VisNEST - Interactive Analysis of Neural Activity Data

Christian Nowke Bernd Hentschel Torsten Kuhlen
Virtual Reality Group, RWTH Aachen University[*]
JARA — High Performance Computing

Jochen M. Eppler Sacha van Albada Maximilian Schmidt Rembrandt Bakker Markus Diesmann
Institute of Neuroscience and Medicine (INM-6), Computational and Systems Neuroscience, Research Center Jülich[†]
JARA — Translational Brain Medicine

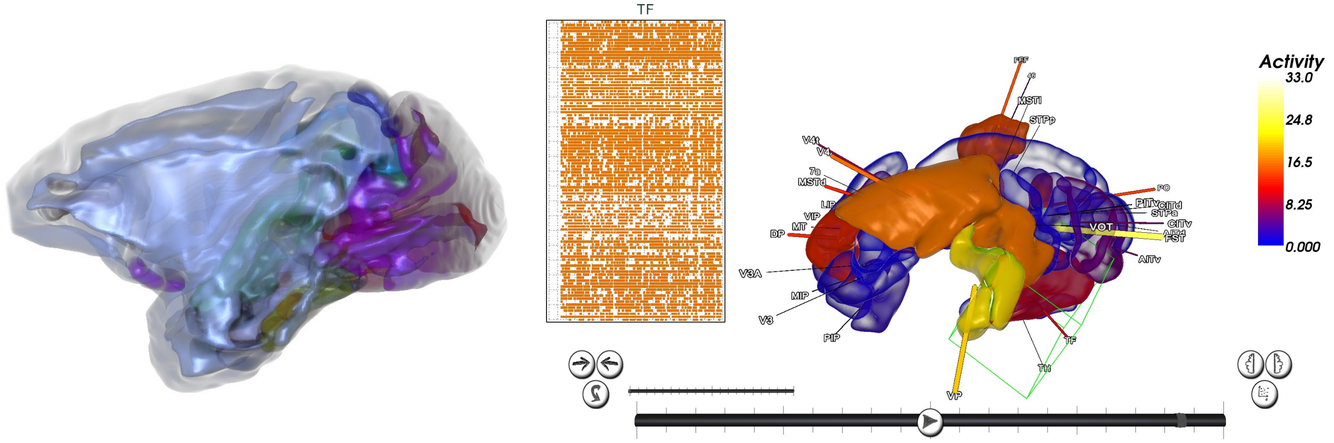

Figure 1: Left: Semi-transparent rendering of Macaque brain areas using order independent transparency. Right: Textual annotations show anatomical area designators. Both, the color of each area and the width of the corresponding annotation's link are used to depict average neural activity. Interactive selection of a brain area (green box) brings up the corresponding dot plot display (top center).

Abstract

Modeling and simulating a brain's connectivity produces an immense amount of data, which has to be analyzed in a timely fashion. Neuroscientists are currently modeling parts of the brain – e.g. the visual cortex – of primates like Macaque monkeys in order to deduce functionality and transfer newly gained insights to the human brain. Current research leverages the power of today's High Performance Computing (HPC) machines in order to simulate low level neural activity. In this paper, we describe an interactive analysis tool that enables neuroscientists to visualize the resulting simulation output. One of the driving challenges behind our development is the integration of macroscopic data, e.g. brain areas, with microscopic simulation results, e.g. spiking behavior of individual neurons.

Index Terms: I.6.5 [Model Development];I.6.6 [Simulation and Modeling]: Simulation Output Analysis—

1 Introduction

Understanding the fundamental structures of the human brain poses several formidable challenges. Among others, the microscopic structure of neural circuits is a main focus of contemporary efforts in neuroscience. To this end, large scale simulations are performed in order to understand the inner workings of our brain. The data underlying this work was generated with NEST, a simulator for networks of point neurons or neurons with few electrical compartments [3]. NEST runs on standard desktop computers, compute clusters, and large-scale HPC facilities, e.g. the IBM BlueGene.

