

# 漸化式を用いた組み込み機器向け 電子ホログラフィ専用計算機の開発

山本 洋太<sup>†</sup> 増田 信之<sup>\*</sup> 角江 崇<sup>†</sup> 下馬場 朋禄<sup>†</sup> 伊藤 智義<sup>†</sup>  
千葉大学大学院工学研究院<sup>†</sup> 東京理科大学基礎工学部<sup>\*</sup>

## 1. まえがき

理想的な3次元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。

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。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 [2, 3]、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 [4]。

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、ハードウェア実装に最適化された漸化式法 [5] を用いて小型な電子ホログラフィ専用計算機の開発を行った。

## 2. 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。 $A_j$  は点群の振幅強度、 $\lambda$  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。

## 3. 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 ( $5\mu\text{m}$  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。ここで、新たな変数として、

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

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

を導入すると  $\theta$  を既知としたとき、隣り合う画素の  $\theta$  は、

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

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。

## 4. FPGA を用いた専用計算機システム

本研究では、式 (5) を用いて図 1 に示す専用計算機を FPGA に実装した。図 1 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なっている。

## 5. 評価

### 5.1 評価環境

開発した専用計算機の環境を表 1 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) が動作している。OS 上では、点群データの管理、FPGA の制御、CGH の画面出

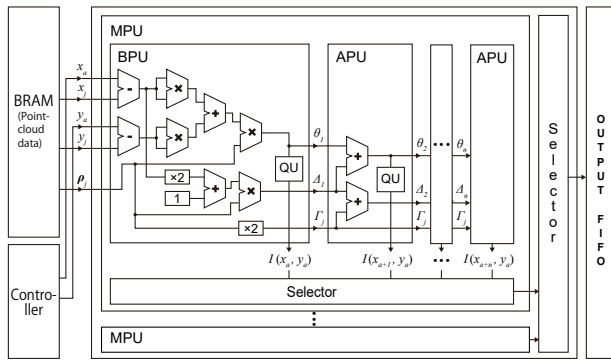


図 1: 専用計算機のブロック図.

力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、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 1: 評価環境.

ファミリ	ZU9EG-2FFB1156I
CPU	Quad Core Cortex-A53 1200MHz
Memory	4GB
OS	Ubuntu 18.04.1 LTS 64bit
Compiler	gcc (Ubuntu/Linaro 7.3.0)

### 5.2 計算時間

計算時間比較のため、デスクトップ PC の CPU と組み込み機器向けの GPU 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 1,920×1,080 画素の CGH を計算するのにかかった時間を表 2 に示した。

表 2: 計算時間の比較.

計算ハード	計算時間 [s]	理論値性能 [TFLOPS]	fps	高速化比 [倍]
FPGA	0.099	-	10	164
GPU: NVIDIA Jetson TX1	12.978	1.0	0.077	1.25
CPU: Intel Xeon	16.252	0.518	0.061	1.00

比較対象の CPU は Intel Xeon CPU E5-2697 v2 2.70GHz、メインメモリ 64GB の CentOS Linux release 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 2 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 [4] のシステムと比べ、6.7 倍の処理速度向上に成功した。

### 5.3 リソース使用率

リソースの使用率について表 3 に示した。

表 3: 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

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

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 2 に示した。図 2 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。

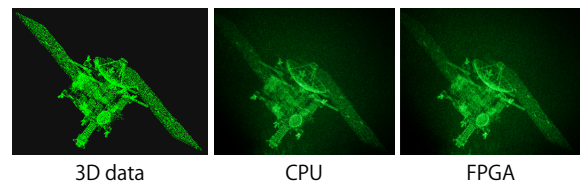


図 2: 再生像の様子.

## 6. まとめと今後の展望

本研究では、小型な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 で実行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 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] Kramida, Y. *et al.*, IEEE Transactions on Visualization and Computer Graphics, Vol. 22, pp. 1912-1931, 2016.
- [2] Niwase, H. *et al.*, Optics Express, Vol. 22, pp. 28052-28057, 2014
- [3] Sugie, T. *et al.*, Nature Electronics, Vol. 1, pp. 254-259, 2018
- [4] Yamamoto, Y. *et al.*, OSA Continuum, (in press)
- [5] Shimobaba, T. *et al.*, Computer Physics Communications, Vol. 138, pp. 44-52, 2001