Comparison of Multiple Large Fluid-Structure Interaction Simulations in Virtual Reality

Daniel Orban* University of Minnesota
Seth Johnson† University of Minnesota
Hakizumwami Birali Runesha‡ University of Chicago
Lingyu Meng§ University of Chicago
Bethany Juhnke¶ University of Minnesota
Arthur Erdman‖ University of Minnesota
Francesca Samsel** University of Texas – Austin
Daniel F. Keefe†† University of Minnesota

1 Introduction

Scientific visualization tools are rapidly embracing the necessary challenge of simultaneously visualizing multiple parameterized simulation data sets [8]. In the new paradigm, scientists hope to understand parameter relationships and stochastic trends that exist in a parameter space [6,7]. At the same time, virtual reality (VR) environments have enabled exciting possible opportunities for exploring and comparing time varying spatial data sets [3]. Although VR offers a unique perspective to view 3D and 4D data, it requires high framerates for interactivity and optimized use of precious GPU memory. Accurate simulations, on the other hand, are often very large due to dynamic unstructured mesh resolutions and small timesteps, making it difficult to simply render even one data set. To solve this, large data visualization frameworks often use data sampling and efficient rendering techniques to engage the GPU [1, 8]. Even then, VR is mostly used to add a stereoscopic view, and is rarely an integral part of interactive data instance comparison [3].

We present a framework for interactively comparing multiple large fluid structure interaction (FSI) simulations by extending a VR application called Bento Box. Our approach uses rendering, sampling, and streaming strategies to optimize memory on the GPU while achieving interactive framerates. We analyze our approach by comparing 10 cardiac lead FSI simulations, totaling 39GB, in two separate VR environments. Surprisingly, visualizing spatial relationships and interaction between data parts in multimodel scenarios (i.e. fluid and structure interactions) is rarely explored [2]. Our contribution provides a case study for building a multi-instance VR comparison tool for FSI data sets that are too large to fit onto a GPU.

2 Fluid-Structure Interaction using Bento Box

Fluid-Structure Interactions (FSI), the coupling of fluid and solid domains, are critical to the success of many engineering applications including developing medical devices, bridges, airplanes, and engines. Consider the case of analyzing how the stiffness and length of cardiac lead affects blood flow and tissue stress in the right atrium of a heart [5]. An engineer must discover what length and stiffness of the lead adequately attaches to the heart wall while minimizing the effect on the surrounding blood flow.

In order to understand the differences between multiple FSI simulations, like the cardiac lead model, we extend a general VR comparison tool called Bento Box. Bento Box allows users to interactively select and drill down to linked regions of interest across several instances. Fig. 1 shows the bimanual 3D interface specifically built for immersive VR environments where users can view and build a comparison grid. The columns in the grid represent instances and the rows linked sub-volumes of interest. The first row shows an overview of each simulation. From any cell, engineers can interactively create linked regions of interest (new rows) by clicking and dragging a bounding box inside the zoomed-in volume. Therefore, users can compare many spatially relevant views side-by-side based on user-defined volumes of interest (i.e. stress where the lead touches the heart and blood flow patterns elsewhere). Time steps can be compared by adding new columns for each visible instance. Users interact by selecting and zooming into a set of cells using a laser pointer. They can then immersively reposition, resize, change color maps, change visible variables, scroll through time, and create new sub-volumes of interest to customize each linked row.

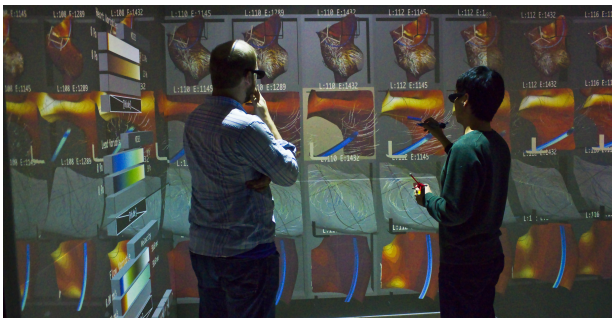

Figure 1: The Bento Box application displaying multiple instances and many views of a parameterized cardiac lead model.

