RC-004

シングル空間光位相変調器用マルチ GPU クラスタシステムによる計算機合成ホログラムの計算高速化

Fast computation of computer-generated hologram using multi-GPU cluster system for a single spatial light modulator

庭瀬 裕章 † 藤原 将人 † 荒木 啓充 † 前田 祐貴 † 中山 弘敬 ¶
Hiroaki Niwase Masato Fujiwara Hiromitsu Araki Yuki Maeda Hirotaka Nakayama
角江 崇 § 下馬場 朋禄 § 伊藤 智義 § 高田 直樹 ‡
Takashi Kakue Tomoyoshi Shimobaba Tomoyoshi Ito Naoki Takada

1. はじめに

現在実用化されている三次元映像は右目と左目用の 2 枚の二次元映像による疑似的な三次元表示である．様々な角度から眺めても正面の立体像しか見ることができない．視覚疲労など健康被害が懸念され長時間の利用は難しい．

一方，ホログラフィは 1947 年に D. Gabor により発明され，三次元物体からの光の波面を忠実に記録・再生できる唯一の技術である．レーザーの発明により大きく進展し，D. Gabor は 1971 年にノーベル賞を受賞している．ホログラフィによる立体像は様々な角度から眺めることができ，視覚疲労もなく長時間利用可能である．このことより，電子化したホログラフィ (電子ホログラフィ) は「究極の立体テレビ」になると考えられている．コンピュータによって作られたホログラムを計算機合成ホログラム (CGH: Computer-Generated Hologram) という．電子ホログラフィは CGH を空間光位相変調器 (SLM: Spatial Light Modulator) に表示し，再生光を SLM に照射することで空中に三次元物体を再生する技術である．しかし，CGH 計算は膨大であるため，未だ実用化されていない．

近年，GPU (Graphics Processing Unit) の浮動小数点演算性能とコストパフォーマンスは著しく向上している．GPU は本来コンピュータグラフィックス用のプロセッサである．CGH 計算は使用するデータ量に比べ演算量が多く並列化に向いている．GPU を用いた電子ホログラフィの研究は盛んに行われている [1,2]．3 枚の GPU ボードを搭載したマルチ GPU 環境の PC と 3 枚の SLM を用いたカラー電子ホログラフィに関する研究も報告されている [3]．複数の SLM とマルチ GPU クラスタシステムによる CGH 計算の高速化についても報告されている [4,5]．しかし，複数の SLM を用いる場合，SLM が非常に高価であり，大規模なシステムとなる．また，光学系の位置調整も容易ではない．解像度 $1,920 \times 1,024$ の CGH 計算を複数の GPU で高速化については未だ実現されていない．SLM の価格を考えれば，1 つの SLM を用いた電子ホログラフィのほうが，コストパフォーマンスもよく，光学系も小規模であり実用的である．

本論文では，10 枚の GPU (NVIDIA GeForce GTX 680) を搭載したマルチ GPU クラスタに SLM を 1 枚接続したシステムを用いた CGH 計算の高速化について述べる．9 枚の GPU ボードを用いて，解像度 $1,920 \times 1,024$ の CGH 計算を本システムにより高速化する方法を提案する．最終的に，提案手法により実効速度約 11 TFLOPS で，CPU (Intel Core i7 930, 8 スレッド使用) に比べて約 980 倍の計算高速化を実現した．

2. 計算機合成ホログラム (CGH)

図 1 に CGH 計算の座標系を示す．三次元物体を点で表し，物体を構成する点数を N_P とする．そのとき，ホログラム面上の点 (x_α, y_α) における光の強度 $I(x_\alpha, y_\alpha)$ は，次式となる [1]．

$$I(x_\alpha, y_\alpha) = \sum_{j=1}^{N_P} \cos\theta \quad (1)$$

$$\theta = \frac{\pi}{\lambda z_j}\left\{(x_\alpha - x_j)^2 + (y_\alpha - y_j)^2\right\} \quad (2)$$

