

組み込み機器向け リアルタイム電子ホログラフィ専用計算機の開発

山本 洋太^{1,a)} 増田 信之² 角江 崇¹ 下馬場 朋禄¹ 伊藤 智義¹

概要：理想的な3次元映像提示手法として電子ホログラフィに対する期待が高まっている。両眼視差方式などの方式と比べ、すべての奥行き手がかりを再現可能であり、観察者に対して負担をかけない特徴を持つ。しかし、電子ホログラフィでは、計算によって作成される Computer-Generated Hologram (CGH) を用いるが、この計算量は膨大であり、実用化へ向けて大きな障壁となっている。リアルタイムにデータを処理し、表示を行うことが困難である。本研究では、ARM CPU と FPGA をワンチップに搭載した Xilinx Zynq UltraScale+ MPSoC シリーズを利用し、リアルタイム電子ホログラフィ実現へ向けた専用計算機の開発を行った。

1. 背景

理想的な3次元映像提示手法として電子ホログラフィに対する期待が高まっている。両眼視差方式などの方式と比べ、すべての奥行き手がかりを再現可能であり、観察者に対して負担をかけない特徴を持つ [1]。電子ホログラフィでは、計算によって作成される Computer-Generated Hologram (CGH) を用いるが、この計算量は膨大であり、実用化へ向けて大きな障壁となっている。リアルタイムにデータを処理し、表示を行うことが困難である。

また、CGH を表示するディスプレイである Spatial Light Modulator (SLM) の解像度とサイズは、現状では十分ではなく、再生可能な3次元像は数 cm 角と非常に小さい。従って、3次元 TV といった大型な画面で電子ホログラフィを活用することは、現状では難しい。しかし、Head Mounted Display (HMD) では、目に近い位置に SLM を設置できるため、SLM のサイズ不足を補うことができる。計算負荷の問題を解決できれば、電子ホログラフィを活用することが可能である。

ホログラフィを利用した HMD の研究としては、光学系を小型に実装するなどの研究が行われている [2]。計算時間の削減を行ったシステムでは大型なものが多く [3,4]、小型なシステムが求められる。そこで、私たちは、HMD 向けの小型な電子ホログラフィ専用計算機の開発を行ってき

た [5]。

本研究では、ARM CPU と FPGA をワンチップに搭載した Xilinx Zynq UltraScale+ MPSoC シリーズを利用し、リアルタイム電子ホログラフィ実現へ向けた組み込みシステム向け専用計算機の開発を行った。

2. 原理

2.1 CGH

3次元物体を M 個の点群で表現すると、CGH 上の各画素は、 $z_j \gg x_j, y_j$ の条件下において、式 (1)、式 (2) で表される。

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

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

ここで、 $\rho_j = 1/2\lambda|z_j|$ 、 $x_{aj} = x_a - x_j$ 、 $y_{aj} = y_a - y_j$ である。 $I(x_a, y_a)$ は、 M 個の点群からの光強度の足し合わせによって表現される。実際に CGH を SLM に表示する際、 $I(x_a, y_a)$ は結果の正負により、0 か 255 の画素値に量子化して表示を行う。 A_j は点群の振幅強度、 λ は参照光の波長である。CGH の画素数を K とすると、 KM に比例する計算量が必要になることがわかる。

2.2 漸化式

CGH を表示する SLM の画素間隔は、一般的に非常に細かく (5 μm 程度)、均等に並んでいる。そのことを利用し、隣接する画素に関し近似式を適用し、計算を簡略化した手法に漸化式法がある [6]。ここで、新たな変数として、

¹ 千葉大学
Chiba University

² 東京理科大学
Tokyo University of Science

a) y-yamamoto@chiba-u.jp

$$\Delta_{\alpha j} = \rho_j(2x_{\alpha j} + 1) \quad (3)$$

$$\Gamma_j = \frac{1}{\lambda z_j} = 2\rho_j \quad (4)$$

を導入する。 $\theta_{\alpha j}$ を元に、 x 軸方向の n 番目の $\theta_{(\alpha+n)j}$ を近似で求めるとき、

$$\theta_{(\alpha+n)j} = \theta_{(\alpha+n-1)j} + \Delta_{\alpha j} + (n-1)(2\Gamma_j) \quad (5)$$

