2P2-14

Proceedings of Symposium on Ultrasonic Electronics, Vol. 34 (2013) pp. 235-236
20-22 November, 2013

A Real-time Sound Rendering System based on the Finite-Difference Time-Domain Algorithm

Tan Yiyu[1†], Yasushi Inoguchi, Yukinori Sato [1], Makoto Otani [2],
Yukio Iwaya[3], Hiroshi Matsuoka[4], Takao Tsuchiya[5]
[1] Res. Center for Adv. Comp. Infra., Japan Adv. Inst. of Sci. & Tech.; [2] Faculty of Eng., Shinshu Univ.; [3] Faculty of Eng., Tohoku Gakuin Univ.; [4]Faculty of Eng., Tohoku Univ.; [5]Dept. of Info. Syst. Design, Doshisha Univ.

1. Introduction

Sound rendering technologies are widely applied in many industrial and scientific fields. Sound rendering applications are computation-intensive and memory-intensive. Traditional sound rendering systems are based on computer simulation, and suffer from arithmetic performance and memory bandwidth. Although the arithmetic performance may be improved by increasing clock frequency of processor or by using parallel techniques, the memory bandwidth is the bottleneck of performance improvement.

General-Purpose Graphic Processing Units were applied to enhance arithmetic performance through coarse-grain parallelism of the arithmetic units in recent sound rendering systems [1-3)]. However, these general-purpose processors are designed for high arithmetic performance without sufficient memory bandwidth, and therefore can not fit the requirement of the memory-intensive sound rendering, especially real-time applications.

To address this problem, FPGA-based sound rendering solutions were proposed to provide direct implementation of sound propagation equations by the configurable logic blocks inside a FPGA chip[4-7)]. The system performance is improved due to the inherit parallelism of a FPGA and small overhead of data access. In this paper, a real-time sound rendering system based on the Hardware-Oriented Finite-Difference Time-Domain (HO-FDTD) algorithm [5)] is proposed and implemented.

2. HO-FDTD Algorithm

In a cubic element, sound wave propagation is governed by the following formula:

$$\frac{\partial^2 P}{\partial t^2} = c^2\left(\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2}\right) \qquad (1)$$

where P denotes the sound pressure, c is the sound speed, x, y, and z are directions of an Cartesian coordinate system. By applying the center differential method in equation (1), and let $\Delta x = \Delta y = \Delta z = \Delta l$, then

$$\begin{aligned} P^{n+1}(i,j,k) &= 2P^n(i,j,k) - P^{n-1}(i,j,k) + \chi^2[P^n(i+1,j,k) \\ &+ P^n(i-1,j,k) + P^n(i,j+1,k) + P^n(i,j-1,k) \\ &+ P^n(i,j,k+1) + P^n(i,j,k-1) - 6P^n(i,j,k)] \end{aligned} \qquad (2)$$

where $\chi = c\Delta t/\Delta l$ denotes the Courant number. For a three dimensional sound space, $\chi \le 1/\sqrt{3}$. To eliminate the multiplication operation, χ is assumed to be $1/2$, then equation (2) is changed as[5)].

$$\begin{aligned} P^{n+1}(i,j,k) &= \frac{1}{4}[P^n(i+1,j,k) + P^n(i-1,j,k) + P^n(i,j+1,k) \\ &+ P^n(i,j-1,k) + P^n(i,j,k+1) + P^n(i,j,k-1) + 2P^n(i,j,k)] - P^{n-1}(i,j,k) \end{aligned} \qquad (3)$$

In equation (3), the multiplication operations are implemented by shift operations.

2.1 Boundary condition

Reflections from boundaries of an acoustic space play a pivotal role for sound rendering, and therefore attention has been given to the problem of formulating numerical approximations of boundaries. A reflective boundary can be modeled as a locally reacting surface by assuming that wave does not propagate in the direction parallel to the boundary surface. Thus when waves travel in a positive x-direction, the boundary condition for the right boundary in terms of pressure is

$$\frac{\partial P}{\partial t} = -c\xi_w \frac{\partial P}{\partial x} \qquad (4)$$

where $\xi_w = Z_w/\rho c$ is the normalized wall impedance. By using the centered finite difference method on equation (4), equation (5) is derived as an expression for a point lying outside of the modeled space, which is referred to as a "ghost point".

$$P^n_{l+1,m,i} = P^n_{l-1,m,i} + \frac{1}{\lambda\xi_w}\left(P^{n-1}_{l,m,i} - P^{n+1}_{l,m,i}\right) \qquad (5)$$

If the boundary reflection coefficient R is defined as $(\xi_w - 1)/(\xi_w + 1)$, equation (2) is rewritten as equation (6) by inserting equation (5), R, and λ .

yiyu-t@jaist.ac.jp inoguchi@jaist.ac.jp

$$P^{n+1}_{l,m,i}=[\frac{1}{4}(2P^{n}_{l-1,m,i}+P^{n}_{l,m+1,i}+P^{n}_{l,m-1,i}+P^{n}_{l,m,i+1}+P^{n}_{l,m,i-1}+2P^{n}_{l,m,i})-\frac{3R+1}{2(1+R)}P^{n-1}_{l,m,i}]/(\frac{3+R}{2(1+R)}) \quad (6)$$

In equation (6), the sound pressure of a node on the boundary is calculated by the sound pressures of its neighbors at previous time steps. The same derivation procedure is also used for the boundary edges and corners.

