

Immersive Visualisation of 3-Dimensional Spiking Neural Networks

Stefan Marks

Received: date / Accepted: date

Abstract Recent development in artificial neural networks has led to an increase in performance, but also in complexity and size. This poses a significant challenge for the exploration and analysis of the spatial structure and temporal behaviour of such networks. Several projects for the 3D visualisation of neural networks exist, but they focus largely on the exploration of the spatial structure alone, and are using standard 2D screens as output and mouse and keyboard as input devices.

In this article, we present *NeuVis*, a framework for an intuitive and immersive 3D visualisation of spiking neural networks in virtual reality, allowing for a larger variety of input and output devices. We apply *NeuVis* to *NeuCube*, a 3-dimensional spiking neural network learning framework, significantly improving the user's abilities to explore, analyse, and also debug the network.

Finally, we discuss further venues of development and alternative render methods that are currently under development and will increase the visual accuracy and realism of the visualisation, as well as further extending its analysis and exploration capabilities.

Keywords spiking neural network · 3-dimensional · visualisation · virtual reality · immersive

1 Introduction

NeuVis, the spiking neural network visualisation tool presented in this article, is designed to work independently of the actual network implementation. Since it was mainly developed and tested in conjunction with *NeuCube*, we will briefly present this specific implementation first. However, we believe that the design implications for visualisations described towards the end of this section apply for any neural network.

Stefan Marks
Auckland University of Technology
55 Wellesley St East
Auckland 1010, New Zealand
Tel.: +64 9 921-9999
E-mail: smarks@aut.ac.nz

NeuCube (Kasabov, 2012) is a specific framework for the learning of spatio- and spectro-temporal data through the use of Spiking Neural Networks. ¹ It is designed for the explicit purpose of incorporating the aspects of spatial, spectral, and temporal dynamics, and their interdependence. In addition, it can also incorporate prior knowledge of the data collection context. *NeuCube* is mainly composed of four main modules: 1) Input Encoding; 2) NeuCube Reservoir; 3) Gene Regulatory Network; 4) Output/Classification; and an optional fifth module, Visualisation, which this article addresses. Figure 1 shows a block diagram of the *NeuCube* spatio-temporal framework. For further details of this architecture we refer to related literature (Kasabov et al, 2016; Scott et al, 2013; Kasabov, 2012).

The primary aspect that needs to be addressed by the visualisation framework presented in this article is that of enabling users to gain an overview and an understanding of a complex, large, and interconnected 3-dimensional structure. At the start of the development of *NeuVis*, *NeuCube* typically consisted of roughly 1500 neurons and about 15000 connections. However, the necessity of processing larger spatio-temporal datasets and the recent developments in hardware simulators such as SpiNNaker (Furber et al, 2014) will soon push those numbers into the areas of hundreds of thousands of neurons and millions of connections. This requires that one of the main design pillars of the visualisation is scalability and throughput.

2 Related Work

In the past, numerous visualisation tools dedicated to the rendering and analysis of neural network structures have been developed and presented, going back as early as 1990 (Wejchert and Tesauro, 1990) for 2-dimensional networks, followed by tools for

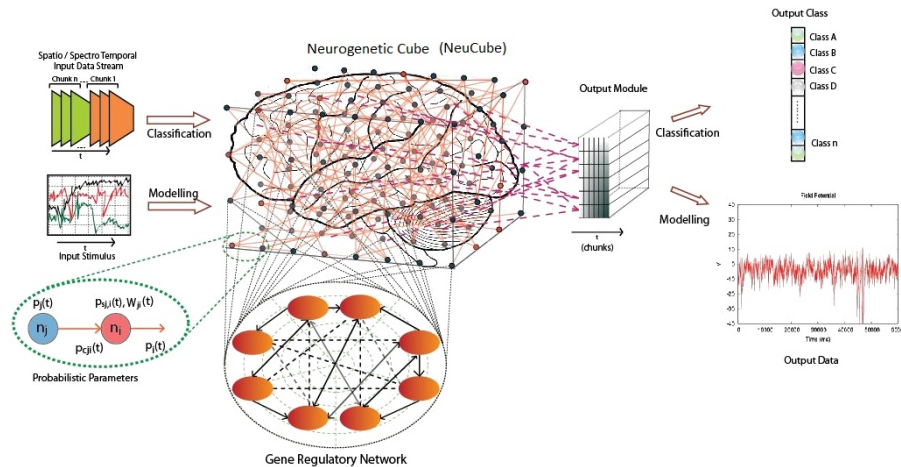


Fig. 1 Overview of *NeuCube*. From (Kasabov et al, 2016).

