RC-010

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

Fast computation of computer-generated-hologram using multi-GPU cluster system for multi-LCD panels

高田直樹† 下馬場朋禄†† 中山弘敬††† 老川稔†† 増田信之†† 伊藤智義††
Naoki Takada Tomoyoshi Shimobaba Hirotaka Nakayama Minoru Oikawa Nobuyuki Masuda Tomoyoshi Ito

1．はじめに

最近，テレビ，映画，ゲーム機など，三次元(3D)立体映像がブームとなっている．しかし，一般的に，右目用と左目用の二次元映像を用いた 3D 表示が中心となっている．一方，ホログラフィ[1]は，三次元物体による物体光を干渉縞として媒体に記録，また，記録された媒体に再生光を照射することで物体光を生成し忠実に三次元物体を再現できる唯一知られた技術である．眼鏡を必要とせず，視覚疲労がないため究極の立体映像技術として期待されている．三次元物体情報を記録した干渉縞をホログラムと呼び，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)[2]という．CGH は，仮想的な三次元物体によるホログラムを作成することができる．CGH を液晶ディスプレイ(LCD：Liquid Crystal Display)などの電子デバイスに表示し，再生光を照射することにより三次元物体を再生することができる(電子ホログラフィ)[3-5]．しかし，CGH は計算量が膨大であり，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おり，GPU を用いた数値計算の高速化に関する研究は盛んに行われている[6-9]．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．GPU で計算した結果を CPU(Central Processing Unit)を介さず，直接，LCD に表示することができる．また，CGH 計算は並列計算に向いていることから，電子ホログラフィの研究に GPU が使用されている[10]．3 枚の LCD パネルを用いた電子ホログラフィに関する研究も報告されており[11,12]，今後，複数の LCD パネルとマルチ GPU クラスタから構成されたシステムによる電子ホログラフィの研究は重要となるものと考えられる．

本論文では，最大 12 枚の LCD パネルの利用を想定し，1 ノードに 3 枚の GPU ボード(NVIDIA Geforce GTX 480)を搭載した 4 ノード構成のマルチ GPU クラスタシステムに CGH 計算を実装した．本システムによる計算高速化について報告する．最終的に，4,096 点で構成された三次元物体による約 20M pixel の CGH 計算において，CPU(Intel Core i7 930, 8 スレッド使用)に対し，1,600 倍の計算高速化を実現した．

2．研究の背景

2.1 Fermi アーキテクチャを搭載した GPU

本論文では，GPU として Fermi アーキテクチャを搭載した NVIDIA 社の Geforce GTX 480 を使用した(表 1)．図 1 に Fermi アーテクチャの概要を示す．Fermi アーキテクチャの GPU ボードは，主に GPU チップとデバイスメモリから構成される．GPU は，複数のマルチプロセッサ(MP)を持つ．1 つの MP 内に 32 個のストリームプロセッサ(SP)と 64KB のコンフィグラブルな L1 キャッシュ/共有メモリ(Shared Memory)を持つ．具体的には，48KB の共有メモリと 16KB の書き込み可能な L1 キャッシュ，もしくは，16KB の共有メモリと 48KB の書き込み可能な L1 キャッシュとして使用することができる． SP がデバイスメモリから読み出す際，L1 キャッシュを使用できるようになったことと，1 つの MP 内の SP の数が増えたことが Fermi アーキテクチャの主な特徴である．MP 毎に，SIMD 処理がなされる．なお，Geforce GTX 480 では，MP の個数は 15 個となっている．

表 1　NVIDIA Geforce GTX 480 の仕様

Processor クロック	1,401 MHz
Streaming Processor 数	480
メモリ容量	1,536 MB GDDR5
メモリバンド幅	177.4GByte/sec