ここで，変数 α はホログラム点を表す．物体点 j の座標を (x_j, y_j, z_j) とした．λ は三次元情報の記録に使用される参照光の波長である．

CGH 上の 1 画素の光の強度を求めるには，式 (2) を $j = 1$ から N_P まで繰り返し計算する必要がある．よって，1 枚の CGH を作成するには，ホログラムの解像度を $W \times H$ とすると，その計算量は $(W \times H) \times N_P$ に比例することになる．

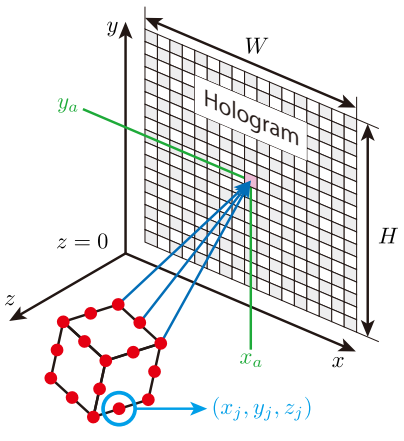

図 1: CGH 計算の座標系

†高知大学大学院総合人間自然科学研究科
‡高知大学教育研究部
§千葉大学大学院工学研究科
¶国立天文台

3. シングル SLM 用マルチ GPU クラスタシステムへの実装

3.1. システム構成

図 2 に本研究で使用したシステムの概略図を示す．本システムでは 4 ノードの PC (PC1〜PC4) からなり，CGH 表示ノード (PC1) と CGH 計算ノード (PC2〜PC4) で構成される．CGH 計算ノードは，各ノードに 3 枚の GPU ボードを搭載したマルチ GPU 環境の PC である．各 GPU で CGH を計算し，計算された CGH を CGH 表示ノード (PC1) に転送する．CGH 表示用ノードは 1 枚の GPU ボード (GPU 0) を搭載し，これに SLM が接続されている．CGH 計算ノードから受け取った CGH は CGH 表示ノードの GPU ボード (GPU 0) により SLM に描画される．SLM に表示された CGH にレーザー等の参照光を照射することにより，図 3 に示すように再生像が空中に表示される．

3.2. 実装

1 つの SLM に表示する CGH を多数の GPU を用いて高速に計算を行う処理を図 4 に示す．CGH 計算に用いる GPU ボード N 枚と CGH 表示に用いる GPU ボード 1 枚の合計 $N+1$ 枚の GPU ボードを用いる．

図 4 においても，図 3 と同様に CGH 表示に用いる GPU ボードを GPU 0 とする．また，CGH 計算に用いる N 枚の GPU ボードを GPU 1〜GPU N とする．1 枚の GPU ボードで三次元動画の 1 つのフレームの CGH を計算する．

図 4 のように，三次元動画の最初のフレーム (Frame 1) の CGH を GPU 1 で計算する．次に，2 番目のフレーム (Frame 2) の CGH を GPU 2 で計算する．同様に，N 番目のフレームの CGH を GPU N で計算する．$N+1$ 番目のフレーム (Frame $N+1$) の CGH は GPU 1 で CGH を計算する．それ以降のフレームの CGH についても同様に各 GPU ボードに割り当てて計算を行う．各 GPU で計算された CGH は GPU 0 へ送る．

GPU 0 は Frame 1 の CGH を受け取り次第，直ちに SLM に描画する．Frame 1 の CGH を SLM に一定時間表示した後，GPU 0 は GPU 2 で計算された Frame 2 の CGH を受け取り SLM に表示する．Frame 2 の CGH を SLM に一定時間表示した後，GPU 0 は次の Frame 3 の CGH を GPU 3 より受け取り同様に処理を行う．これを繰り返し，図 4 に示すように処理がなされる．図 4 の処理を図 3 に示すシステムに実装した．図 4 の処理を実現するため MPI (Message Passing Interface) を使用した．