¹Available from the Knowledge Engineering and Discovery Research Institute (KEDRI), Auckland University of Technology, <http://www.kedri.aut.ac.nz/neucube>

the 3-dimensional domain a few years later. The latter can roughly be categorised either as tools with a focus on analysis, or as tools with a focus on 3D spatial exploration.

BrainGazer (Bruckner et al, 2009), BrainBrowser (Sherif et al, 2015), BrainNet (Xia et al, 2013) and Neuron Navigator (NNG) (Lin et al, 2011) are examples of the category of tools for analysis. They are designed for and used in context with models of realistic biological neural networks such as the human connectome or the brain of the fruitfly *Drosophila melanogaster* (Armstrong and van Hemert, 2009). These tools focus on accurate spatial rendering of pathways and implement interaction metaphors for selecting, tracing, and querying labelled neuron clusters. Photorealistic rendering is not the main objective. Colour, transparency, and labels make up the main visual language. Due to the output medium being a 2D screen, movement and interaction metaphors for rotating and moving the 3D virtual volume are important for perception of depth and the spatial structure. Those interactions are usually initiated and controlled by mouse movement and keyboard controls, so they are indirect and have the potential of being less intuitive than natural 3D object interaction.

By using different output and input mechanisms, it is possible to make the interaction more intuitive. von Kapri et al (2011) are using a Computer Assisted Virtual Environment (CAVE) to visualise the spatial structure and activity of a spiking neural network. Stereoscopic rendering and tracking of the user's eye position allows for an easy perception of the spatial structure and intuitive control of the viewpoint. However, due to the limited space within a cave environment, navigation within the workspace restricted by the dimensions of the CAVE. Furthermore, this specific visualisation framework does not provide deeper means of analysis and interaction other than viewing and exploring the positions and activity of spiking neurons, therefore falling into the second category of tools.

Similarly, Ridder et al (2015) describe a tool to convert output from a 2D based analysis tool into a 3-dimensional virtual reality environment, but interaction is merely limited to viewing from different perspectives with different representations of the dataset spatially adjacent to each other. However, there is no advanced interaction possible with the data (yet).

In the following sections, we describe the requirement analysis, implementation, and evaluation of a tool that combines the 3-dimensional rendering of complex neural network structures with a 3D virtual environment that allows for an easy navigation and interaction for the purpose of exploration *and* analysis of the network.

3 Requirement Analysis

One of the main drivers for the development of *NeuVis* were the disadvantages of the previous visualisation built into *NeuCube* which is using an orthographic perspective and only offers a limited set of interaction and navigation. As apparent in Figure 2, an orthographic perspective is not ideal for the display of a large amount of neurons. Only the points directly at the front of the view volume are visible and, depending on the view angle, can completely occlude the ones behind.

Camera navigation is only limited to a rotation outside of the *NeuCube* volume, providing the user no means of zooming in and looking at details inside. This is also

a limiting factor when it comes to any selection of neurons, which would be possible only from the outer regions of *NeuCube* as well.

Analysis tools have to be invoked using menus and setting dialogues, with a severe delay between activation and an actual visible output on the screen. It is not possible to compare several visualisation settings next to each other, except by taking screenshots.

We interviewed data analysts of KEDRI to compile a list of minimum requirements for *NeuVis*, as well as a “wishlist” of features that would increase their analysis capabilities with respect to structure, feature extraction, temporal behaviour, and clustering analysis.

Selection It should be easily possible to select/de-select individual neurons or groups of neurons.

Temporal Control The user should be able to easily control playback of spiking patterns. This includes control mechanisms such as start, stop, rewind, fast forward, slow down.

Neuron and Connection Usage This analysis functionality is helpful to detect areas of high and low neuron activity and allows for the fine-tuning of *NeuCube*.

Input Cluster Analysis During the learning process, connection patterns form in specific ways around the input neurons. An analysis of those input clusters is an important tool for the understanding of the *NeuCube* learning process.