と表される。式 (5) は加算とビットシフトによる乗算のみで計算可能であるため、FPGA 実装したときに、リソースの消費を抑え、並列数を増加させ、高速化できる利点がある。

3. 使用ハードウェア

リアルタイム電子ホログラフィ専用計算機開発の初期段階として、Xilinx 社が提供する Zynq UltraScale+ MPSoC ZCU102 評価キット (以下、ZCU102 と呼ぶ) を用いた。ZCU102 は、ARM CPU と FPGA をワンチップに搭載搭載した System-on-a-chip (SoC) である。搭載されている論理 FPGA の仕様 (表 1, 表 2) をそれぞれ示す。

表 1 ZCU102 の仕様。

搭載ボード	Zynq UltraScale+ MPSoC ZCU102 評価キット
ファミリ名	Zynq UltraScale+ MPSoC
デバイス	ZU9EG-2FFB1156I

表 2 ZCU102 の論理回路リソース。

Resource	Available
LUT	274080
LUTRAM	144000
FF	548160
RAM	912
DSP	2520
BUFG	404

また、ARM CPU の環境を表 3 に示した。本研究のシステムでは、ARM CPU で Ubuntu 18.04.1 LTS 64bit (Linux Kernel 4.14.0) を動作させた。OS 上では、点群データの管理、FPGA の制御といった役割を行う。ARM CPU は、一般的な PC と比べ処理性能は低いですが、汎用的に動作する環境となっている。アプリケーションとして拡張性が期待できる。

表 3 評価環境。

CPU	Quad Core Cortex-A53 1200MHz
Memory	4GB
OS	Ubuntu 18.04.1 LTS 64bit
Compiler	gcc (Ubuntu/Linaro 7.3.0)

4. リアルタイム電子ホログラフィ計算システム

4.1 計算システム全体

リアルタイム電子ホログラフィ計算システムとして、図 1 に示すシステムの開発を行った。

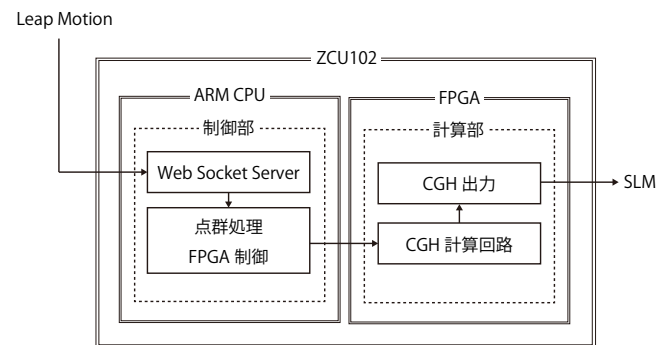


図 1 リアルタイム電子ホログラフィ計算システムのブロック図。

図 1 中において、機能の役割は制御部と計算部に分けた。制御部は、ARM CPU を中心として動作させた。ARM CPU を用いて、FPGA の制御、CGH を計算するための点群データの管理などを行わせた。計算部は、FPGA を中心として動作させた。FPGA で CGH 計算に特化した専用計算回路を構築し、計算に利用することで CGH の計算にかかる時間の削減を行った。ARM CPU で点群データを処理し、FPGA に構築した専用計算回路を利用し計算を行い、CGH の表示を行うシステムとなっている。

4.2 制御部 (ARM CPU 部分)

ARM CPU を中心に動作する制御部に関して詳細に述べる。ARM CPU では、FPGA 制御のために 2 つのスレッドを動作させた。

1 つ目のスレッドでは、CGH をインタラクティブに変化させるためのデータの取得を行わせた。データ取得のため、Web Socket サーバが動作し、Wi-fi を利用し、ネットワーク越しに、座標データの取得を行う。今回は、リアルタイム性の検証のため、Leap Motion からデータを取得し、インタラクティブに CGH を計算し、表示するシステムを開発した。Leap Motion は別に用意した PC に接続し、Web Socket を用いて Leap Motion から座標データを取得し、その座標値を元にインタラクティブに CGH を計算するアーキテクチャとした。