すべての CGH 計算ノードにおいて，CGH 計算に使用する三次元物体上の物体点位置座標データを取得できるように NFS (Network File System) を使用した．

3.3.GPU 間の CGH 転送

CGH によるリアルタイム動画再生を実現するためには，1 秒間に 30 フレームの CGH を表示しなければならない．つまり，図 4 に示す時間間隔 T が約 33 ms 以内で行われなければならない．前節に示された処理を用いた場合，GPU 間での通信速度がボトルネックとなる可能性が高い．そのため，GPU 間の CGH 転送時間

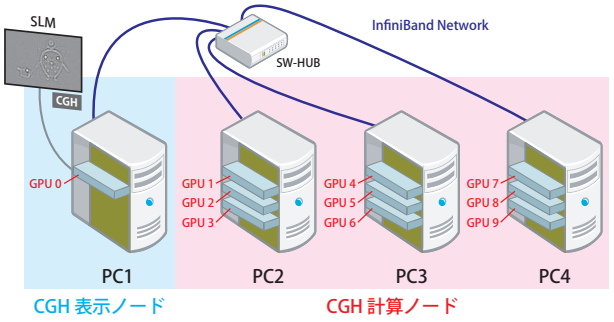

図 2: シングル SLM 用マルチ GPU クラスタシステム

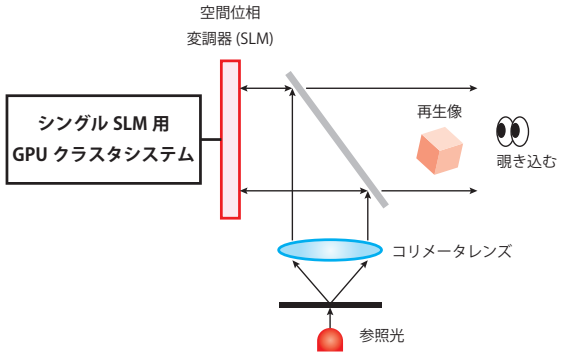

図 3: CGH による三次元映像の再生

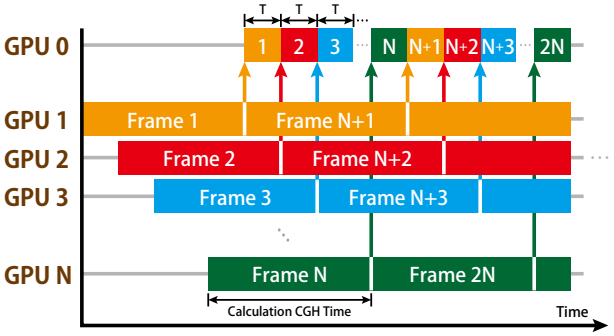

図 4: 本システムで行われる CGH 計算処理

について検討する必要がある．

前節に示す計算処理において，GPU 間で転送する CGH データは 1 画素あたり 4 バイトである．CGH の解像度を $1,920 \times 1,024$ 画素とし，ネットワークに Gigabit Ethernet を使用した場合，1 枚の CGH データ転送時間は次のようになる．

$$32[\mathrm{bit}] \times 1,920[\mathrm{pixel}] \times 1,024[\mathrm{pixel}] \div 1[\mathrm{Gbps}] \approx 63[\mathrm{ms}] \quad (3)$$

本システムを用いてリアルタイム三次元動画再生を実現するためには高速なネットワークが必要となる．そこで，本研究では高速なネットワークとして InfiniBand QDR を用いる．InfiniBand QDR では理論性能は 40 Gbps である．仮に実効速度が約 20 Gbps であっても 1 枚の CGH の転送時間は約 3 ms となりリアルタイム三次元動画再生が可能である．

