Parallel Processing and Immersive Visualization of Sonar Point Clouds

Alessandro Febretti*
Electronic Visualization Lab,
University of Illinois at Chicago

Kristof Richmond†
Frontier Astronautics

Peter Doran‡
Dept. of Environmental Sciences,
University of Illinois at Chicago

Andrew Johnson◊
Electronic Visualization Lab,
University of Illinois at Chicago

Figure 1. A team of environmental scientists analyzing sonar data from West Lake Bonney, Antarctica using our visualization tool inside the CAVE2 hybrid immersive environment. The full dataset consists of 350 million sonar points, which can be filtered interactively based on attributes such as ping return time, beam takeoff angle and timestamp. The point cloud is overlaid with a secondary depth data source (the white columns) for cross-checking sonar depth accuracy.

Abstract

The investigation of underwater structures and natural features through Autonomous Underwater Vehicles (AUVs) is an expanding field with applications in archaeology, engineering, environmental sciences and astrobiology. Processing and analyzing the raw sonar data generated by automated surveys is challenging due to the presence of complex error sources like water chemistry, zero-depth variations, inertial navigation errors and multipath reflections. Furthermore, the complexity of the collected data makes it difficult to perform effective analysis on a standard display. Point clouds made up of hundreds of millions to billions of points are not uncommon. Highly interactive, immersive visualization is a desirable tool that researchers can use to improve the quality of a final sonar-based data product.

In this paper we present a scalable toolkit for the processing and visualization of sonar point clouds on a cluster-based, large scale immersive visualization environment. The cluster is used simultaneously as a parallel processing platform that performs sonar beam-tracing of the source raw data, and as the rendering driver of the immersive display.

1 Previous Work

3D Mapping with underwater robots ([1], [2]) has received relatively little attention when compared to 3D mapping in other environments. Whereas in other settings it is possible to use laser scanners or other optical methods to analyze the local and remote geometry of the obstacles and features surrounding an autonomous vehicle, the optical properties of underwater environments make this impossible. Sonar is still the best technology for underwater location, navigation and scanning.

Point cloud visualization has been extensively explored in previous works, with a particular focus on photorealistic reconstruction of environments using LIDAR scans ([3], [4]). In our work, we are less interested in displaying a photorealistic representation of a sonar scan. We want our visualization to convey meaningful information about the scanned geometry and the scanning process itself, for instance to understand the relation between the AUV attitude or sonar beam angle and possible errors in the reconstruction. We also want to be able to filter the visualized point cloud in real time based on one or more attributes in the source raw data, a feature that is typically available only as an offline processing step in standard LIDAR visualization tools.

2 Context

The processing and visualization toolkit described in this paper was created to support the analysis of a particularly challenging underwater environment: Lake Bonney, in the McMurdo Dry Valleys, Antarctica. NASA funded the exploration of this lake using the Environmentally Non-Disturbing Underwater Robotic ANTarctic Explorer (ENDURANCE) AUV. NASA hopes to build upon lessons learned during testing for exploring objects in our solar system known to harbor sizable bodies of water, such as Jupiter's moon, Europa.

ENDURANCE operated during two Antarctic summer seasons on 3 distinct science objectives: water chemistry profiling, bathymetry scanning, and glacier exploration. This work concentrates on processing data from bathymetry and glacier dives, but a full coverage of the mission science objective can be found in [5] and [6]. For the purpose of bathymetry reconstruction, the source data consisted of about 350 Million distinct sonar range returns, plus navigation data and AUV attitude information at 0.2 second intervals.

Both the processing and visualization tools presented in this work have been evaluated on the CAVE2 system [7], a hybrid immersive environment with a 72Mpixel display driven by a cluster of 36 16-core Xeon processor machines with 20Gb/s shared network connectivity.

* afebre2@uic.edu
† kristof.richmond@stoneaerospace.com
‡ pdoran@uic.edu
◊ aej@evl.uic.edu

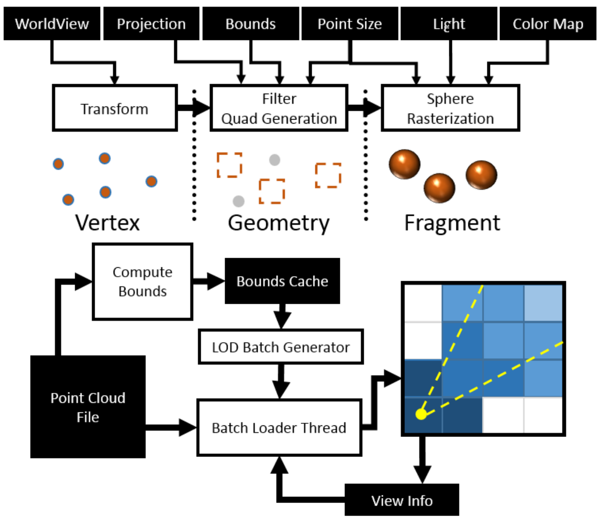

Figure 2. *(top)* structure of the GPU pipeline used by the point cloud visualization tool. *(bottom)* Main components of the point batch loading system.

3 Sonar Beam Processing and Visualization

The basic measurement of most sonar systems is the time a sound pressure wave takes to reach a reflector and return. The speed of sound in water varies with changes in temperature, salinity and pressure, causing the trajectory of sound waves to bend due to refraction. To correct for beam bending, sonar data is processed by a ray-tracing algorithm that adjusts the beam trajectory as it goes through water layers with different sound speed characteristics. After beam tracing, the 3D points generated from multiple missions are merged into a single point cloud: overlapping data is used to filter noise and to generate final output points at the desired resolution level.