In spite of the multitude of data sets resulting from such simulation studies, little has been done to support neuroscientists with interactive tools for the inspection and analysis of these data. This is a demanding challenge just because of the shear amount of data to browse. Therefore, we are investigating methods to address specific domain questions like: *"Which brain areas show similar activity levels for a given time step in the simulation?"*. While this question could be answered with traditional techniques, arguably a visual representation of activity, along with morphology data, would greatly help to gain an answer to this and similar questions more quickly, thus reducing overall time to insight.

This paper addresses three specific questions that have been established as key requirements by domain scientists. First, users need to be able to explore spatial relationships among simulated brain areas. Second, we need to enable users to link low-level, neural activity to geometrical data. Finally, a successful tool should seamlessly integrate techniques from the domain scientists' daily analysis workflow. In this case, we chose to incorporate dot plots, which show the spiking behavior of a set of neurons over time.

2 Implementation

We implemented a prototypical visualization application that has been used to simultaneously visualize 32 vision related areas of a Macaque brain. Its design was focused on enabling users to explore the spatial relationship among all simulated areas and see the average neural activity of each area while being able to interactively browse the time-dependent simulation results. It provides the ability to freely navigate and rotate around brain areas of interest, to show activity levels of simulated neurons, and to present dot plots of neural activity per area. Users may interactively select areas of

[*] e-mail: {nowke, hentschel}@vr.rwth-aachen.de

[†] e-mail:{j.eppler, s.van.albada, max.schmidt, r.bakker, diesmann}@fz-juelich.de

interest in order to get fine-grained control over the visualization's focus and to help them reduce information overload. The system is based on ViSTA, a platform independent Virtual Reality framework [1], which allows us to seamlessly scale the visualization application from standard desktop workstations to large CAVE-like virtual environments.

2.1 Input Data

The simulated neural model underlying this work mimics the visual cortex of a Macaque monkey. The specific data set used for this paper consists of 32 individual areas with known functional connections to the visual cortex. Connectivity information was taken from anatomical records [5]. For each brain area, the simulation outputs the firing patterns of all neurons in that area over time. In addition to low-level spiking activity, average spike rates for each area are recorded. The geometry was taken from the Scalable Brain Atlas [2]. Each brain area was reconstructed from volumetric data slices to a polygonal mesh, in total containing roughly about 2.4 million triangles.

2.2 Visualization Methods

Since the shape of individual brain areas is a geometrically complex structure, it is of importance to recognize the area's shape and its activity regardless of the current perspective. Furthermore, brain areas occlude each other; thus, it is hard to recognize their spatial relation and shape. To this end, a transparency rendering technique has been deemed useful.

For transparency modulation we chose to focus on highlighting the complex geometrical structure of the brain by using ideas of *angle-based transparency* [4]. Angle-based transparency modulates an α-value based on local view direction and surface normal, which results in surface regions orthogonal to the viewer becoming more transparent, and less transparent where the surface of the region curves away from the view. However, it provides no possibility to control opaqueness dependent on average spiking data. Furthermore, we are interested in the opposite effect, so that regions orthogonal to the viewer become transparent in order to see occluded areas more easily. Therefore, we introduce a new α-modulation method which combines aspects of angle-based transparency and a modulation with the Gompertz function based on activity data:

$$a = (a_{base} + \overline{a}) \; \mathit{and} \; a = 1 \; \mathit{iff} \; (a_{base} + \overline{a}) > 1 \tag{1}$$

$$\alpha_1 = ae^{be^{ca}} \tag{2}$$

$$\alpha_2 = 1 - \|\hat{n} \cdot \hat{v}\| \tag{3}$$

$$\alpha_{final} = \alpha_1 + \alpha_2 \; \mathit{and} \; \alpha_{final} = 1 \; \mathit{iff} \; \alpha_1 + \alpha_2 > 1 \tag{4}$$

where a is the sum of the base transparency a_{base} and the average spiking activity denoted $\overline{a}$, which is normalized to unit interval, for the current simulation time step (Eq. 1). In Eq. 2 the transparency portion for the activity data is calculated. Parameters b and c define an x-axis displacement and a growth rate but are not used to convey data values. Both values are kept constant with $b = -7$ and $c = -5$.