表 1: NVIDIA GeForce GTX 680 の仕様

GPU クロック	1,006 MHz
CUDA コア数	1,536 個
メモリ容量	2,048 MB DDR5
メモリバンド幅	192.2 GB/s
最大浮動小数点数演算性能 (単精度)	3.09 TFLOPS

4. 結果

4.1. 性能評価

GPU ボードとして NVIDIA GeForce GTX 680 を使用する (表 1)．表 1 に示す最大浮動小数点数演算性能は理論性能を示している．1 クロックで積和演算を実行できることから，1 クロックで 2 演算行うものとして理論性能を求めた [6]．図 2 に示すシステムにおいて，各ノードのスペックを表 2 に示す．ネットワークとして InfiniBand QDR (HCA: Mellanox MHQH19B-XTR, HUB: Mellanox MIS5031Q-1SFC) を使用した．OS として CentOS 6.3 (x86_64)，開発環境として CUDA 5.0 SDK (Software Development Kit) [7]，OpenGL，MPI ライブラリとして Open MPI v1.7.2 を用いた．

表 3 に，本手法による $1,920 \times 1,024$ 画素の CGH を GPU 0 で表示する時間間隔 T (図 4) を示す．図 2 に示された CGH 計算ノードの GPU ボードの枚数を示す．三次元物体を構成する物体点数に比例にして時間間隔 T は増加している．なお，Full HD の解像度は $1,920 \times 1,080$ 画素であるが，$1,920 \times 1,024$ 画素を用いたのは GPU の演算性能を十分引き出すためである [5]．表 4 は表 3 の時間間隔 T から換算したフレームレートを示す．CGH 計算ノードの 9 枚の GPU ボードを用いたとき，20,480 点から構成される三次元物体の CGH を 1 秒間に約 33 枚表示できることを示しており，リアルタイムで動画再生できることを意味する．表 3 より求めた本手法による実効性能を表 5 に示す．ここで，式 (2) において，$\pi/\lambda z_j$ をあらかじめ計算しておく．また，cos 関数は GPU の SFU (Special Function Unit) により 1 クロック 1 演算で行われる [7]．このことより，式 (1)，(2) の演算数は 7 演算となり，これを用いて実効性能を求めた．表 5 より，1 枚の GPU ボードで CGH 計算した場合において約 1.4 TFLOPS の演算速度を実現していることがわかる．本手法により計算ノードの 9 枚の GPU ボードを用いたとき，約 11 TFLOPS の演算速度を達成している．表 6 に 1 枚の GPU ボードで計算した場合に対する本手法による高速化を示す．計算ノードの 9 枚の GPU ボードを CGH 計算に使用した場合，1 枚の GPU ボードの速度に対して約 8 倍の高速化を実現している．表 7 に 1 個の CPU (Intel Core i7 930) による CGH 計算時間と本手法を用いた計算ノード 9 枚の GPU ボードによる CGH 計算時間との比較を示す．ここで，表 7 に示す CPU による CGH 計算時間には SLM への CGH 描画処理は含まない．CPU による CGH 計算プログラムは C 言語で作成し，Intel C コンパイラ v14.0.0.1 (最適化オプション: `-O3`) を使用し，Open MP により 8 スレッドで計算した．本手法による計算ノードの 9 枚の GPU ボードを用いた CGH 計算は，CPU に比べ約 980 倍の計算高速化を実現した．

表 2: 各ノードのスペック

CPU	Intel Core i7 930 (Clock speed: 2.80GHz)
メインメモリ	6 GB (2 GB × 3 枚) DDR3-1333
マザーボード	ASUS P6T7 WS SuperComputer

表 3: CGH 表示時間間隔

物体点数	CGH 表示時間間隔 T [ms]			
	1 GPU	3 GPUs	6 GPUs	9 GPUs
10,240	102.1	37.6	21.5	15.1
20,480	203.1	74.5	39.5	30.4
40,960	412.5	146.5	78.6	55.4
61,440	614.2	220.6	120.6	77.8
81,920	825.2	292.6	155.6	103.6
102,400	1030.5	365.6	194.6	130.4