3. System Performance

To verify and estimate the performance of the proposed sound rendering system, a three-dimensional sound rendering system with 32×32×16 grids was investigated and implemented by a processor-based FPGA machine TD-SPP3000. The reflection coefficient of boundaries is 0.95. The incident and observation points are the middle of the sound space. For comparison, the related system is developed by C++ programming language, and executed on a personal computer (PC) with 16GB RAM and an AMD Phenom 9500 Quad-core processor running at 2.2 GHz. The reference C++ codes are compiled and optimized for the maximum speed with option of /O2.

3.1 System architecture

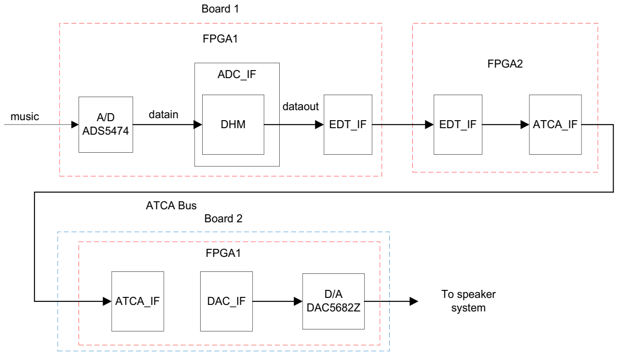

Fig. 1 System architecture

As shown in **Fig. 1**, the FPGA-based sound rendering system consists of two FPGA boards, and each FPGA board contains two FPGA chips. In Fig. 1, the incidence signal, such as a song, is sampled by a high-speed A/D board (ADS5474), which is attached to the FPGA 1 on the board 1. Then the sampled data are as incident data and processed by the DHM module. The sound pressures at the observation point are sent to the FPGA 2 through the extended data transfer module (EDT_IF). Finally, they are transferred to the D/A board (DAC5682Z) on the board 2 through the ATCA bus, and output to drive the speaker system. The whole procedure is handled at real-time. The hardware system are extended by modifying the data transfer interface between FPGAs (EDT_IF) and FPGA boards (ATCA_IF) to make multiple FPGAs work in parallel to enlarge the simulated area.

3.2 System stability

Since data are fixed-point in the hardware system, computational errors occur due to data truncation. They may be accumulated during calculation and result in system divergent and unstable. **Fig. 2** shows the impulse response of the investigated sound space obtained by the FPGA-based sound rendering system. The system becomes stable after 200 time steps, which indicates that the hardware system is convergent.

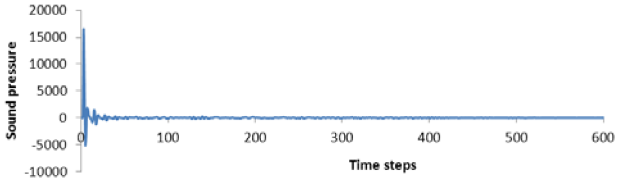

Fig. 2 Impulse response

3.3 Rendering time

Table I shows the rendering time taken by the software simulation on PC and FPGA-based system when three-minute Beethoven music is as an incidence. The rendering results are output at real-time in the FPGA-based system while they are taken long time in the computer-based software simulation. This owes much to small data access overhead and parallel processing inside FPGA.

Table I Rendering time

Node	FPGA system	Software simulation
32 × 32×16	real-time	61.021 minutes

Acknowledgment

This project is supported by the National Institute of Information and Communication Technology.

References

1. T. Ishii, T. Tsuchiya, and K. Okubo: Jpn. J. Appl. Phys. 52 [7] (2013) 07HC11.
2. T. Tsuchiya: Jpn. J. Appl. Phys. 49 (2010) 07HC10.
3. L. Savioja: Proc. Int. Conference on DAFx, 2010, p. 1.
4. T. Yiyu, Y. Inoguchi, E. Sugawara, et al: J. Sound Vib. 330 (2011) 4302.
5. T. Yiyu, Y. Inoguchi, Y. Sato, et al: Jpn. J. Appl. Phys. 52 [7] (2013) 07HC03.
6. T. Yiyu, Y. Inoguchi, et al: Proc. Int. Conference on DAFx, 2012, p. 93.
7. T. Yiyu, Y. Inoguchi, et al: Proc. Int. Conference on ITNG, 2012, p. 484.