Twenty Thousand Parallel Special-purpose Computer for Phase-type Electroholography Using the Hilbert Transform

Yota Yamamoto[1], Tomoyoshi Shimobaba[2], Yukinobu Taniguchi[1], and Tomoyoshi Ito[2]

yy-yamamoto@rs.tus.ac.jp
[1]Faculty of Engineering, Tokyo University of Science, 6-3-1 Niijuku, Katsushika-ku, Tokyo 125-8585, Japan
[2]Graduate School Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
Keywords: Holography, 3D display, Special-purpose Computer, Field-programmable gate array

ABSTRACT

Electroholography is gaining popularity as an ideal three-dimensional display technology, but it requires high-performance computers for practical use. In this paper, we used the Hilbert transform to build a special-purpose computer for phase-only electroholography. As a result, we succeeded in speeding up calculation 400 times faster than commercial CPUs.

1 Introduction

Electroholography [1] has been gaining attention as an ideal three-dimensional (3D) display technology because it can reproduce all the depth cues. Thus, electroholography can overcome 3D sickness, also known as vergence-accommodation conflict (VAC) [2], which is caused by the lack of depth cues in traditional stereoscopic displays (such as head-mounted displays). Computer-generated holograms (CGHs), which are recorded interference patterns with three-dimensional information, are used in electroholography. CGH is generated by computer calculation, and it is necessary to develop a system capable of rapidly computing a large amount of 3D information [3].

This study developed special-purpose computers for electroholography using field-programmable gate arrays (FPGAs) [4,5] to reduce computation time. CGH is classified into two types: amplitude and phase. The amplitude type has a lower computational cost than the phase type and can be computed faster. On the contrary, although the computational cost of the phase type is high, it can produce brighter and better-reproduced images than the amplitude type [6]. We have been developing specialized computers for both methods. However, previous work discovered that phase-type special-purpose computers have nearly half the computational performance of amplitude types due to the increased size of the required circuits [4,5].

This paper used the Hilbert transform to create a phase-type CGH calculation circuit [7]. The scale of the circuit was greatly reduced using the Hilbert transform, and we were able to create a special-purpose computer with ~20,000 parallel cores. The performance is 400 times faster than that of a CPU (Intel Core i9-9900K) and 8.5 times faster than that of a GPU (NVIDIA RTX 2080 Ti). We also describe the architecture and computational performance of the developed computer.

2 Calculation Method

We used CGH based on point cloud 3D models. In M points cloud, the formula under the condition of $\mathrm{z_j} \gg \mathrm{x_j}, \mathrm{y_j}$ is expressed as

$$u_c(x_a, y_a) = \sum_{j=1}^{M} A_j \exp(i2\pi\theta_{aj}), \quad (1)$$

$$\theta_{aj} = \rho_j(x_{aj}^2 + y_{aj}^2), \quad (2)$$

where $\rho_j = 1/(2\lambda|z_j|)$, $x_{aj} = x_a - x_j$, $y_{aj} = y_a - y_j$, x_a, y_a is coordinates on the CGH, x_j, y_j, z_j represents the coordinates of the point cloud. A_j is the amplitude intensity of the point cloud (fixed to 1), and λ is the reference light's wavelength.

The CGH obtained using Eq. (1) and Eq (2) is called a complex hologram. Complex holograms cannot be displayed at once in commercial displays but must be displayed with phase or amplitude distributions selected.

In the case of phase-type holograms, the phase distribution is extracted using the following equation:

$$u_p(x_a, y_a) = \tan^{-1}\frac{\mathrm{Im}\{u_c\}}{\mathrm{Re}\{u_c\}}, \quad (3)$$

where $\mathrm{Re}\{u_c\}$ and $\mathrm{Im}\{u_c\}$ are functions that extract the real and imaginary parts from the complex distribution, respectively.

Amplitude holograms can be created by extracting the real part using $\mathrm{Re}\{\mathrm{u_c}\}$. However, due to the computational cost, it can be calculated directly using Eq. (4) instead of Eq. (1), Eq. (2) and $\mathrm{Re}\{\mathrm{u_c}\}$.

$$u_a(x_a, y_a) = \sum_{j=1}^{M} A_j \cos[2\pi\theta_{aj}]. \quad (4)$$

2.1 Recurrence Relation Algorithm

There is a simple calculation method for FPGA implementation named recurrence relation algorithm [8]. Here we defined

$$\Gamma_{\mathrm{j}} = \frac{1}{\lambda \mathrm{z_j}} = 2\rho_{\mathrm{j}}, \quad (5)$$

$$\Delta_{0j} = \rho_j\{2(x_0 - x_j) + 1\}. \quad (6)$$

In the recurrence relation algorithm, initially Eq. (2) as θ_{0j} is calculated. In the n-th θ_{nj} in the x-axis direction, θ_{nj} is formulated with the recurrence relation algorithm as follows:

$$\theta_{nj} = \theta_{(n-1)j} + \Delta_{(n-1)j}. \qquad (7)$$

Also, we update Δ_{0j} value using the following equation:

$$\Delta_{nj} = \Delta_{(n-1)j} + \Gamma_j. \qquad (8)$$

We can calculate θ_{nj} by simply repeating simple Eq. (7) and Eq. (8).

2.2 Hilbert Transform

The Hilbert transform using the 1D Fast Fourier Transform (FFT) is expressed as:

$$\hat{h}(x) = \mathrm{FFT}^{-1}[\mathrm{FFT}[u_a(x)]H(f)], \qquad (9)$$

$$\mathrm{H}(f) = \begin{cases} 1\ (f = 0) \\ 1/2\ (f < W) \\ 0\ (otherwise) \end{cases} \qquad (10)$$

FFT and FFT^{-1} denote the forward and reverse transformations of the 1D FFT, respectively. W denotes the width of the image. The Hilbert transform can generate complex holograms from amplitude-type CGH. From the complex hologram obtained using the Hilbert transform, the phase CGH can be calculated by extracting the phase distribution using Eq. (3).