Connection Length Analysis By analysis of the average connection length between neurons, the learning process of *NeuCube* can be optimised even further.

Feature Cluster Analysis Clusters of neurons encode certain features of the dataset. Visualisation of the spatial structure and connectivity of those clusters is a further helpful tool for the understanding of *NeuCube*.

In addition to this list, the following design pillars were also considered in the design of the *NeuVis* (see Sections 1 and 2 for the rationale).

Scalability The tool needs to be capable of visualising and analysing datasets with millions of neurons and connections.

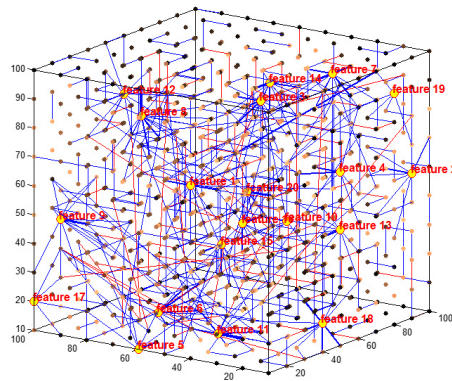


Fig. 2 Visualisation of *NeuCube* using an orthographic view. Note the loss of depth perception and how neurons in the front cover those behind them. This dataset has 1470 neurons, but only a fraction of them is visible due to occlusion. This is detrimental to any selection mechanism for neurons that are located inside the *NeuCube* volume. The display is also cluttered with the large number of input neuron labels (red)

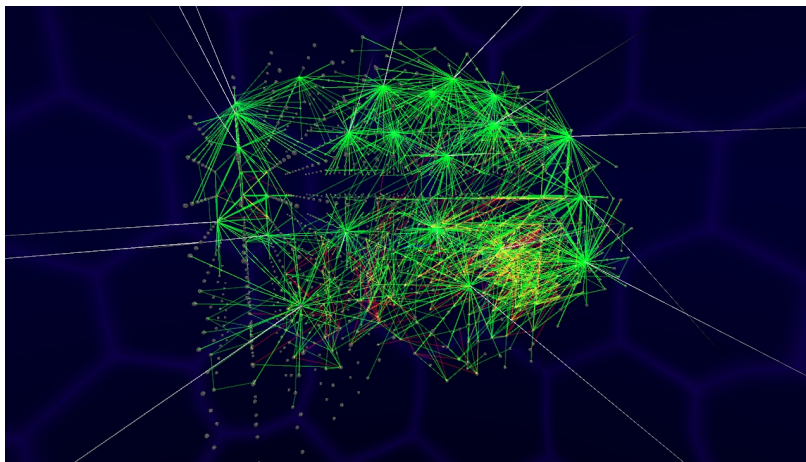


Fig. 3 Visualisation of a *NeuCube* dataset. Neurons are shown as stylised spheres. Connections are rendered green for excitatory and red for inhibitory connections. Thresholding is applied to only show the strongest connections.

Intuitiveness The tool needs to be intuitive and easy to use, preferably with interaction metaphors that do not require any lengthy explanation. This also includes navigation, which should be as direct as possible.

4 Implementation

4.1 Rendering

We created a specialised neural network renderer using OpenGL as a graphics library in conjunction with GLSL (OpenGL Shading Language) shaders in order to be able to render several million neurons together with their connections at a stable framerate of 60 fps (see Figure 3).

In this visualisation, neurons are displayed as stylised spheres, and connections are rendered as lines with different colour for excitatory and inhibitory connections and with different opacities indicating the strength of the connection. Spiking activity is shown as signals travelling along the connections, and neurons are changing size and colour based on their potential. The visualisation is purposefully designed as an analysis tool, therefore not focusing on a realistic depiction of biological neurons and their axons.

Data transfer between the CPU and the graphics card is optimized to a bare minimum for the visualisation. Only the neuron states and positions are transferred to the Graphics Processing Unit (GPU). Vertex Shaders pre-process the data considering the camera position and the neuron activity. Geometry Shaders convert this pre-processed information to simplified “billboard” coordinates which are then sent to Fragment Shaders that construct the final image, applying lighting and shape to the neurons and connections.