表 4: フレームレート

物体点数	フレームレート [fps]			
	1 GPU	3 GPUs	6 GPUs	9 GPUs
10,240	9.8	26.6	46.6	66.2
20,480	4.9	13.4	25.3	32.9
40,960	2.4	6.8	12.7	18.1
61,440	1.6	4.5	8.3	12.9
81,920	1.2	3.4	6.4	9.7
102,400	1.0	2.7	5.1	7.7

表 5: 実効性能

物体点数	実効性能 (TFLOPS)			
	1 GPU	3 GPUs	6 GPUs	9 GPUs
10,240	1.38	3.75	6.56	9.31
20,480	1.39	3.79	7.14	9.28
40,960	1.37	3.85	7.17	10.18
61,440	1.38	3.83	7.01	10.87
81,920	1.37	3.85	7.25	10.88
102,400	1.37	3.85	7.24	10.81

表 6: 1 GPU に対する複数 GPU の CGH 計算高速化

物体点数	Speed-Up (vs. 1 GPU)		
	3 GPUs	6 GPUs	9 GPUs
10,240	2.72	4.76	6.74
20,480	2.73	5.16	6.69
40,960	2.82	5.24	7.45
61,440	2.78	5.09	7.89
81,920	2.82	5.30	7.96
102,400	2.82	5.30	7.90

表 7: CPU に対する本手法による CGH 計算時間の比較

物体点数	CGH 計算時間		高速化率
	1 CPU [s] (8 threads)	9 GPUs [ms]	
10,240	12.7	15.1	841
20,480	25.4	30.4	836
40,960	51.1	55.4	922
61,440	76.1	77.8	978
81,920	101.4	103.6	979
102,400	126.7	130.4	972

表 8: 計算モデルの再生 (計算ノードの 9 GPU を使用)

計算モデル	物体点数	計算時間 [ms]	フレームレート [fps]
恐竜	11,646	15.1	66.2
チェス	44,647	58.5	17.1
メリーゴーランド	95,949	123.5	8.1

4.2. 提案手法による再生像

3 種類の計算モデルについて提案手法を用いて三次元動画像再生を行った．これらの動画像の再生には，計算ノードの 9 つの GPU を解像度 $1,920 \times 1,024$ の CGH 計算に用いた．3 種類の計算モデルの物体点数，計算時間およびフレームレートを表 8 に示す．

約 1 万点の物体点からなる “恐竜” の計算モデル (図 5(a)) では 66 枚の CGH 計算をすることができ，リアルタイム再生を実現している．この三次元動画像のスナップショットを図 5(b) に示す．約 4 万 5 千点の物体点からなる “チェス” の計算モデル (図 6(a)) では，17 枚の CGH 計算をすることができる．三次元動画像のスナップショットを図 6(b) に示す．約 10 万点の物体点からなる “メリーゴーランド” の計算モデル (図 7(a)) では，8 枚の CGH 計算をすることができる．この三次元動画像のスナップショットを図 7(b) に示す．本手法により鮮明な再生像を得ることができている．

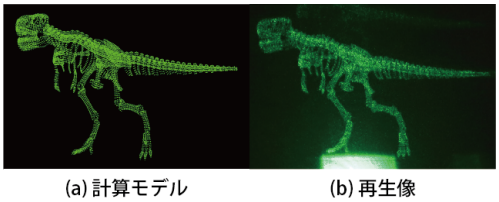

図 5: 再生像 (恐竜: 11,646 点)

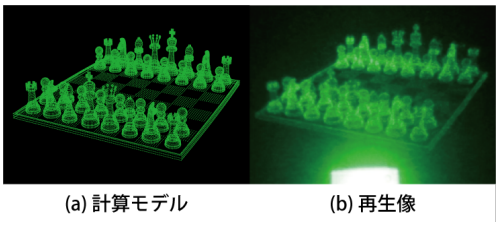

図 6: 再生像 (チェス: 44,647 点)