3 Methods

Due to large data instance sizes, we can not simply load all the data onto the GPU. In order to make Bento Box possible for multiple large FSI simulations, we implement different rendering, sampling, and streaming strategies for the solid and fluid domains.

The solid domain represents variables like position, displacement, stress, and pressure. A triangular mesh is first generated from the unstructured grid primitives provided by the simulation. Positional attributes like displacement and stress are stored in separate vertex arrays on the GPU. We then use standard mesh rendering techniques to show the solid volume textured by a user selected attribute, while a vertex shader displaces vertex positions based on the time step.

Since it is not possible to load all the solid data on to the GPU, we use a temporal sampling technique in order to maximize the mesh resolution for accuracy. Perceptual research suggests static views more helpful than animated views for data analysis tasks like comparison [4], so we assume solid data is streamed in on demand. This, however, presents a significant challenge as achieving efficient GPU memory management and fast update speeds is a complex balancing act. Our solution stores variables in their own vertex

*e-mail: dtorban@umn.edu
†e-mail: joh08230@umn.edu
‡e-mail: runesha@uchicago.edu
§e-mail: Lmeng7@hawk.iit.edu
¶e-mail: toure023@umn.edu
‖email: agerdman@umn.edu
**e-mail: figs@cat.utexas.edu
††e-mail: dfk@umn.edu

Table 1: Characteristics and memory usage for the ten data instances.

ID	Raw (MB)	Solid (MB) Processed	Solid (MB) GPU	Fluid (MB) Proc. / GPU	Total (MB) Processed	Total (MB) GPU
108-1145	5,252.6	892.6	7.0	115.2	1,007.8	122.2
108-1289	4,724.4	682.2	7.0	115.2	797.4	122.2
108-1432	4,911.5	682.2	7.0	115.2	797.4	122.2
110-1145	2,933.7	660.9	6.7	115.2	776.2	122.0
110-1289	2,933.7	660.9	6.7	115.2	776.2	122.0
110-1432	2,933.7	660.9	6.7	115.2	776.2	122.0
112-1145	4,926.1	687.5	7.0	115.2	802.7	122.3
112-1289	4,934.6	687.5	7.0	115.2	802.7	122.3
112-1432	4,934.6	687.5	7.0	115.2	802.7	122.3
116-1145	1,588.9	832.4	6.9	115.2	947.6	122.1
	40,073.6	7,134.7	69.1	1,152.3	**8,287.0**	**1,221.4**

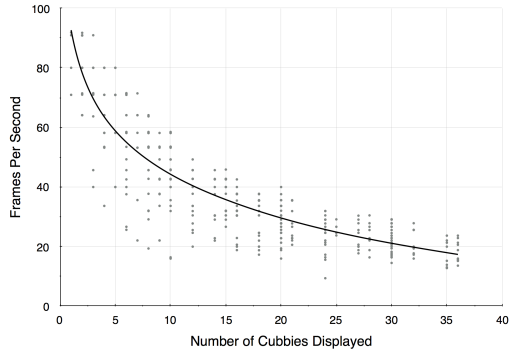

Figure 2: Framerate data based on visual grid size.

arrays on the GPU, which are indexed by visible time step and then by position. Updates to the GPU from the CPU, therefore, only happen if a time step changes or different variables are selected. When a variable changes, the full array needs to be updated, and when a time step changes, only the specific time step information needs to be updated. Therefore, the indexing scheme is optimized for time step changes, allowing for efficient solid domain animation.

For the fluid domain we use a spatial path line sampling and instanced particle rendering. Path lines are generated offline using an accelerated cell location data structure. We use random sampling for choosing the particle seed locations and advect them backwards and forward in time. The end result is hundreds of thousands particle paths across all data instances, stored as data buffers on the GPU. The GPU also stores path value buffers for data variables such as velocity and pressure that may be used for particle visualization.