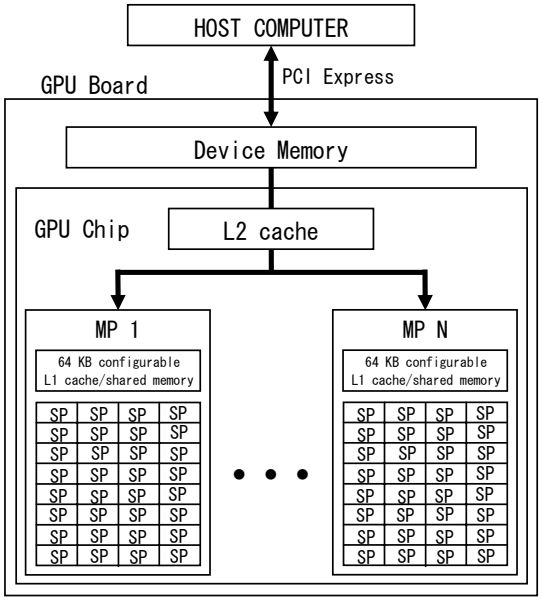

図 1　Fermi アーキテクチャの概要

GPU プログラム開発環境として NVIDIA 社から提供されている CUDA Toolkit [13]を用いる場合，GPU で処理される関数を kernel と呼ぶ．kernel は，PCI-Express バスを経由して，ホスト PC から GPU ボードに転送され，GPU 上で kernel が動作する．GPU 上で行う処理は thread，thread

†　湘北短期大学情報メディア学科
††　千葉大学大学院工学研究科
†††　国立天文台

block，grid と呼ばれる単位に分けられる．1 つの SP に割り当てられる処理を thread，thread のまとまりを thread block と呼び，1 つの MP に 1 つの thread block が割り当てられる．thread block は最大で三次元の thread 配列を含むことができる．同じサイズの thread block をまとめたものを grid と呼び，grid は最大で二次元の thread block 配列を含むことができる．また，grid はホスト PC から GPU に実行を指令する単位であり，grid 内の全 thread は同じカーネルを実行する．

2.2 CGH の作成

図 2 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 とする．そのとき，ホログラム面上の点 α の位置座標(x_α, y_α)における光の強度 $I(x_\alpha, y_\alpha)$ は，次式となる．

$$I(x_a, y_a) = \sum_{j=1}^{N} \cos\theta, \qquad (1)$$

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

ここで，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．なお，三次元物体の再生時に使用される再生光は，参照光と同じ波長の光を使用する．

式(2)において，平方根が存在するため計算量は多くなる．そこで，物体の z 座標を，x, y 座標に比べて十分大きくとり，式(2)にテイラー展開の 1 次近似を適用する．これにより，次のフレネル近似式を導出することができる[14].

$$\theta = \frac{\pi}{\lambda z_j}\left\{(x_\alpha - x_j)^2 + (y_\alpha - y_j)^2\right\} \qquad (3)$$

式(3)は式(2)に比べると計算量が低減している．本研究では，式(1)及び(3)を使用する．

CGH 上の 1 画素の光の強度を求めるには，式(1)と式(3)を j=1 から N 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×H とすると，その計算量は$(W$×$H)$×N に比例することになる．一方，CGH 計算で使用するデータは物体点の位置座標と出力画像であるため，メモリ使用量は W×H+N に比例する．1 枚の CGH の計算量とメモリ使用量を表 2 に示す．

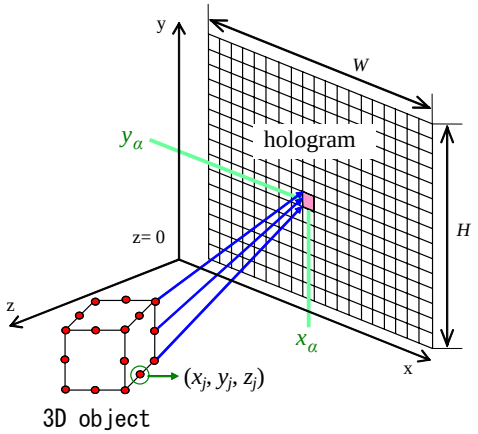

図 2　CGH 計算の座標系

表 2　1 枚の CGH 作成に必要な計算量とメモリ使用量

計算量	メモリ使用量
$O(WHN)$	$O(WH+N)$

表 3　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 (2GB×3 枚) DDR3-1333
マザーボード	ASUS P6T7 WS Supercomputer
ハードディスク	500 GB

