Sound Rendering and its Acceleration Using FPGA

Yiyu Tan
RIKEN Center for Computational Science
Kobe, Hyogo, Japan
tan.yiyu@riken.jp

Toshiyuki Imamura
RIKEN Center for Computational Science
Kobe, Hyogo, Japan
imamura.toshiyuki@riken.jp

INTRODUCTION

Sound rendering exhibit numerical methods to model sound propagation phenomena in spatial and time domain, and the FDTD method has become an essential approach in room acoustics owing to its high accuracy and ease of implementation and parallelization since it was introduced to analyse acoustical behaviour by O. Chiba et al. and D. Botteldooren et al. [1–3]. However, the FDTD method is computationally intensive and memory-intensive as the problem size is increased because oversampling in spatial grids is required to suppress numerical dispersion. In recent years, GPUs and FPGAs were applied to accelerate sound rendering. In particular, FPGAs become promising in sound rendering because of its easy customization of the I/O interfaces for real-time applications. In this research, a FPGA-based sound renderer is developed, and large external memory is applied to extend the rendering sound space.

SYSTEM DESIGN AND IMPLEMENTATION

Rendering algorithm. The hardware-oriented FDTD algorithm [4-6] is applied, in which 7-point stencil is used to compute the sound pressure of a grid. Equation 1 shows the formula to compute sound pressure of a grid from its six neighbors, in which *D1* and *D2* are parameters selected according to the grid position and boundaries.

$$P_{i,j,k}^{n} = D1 \times [\, P_{i-1,j,k}^{n-1} + P_{i+1,j,k}^{n-1} + P_{i,j-1,k}^{n-1} + P_{i,j+1,k}^{n-1} + P_{i,j,k-1}^{n-1} + P_{i,j,k+1}^{n-1} + 2P_{i,j,k}^{n-1}] - D2 \times P_{i,j,k}^{n-2} \quad (1)$$

System design and implementation. Sound rendering is memory-intensive, and it is impossible to keep all data in the on-chip block RAMs inside FPGA as the sound space becomes large. Instead, the large external on-board RAMs are adopted to extend the rendering sound volume in this study. Furthermore, a sliding window-based data buffering system is investigated to reduce the demand of memory bandwidth and speed up data access to/from the external on-board memory. The system is designed by using OpenCL, compiled by the Intel FPGA SDK for OpenCL 17.1, and implemented by the FPGA board DE5a-NET from the Terasic Company. The whole system utilizes 70701 logic blocks, 152 DSP blocks, and 891 RAM blocks after implementation.

PERFORMANCE EVALUATION

Fig. 1 shows the rendering time in the case of different grid scales and data being single precision floating-point by the proposed FPGA-based rendering system and the software simulation performed on a desktop machine with 128 GB DDR4 memory and an Intel i7-7820X processor running at 3.6 GHz. The related C++ codes for the software simulation are parallelized by OpenMP to use all eight processor cores. The FPGA system runs at 267 MHz, and it takes almost half of the rendering time of the software simulations executed on the desktop machine.

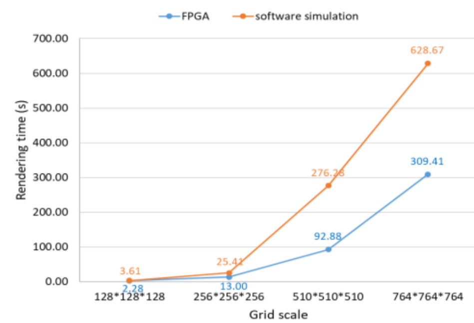

Fig. 1 Rendering time

CONCLUSION

In this work, a FPGA-based sound renderer is investigated to speed up computation. Compared with the software simulation performed on the desktop machine, the FPGA-based sound renderer doubles the computation performance even though it runs at much lower clock frequency and has much smaller external memory over the desktop machine. In future work, more computing units will be applied to improve computing performance.

ACKNOWLEDGMENTS

Thanks for Intel's donation of the FPGA board DE5a-NET and the related EDA tools through University Program. This work was supported by the I-O DATA Foundation and JSPS KAKENHI Grant Number JP19K12092.

REFERENCES

[1] D. Botteldooren (1994). Acoustical Finite-difference Time-domain Simulation in a Quasi-Cartesian Grid. Journal of the Acoustical Society of America, 95, 2313-2319.
[2] D. Botteldooren (1995). Finite-difference Time-domain Simulation of Low-frequency Room Acoustic Problems. Journal of the Acoustical Society of America, 98, 3302-3308.
[3] O. Chiba, T. Kashiwa, H. Shimoda, et al (1993). Analysis of Sound Fields in Three Dimensional Space by the Time-dependent Finite-difference Method based on the Leap Frog Algorithm. Journal of the Acoustical Society of America, 49, 551-562.
[4] T. Yiyu, Y. Inoguchi, M. Otani, et al (2018). A Real-Time Sound Field Rendering Processor. Applied Science, 8(35).
[5] Y. Y. Tan, Y. Inoguchi, Y. Sato, et al (2013). A Hardware-oriented Finite-difference Time-domain Algorithm for Sound Field Rendering. Japanese Journal of Applied Physics, 52, 07HC03.
[6] Y. Y. Tan, Y. Inoguchi, Y. Sato, et al (2014). A Real-time Sound Rendering System Based on the Finite-difference Time-domain Algorithm. Japanese Journal of Applied Physics, 53, 07KC14.