2 つ目のスレッドでは、FPGA に構築した専用計算機の制御を行わせた。点群の転送、1 つ目の Web Socket サーバスレッドが取得した座標値の転送、FPGA の計算制御を行う。

このように、ARM CPU では、2 つのスレッドを利用し、

制御全般を行うシステムを開発した。Leap Motion 用のドライバが今回利用している，ARM CPU 向けには提供されていなかったため，別に用意した PC に Leap Motion を接続し利用した。しかし，Web Socket を利用し，取得したデータを処理し，表示する部分までは，スタンドアロンのシステムの開発を行った。

4.3 専用計算機システム (FPGA 部分)

FPGA を中心に動作する計算部に関して詳細に述べる。FPGA は，主に CGH の計算と CGH の表示を行う。

CGH の計算としては，式 (5) を用いて図 2 に示す専用計算機を FPGA に実装した。図 2 中，Basic Phase Unit (BPU) は，式 (2) の計算を行うユニットである。Additional Phase Unit (APU) は，式 (5) の計算を行うユニットである。Multi Phase Unit (MPU) は，1 個の BPU と 1919 個の APU を内包した CGH 上の横 1 ライン分の画素を計算するユニットである。Quantization Unit (QU) は，各点群からの光強度の足し合わせを行い，結果の正負から 0 か 255 の画素値を出力するユニットである。漸化式を利用し，並列数を増やし，計算時間を削減するアーキテクチャとした。

CGH の表示は，OS が確保した物理メモリ上のフレームバッファへ FPGA から直接書き込むことによって CGH の表示を行った。CPU による仲介は行わないため，画像表示のための遅延は発生しないようになっている。

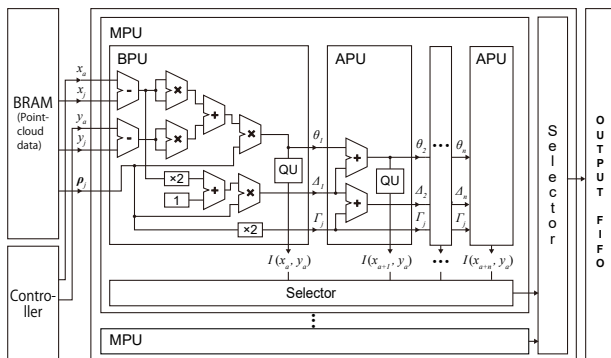


図 2 専用計算機のブロック図。

5. 評価

5.1 計算時間の比較

計算時間比較のため，デスクトップ PC の CPU (Intel Xeon) と組み込み機器向けの GPU (NVIDIA Jetson TX1) において CGH を求めるプログラムを作成し，計算時間の比較を行った。65,000 点の点群から $1,920 \times 1,080$ 画素の CGH 1 フレームを計算するのにかかった時間を表 4 に示した。

比較対象の CPU は Intel Xeon CPU E5-2697 v2 2.70 GHz，メインメモリ 64GB の CentOS Linux release

表 4 計算時間の比較。

計算ハード	計算時間 [s]	fps	高速化比 [倍]
FPGA	0.099	10	164
NVIDIA Jetson TX1	12.978	0.077	1.25
Intel Xeon	16.252	0.061	1.00

7.1.1503 (Core) redhat が動作するシステムである。コンパイラは，Intel C compiler 16.0.1.150 を用い，すべてのコアを利用し，並列で実行した。GPU 環境は，組み込み機器向けの GPU である NVIDIA Jetson TX1 を利用し，CUDA 8.0 を用いて実装した。

表 4 より，CPU と比較して 164 倍，組み込み機器向けの GPU と比較して 131 倍の高速化を達成した。

また，式 (2) のみを用いた先行研究 [5] のシステムと比べ，6.7 倍の処理速度向上に成功した。

5.2 フレームレート評価

フレームレート評価のため，図 3 に示す光学系に今回開発したリアルタイム電子ホログラフィ計算システムを接続した。65,000 点の点群から，単位時間あたりに処理可能な CGH のフレーム数を ARM CPU で作成したプログラムから数えることにより，システム全体の fps の評価を行った。実際にインタラクティブに動かしている様子を図 4 に示した。その結果，60s あたり，600 フレームの CGH を処理できていた。