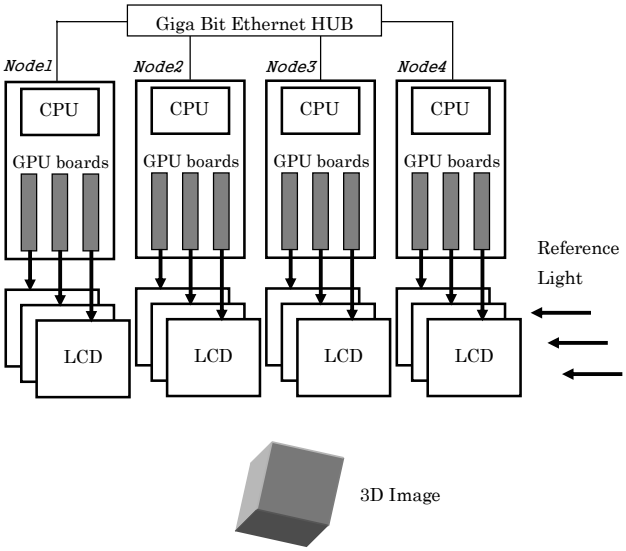

図 3　マルチ GPU クラスタシステム

式(1)の計算において，各 α 間に相関はなく，CGH 上の各画素に対して独立に計算することができ，並列計算が有効である．

3. マルチ GPU クラスタシステムへの実装

3.1 マルチ GPU クラスタシステム

図 3 に本研究で使用するマルチ GPU クラスタシステムを示す．本システムは，4 ノードの PC から構成される．各ノードは，3 枚の GPU ボード(NVIDIA Geforce GTX 480)を搭載している．各ノードのスペックを表 3 に示す．表 2 より CGH 計算で使用するメモリ量は計算量に比べ少ない．ノード間通信も物体点の座標データのみであるため，高速なネットワークを必要としない．よって， 1Gbps のギガビット・イーサネットをネットワークに使用した．各ノードから，物体点の座標データが取得できるように Network File System (NFS)を使用した．

本来，電子ホログラフィでは，画素間隔が 10 μ m 以下の高精細な専用の LCD パネルが用いられる．これに再生光を照射することにより，図 3 のように三次元物体が再生される．本論文では，マルチ GPU クラスタシステムによる CGH 計算の高速化を検討することを目的とする．そのため，利便性を考慮し，電子ホログラフィ用の高精細な LCD パネルの代わりに，従来から PC に用いられている LCD を用