At first glance, this may appear to be a redundant procedure. However, when implemented on an FPGA, the Hilbert transform circuit requires fewer resources than a circuit that directly calculates Eq. (1). Furthermore, as the Hilbert transform is independent for each line of the CGH, the amplitude-type CGH and the Hilbert transform can be computed concurrently. The computation time for the Hilbert transform can eventually be hidden. As a result, a FPGA-based special-purpose circuit can compute the phase-type CGH faster than it can compute Eq. (1).

3 Implementation of a Special-purpose Calculation Circuit

For the special-purpose computer implementation, we used the Xilinx Alveo U250 (U250). As an expansion board, the U250 is connected to a PC. The U250 implements phase-type CGH calculation circuits that use the Hilbert transform as a calculation accelerator from the PC. Figure 1 depicts the block diagram of the developed special-purpose computer.

In Fig. 1, the recurrence relation unit (RRU) is an amplitude-type CGH calculation circuit that uses the recurrence relation algorithm. The Hilbert transform unit performs the Hilbert transform that reads CGH pixel data from the CGH RAM and writes it back to the CGH RAM after the RRU finishes calculating one CGH line. In this paper, 10 RRUs are installed in the FPGA; thus, ~20,000 (1,920 (one RRU) × 10 Units) pixels can calculate in parallel.

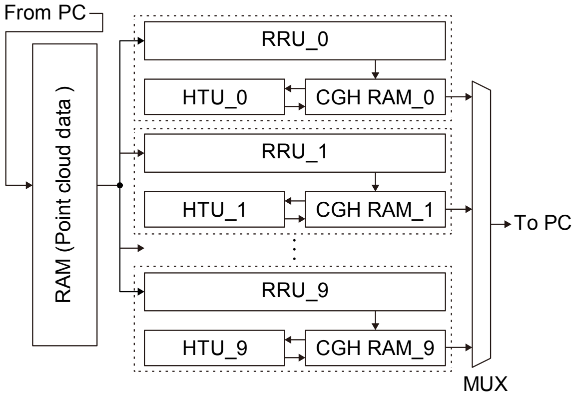

Fig. 1 Block diagram of a phase-type CGH calculation circuit using Hilbert transform, where MUX stands for multiplexer.

4 Results

A comparison of the time required to calculate a phased CGH of 1,920×1,080 pixels from 50,000 points is shown in Table 1.

Table 1 Comparison of computation time. fps indicates frames per second.

Calculation Hardware	Calculation time [ms]	fps
U250: with the Hilbert transform (this work)	27	37
U250: without the Hilbert transform	58	17
GPU (NVIDIA RTX 2080 Ti)	230	4.4
CPU (Intel Core i9-9900K)	11,014	0.091

From Table 1, we achieved a speedup of 400 times compared to the CPU and 8.5 times faster than the GPU. Also, we succeeded in speeding up 2.1 times faster than our previous method, without the Hilbert transform but calculating Eq. (1).

5 Discussion and Conclusions

In this study, we used the Hilbert transform to create a special-purpose computer for phase-type electroholography. As a result, we succeeded in implementing ~20,000 parallel special-purpose computers. Furthermore, we achieved a speedup of 400 times compared to the CPU, 8.5 times faster than the GPU, and 2.1 times faster than our previous method [5].

We plan to build a computation cluster using multiple FPGAs in the future. We would also like to build a computer system for CGH that has 4K and 8K pixels.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Numbers 19H01097 and 21K21294.

References

[1] P. S. Hilaire, S. A. Benton, M. E. Lucente, M. L. Jepsen, J. S. Kollin, H. Yoshikawa, J. Underkoffler,

"Electronic display system for computational holography," Proc. SPIE, Vol. **1212**, pp. 174-182 (1990).
[2] G. Kramida, "Resolving the vergence-accommodation conflict in head-mounted displays," IEEE Trans. Vis. Comput. Graph., Vol. **22**, No. 7, pp. 1912-1931 (2016).
[3] M. Lucente, "Interactive three-dimensional holographic displays: seeing the future in depth," ACM Trans. Graph., Vol. **31**, pp. 63-67 (1997).
[4] T. Sugie, T. Akamatsu, T. Nishitsuji, R. Hirayama, N. Masuda, H. Nakayama, Y. Ichihashi, A. Shiraki, M. Oikawa, N. Takada, Y. Endo, T. Kakue, T. Shimobaba, T. Ito, "High-performance parallel computing for next-generation holographic imaging," Nat. Electron., Vol. **1**, pp. 254-259 (2018).
[5] T. Nishitsuji, Y. Yamamoto, T. Sugie, T. Akamatsu, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, T. Ito, "Special-purpose computer HORN-8 for a phase-type electro-holography," Opt. Express, Vol. **26**, Issue 20, pp. 26722-26733 (2018).
[6] L. B. Lesem, P. M. Hirsch, J. A. Jordan, "The kinoform: a new wavefront reconstruction device," IBM J. Res. Dev., Vol. **13**, No. 2, pp. 150-155 (1969).
[7] T. Shimobaba, T. Kakue, Y. Yamamoto, I. Hoshi, H. Shiomi, T. Nishitsuji, N. Takada, T. Ito, "Hologram generation via Hilbert transform," OSA Contin., Vol. **3**, pp. 1498-1503 (2020).
[8] T. Shimobaba, T. Ito, "An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition," Comput. Phys. Commun., Vol. **138**, No. 1, pp. 44-52 (2001).