matrix describes the quality per distance. every data point is assigned a quality factor based on its distance from the sensor i.e. its TOF value. This means that points points which are closer to the sensor are more important than those which are far from it. The factors per TOF range are mentioned below:-

1. (0-100) - 20
2. 100-200 - 8
3. 200-300 - 4
4. 300-400 - 2
5. 400-500 - 1

Hence, Based on the observations from Spatial Quality matrix table 6.3 we can conclude that with $Vth = 200mV$ we lose Quality of around 25 - 27 % while Spatial Quality of around 20 - 23 %.

6.4 Robustness

This section throws light on the effect of variations on 4 major input parameters (described below) on the system's functionality and how robust is the system to those variations.

6.4.1 Input LIDAR data frequency

Input data frequency of LIDAR (Computational bound): The digital circuit provides no major latency bottleneck to the input LIDAR stream that varies from 20 to 200 Hz. While the SNN system is bounded by the axon delay limits (which has to be matched

Experiment	Quality reduction (%) $Vth = 120mV$	Quality reduction(%) $Vth = 200mV$
Exp Car moving 1	10.45	25.30
Exp Car moving 2	11.23	26.43
Exp Car moving 3	10.50	25.54
Exp Car moving 4	12.30	27.35
Exp Car moving 5	10.56	25
Exp pedestrian moving 1	5.30	6.63
Exp pedestrian moving 2	5.45	7.34
Exp pedestrian moving 3	4.54	7.54
Exp pedestrian moving 4	5.43	6.65
Exp pedestrian moving 5	4.65	6.45

Table 6.2: LiDAR Data Quality table

Experiment	Quality reduction (%) $Vth = 120mV$	Quality reduction(%) $Vth = 200mV$
Exp Car moving 1	10.14	20.87
Exp Car moving 2	11.15	22.34
Exp Car moving 3	10.24	21.67
Exp Car moving 4	12.12	22.56
Exp Car moving 5	10.17	20.67
Exp pedestrian moving 1	5.13	5.95
Exp pedestrian moving 2	5.15	6.14
Exp pedestrian moving 3	4	6.23
Exp pedestrian moving 4	5.30	5.35
Exp pedestrian moving 5	4.13	5.65

Table 6.3: LiDAR Data Spatial Quality table

with the Frame rate). The adapted axon delay circuit 4.1.3 can provide a delay from 1ms to 50 ms which can clearly accommodate maximum LIDAR frame time period of 5 ms. Hence, the system works at all the input LIDAR data frequencies.

6.4.2 Time Constant τ variations

.

Experiments were conducted on all the available dataset keeping V_{Th} constant at 200 mV and varying the value of τ linearly across the neural distribution as shown in figure 6.13. The experiments allowed variation of +-20% of variation in τ values. The results shown in table 6.2 suggests that small changes in τ doesn't affect the overall system output. Hence, the system is robust against small τ variations

6.4.3 Threshold Voltage V_{Th} variations

. Experiments were conducted on all the available data set keeping τ constant at 25 and varying the value of V_{Th} linearly across the neural distribution as shown in figure

