From Big Data to Big Displays
High-Performance Visualization at Blue Brain

Stefan Eilemann, et.al.

Blue Brain Project, Ecole Polytechnique Federale de Lausanne

Abstract. The Blue Brain project has been investing in high-performance visualization (HPV) to complement its HPC strategy since its inception in 2007. In 2011, this strategy has been accelerated to develop innovative visualization solutions through through increased funding and strategic partnerships with other research institutions.
We present the key elements of this HPV ecosystem, which integrates C++ visualization applications with novel collaborative display systems. We motivate how our strategy of transforming high-performance visualization engines into services enables a variety of use cases, not only for the integration with high-fidelity displays, but also to build service oriented architectures, to link into web applications and to provide remote services to python applications.

1 Collaborative Display Systems

Collaborative display systems are the integration point of the Blue Brain visualization infrastructure. They are the evolution of existing visualization systems. Compared to the current single-user or single-presenter systems, collaborative display systems enable real team work through a combination of size, resolution and user friendly implementation. Compared to immersive visualization systems like the CAVE, they provide a simple to use environment for high-fidelity visualization. For all use cases, the increased display size and resolution allows improved data exploration for 2D and 3D content.

1.1 Tiled Multitouch Display Walls

The core of the Blue Brain visualization infrastructure are multiple high-resolution tiled display walls driven by our Tide software [1]. All walls are equipped with a multitouch user interface and can be remote controlled from any web browser. The walls are build using thin-bezel, 55 inch, Full HD LCD panels with a hardened glass sheet. We use 4×3 and 3×3 configurations for a total of 24 and 18 Megapixel resolution, respectively. The display size of over five meter diagonally (four meter for 3×3) allows team-size collaboration (up to ten people) or project-wide presentations (up to a hundred people). Figure 1 shows one wall during a project-wide presentation with multiple interactive applications.

In the last years we created a production-ready setup of monoscopic tiled display walls. On one hand, while there is substantial research on tiled display wall software,

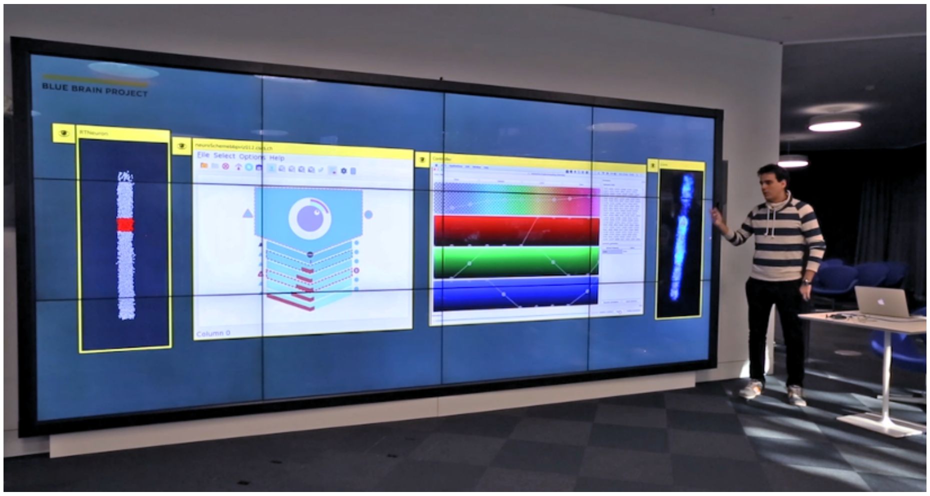

Fig. 1: Blue Brain 4×3 tiled display wall

we found that most solutions where not ready to be used in 24×7 unattained environment. On the other hand, the technology has been commoditized to make these type of installations affordable to medium-sized institutions which allowed us to build the software integration for a reasonable startup cost. The multitouch user interface implements a low entry barrier for new users, which is a unique capability of our solution.

1.2 OpenDeck

OpenDeck is our next-generation visualization system, aiming to integrate the success of tiled display walls with a seamless transition to fully immersive environments. We are currently in the process of installing the system which consists of a semi-cylindrical back-projection screen with a 41 Megapixel usable resolution on a 36 m^2 surface (Figure 2). Like the display walls, it is equipped with multitouch capabilities which makes it usable as a monoscopic collaboration system from the first day of installations. Unlike tiled display walls, it is active stereo capable and is equipped with a 3D tracking system for immersive rendering. For increased immersion, a lower resolution front projection system fills in the floor area. OpenDeck will allow us to run our applications based on Equalizer [4] once the system is installed.

The OpenDeck will provide a unique environment for the research and development of new visualization techniques. The main area of research will be mixed reality systems, such as touch interface implementation in an immersive environment, and the transition from 2D content to full immersion. The OpenDeck immersive infrastructure with multitouch will open a set of questions along immersive touch user interfaces, transitions and mixing of monoscopic to immersive usage, the combination of tracked and touch devices, multi-user immersion, latency for remote immersive rendering as well as multi-site collaboration.

