A FPGA implementation of the two-dimensional Digital Huygens' Model

Conference Paper · December 2010

DOI: 10.1109/FPT.2010.5681441 · Source: DBLP

8 authors, including:

Tan Yiyu
Japan Advanced Insitute of Science and Technology, Nomi, Ishikawa, Japan

Yukinori Sato
Toyohashi University of Technology

Y. Inoguchi
Japan Advanced Institute of Science and Technology

Makoto Otani
Kyoto University

A FPGA Implementation of the Two-Dimensional Digital Huygens' Model

Tan Yiyu [#1], *Yukinori Sato* [#1], *Eiko Sugawara* [#2], *Yasushi Inoguchi* [#1],
Makoto Otani [#3], *Yukio Iwaya* [#4], *Hiroshi Matsuoka* [#4], *Takao Tsuchiya* [#5]

[#1] *Center for Information Science, Japan Advanced Institute of Science & Technology,*
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan

[1] yiyu-t@jaist.ac.jp, inoguchi@jaist.ac.jp

[#2] *Akita National College of Technology*

[#3] *Department of Information Engineering, Shinshu University*

[#4] *Research Institute of Electrical Communication, Tohoku University*

[#5] *Faculty of Science and Engineering, Doshisha University*

***Abstract*— Modeling acoustical behavior in a room is complicated and computationally intense. Many methods have been proposed to analyze the distribution of sound field by using computer simulation. However, the procedure is time-consuming with sound space increasing. In this paper, a hardware solution based on Digital Huygens' Model (DHM) and Field Programmable Gate Array (FPGA) technology is proposed to simulate and rebuild the distribution of sound field in a room. Two schemes of DHM are derived to analyze the sound propagation in a 2D space and implemented by FPGA. In a 2D space with 35 × 35 nodes and surrounded by rigid walls, the results got by hardware meet well with the analytical results in case of different incidences except having three-cycle delays. The designed hardware system consumes about 0.016s to process the computations during 10000 time steps while the software solution developed by C++ programming language costs about 0.14s. The hardware implementation occupies 76% of LUTs and 39% of FDCs in a Xilinx FPGA chip XC5VLX330T-FF1738.**

I. INTRODUCTION

Understanding and predicting acoustical behavior in a room becomes more and more needed in some applications. But modeling the acoustical phenomena accurately is complicated and hard because it is significantly affected by the geometry of the room and the boundary properties of the walls, ceiling and floor. Now analysis and simulation of sound fields have become familiar as a result of the progress in computer technology. Many methods have been proposed, such as the geometrical based methods like acoustical ray tracing [1], image source method [2], beam tracing method [3], acoustic radiosity [4-5]; and the wave-based methods like Finite Element Method (FEM) [6], Boundary Element Method (BEM) [7], Finite Difference Time-Domain (FDTD) method [8-9]. Especially the FDTD method has become a powerful method to analyze a wide variety of sound phenomena since it provides a direct time-domain solution to sound propagation with relatively good accuracy. However, analyzing the distribution of sound field is a computation-intense and data-intense work. It is time-consuming and requires abundant computational resources, which limits its applications where real-time computation, small size, and low power consumption are required. In recent years, some physical equivalent methods have been derived from the wave equations of sound propagation to investigate acoustical behavior, such as DHM, digital waveguide mesh [15-16].

DHM is based on the transmission line matrix method, and was proposed by Y. Kagawa and T. Tsuchiya [10-12][18]. In it, sound space was divided into small acoustic tubes and the neighbour tubes were connected in nodes. When a sound pressure pulse travelled on the space, due to the impedance discontinuity at nodes, parts of the pulse were launched into the adjacent tubes while others were reflected back into the original tube where the pulse came from. The transmitted pulses were then as the incidences of the adjacent tubes and the scatterings occurred again. This procedure is actually the discrete implementation of the Huygens' principle, and equivalent to the case that a voltage impulse travels over an orthogonal mesh made of pairs of transmission lines. The sound behavior at every node is investigated by the theory of transmission line matrix. Compared with other methods, DHM is simple and easy to implement on a computer or by hardware.

This paper will present an implementation of two-dimensional DHM by using FPGA. The rest of this paper is organized as follows. Section II will introduce DHM briefly. Section III will describe system analysis and simplification. The system implementation results by using FPGA are presented in Section IV. Section V will discuss the simulation results by hardware and software and performance estimation. Finally, conclusions are drawn in Section VI.