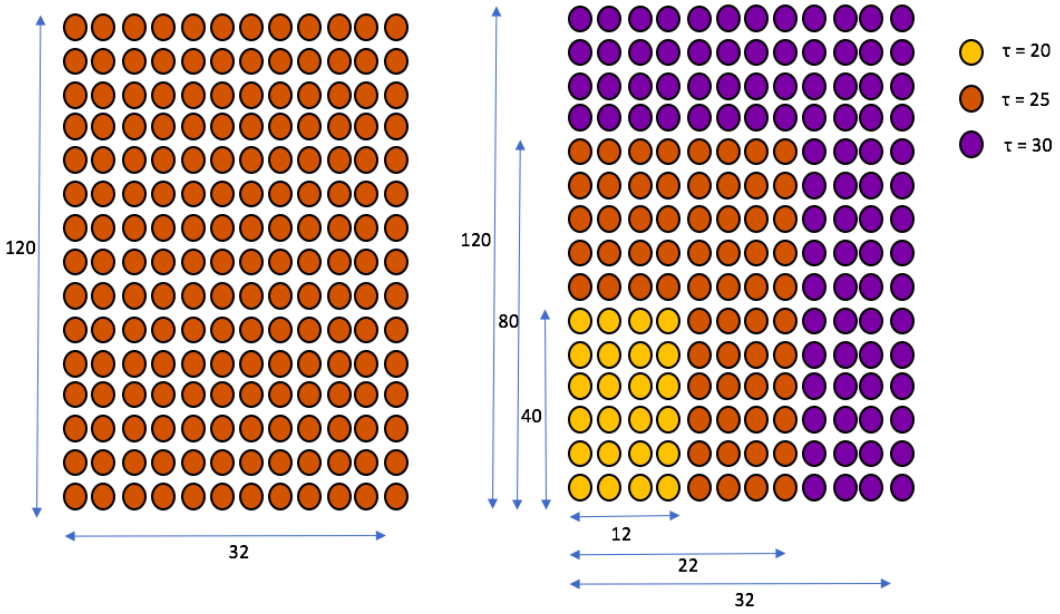


Figure 6.13: Variation of τ throughout the neural hardware space - LiDAR frame -120x32

Experiment	change in Data reduction (%)
Exp Car moving	0.005
Exp Car moving	0.005
Exp Car moving	0.004
Exp Car moving	0.005
Exp Car moving	0.005
Exp pedestrian moving 1	no significant change
Exp pedestrian moving 2	no significant change
Exp pedestrian moving 3	no significant change
Exp pedestrian moving 4	no significant change
Exp pedestrian moving 5	no significant change

Table 6.4: change in data reduction of spatially variable τ w.r.t constant τ at 25

6.14. The experiments allowed variation of +-20% of variation in V_{Th} values. The results shown in table 6.3 suggests that small changes in V_{Th} doesn't affect the overall system output. Hence, the system is robust against small V_{Th} variations

6.4.4 Axon delay (section: 4.1.3)

Axon delay acts as temporary spike memory. It is responsible for concurrent processing of spikes from consecutive frames. This enables the system to detect changes in current and the previous inputs. If these spikes are shifted then then the current produced due to these changes will change accordingly and hence will affect the output spike rate.

Benefit of Spike encoding is that the output is affected by an average rate rather

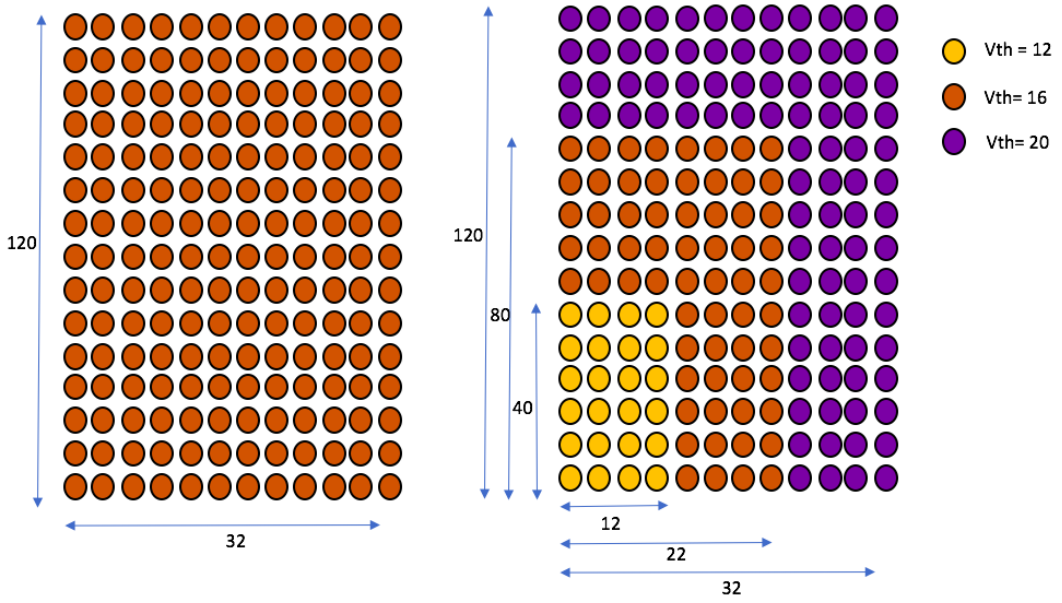


Figure 6.14: Variation of V_{Th} throughout the neural hardware space - LiDAR frame 120x32 pixels

Experiment	change in Data reduction (%)
Exp Car moving	0.012
Exp Car moving	0.020
Exp Car moving	0.011
Exp Car moving	0.017
Exp Car moving	0.014
Exp pedestrian moving 1	no significant change
Exp pedestrian moving 2	no significant change
Exp pedestrian moving 3	no significant change
Exp pedestrian moving 4	no significant change
Exp pedestrian moving 5	no significant change

Table 6.5: change in data reduction of spatially variable V_{Th} w.r.t constant V_{Th} at 160

than individual spikes (as mentioned in the equation 5.3). Hence, +-(10 to 20%) phase shift doesn't have drastic affect on the pre-synaptic current. Therefore the system output is not much affected due small phase shifts in delayed input spike trains.