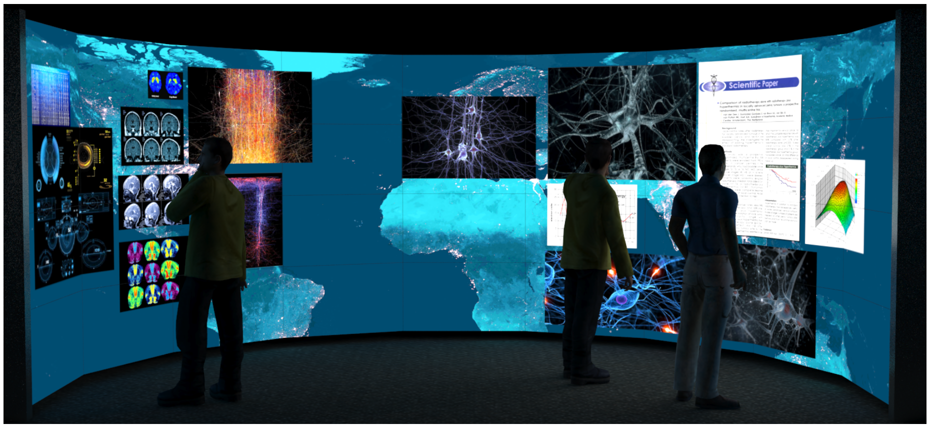

Fig. 2: OpenDeck concept rendering

1.3 Tide

Tide (Tiled Interactive Display Environment) is the software driving the Blue Brain tiled display walls and OpenDeck. It provides multi-window, multi-user touch interaction on large surfaces — think of a giant collaborative wall-mounted tablet. Tide is a distributed application that can run on multiple machines to power display walls or projection systems of any size. Its user interface is designed to offer an intuitive experience on touch walls. It works just as well on non touch-capable installations by using its web interface from any web browser. Figure 1 shows Tide on a 4×3 display wall and Figure 3 shows the Tide web interface in a browser.

Compared to other solutions [3, 6, 7], the development of Tide was driven by the need of a stable and easy to use implementation. Tide supports three types of content: files (high-resolution images, movies, pdfs), built-in applications (web browser, whiteboard) and remote applications using the Deflect library (DesktopStreamer, Equalizer-based applications, Brayns). The multitouch user interface can handle multiple users manipulating different windows and their content.

1.4 Deflect

Deflect is the client library for Tide. It provides an API for pixel streaming to Tide and for receiving events from Tide. The pixel streaming allows synchronized parallel streaming from a parallel rendering application as well as monoscopic and stereoscopic streams. Various events allow the application to react to multi-touch input from the wall.

Deflect is integrated into the Equalizer parallel rendering framework [4], enabling transparent usage of Equalizer applications on Tide walls. Furthermore, the DesktopStreamer application mirrors the desktop of other machines onto a wall window and allows interaction with the remote desktop. Other rendering applications, such as our interactive raytracing engine Brayns [2] are easily integrated with Deflect and Tide.

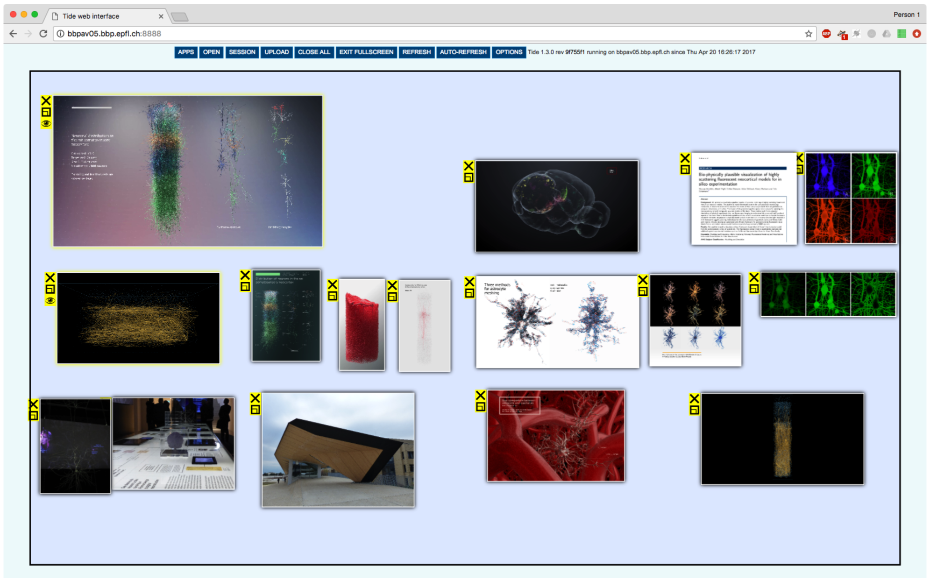

Fig. 3: Tide web interface

2 Messaging/Services

3 Rendering Applications

3.1 Interactive Raytracing

3.2 Out-of-Core Volume Rendering

Real-time rendering of volumetric data suffers from the curse of dimensionality. The data size of regular grids increases cubically as the number of dimensions increase. To render such data interactively, out-of-core (OOC) storage using multi-resolution algorithms are commonly employed.