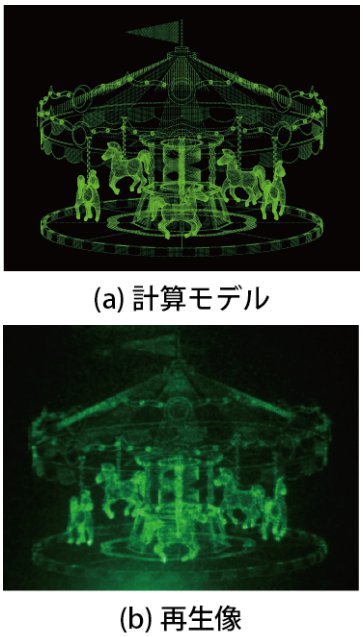

図 7: 再生像 (メリーゴーランド: 95,949 点)

5. まとめ

解像度 $1,920 \times 1,024$ の CGH 計算を多数の GPU を用いて高速に計算する方法提案した．本手法を用いて 9 枚の GPU ボードで CGH 計算したところ，約 2 万点の物体点からなる三次元物体の CGH 計算および再生をリアルタイムで実現できることが確認された．本手法による実効速度は約 11 TFLOPS で，CPU に比べて約 980 倍の計算高速化を達成した．

今後，本手法に残像効果を利用した時空間分割多重法 [8] を適用してさらなる計算高速化を実現する予定である．

謝辞

本研究の一部は，日本学術振興会の科研費・基盤研究 (C) (課題番号 24500071)，基盤研究 (A) (課題番号 25240015) および公益財団法人栢森情報科学振興財団平成 25 年度研究助成によって行われた．

参考文献

[1] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, “Computer generated holography using a graphics processing unit,” Optics Express, 14, pp.603–608 (2006).

[2] A. Shiraki, N. Takada, M. Niwa, Y. Ichihashi, T. Shimobaba, N. Masuda, and T. Ito, “Simplified electroholographic color reconstruction system using graphics processing unit and liquid crystal display projector,” Optics Express, 17, pp.16038–16045 (2009).

[3] H. Nakayama, N. Takada, Y. Ichihashi, S. Awazu, T. Shimobaba, Nobuyuki Masuda and Tomoyoshi Ito, “Real-time color electroholography using multi graphics processing units and multi high-definition liquid-crystal display panels,” Applied Optics, 49, pp.5993–5996 (2010).

[4] 高田直樹, 下馬場朋禄, 中山弘敬, 老川稔, 増田信之, 伊藤智義, “マルチ LCD 用マルチ GPU クラスタシステムによる計算機合成ホログラムの計算高速化,” 第 10 回情報科学技術フォーラム講演論文集, 第 1 分冊, RC–010，pp.117–122 (2011).

[5] N. Takada, T. Shimobaba, H. Nakayama, A. Shiraki, N. Okada, M. Oikawa, N. Masuda and T. Ito, “Fast high-resolution computer-generated hologram computation using multiple graphics processing unit cluster system,” Applied Optics, 51, pp.7303–7307 (2012).

[6] 高田直樹 他, “GPU プログラミング入門 — CUDA5 による実装,” (伊藤智義 編), 講談社 (2013).

[7] NVIDIA, “NVIDIA CUDA Computational unified device architecture programming guide version 5.0,” NVIDIA (2012).

[8] H. Niwase, N. Takada, H. Araki, H. Nakayama, A. Sugiyama, T. Kakue, T. Shimobaba, T. Ito, “Real-time spatiotemporal division multiplexing electroholography with a single graphics processing unit utilizing movie features,” Optics Express, 22, pp.28052–28057 (2014).