6.4.5 Conclusion

In conclusion, the system is robust to small variations in major input parameters namely V_{th}, τ, LIDAR input data frequency and axon delay

Part III
Conclusion and future work

7 Conclusion and Future work

7.1 Conclusion

The proposed Neuromorphic Retina for LIDAR is able to perform three major functions:-

1. Detect event in the scene.
2. Detect velocity of the moving objects in the scene.
3. Directional selectivity

Hence, this unique pre-processing unit is able to extract and encode movement happening at particular distance, particular angle and with particular velocity into unique spike so that the information about the dynamic environment can be efficiently classified and processed by an event based power efficient neuromorphic processing unit.

Retina is able to filter out redundant static information from the LIDAR data stream thereby reducing data throughput as well as remove noise caused due to luminous reflections. The results show that the retina is able to reduce around 50 - 70 % of data with 5 - 22 % spatial quality loss (based on scenario). This has tremendous impact on system latency and power consumption due to drop in memory accesses. The system is robust to small variations in major input parameters which increases the overall system reliability.

7.2 Future work

There are two major areas where more work is needed:-

1. Reduction in number of spiking neurons
2. SNN based classification and learning

7.2.1 Reduction in number of spiking neurons by averaging nearby pixels

Shown below (fig: 7.1) is an architecture that was designed in order to average (spike rate) the incoming spike train from a window of 4 nearby pixels. AS shown in the figure: below, the system generates output spike train with a frequency which is almost average of the four input spike train frequencies

Due to this architecture, the neural requirements drops by 4 times but the major problem is the dynamic input range. The proposed system is able to take input TOF values from 0 to 100 which is quit less. Hence, a better architecture has to be designed in order to increase the input dynamic range.