II. TWO DIMENSIONAL DIGITAL HUYGENS' MODEL

In the two dimensional DHM, sound space is divided into a number of sound tubes with same length Δl, and each tube is described by an element consists of four transmission lines W, N, E and S (shown in Fig. 1) [10], which are connected by a node. If the impedance of a transmission line is Z_0, then the equivalent impedance from one line to the other three lines at the node is $Z_0/3$ because the three lines are connected in parallel at the node. The reflection coefficient at the node is given by

$$\Gamma = \frac{Z_0/3 - Z_0}{Z_0/3 + Z_0} = -\frac{1}{2}$$

978-1-4244-8983-1/10/$26.00 ©2010 IEEE

Thus when an incident pulse P is applied to the line W at time $t = K\Delta t$, where K=1,2,3,…, $\Delta t = \Delta l / C_T$ and C_T is the propagation speed of sound in sound tubes, due to the impedance discontinuity at the connecting node, impulses with magnitude ½P will be transmitted into the lines E, N, and S, respectively, while an impulse with magnitude -½P will be reflected back to the line W.

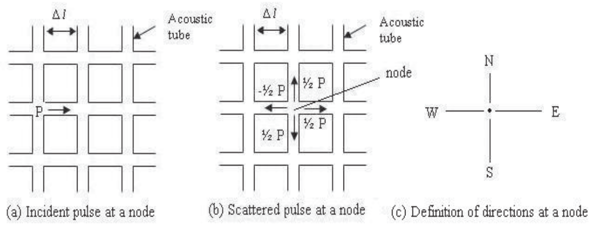

Fig. 1 Digital Huygens' model in a 2D space

If four impulses are incident on the lines E, W, N, and S simultaneously, the scattering matrix, which is shown in equation (1), is obtained by applying the superposition principle to the previous case of a single pulse [13-14].

$$\begin{bmatrix} {}_K S_E \\ {}_K S_S \\ {}_K S_W \\ {}_K S_N \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} {}_K P_E \\ {}_K P_S \\ {}_K P_W \\ {}_K P_N \end{bmatrix} \qquad (1)$$

Where ${}_K S_m$ and ${}_K P_m$ (the subscript m denotes the four lines E, S, W, N) are the scattered pulse and incident pulse on the different lines at time t, respectively. K is the discrete time steps. In general, time t is represented by K for simplicity. The scattered pulses then become the incident pulses to the adjacent elements. This procedure is same as the propagation of waves described by Huygens' principle when the sound space is divided into a mesh network connected by elements.

If the sound pressure $P(i,j)$ of a node at the position (i, j) is defined as half of the sum of the incident pulse magnitudes, then

$${}_K P(i,j) = \frac{1}{2}\left({}_K P_E(i,j) + {}_K P_S(i,j) + {}_K P_W(i,j) + {}_K P_N(i,j) \right) \qquad (2)$$

By using equation (2), equation (1) can be rewritten as

$${}_K S_m(i,j) = {}_K P(i,j) - {}_K P_m(i,j) \qquad (3)$$

At a node, the scattered pulses are as the incident pulses to the corresponding directions of its neighbours at the next time step. For a node at the position (i, j), the relations between ${}_K S$ and ${}_{K+1} P$ are shown as follows:

$$\begin{aligned} {}_{K+1} P_3(i,j) &= {}_K S_W(i+1,j) \\ {}_{K+1} P_W(i,j) &= {}_K S_E(i-1,j) \\ {}_{K+1} P_N(i,j) &= {}_K S_S(i,j+1) \\ {}_{K+1} P_S(i,j) &= {}_K S_N(i,j-1) \end{aligned} \qquad (4)$$

If a node locates on a rigid wall boundary, its scatterings and incidences have the following relations due to fully reflection.

$${}_K S_m(i,j) = {}_{K+1} P_m(i,j) \qquad (5)$$

Thus according to the initial incidence and equations (2), (3), (4), (5), the scatterings and sound pressures of all nodes are calculated through a time iterative based on the time step $\Delta l/C$. From equations (2) and (3), eight operations (three additions, four subtractions, one right shift) are needed to calculate the sound pressure and scatterings at a node.

III. System Analysis and Simplification

Before hardware implementation, the related algorithm and system architecture are carefully investigated to understand the system data flow, logical structure, data width and range, input/output interface and memory hierarchy. All these help to optimize the designed hardware system.

A. *Algorithm Analysis*

From equations (1), (2), (4), during computation, the sound pressure and scatterings of a node are needed to obtain and store for further computation. As shown in Fig. 2, DHM starts with initializing all its parameters and generating the connectivity of the mesh network for the sound space according to equation (4), Δl and the space dimension size. Every computing cell has associated the computation procedure shown in Fig. 2 with four incident inputs and four scattering outputs. The excitation data are external inputs and loaded into the input cells at every time step. The calculation procedure updates the sound pressure and the scatterings in four lines of each node. And the scatterings of a node are applied as the incidences of its neighbours in the next time step. The updating procedure is then repeated until the calculation time is over. The sound pressure of the observed point will output at any time.