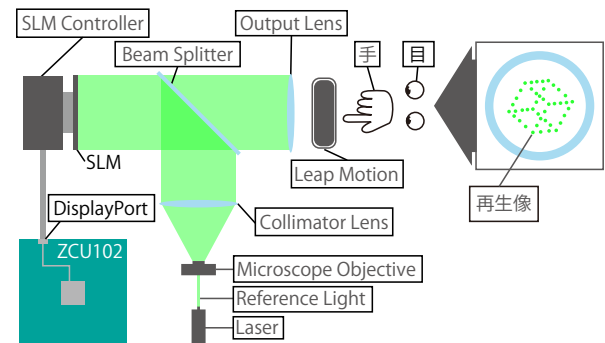


図 3 光学系の様子。

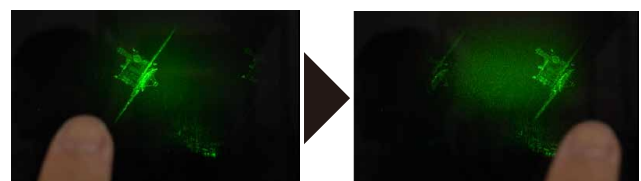


図 4 インタラクティブに動かしている様子。指の動きに合わせて再生像が移動している。

表 5 システム全体のリソースの使用率 (%).

	FF	LUT	BRAM	DSP
Total [%]	38.55	64.39	9.10	76.47

5.3 リソース使用率

リソースの使用率について表 5 に示した.

また, 動作周波数は 375MHz, 並列数は 3840 並列という結果となった.

5.4 再生像の評価

65,000 点の点群の様子と, その 3D データから CPU, FPGA を用いて計算した CGH の再生像を図 5 に示した. 図 5 より, CPU, FPGA において, 肉眼では画質の差が確認できなかった.

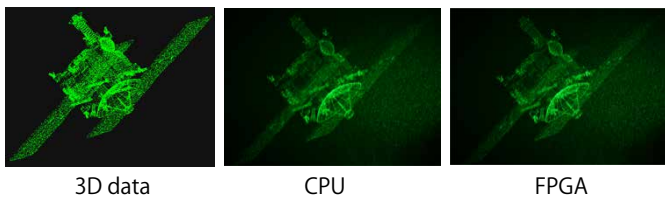


図 5 再生像の様子.

6. まとめと今後の展望

本研究では, 小型なリアルタイム電子ホログラフィ専用計算機の開発を行った. その結果, 漸化式法を利用し, 65,000 点の点群を 10fps でインタラクティブに再生可能なシステムの開発に成功した. 同等の画質を持つ再生像で, デスクトップ向けの CPU や組み込み機器向けの GPU と比べ 100 倍以上の高速化を達成した.

今後は, より応用的なシステムとして, SLM など光学系と計算システムが一体化したシステムの開発を行っていきたい.

参考文献

- [1] G. Kramida, “Resolving the vergence-accommodation conflict in head-mounted displays,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1912–1931, July 2016.
- [2] E. Murakami, Y. Oguro, and Y. Sakamoto, “Study on compact head-mounted display system using electroholography for augmented reality,” *IEICE Transactions on Electronics*, vol. E100.C, no. 11, pp. 965–971, 2017.
- [3] H. Niwase, N. Takada, H. Araki, H. Nakayama, A. Sugiyama, T. Kakue, T. Shimobaba, and T. Ito, “Real-time spatiotemporal division multiplexing electroholography with a single graphics processing unit utilizing movie features,” *Opt. Express*, vol. 22, no. 23, pp. 28 052–28 057, Nov 2014.
- [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, and T. Ito, “High-performance parallel computing for next-generation holographic imaging,” *Nature Electronics*, vol. 1, pp. 254–259, 04 2018.

- [5] Y. Yamamoto, N. Masuda, R. Hirayama, H. Nakayama, T. Kakue, T. Shimobaba, and T. Ito, “Special-purpose computer for electroholography in embedded systems (in press),” *OSA Continuum*.
- [6] T. Shimobaba and T. Ito, “An efficient computational method suitable for hardware of computer-generated hologram with phase computation by addition,” *Computer Physics Communications*, vol. 138, no. 1, pp. 44 – 52, 2001.