In order to optimize the memory on the GPU, we only store one simple axis-aligned glyph mesh. The Bento Box application specifies the number of particles to render based on zoom factor or desired particle density. From there, we render the glyph using instanced rendering, which allows thousands of particles to be drawn with one draw call. Based on the instance number, we look up the particle path in the vertex shader and modify the glyph vertex positions to follow the path at the desired time step. This fluid domain rendering method makes it possible to visualize the flow at any scale and any time from the same pre-calculated data. Thus, changing or adding a visible time step does not use any additional GPU memory or require additional CPU-GPU memory updates.

4 Data and Results

The cardiac lead data sets are built using the ABAQUS solver. The bounding geometry of the right atrium is a smoothed version of a real heart anatomy captured via CT scan, and the cardiac lead is modeled as a uniform wire entering the right atrium through the superior venae cavae and exiting through the tricuspid valve. Characteristics for ten visualized data instances are reported in Table 1. After processing the 39 GB of raw data, the amount of memory needed to accurately visualize the solid and fluid attributes is over 8 GB, exceeding a 4 GB GPU hardware limit on our 4-wall cave environment, a 2 processor Intel(R) Xeon(R) CPU E5–2640 @2.50GHz machine with two NVIDIA Quadro K5000 cards and 192 GB of RAM. Streaming combined with the pathline sampling of the fluid, provides an extremely low memory footprint on the GPU, allowing us to visualize many instances and variables.

Using the Bento Box application as a testbed, we also report some rendering performance measures, summarized in Fig. 2. These timings were recorded on a 4 core processor Intel(R) CORE(TM) i7–7700HQ CPU @2.80GHz machine with 16 GB of RAM and a NVIDIA GeForce GTX 1070 graphics card, which was configured to drive an HTC Vive with a resolution of 2160×1200 pixels. The data sets are streamed into memory from a 128 GB M.2 PCIe SSD. The scatter plot in Fig. 2 shows a systematic sampling of Bento Box grid configurations that are possible for this 10-instance data ensemble. All possible grid arrangements that result in a total of 40 cells or less were sampled. The trend is above 30 frames-per-second for Bento Box arrangements of about 20 cells or less, and is in the 40–50 frames-per-second range for smaller arrangements.

5 Conclusion

Our extension to the Bento Box application allows the comparison of multiple large FSI simulations at interactive rates, a rare feat according to the scientific visualization literature. In fact, our sampling algorithms were able to reduce 39GB of simulation data to 1.22GB on the GPU, implying that the number of instances can scale further. We hope that this case study can inspire more work in combining large data visualization techniques with interactive exploration.

Acknowledgments

This work was supported in part by grants from the National Science Foundation (IIS-1251069,IIS-1218058) and the National Institutes of Health (1R01EB018205-01). Thanks to Bogdan Tanasolu, Georgi Subashki, and Shan "Sandy" Wang for simulation assistance and data wrangling. Thanks to Dr. Paul Iaizzo and the University of Minnesota Visible Heart Lab for access to heart geometry data.

References

[1] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717, 2005.

[2] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE transactions on visualization and computer graphics*, 19(3):495–513, 2013.

[3] K. Kim, J. V. Carlis, and D. F. Keefe. Comparison techniques utilized in spatial 3d and 4d data visualizations: A survey and future directions. *Computers & Graphics*, 67:138–147, 2017.

[4] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.

[5] H. B. Runesha, B. F. Tanasoiu, G. Subashki, A. G. Erdman, and D. F. Keefe. Fluid–structure interaction simulation of cardiac leads in the heart: Developing a computational model for use in medical device design. *Journal of Medical Devices*, 10(3):030959, 2016.

[6] D. Schroeder, F. Korsakov, C. M.-P. Knipe, L. Thorson, A. M. Ellingson, D. Nuckley, J. Carlis, and D. F. Keefe. Trend-centric motion visualization: Designing and applying a new strategy for analyzing scientific motion collections. *IEEE transactions on visualization and computer graphics*, 20(12):2644–2653, 2014.

[7] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[8] D. Vohl, D. G. Barnes, C. J. Fluke, G. Poudel, N. Georgiou-Karistianis, A. H. Hassan, Y. Benovitski, T. H. Wong, O. L. Kaluza, T. D. Nguyen, et al. Large-scale comparative visualisation of sets of multidimensional data. *PeerJ Computer Science*, 2:e88, 2016.