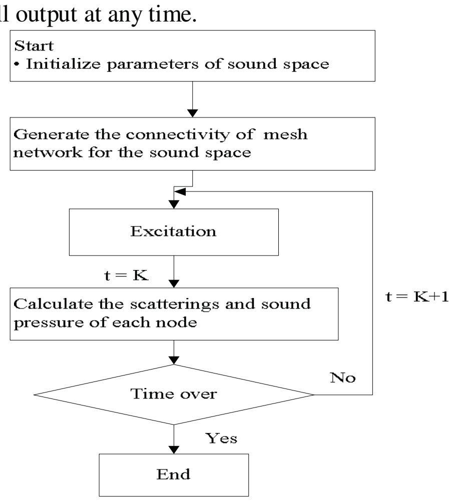

Fig. 2 Flow diagram of the 2D DHM

B. *System Simplification*

At every node, a computing cell is designed to calculate the sound pressure and scatterings according to the incident pulses. When the sound space area increases, more computing cells are needed and the consumed hardware resources increase significantly. Thus the structure of computing cell has a great affect on the final performance of system. An optimal computing cell may be developed through the based algorithm and circuit design techniques. The based algorithm is the principle factor to improve system performance while circuit design techniques are complements. From the above analysis, each computing cell costs three adders, four subtracters, and one shifter. By inserting equations (3), (4) into equation (2), and eliminate the pulses ${}_{K-1}P_m(i, j)$ (m=E, S, W, N), equation (2) is rewritten as.

$$ {}_{K}P(i,j)=\frac{1}{2}\left({}_{K-1}P(i+1,j)+{}_{K-1}P(i-1,j)+{}_{K-1}P(i,j+1)\right.$$
$$\left.+{}_{K-1}P(i,j-1)-\sum{}_{K-2}S_m(i,j)\right) \qquad (6)$$

The scattering matrix is symmetrical in equation (1), thus

$$\sum{}_{K}S_m(i,j)=\sum{}_{K}P_m(i,j) \qquad (7)$$

Equation (6) is expressed as equation (8) by substituting equation (7) [17].

$$ {}_{K}P(i,j)=\frac{1}{2}\left({}_{K-1}P(i+1,j)+{}_{K-1}P(i-1,j)+{}_{K-1}P(i,j+1)\right.$$
$$\left.+{}_{K-1}P(i,j-1)\right)-{}_{K-2}P(i,j) \qquad (8)$$

Equation (8) indicates that the sound pressure of a node can be obtained through the sound pressure of its neighbours. This formula is the same as the FDTD expression for the 2D wave propagation. The difference between them is that DHM is a physical model while FDTD is only a numerical method.

If a node *(i, j)* is on a rigid wall boundary, the related fields in equation (8) will be replaced by the results of sound pressure in the previous time step. For example, when the east boundary is a rigid wall and a node locates on it, equation (9) is used to calculate its sound pressure.

$$ {}_{K}P(i,j)=\frac{1}{2}\left({}_{K-1}P(i-1,j)+{}_{K-1}P(i,j-1)+{}_{K-1}P(i,j)\right.$$
$$\left.+{}_{K-1}P(i,j+1)\right)-{}_{K-2}P(i,j) \qquad (9)$$

Compared with the original scheme, equation (8) or (9) needs only five operations (three additions, one subtraction, and one right shift) and the implemented circuit is much simpler.

IV. System Implementation

The hardware-based DHM system is like an array with each node having a computing cell and its structure diagram is shown in Fig. 3. A uniform computing cell is designed to calculate the scatterings and sound pressure of a node according to its incident pulses. The sound pressure of the observation node is outputted through the hardware logic directly. In order to reduce circuit complexity, data are 32-bit fixed-point. The number of computing cells in every row and column is determined by the dimensions of sound space and Δl. All cells are cascaded together and implemented by pipelining to improve computation speed. The whole system is implemented by a processor-based FPGA machine TD-SPP3000 from Tokyo Electron Device Limited Corporation, which has two FPGA boards and one CPU board. Two Xilinx XC5VLX330T-FF1738 FPGA chips and 512Mbit SRAM are included in every FPGA boards. One FPGA chip is connected to the Advanced TCA bus and another is connected to the compact PCI bus. The two FPGA chips can exchange data directly by programming. The FPGA chips in different boards communicate through the Advanced TCA bus or the compact PCI bus. The processor board contains an Intel Pentium M processor (1.4 GHz) and 504MB RAM, which mainly provides an environment to debug system. Typically, a system with 1225 nodes in the simplified scheme is implemented by a FPGA chip, where 76% of LUTs and 39% of FDCs are occupied and after implementation, the system maximum frequency is about 137MHz. Since the current DHM system is relatively simple, only one FPGA board is needed.