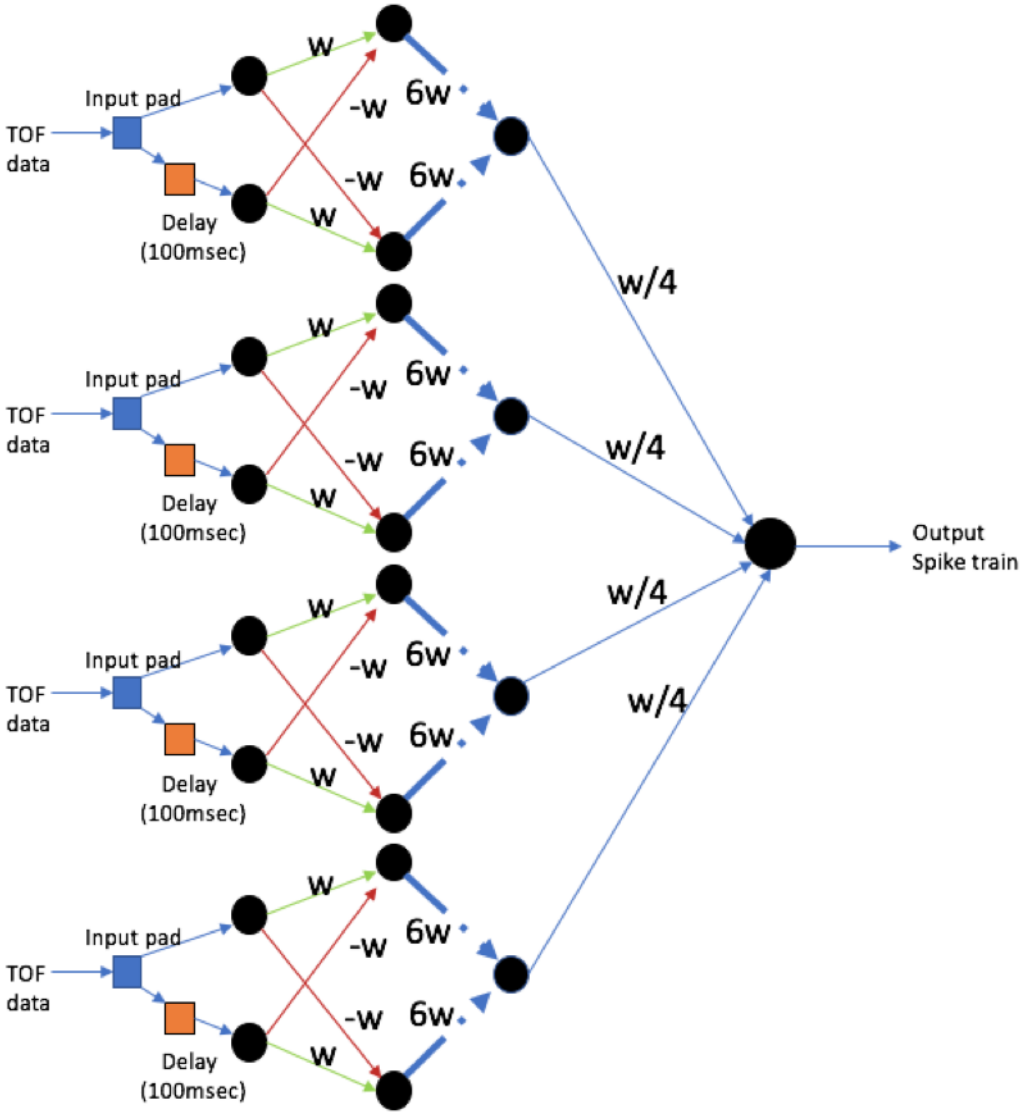


Figure 7.1: Neural network model for averaging 4 neighboring pixels

7.2.2 Classifications and optical flow predictions

Various learning methods are available for classifying real-time spatio-temporal patterns. Hence, LIDAR data by nature supports the dynamic movement information in the scene. The proposed SNN model provide both the velocity as well as distance encoded information in terms of spikes which can be further utilized for classifying and understanding the dynamics of the scene as shown in the figure: 7.4