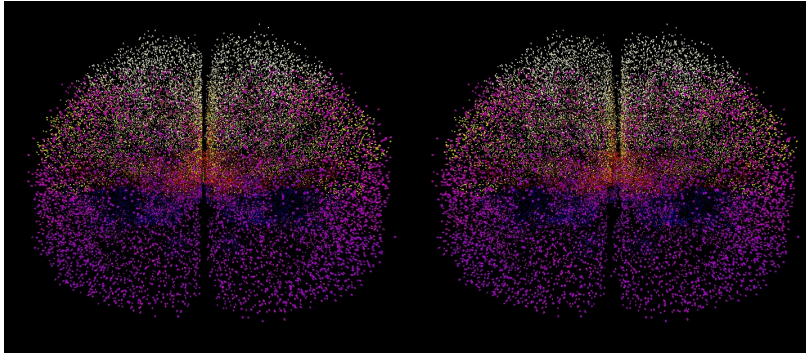


Fig. 4 Stereoscopic rendering of the 1.5 Million neurons of the Talairach Atlas (Lancaster et al, 1997, 2000), used as a benchmark for the performance of the rendering framework.

Through this technique of “billboard rendering” (Maciel and Shirley, 1995), the visualisation can display several million neurons in an interactive framerate of 60 fps, even in a stereoscopic rendering context such as when using a head mounted display (HMD) like the Oculus Rift (Oculus VR, 2016) Figure 4 shows a visualisation of the Talairach Atlas dataset which contains 1.5 million neurons (Lancaster et al, 1997, 2000).

4.2 Interaction

NeuVis is designed to be used in a variety of settings, e.g., on a desktop computer with 2D screen and mouse, on a computer with a HMD and a wireless game controller, or in a motion capture studio with gesture or tracked controller input (see next Section).

We therefore implemented an interaction abstraction layer, that separates the *Neu-Cube* simulation and the computer graphics component from the functionality necessary to control and invoke the visualisation. By doing so, a neuron can be selected by, e.g., directly clicking it with the mouse cursor, by looking at it with the HMD



Fig. 5 User selecting a neuron via a 3D tracked joystick (left image) and a glove (centre image) to observe the activity of a specific neuron (right image).

for more than one second, by clicking it with a 3D tracked joystick, or by applying a “grab” gesture using a motion tracked glove (see Figure 5) (Footitt et al, 2014).

4.3 Analysis

Based on the feature list mentioned in Section 3, the following analysis modes were implemented in *NeuVis*:

Connection Weight Threshold This mode allows to show and hide connections based on a lower and upper threshold weight (see Figure 6).

Connection Length Threshold This mode allows to view the distribution and density of long, medium, or short connections (see Figure 7).

Connection Usage Threshold This mode allows to visualise connections by the amount of spikes they have transmitted. By filtering with a low threshold, it is possible to detect the density and distribution of less important connections (see Figure 8).



Fig. 6 Screenshots of *NeuVis* with the connection weight threshold set to show all connections (left) or only connections with a high weight (right), revealing connectivity clusters.

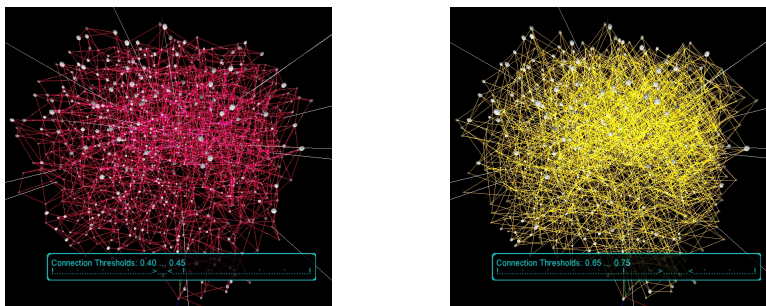


Fig. 7 Screenshots of *NeuVis* with the length weight threshold set to show short connections (left) or longer connections (right), allowing an overview of their distribution and density.