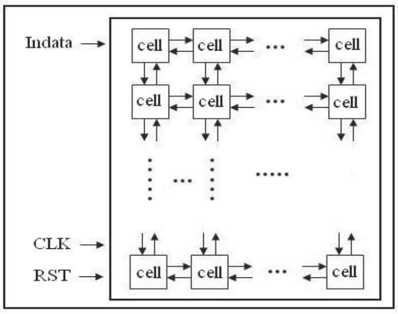

Fig. 3 System diagram

The hardware resources needed by the system with different number of nodes in the original and simplified schemes are shown in Fig. 4. Averagely, each node consumed 228 look-up tables (LUTs) and 128 D flip-flops (FDCs) in the original scheme, while it costs 130 LUTs and 68 FDCs in the simplified scheme. In the original scheme, every computing cell consists of three adders, four subtracters, and one shifter while it is composed of three adders, one subtracter, and one shifter in the simplified scheme. Thus in the original scheme, except the hardware logics are needed to obtain the sound pressure of a node, the other three subtracters and four D flip-flops are cost to calculate and store its scatterings, which results in the increasing of the hardware resources.

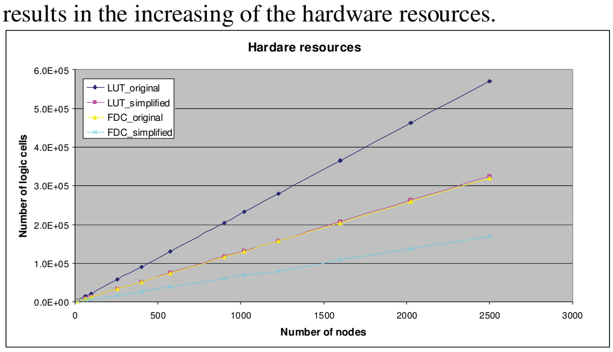

Fig. 4 Hardware resources needed in the original and simplified scheme

V. Results Comparison and Performance Estimation

To verify the validity of the proposed DHM and its implementation, a 2D 35 × 35 model space surrounded by rigid walls is investigated. For comparison, the simplified scheme of DHM is developed by the C++ programming language and simulated in a computer with an AMD Phenom 9500 Quad-core processor (1.8 GHz) and 4GB RAM. The simplified scheme is implemented by a Xilinx FPGA XC5VLX330T-FF1738. The system is incident by a single-shot sine wave and a Gaussian wave, respectively. On the incident pulses, the sample rate is 16 and the magnitude is 10^8. The incident position is on the west branch of node (0, 0), and the observation point is at the node (6, 15). The number of time steps is 163. Figures 5-6 show the calculation results in case of different incidents by software and hardware solutions. The results achieved by hardware and software are almost same except having three cycles delay.

The computational time to process 10000 time steps by software is 0.14s for the single-shot sine wave, and the Gaussian wave pulses, respectively. In the designed hardware running at 50MHz, the time is about 0.016s to handle the same time steps for the different incident pulses. Generally, the hardware system completes the computations for each time step in one clock cycle due to its pipelining structure and FPGA parallelism. However, it needs some extra cycles to initialize system and set up computational environment when the computation starts.

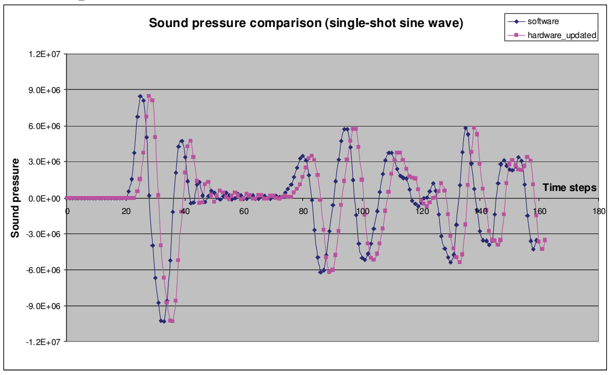

Fig. 5 Sound pressure comparison (single-shot sine wave)