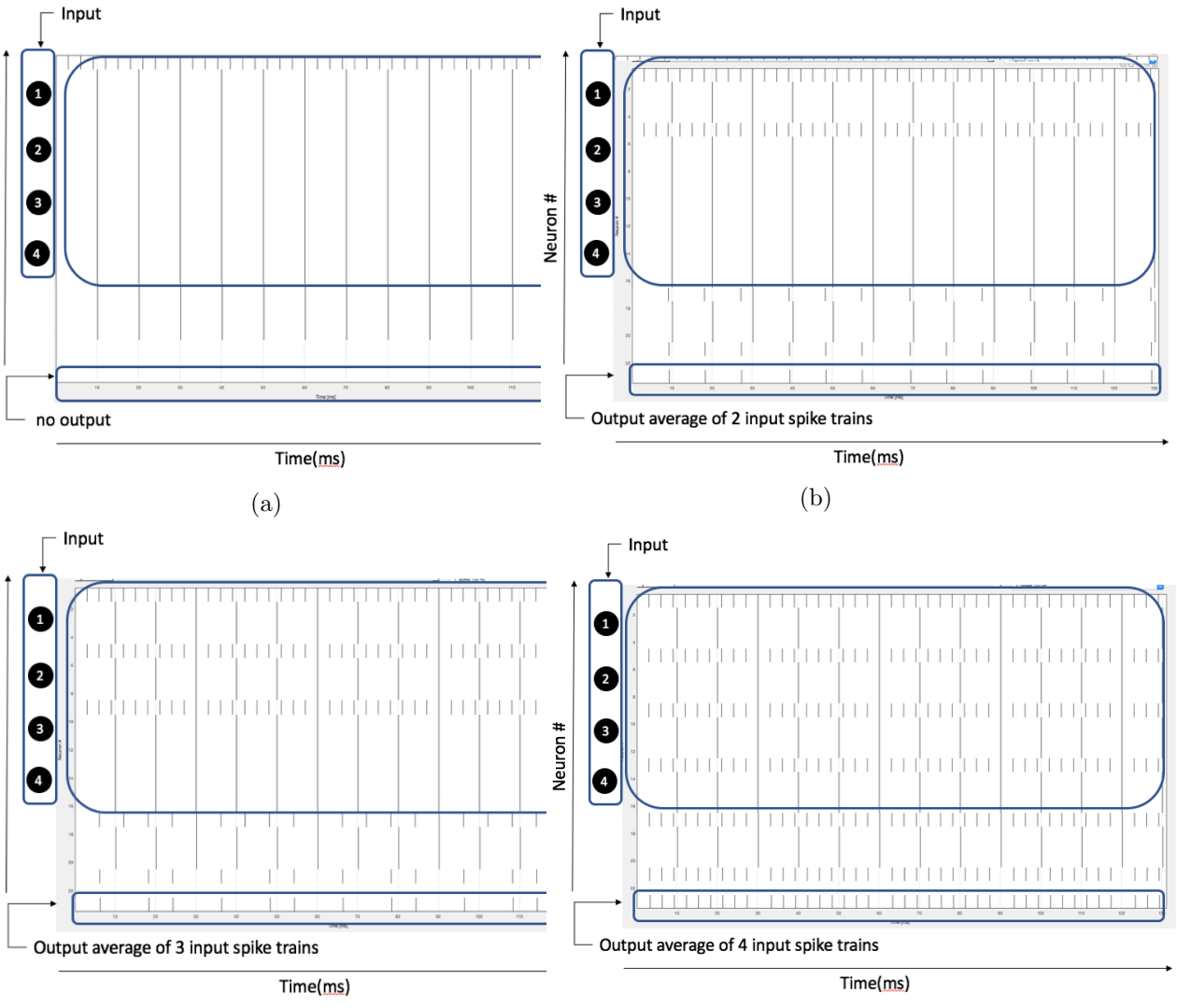


Figure 7.2: output waveforms when 1 input neuron is active (a), when two input neurons are active (b), when 3 are active (c) and when all the 4 are active (d)