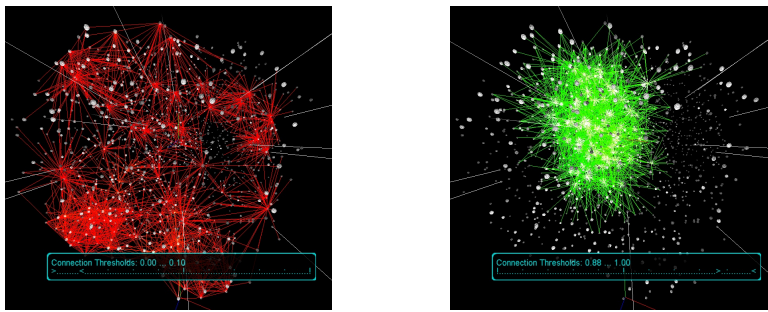


Fig. 8 Screenshots of *NeuVis* with the connection usage threshold set to show rarely used connections (left) or frequently used connections (right), allowing an overview of their distribution and density.

4.4 Sentience Lab Framework

Similar to the work of von Kapri et al (2011), we believe in the potential of being surrounded by the data you are about to analyse and to intuitively move among it. In contrast to the CAVE used in their work, we utilise a setup with a larger area for movement and head mounted displays as output devices.

Our immersive virtual reality research facility, *Sentience Lab*, comprises of the following major components: 1) a wide area motion capture suite, 2) wireless and head mounted displays, and 3) several types of render engines (see Figure 9).

The motion capture system uses an infrared optical marker based solution. It is currently installed in a dedicated room that allows for a capture volume of $6\text{ m} \times 6\text{ m} \times 6\text{ m}$, but can be configured differently to cover an even larger volume. A total of 24 cameras enable a positional accuracy of up to 0.1 mm at 120 fps. The cameras are connected to a computer with software that captures and processes the location of markers in the space and can reconstruct the position and orientation of sets of markers in space, e.g., markers attached to a HMD (see Figure 10). This information stream is then broadcast into the network by a specialised motion capture server and can be received via wired or wireless connections. For maximum speed and minimum latency, we employ a Gigabit wired network and an 802.11ac wireless access point.

Computers with render engines use the information provided in this data stream to render scenes based on the positions and orientations of the headsets and other objects like handheld pointers or joysticks. Wired or wireless HMDs then display the video stream to the user on a stereoscopic display. Due to the positional and rotational tracking, of the HMD, the user can simply change the view of the virtual scene by moving the head or walking around.

Rendering of the visualisation can be implemented in three ways. The first option is a small but portable render machine, e.g., a laptop or a small computer that receives the motion capture data wirelessly and renders directly to the HMD (see Figure 9, lower part). The advantage of this solution is a fast response of the display to user movement, which is vital to avoid motion sickness (see Section 5.2). The disadvantage is the weight and power consumption of such a render unit, requiring a compromise between render quality, runtime, and wearability.

The second option is a dedicated render machine which is connected to the wired network and transmits the video signal to a tethered HMD. The advantage of this solution is that the render machine itself can be equipped with a high-performance graphics card which enables high and stable framerates and therefore achieves a better visual quality. The disadvantage is the limited range and potential for accidents caused by the cable.

The third option is to replace the tether of the second option by a wireless signal transmission system (see Figure 9, centre part).

The wireless HDMI transmitter we employed to implement the third variant can transmit a standard 1080p video signal with up to 75 Hz refresh rate and a maximum of 2 ms signal delay. The headset is a standard Oculus Rift DK1 and its 1280×800 pixel resolution and 60 Hz framerate, connected to the HDMI receiver and a battery in a small pouch that can be attached to a belt (see Figure 11).

We attached 3D-printed custom marker mounts to the headset to enable full positional and rotational tracking (see Figure 10). Six markers are more than sufficient to prevent loss of tracking due to occluded markers. By being able to configure the location of some markers, it is possible to have several independently tracked headsets in the motion capture space at the same time, allowing for collaboration between several users, either in the same physical space, or even remotely in different physical facilities.