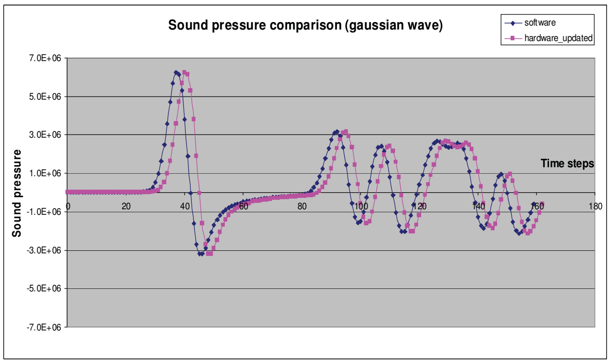

Fig. 6 Sound pressure comparison (Gaussian wave)

VI. Conclusion

Analyzing the distribution of sound field in a room is a computationally intense and data intense work. This paper proposed a hardware solution based on the DHM and FPGA technology to speed up the computation. As a result, both schemes of the DHM are implemented by Xilinx FPGA, and the simplified scheme consumes much smaller hardware resources than the original scheme. The simulation results achieved by hardware match well with the analytical results except having three cycles delay.

In a 2D sound space with 35 × 35 nodes and surrounded by rigid walls, the computational time in the designed hardware system running at 50MHz is much smaller than that in the software implementation running on a computer with an AMD Phenom 9500 Quad-core processor. DHM and its FPGA implementation achieve significant speedup for the sound field analysis due to its simplicity and FPGA parallelism.

Acknowledgment

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

References

[1] A. Krokstad, S. Strom, and S. Sorsdal, "Calculating the acoustical room response by the use of a ray tracing technique", *Journal of Sound Vibration*, vol. 8, no. 1 pp. 118–125, 1968.

[2] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics", *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[3] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, et al, "A beam tracing method for interactive architectural acoustics", *Journal of the Acoustical Society of America*, vol. 115, pp.739–756, 2004.

[4] E. M. Nosal, M. Hodgson, and I. Ashdown, "Investigation of the validity of radiosity for sound field prediction in cubic rooms", *Journal of the Acoustical Society of America*, vol. 116, pp. 3505–3514, 2004.

[5] T. Lewers, "A combined beam tracing and radiant exchange computer model of room acoustics", *Applied Acoustics*, vol. 38, pp. 161–178, 1993.

[6] J. N. Reddy, *An Introduction to the Finite Element Method*, 3rd ed., McGraw-Hill, USA, 2006.

[7] P. K. Banerjee, and R. Butterfield, *The Boundary Element Methods in Engineering Science,* McGraw-Hill, USA, 1994.

[8] D. Botteldooren, "Acoustical finite-difference time-domain simulation in a quasi-cartesian grid", *Journal of the Acoustical Society of America*, vol. 95, no. 5, pp. 2313–2319, 1994.

[9] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems", *Journal of the Acoustical Society of America*, vol. 98, no. 6, pp. 3302–3308. 1995.

[10] Y. Kagawa, T. Tsuchiya, B. Fujii, and K. Fujika, "Discrete Huygens' model approach to sound wave propagation", *Journal of Sound and Vibration*, vol. 218, no. 3, pp. 419–444, 1998.

[11] T. Tsuchiya, and Y. Kagawa, "Digital equivalent circuit for acoustic field based on Discrete Huygens' modeling", *Japanese Journal of Applied Physics*, vol. 44, no. 6B, pp. 4297-4300, 2005.

[12] T. Tsuchiya, "Numerical simulation of sound wave propagation with sound absorption using digital Huygens' model", *Japanese Journal of Applied Physics*, vol. 46, no. 7B, pp.4809-4812, 2007.

[13] D. D. Cogan, *Transmission Line Matrix Techniques for Diffusion Applications,* Gordon and Breach Science Publishers, 1998.

[14] C. Christopoulos, *The Transmission Line Modeling Method*, IEEE Press, 1995.

[15] G. R. Campos and D. M. Howard, "On the computational efficiency of different waveguide mesh topologies for room acoustic simulation", *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5 pp. 1063-1072, 2005.

[16] S. A. Van Duyne, and J. O. SMITH, "The 2D digital waveguide mesh", In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1993, pp. 177–180.

[17] Tan Yiyu, Yukinori Sato, Eiko Sugawara, Yasushi Inoguchi, Makoto Ohya, Yukio Iwaya, Hiroshi Matsuoka, Takao Tsuchiy, "A real-time sound field renderer based on digital Huygens' model", submitted to *Journal of Sound and Vibration.*

[18] Takao Tsuchiya, Eiko Sugawara and Yasusi Inoguchi, "On the numerical simulation of sound wave propagation using field programmable gate array device", *Journal of the Acoustical Society of America,* vol. 120, no. 5, pp. 3218-3218, 2006.