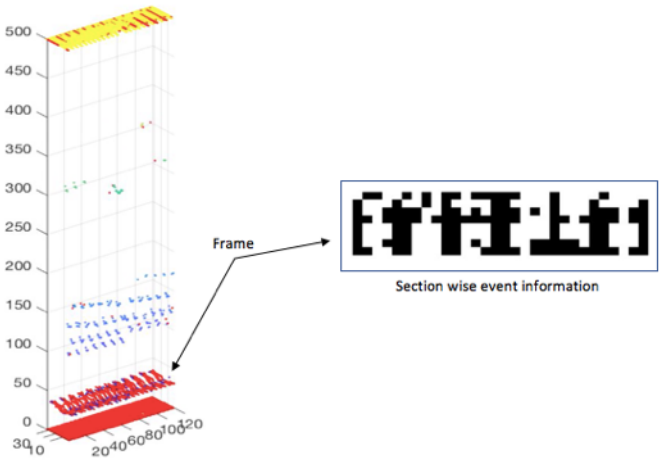


Figure 7.3: Sectionwise event information

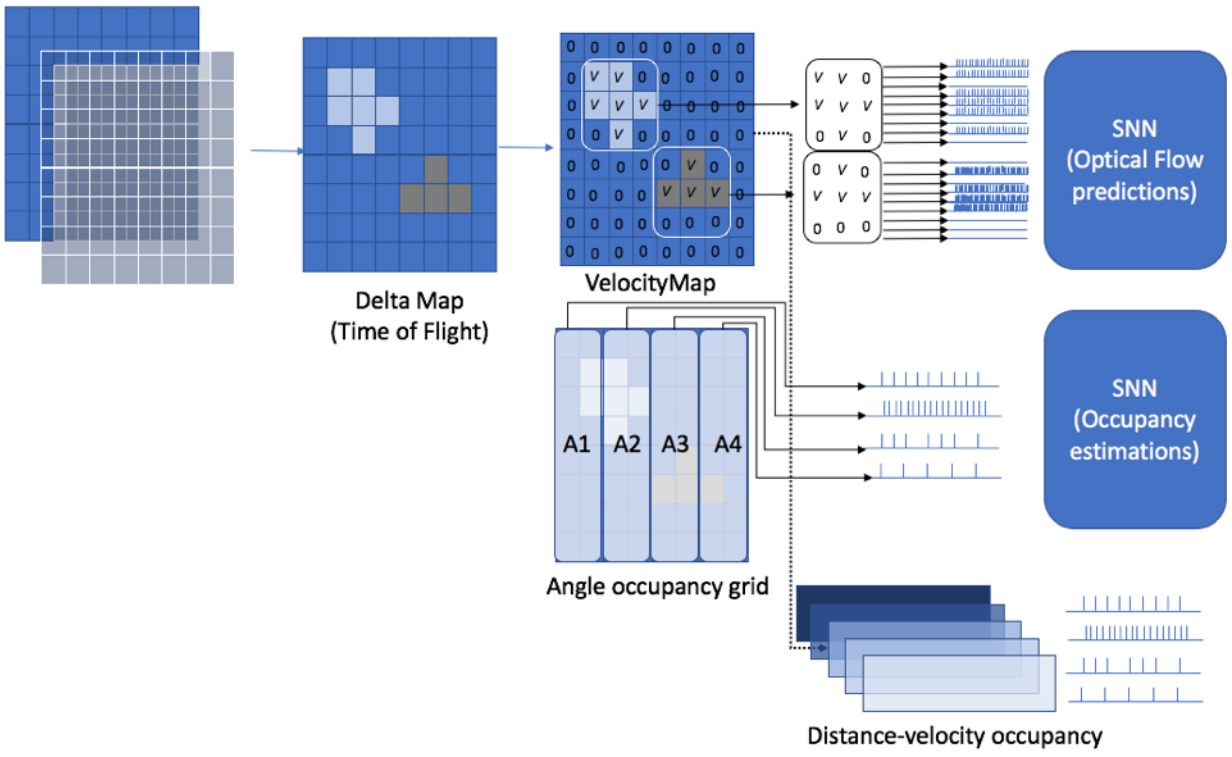


Figure 7.4: SNN based model planning to better classify the dynamics of the scene for autonomous vehicles

8 Appendix

8.1 Axon Delay circuit

8.1.1 Ramp generator

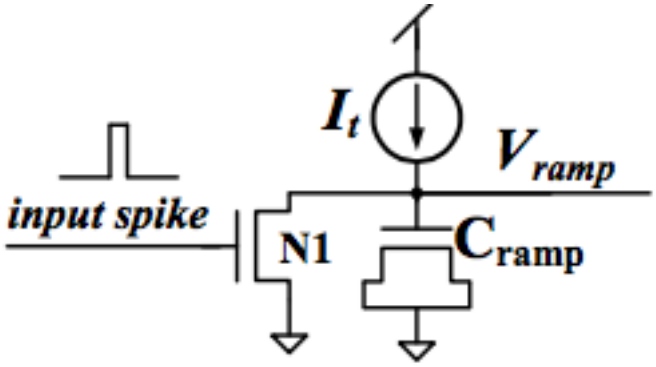


Figure 8.1: Ramp generator [7]

On the activation from input spike, the ramp signal resets ($V_{ramp} = 0$) and starts increasing towards V_{dd}. The instantaneous value of increasing ramp signal is directly proportional to time. Hence, this elapsed time represents delay. In further stages this ramp signal is compared with a programmable voltage in order to spike out a pulse after a fix interval of time.

The typical values used in the circuits as described in [7] are $C_{ramp} = 2pF$ and $I_t = 200pA$. The ramp generator can generate voltage from 100mV to 2.5V which translates to delay of 1ms to 5 ms. Hence, 50mV represents 1 ms delay.

8.1.2 Delay Circuit

The delay circuit consists of two modules. First, Delay storage circuit which is responsible to store the programmable delay voltage value 8.2. Second, spike generator 8.3 that compares the instantaneous ramp signal with the stored programmable delay value so as to generate a spike by enabling a SR latch.

The storage cell consists of a $2pF$ capacitor that is able to store the programmable voltage for about 500 sec without refreshing. The refreshing is managed by a separate adapter circuit if required. This voltage is buffered across an operational transconductance amplifier (OTA). The spike generator has SR latch that is reset when the spike

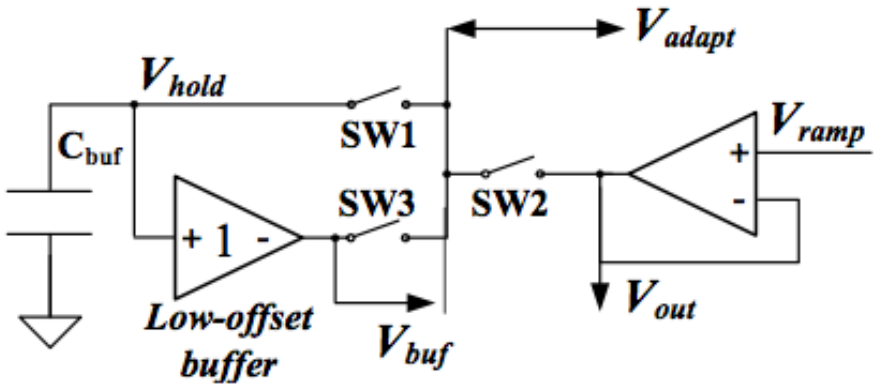


Figure 8.2: Delay Storage circuit [7]

arrives and is set when the ramp voltage exceeds the stored voltage.