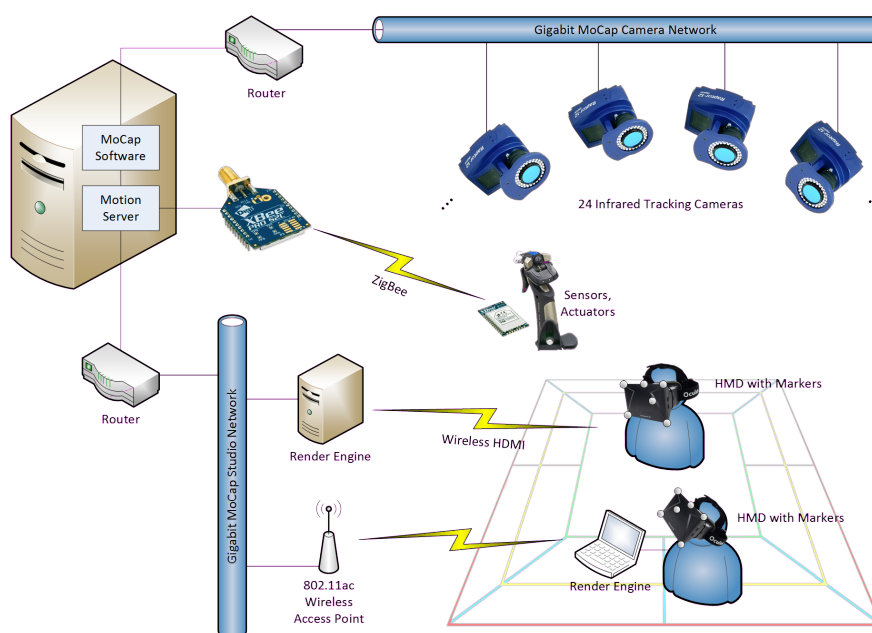


Fig. 9 Diagram of the *Sentience Lab* framework used to enable fully immersive movement and interaction within a virtual environment for the visualisation and analysis of *NeuCube*. 24 cameras allow for sub-millimetre precision tracking of several users within the same physical space. Wireless data transmission enables untethered operation of render engines and head mounted displays. Wireless sensors integrated in tracked controllers, e.g., joysticks, allow for intuitive control and interaction within the virtual environment.

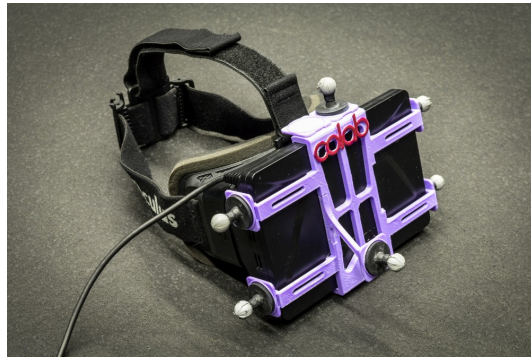


Fig. 10 Head mounted display with markers attached for positional and rotational tracking.



Fig. 11 Left: Components of the wireless HDMI transmitter system used for un-tethering users within the motion capture space (from left to right: Battery, transmitter, Oculus Rift breakout box, HDMI receiver). Right: Belt pouch containing all receiver and power components. The sender unit in front is connected to the stationary render computer.

To allow for intuitive control of the immersive 3D visualisation, we integrated wireless sensor technology into controllers, e.g., joysticks, that allow not only for the tracking of their position and orientation, but also for events such as pressing a button or triggering a vibration motor within the controller.

5 Results

The following section describes the observations of users exploring and analysing *NeuCube* with the help of *NeuVis*. Due to the limited number of *NeuCube* experts within our institution, quantitative studies are currently difficult to conduct. For this article, we rely on observations and qualitative data of around 15 specialists of KEDRI and about 30 non-specialists (visitors to *Sentience Lab*).

5.1 Intuitiveness

All specialist users unanimously reported that using *NeuVis* was 1) a clear improvement to the limited previous 2.5-dimensional version, and 2) that navigation within

NeuCube, selection of neurons, and identifying clusters of connectivity was intuitive and easy to accomplish.

For a desktop setting with a 2D screen, mouse and keyboard, the controls were configured such that the keys W and S would move the virtual camera forwards and backwards, while A and D would move it sideways. The cursor keys would tilt the camera around its vertical and horizontal axis. This provides a sort of flight control, specifically tailored for people who have experience with video games where WASD or cursor key control is an accepted standard. We purposefully did not use the mouse for camera control because in a desktop setting, the mouse cursor is more suitable for selection.

In the motion capture setting, the intuitiveness was even more pronounced. After a short period of getting used to the HMD, users freely walked around and explored *NeuCube*, making use of natural movements like looking from underneath or from above, using the joystick controller intuitively and without any disorientation or other difficulties.