Livre is an interactive volume rendering engine available under a permissive open source license. Our main contribution are: an state-of-the-art implementation of an octree-based level-of-detail selection, a task parallel rendering pipeline, a multi-GPU based parallel rendering engine, and an easily extensible renderer through the use of plugin data sources. Furthermore we propose to create volume representations of different input data using on-the-fly, interactive transformations of the data, skipping costly preprocessing steps.

Our system brings together state-of-the art algorithms to create a volume rendering engine capable of handling extremely high-resolution volumes using a high degree of parallelism, both on a single system as well as in a distributed cluster. We employ a GPU-based ray casting algorithm to compute the radiance absorption of the given volumetric data. The computation is executed per pixel on the pixel shader hardware of the GPU.

In our out-of-core data access layer, multi-resolution data is represented as an octree data structure. This representation accelerates the selection of the proper level-of-detail and to track the status of the LODs (in CPU memory, in GPU memory, not loaded). While rendering, view-based LOD selection is performed using the on the screen-space-error (SSE) [5] technique. According to the requested maximum SSE, the selected LODs represent the given volume the best on the screen.

The creation of volume bricks, their upload to the GPU and the rendering are executed in separate tasks. These tasks run asynchronously, that is, they do not block each other. This is particularly important for the render thread, which has to react to user input as quickly as possible to reduce input latency, and renders at interactive frame rates. The data loader thread can take seconds to generate the brick, for example in the field voxelization data source when sampling millions of events.

Livre uses a plugin mechanism to access the volume data. In this mechanism, data sources are implemented as shared libraries and are loaded on application startup based on the URI of the input data. Loading the shared libraries on runtime allows to add third-party data loaders to an existing Livre application by simply placing them into the shared library directory. Data sources have to provide the requested volume bricks, that is, there is no defined file format or even requirement to read the input data from a file system. This flexibility of the plugin approach lead to novel volume rendering use cases, where volume representations are created on the fly from different input data sets.

4 Discussion and Conclusion

Acknowledgments

This publication was supported by the Blue Brain Project (BBP), the Swiss National Science Foundation under Grant 200020-129525, the King Abdullah University of Science and Technology (KAUST) through the KAUST-EPFL alliance for Neuro-Inspired High Performance Computing, the Spanish Ministry of Science and Innovation under grant (TIN2010-21289-C02-01/02), the Cajal Blue Brain Project, Hasler and HBP.

References

1. Blue Brain Project. Tide: Tiled Interactive Display Environment. https://github.com/BlueBrain/Tide, 2016.
2. Blue Brain Project. Brayns: Interactive raytracing of neuroscience data. https://github.com/BlueBrain/Brayns, 2017.
3. T. A. DeFanti, J. Leigh, L. Renambot, B. Jeong, A. Verlo, L. Long, M. Brown, D. J. Sandin, V. Vishwanath, Q. Liu, M. J. Katz, P. Papadopoulos, J. P. Keefe, G. R. Hidley, G. L. Dawe, I. Kaufman, B. Glogowski, K.-U. Doerr, R. Singh, J. Girado, J. P. Schulze, F. Kuester, and L. Smarr. The optiportal, a scalable visualization, storage, and computing interface device for the optiputer. *Future Gener. Comput. Syst.*, 25(2):114–123, Feb. 2009.
4. S. Eilemann, M. Makhinya, and R. Pajarola. Equalizer: A scalable parallel rendering framework. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):436–452, May/June 2009.

5. S. Guthe and W. Strasser. Advanced techniques for high-quality multi-resolution volume rendering. *Computers & Graphics*, 28(1):51–58, 2004.
6. G. P. Johnson, G. D. Abram, B. Westing, P. Navr'til, and K. Gaither. DisplayCluster: An Interactive Visualization Environment for Tiled Displays. In *2012 IEEE International Conference on Cluster Computing*, pages 239–247, Sept 2012.
7. T. Marrinan, J. Aurisano, A. Nishimoto, K. Bharadwaj, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh. SAGE2: A new approach for data intensive collaboration using Scalable Resolution Shared Displays. In *Collaborative Computing: Networking, Applications and Worksharing*, pages 177–186, 2014.