Additionally, the base transparency a_{base} in Eq. 1 can be adjusted interactively, thus, enabling the user to inspect an area even when according to activity data the area should not be visible. Because we render geometry transparently we have to take back-facing polygons into consideration as well, hence the absolute value of the dot product between the surface normal $\hat{n}$, and the view direction $\hat{v}$ is taken to determine the second transparency portion (Eq. 3). At last, both α-values are combined in Eq. 4 and passed to rendering. For visual depth clues and lighting we use phong illumination.

To assure correct alpha blending, we use an order independent transparency method based on dedicated random access memory buffers on the graphics hardware. The order independent transparency algorithm builds a linked list of (r, g, b, a, z)-tuples, where z denotes the local depth value. After the rasterization pass, all fragments are sorted in depth-order and blended appropriately as depicted in Fig. 1 (left) [6].

To visualize average neural activity per area, we apply standard color coding. We determine *min* and *max* activity scalars for the whole simulation data and assign color values through interpolation to its respective activity potential. In contrast to simple activity plots, this depiction puts the activity data in a geometrical context which is important in order to enable the analysis of cross-area interactions and is one of our requirements. For the purpose of relating morphology of brain areas to simulation results, each brain area is depicted by an annotation showing the area's anatomical designator. Because the scenery can be changed by users, we interactively adapt annotation positions [7]. In order to be able to show activity levels even in case of occluded areas, we additionally encode this information via the connecting line between annotation and area: its width is modulated with the local activity level and the color is chosen according to the activity lookup table. One further key requirement is linking macroscopic data, e.g. the average spike rate per neuron, to fine-grained microscopic features, e.g. the spiking behavior of individual neurons. We address this issue by interactively displaying dot plots for selected brain areas. Users are able to select individual areas by directly pointing at them using a 6-degree of freedom input device or mouse. The selected area is highlighted – cf. Fig. 1 (right) – and the corresponding spike trains for all the neurons in that area are displayed by means of a dot plot.

In addition to this selection feature, scientists are able to browse through time via standard video cassette recorder controls and navigate around the data set. This enables them to explore spatial relationships among simulated areas and covers one of the primary requirements stated by the domain scientists.

3 Summary and Future Work

In this paper, we discussed the design of the VisNEST visualization application which helps neuroscientists to interactively analyze large-scale neural activity data. Currently, data exchange is solely file-based. However, in order to further speed up overall turnaround times, we would like to explore the possibility to remotely connect to running NEST simulations. This would allow domain experts to inspect their neural networks and decide whether or not to abort the simulation run due to e.g. bad modeling. Additionally, they could validate the inner workings of their model before simulation concludes.

References

[1] I. Assenmacher and T. Kuhlen. The ViSTA Virtual Reality Toolkit. In *Proceedings of the IEEE VR 2008 Workshop Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, pages 23–28. Shaker Verlag, 2008.

[2] R. Bakker and G. Bezgin. Scalable brain atlas. *http://scalablebrainatlas.incf.org/*, June 2012.

[3] M.-O. Gewaltig and M. Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[4] M. Hummel, C. Garth, B. Hamann, H. Hagen, and K. I. Joy. IRIS: Illustrative Rendering for Integral Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1319–1328, 2010.

[5] R. Kötter. Cocomac (collations of connectivity data on the macaque brain). *http://cocomac.org*, Feb. 2008.

[6] Loren Carpenter. The A-Buffer, An Antialiased Hidden Surface Method. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 103–108, 1984.

[7] S. Pick, B. Hentschel, M. Wolter, I. Tedjo-Palczynski, and T. Kuhlen. Automated Positioning of Annotations in Immersive Virtual Environments. In *Proceedings of the Joint Virtual Reality Conference of EuroVR - EGVE - VEC*, pages 1–8, 2010.