5.2 Simulator Sickness

One of the major issues in fully immersive VR using HMDs is that of latency between physical motion of the head and the actual change in the visible image according to that motion. “Cue conflict”, i.e. a difference in motion perceived by the vestibular organ and as seen by the eyes can easily lead to nausea and other negative physiological effects (Kolasinski, 1995). The Best Practices Guide of Oculus VR provides a very useful and exhaustive list of factors that need to be addressed during the design of VR scenarios (Oculus VR, 2015). One of the most crucial recommendations is to keep the latency below 20 ms.

Surprisingly, we found that users are less prone to simulator sickness within the immersive VR space compared to configurations where the user is confined to a stationary location, e.g., an office chair using a desktop computer, and has to move their camera by means of a joystick or mouse/keyboard controls. A high latency seems to be more critical in the latter case compared to when the user is directly in full control of every movement. We currently assume that due to the natural inertia of the human body, movement is less rapid and also less “clean” compared to movement controlled by external input devices. This hypotheses still need to be verified by quantitative and qualitative studies.

However, as noted in the subsection above, we had around 45 people experiencing *NeuVis* in *Sentience Lab* to this date without major motion sickness problems as a result.

5.3 Analysis Capabilities

The capability to repeatedly play back neuron firing patterns and to observe their propagation through the system, to slow down or speed up the playback rate, provided valuable insights into timing related issues and properties of *NeuCube* instances based on EEG or fMRI datasets.

Another outcome of using *NeuVis* was the discovery of minor bugs in some *NeuCube* structures, such as duplicate input neurons, or wrong EEG electrode mappings. These bugs were not apparent beforehand mostly because of the inadequacy of the orthogonal 2.5-dimensional projection (see Figure 2).

6 Conclusion and Future Work

In this article, we have presented *NeuVis*, a generic visualisation and analysis framework for spiking neural networks, and its specific application to *NeuCube*.

NeuVis is designed with flexibility and scalability in mind, allowing for a range of input and output device combinations. Furthermore, it integrates a large variety of analysis tools that are helpful in gaining insights to *NeuCube* that were not possible with the previous visualisation tool.

Its full potential is achieved when using it in a motion capture setup with stereoscopic rendering, improving the spatial perception and making navigation as intuitive as moving the user's own head and body.

While we agree that using a full motion capture system is a very costly investment for a visualisation, the flexible design of *NeuVis* addresses this point by allowing different input and output devices such as mouse, gamepad, stereoscopic HMDs, or motion tracking data from different sources, such as an HTC Vive (HTC Corporation, 2016).

In the future, we are going to explore additional output and tracking mechanisms such as the HTC Vive which is much more affordable, covers a bigger capture volume than the Oculus Rift system and is also shipped with motion tracked hand controllers. We will also further explore improved interaction metaphors for the operation of the visualisation, and implement advanced analysis tools for a better understanding of *NeuCube*.

Acknowledgements Thanks to the staff at KEDRI and specifically Nathan Scott and Prof. Nikola Kasabov for his support in the development of this visualisation. Also thanks to Javier Estevez, MoCap technician at AUT, for his support in developing the *Sentience Lab* framework.

References

- Armstrong JD, van Hemert JI (2009) Towards a virtual fly brain. *Philos Transact A Math Phys Eng Sci* 367:2387–2397
- Bruckner S, Šoltészová V, Gröller ME, Hladůvka J, Bühler K, Yu JY, Dickson BJ (2009) BrainGazer - visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics* 15(6):1497–1504, DOI 10.1109/TVCG.2009.121
- Footitt J, Brown D, Marks S, Connor A (2014) An Intuitive Tangible Game Controller. In: Nesbitt K, Blackmore K (eds) *The 10th Australasian Conference on Interactive Entertainment (IE 2014)*, Newcastle, NSW, Australia, DOI 10.1145/2677758.2677774