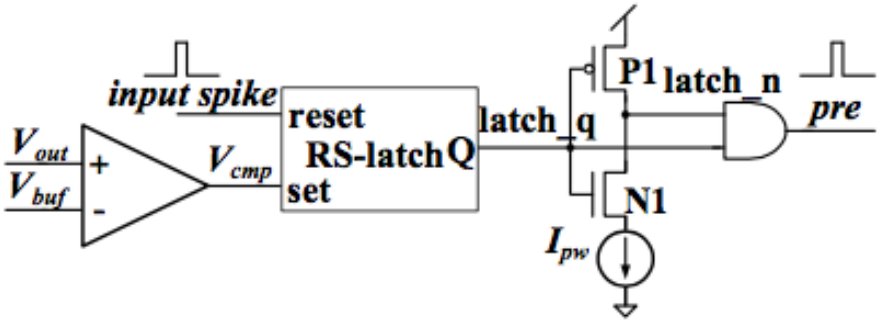


Figure 8.3: Spike Generator [7]

The system uses 3 OTA buffers and RS latch which consumes the highest area, the ramp generator consumes $1600um^2$ of area at 0.6um technology.

8.1.3 Axon delay Resource utilization

In terms of transistor count a module based list has been provided below:-

1. Ramp Generator: 1 current source, 1 capacitor and one NMOS transistor is equivalent to 11 transistors in total [7]

2. comparator: Size of operational transconductance based comparator can vary from 5 transistors [31] to 15 transistors [32].

3. Standard SR latch, Invertor and AND gates are 8, 2 and 4 transistors design respectively.

Hence in terms of transistor count for a single pixel the axon delay requires approximately 30 to 40 transistors which is equal to 115200 to 153600 transistors for the

entire design (120x32 pixels)

While in therms of area consumption we focused towards two biggest components namely ramp generator and the comparator. Based on 0.5 μm technology used in [7], the ramp generator occupies $1600\mu m^2$ area while the comparator occupies $1500\mu m^2$ [33]. Hence, the net area consumption per pixel by the biggest components in axon-delay circuit is around $3100\mu m^2$ which is equivalent to around $12mm^2$ for the entire design.

8.1.3.1

Area comparison with SRAM based DSP implementation

The area consumption of an SRAM cell ($0.5\mu m - technology$) with 12 bits per pixel is around $567\mu m^2/bit * 12 = 6800\mu m^2$ [34]. Hence if we compare the area consumed by an SRAM unit with the area consumed by major components of axon delay cell then we can say that axon delay cell consumes approximately 50% less area than SRAM cell per pixel.

Bibliography

[1] velodyne. velodyne.com.

[2] Norbert Druml, Ievgeniia Maksymova, Thomas Thurner, Diederik Van Lierop, Marcus Hennecke, and Andreas Foroutan. 1d mems micro-scanning lidar. 09 2018.

[3] Md Nasim Khan. *Machine and Deep Learning Techniques for Real-Time In-Vehicle Fog Detection and Speed Behavior Investigation Utilizing the SHRP2 Naturalistic Driving Study Data*. PhD thesis, 05 2018.

[4] Devin Soni. Spiking neural networks, the next generation of machine learning.

[5] Eralp Kolaasiolu. Energy efficient feature extraction for single-lead ecg classification based on spiking neural networks. Master's thesis, TU Delft, 2018.

[6] David Corvoysier. Leaky integrate and fire neuron with tensorflow.

[7] R. Wang, G. Cohen, T. J. Hamilton, J. Tapson, and A. van Schaik. An improved avlsi axon with programmable delay using spike timing dependent delay plasticity. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 1592–1595, May 2013.

[8] Shashanka M R. Study of deep learning based perception for autonomous vehicles.

[9] Yecheng Lyu, Lin Bai, and Xinming Huang. Real-time road segmentation using lidar data processing on an fpga. 11 2017.

[10] Wikipedia. Motion perception.