Our parallel sonar processing framework is based on an open source sonar processing toolkit previously developed by us, called dttools[1]. We extended the toolkit with two additional tools that use a Message Passing Interface (MPI) implementation to distribute the beam tracing and point cloud merging on a cluster. The new tools accept the same configuration files and command line arguments as their non-cluster variants, so they can be substituted easily in existing processing pipelines.

3.1 GPU Point Cloud Rendering and Filtering

Figure 2 (top) illustrates the general structure of the GPU component of our visualization tool. The pipeline processes a vertex buffer containing point positions decorated with attribute data for each point. We use conditional primitive emission inside a geometry shader to discard points that are not within the attribute bounds set by the user. For each passed point, the geometry shader emits a screen-space quadrilateral, which is then rendered as a high quality sphere by a fragment shader.

To increase the scalability of our visualization tool and further improve frame rate, we use a level-of-detail (LOD) point batch loader (Figure 2, bottom). Upon loading, data from a point cloud file is split in batches of a target size. Based on observer distance, each batch is loaded at a different decimation level. Batches are queued for loading on a separate thread, so the visualization remains interactive regardless of the speed at which LODs update.

The visualization tool has been implemented on top of a scalable application framework called Omegalib [8]. Omegalib makes it possible to run our visualization both on standard computers and on cluster-based immersive environments.

1. http://dttools.googlecode.com

4 Results

Figure 1 shows the ENDURANCE team using our visualization tool inside the CAVE2 hybrid immersive environment. Through the tool, the team could display multiple views of the data, overlay different depth information sources for visual cross-checking and perform virtual measurements and place fiducial markers for depth contour generation. As the meeting progressed and users identified errors in the data, they were able to perform quick 'experiments' by modifying some of the beam tracing or noise filtering parameters and re-import the data into the visualization to assess the effect of the changes.

We evaluated the performance of our parallel tools on the 36-node cluster running CAVE2, varying the number of cores used, from 8 to 512. Both the beam tracing and merge steps reach a maximum speedup at 128 cores. Past this point, data communication becomes a bottleneck and prevents further improvement of the parallel performance. At 128 cores, beam tracing the 350 million points of the ENDURANCE dataset takes approximately 4 minutes, compared to 1 hour needed by a sequential run. For the merge step, we measure the speedup for two clustering resolutions: 1 meter and 2 meters. While 1-meter resolution is desired for the final data product, the 2-meter resolution in enough to perform quality assessment of the data and identify possible merging issues. At 128 cores, merging all the ENDURANCE dives at 2-meter resolution requires about 5 seconds.

5 Conclusion

Our parallel processing and visualization tools have proven effective in speeding up the analysis of a challenging dataset such as the one generated by the ENDURANCE mission. We are planning to further improve the tools for use in an upcoming Antarctic mission, involving a much larger survey volume. To the best of our knowledge, this is the first toolkit specifically designed to support the integrated processing and visualization of sonar data on a cluster-driven display. The toolkit is open source and can be downloaded from the public dttools and Omegalib repositories. Our point cloud visualization tool can be used to visualize data coming from other sources as well, i.e. LIDAR scans or depth cameras.

References

[1] C. Forney, J. Forrester, B. Bagley, W. McVicker, J. White, T. Smith, J. Batryn, A. Gonzalez, J. Lehr, T. Gambin, and others, "Surface reconstruction of Maltese cisterns using ROV sonar data for archeological study," *Advances in Visual Computing*, pp. 461–471, 2011.

[2] N. Fairfield, G. Kantor, and D. Wettergreen, "Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels," *Journal of Field Robotics*, vol. 24, no. 1–2, pp. 03–21, 2007.

[3] M. Wimmer and C. Scheiblauer, "Instant points: Fast rendering of unprocessed point clouds," *IEEE VGTC conference on Point-Based Graphics*, 2006.

[4] O. Kreylos, G. Bawden, and L. Kellogg, "Immersive visualization and analysis of LiDAR data," *Advances in visual computing*, pp. 846–855, 2008.

[5] K. Richmond, A. Febretti, S. Gulati, C. Flesher, B. P. Hogan, A. Murarka, G. Kuhlman, M. Sridharan, A. Johnson, W. C. Stone, J. Priscu, P. Doran, C. Lane, D. Valle, C. Science, S. M. St, L. J. Hall, and W. T. S. Chicago, "Sub-Ice Exploration of an antarctic lake: results from the Endurance Project," in *17th International Symposium on Unmanned Untethered Submersible Technology (UUST11)*, 2011.

[6] A. Febretti, K. Richmond, and S. Gulati, "Poisson reconstruction of extreme submersed environments: The ENDURANCE exploration of an under-ice Antarctic Lake," *Advances in Visual* …, 2012.

[7] A. Febretti, A. Nishimoto, T. Thigpen, J. Talandis, L. Long, J. D. Pirtle, T. Peterka, A. Verlo, M. D. Brown, D. Plepys, D. Sandin, L. Renambot, A. Johnson, and J. Leigh, "CAVE2 : A Hybrid Reality Environment for Immersive Simulation and Information Analysis," 2012.

[8] A. Febretti and A. Nishimoto, "Omegalib: A multi-view application framework for hybrid reality display environments," *IEEE Virtual Reality (VR)*, 2014.