- Furber S, Galluppi F, Temple S, Plana L (2014) The SpiNNaker Project. *Proceedings of the IEEE* 102(5):652–665, DOI 10.1109/JPROC.2014.2304638
- HTC Corporation (2016) Vive. URL <https://www.vive.com/anz/>
- von Kapri A, Rick T, Potjans TC, Diesmann M, Kuhlen T (2011) Towards the visualization of spiking neurons in virtual reality. *Studies in Health Technology and Informatics* 163:685–687
- Kasabov N (2012) NeuCube EvoSpike architecture for spatio-temporal modelling and pattern recognition of brain signals. In: *Artificial Neural Networks in Pattern Recognition*, Springer, pp 225–243, DOI 10.1007/978-3-642-33212-8_21, URL http://dx.doi.org/10.1007/978-3-642-33212-8_21
- Kasabov N, Scott NM, Tu E, Marks S, Sengupta N, Capecchi E, Othman M, Doborjeh MG, Murli N, Hartono R, Espinosa-Ramos JI, Zhou L, Alvi FB, Wang G, Taylor D, Feigin V, Gulyaev S, Mahmoud M, Hou ZG, Yang J (2016) Evolving spatio-temporal data machines based on the neucube neuromorphic framework: Design methodology and selected applications. *Neural Networks* 78:1–14, DOI <http://dx.doi.org/10.1016/j.neunet.2015.09.011>, URL <http://www.sciencedirect.com/science/article/pii/S0893608015001860>, special Issue on Neural Network Learning in Big Data
- Kolasinski EM (1995) Simulator Sickness in Virtual Environments. Tech. rep., Defense Technical Information Center, URL www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA295861
- Lancaster J, Rainey L, Summerlin J, Freitas C, Fox P, Evans A, Toga A, Mazziotta J (1997) Automated Labeling of the Human Brain. *Human brain mapping* 5(4):238–242, DOI 10.1002/(SICI)1097-0193(1997)5:4<238::AID-HBM6>3.0.CO;2-4, URL <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2860189/>
- Lancaster JL, Woldorff MG, Parsons LM, Liotti M, Freitas CS, Rainey L, Kochunov PV, Nickerson D, Mikiten SA, Fox PT (2000) Automated Talairach atlas labels for functional brain mapping. *Human Brain Mapping* 10(3):120–131
- Lin CY, Tsai KL, Wang SC, Hsieh CH, Chang HM, Ann-Shyn Chiang (2011) The neuron navigator: Exploring the information pathway through the neural maze. In: 2011 IEEE Pacific Visualization Symposium (PacificVis), pp 35–42, DOI 10.1109/PACIFICVIS.2011.5742370
- Maciel PWC, Shirley P (1995) Visual Navigation of Large Environments Using Textured Clusters. In: *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, ACM, New York, NY, USA, I3D '95, pp 95–ff., DOI 10.1145/199404.199420, URL <http://doi.acm.org/10.1145/199404.199420>
- Oculus VR (2015) Best Practices Guide. Tech. rep., Oculus VR, URL <http://developer.oculus.com/best-practices>
- Oculus VR (2016) Oculus Rift. URL <https://www3.oculus.com/en-us/rift/>
- Ridder Md, Jung Y, Huang R, Kim J, Feng DD (2015) Exploration of Virtual and Augmented Reality for Visual Analytics and 3d Volume Rendering of Functional Magnetic Resonance Imaging (fMRI) Data. In: *Big Data Visual Analytics (BDVA)*, 2015, pp 1–8, DOI 10.1109/BDVA.2015.7314293

- Scott N, Kasabov N, Indiveri G (2013) NeuCube neuromorphic framework for spatio-temporal brain data and its python implementation. In: *Neural Information Processing*, Springer, pp 78–84, DOI 10.1007/978-3-642-42051-1_11
- Sherif T, Kassis N, Rousseau ME, Adalat R, Evans AC (2015) BrainBrowser: distributed, web-based neurological data visualization. *Frontiers in Neuroinformatics* 8:89, DOI 10.3389/fninf.2014.00089, URL <http://journal.frontiersin.org/article/10.3389/fninf.2014.00089/full>
- Wejchert J, Tesauro G (1990) Neural network visualization. In: Touretzky DS (ed) *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 465–472, URL <http://dl.acm.org/citation.cfm?id=109230.109287>
- Xia M, Wang J, He Y (2013) BrainNet Viewer: A Network Visualization Tool for Human Brain Connectomics. *PLOS ONE* 8(7):e68,910, DOI 10.1371/journal.pone.0068910, URL <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0068910>