[11] Stefanie Anna Baby, Bimal Vinod, Chaitanya Chinni, and Kaushik Mitra. Dynamic vision sensors for human activity recognition. pages 316–321, 11 2017.

[12] D. Tpfer, J. Spehr, J. Effertz, and C. Stiller. Efficient road scene understanding for intelligent vehicles using compositional hierarchical models. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):441–451, Feb 2015.

[13] Ievgeniia Maksymova, Christian Steger, and Norbert Druml. Review of lidar sensor data acquisition and compression for automotive applications. 09 2018.

[14] Victor Vaquero, A Sanfeliu, and Francesc Moreno-Noguer. Deep lidar cnn to understand the dynamics of moving vehicles, 08 2018.

[15] Ievgeniia Maksymova, Christian Steger, and Norbert Druml. Review of lidar sensor data acquisition and compression for automotive applications. 09 2018.

[16] Alex Staravoitau. Visualizing lidar data.

[17] MATLAB. matlab optical flow.

[18] MATLAB. matlab sementic segmentation.

[19] Shi Dong and David Kaeli. Dnnmark: A deep neural network benchmark suite for gpus. In *Proceedings of the General Purpose GPUs*, GPGPU-10, pages 63–72, New York, NY, USA, 2017. ACM.

[20] OD Gary Heiting. The retina: Where vision begins.

[21] Stefanie Anna Baby, Bimal Vinod, Chaitanya Chinni, and Kaushik Mitra. Dynamic vision sensors for human activity recognition. *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2017.

[22] Rafael Serrano-Gotarredona, Matthias Oster, Patrick Lichtsteiner, Alejandro Linares-Barranco, Rafael Paz-Vicente, Francisco Gmez-Rodrguez, Haavard Kolle Riis, Tobi Delbruck, Shih-Chii Liu, S Zahnd, Adrian Whatley, Rodney Douglas, Philipp Hfliger, Gabriel Jimenez, Anton Civit, Teresa Serrano-Gotarredona, Antonio Acosta-Jimenez, and Bernab Linares-Barranco. Aer building blocks for multilayer multi-chip neuromorphic vision systems. 01 2005.

[23] Evangelos Stromatias, Miguel Soto, Teresa Serrano-Gotarredona, and Bernab Linares-Barranco. An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Frontiers in Neuroscience*, 11:350, 2017.

[24] Jayram Moorkanikara Nageswaran, Micah Richert, Nikil Dutt, and Jeff Krichmar. Towards reverse engineering the brain: Modeling abstractions and simulation frameworks. pages 1–6, 09 2010.

[25] R. Wang, J. Tapson, T. J. Hamilton, and A. van Schaik. An analogue vlsi implementation of polychromous spiking neural networks. In *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 97–102, Dec 2011.

[26] Mireya Zapata, Jordi Madrenas, Miroslava Zapata, and Jorge Alvarez Tello. *Axonal Delay Controller for Spiking Neural Networks Based on FPGA*, pages 284–292. 06 2019.

[27] Monica Hanslien, Kenneth H. Karlsen, and Aslak Tveito. A maximum principle for an explicit finite difference scheme approximating the hodgkin-huxley model. *BIT Numerical Mathematics*, 45(4):725–741, Dec 2005.

[28] Nikola K. Kasabov. *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*, volume 7 of *1*. Springer-Verlag Berlin Heidelberg, 3 edition, 7 2019.

[29] QingXiang Wu, T. M. McGinnity, Liam Maguire, Jianyong Cai, and G. D. Valderrama-Gonzalez. Motion detection using spiking neural network model. In De-Shuang Huang, Donald C. Wunsch, Daniel S. Levine, and Kang-Hyun Jo, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 76–83, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[30] M. Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14, Feb 2014.

[31] Chetan Thakur, Runchun Wang, Tara Hamilton, Ralph Etienne-Cummings, Jonathan Tapson, and Andr van Schaik. An analogue neuromorphic co-processor that utilizes device mismatch for learning applications. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, PP, 09 2017.

[32] Mehdi Noormohammadi Khiarak, Vahid Khojasteh Lazarjan, and Khosrow HajSadeghi. New operational transconductance amplifiers using current boosting. pages 109–112, 08 2012.

[33] A. Mahdy, R. A. Rassoul, and N. Hamdy. A high - speed analog comparator in 0.5 m cmos technology. In *2008 National Radio Science Conference*, pages 1–7, March 2008.

[34] ON semiconductor. 0.5 m process technology.