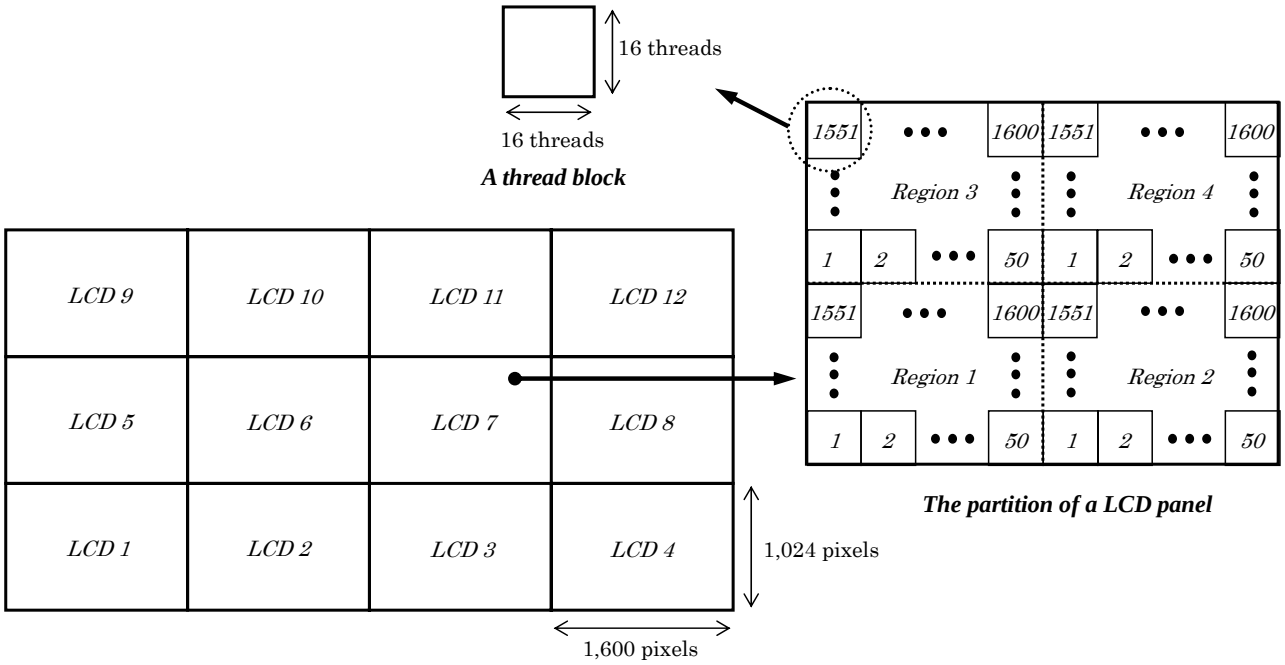

図 4 マルチ GPU クラスタシステムを用いた CGH 計算

いて検討を行うことにした．なお，両者において，CGH 表示の処理時間の差異は全く生じない．

本システムにおいて，各 GPU ボードに 1 台の LCD を接続し，GPU で作成した CGH を，CPU を介さずに直接 LCD に表示させた．解像度 1,600×1,024 の LCD を合計 12 台使用した．よって，本システムで作成する CGH の全画素数は，約 20M pixel (≈12 LCDs×(1,600×1,024 pixels/ LCD))となる．

3.2 CGH 計算の実装

図 4 にマルチ GPU クラスタシステムを用いた CGH 計算の概要を示す．図 4 において，本計算で使用する 12 台の LCD は，LCD 1~12 で示されている．2.2 節より，式(1)の計算は，各画素に対して並列計算が可能である．よって，各 LCD に表示される CGH は，接続された 1 枚の GPU ボードで計算が行われ，作成される．つまり，12 台の LCD に表示される約 20M pixel の CGH は，12 枚の GPU ボードで並列計算が行われ，作成されることになる．さらに，各 LCD に表示される CGH においても，接続された 1 枚の GPU ボード上の GPU チップ内に存在する 480 個の SP によって並列計算が行われ，作成される．

1 枚の GPU ボードで行われる各 LCD の CGH 計算を，図 4 の LCD 7 に示す．LCD 7 において，領域を 4 つに分割する．分割された各領域(Region 1~4)は，1,600 個のブロックで構成される．Region 1~4 において，同じ番号が付けられたブロックは 4 個存在する．これらの 4 つのブロック内の全画素を，一つの thread block で計算させる．つまり，1 つの thread で Region 1~4 の 1 画素ずつ，すなわち，1 つの thread で合計 4 画素について式(1)及び式(3)の計算を行う．なお，GPU による CGH 計算において，thread block を二次元の thread 配列(16,16)で構成し，grid を二次元の thread block 配列(50,32)(=(1600/(16×2), 1024/(16×2)) で構成する．

各 GPU ボードにおいて，三次元物体点の位置座標データを global memory に格納する．GPU 上で実行される CGH 計算の kernel において， 三次元物体点の位置座標データを global memory から共有メモリにコピーし，共有メモリに格納されたデータを計算に用いる．ここで，coalesced memory access を用いた[13]．式(1)より，CGH 計算の kernel において物体点 j に対し，j=1~N のループ処理が必要である．GPU による計算を高速化させるため，このループに対し，ループアンローリングを適用した．

次に，マルチ GPU クラスタシステムにおいて，各 GPU ボードでの処理を同期させるために，MPI(Message Passing Interface)を使用した．また，各 GPU によって計算された CGH を各 LCD へ表示するためにグラフィックス API(Application Program Interface)として OpenGL を使用した．マルチ GPU クラスタシステムによる CGH 計算は，次のように行われる．

1) 各ノードにおいて，NFS により共有しているディレクトリから物体点の位置座標データを取得する．このとき，MPI のバリア同期を用いて，全部のノードで物体点の位置座標データを取得するのを待つ．
2) 各 GPU ボードにおいて，各画素に対して式(1)及び式(3)を並列計算する(図 4).
3) Pixel buffer object(PBO)[13]を用いて，各 GPU ボードで計算された CGH を CPU に戻すことなく直接 LCD に出力する．

1) ～ 3)の処理を繰り返すことで，電子ホログラフィによる 3D 動画再生を行うことができる．

表 4　マルチ GPU クラスタシステムによる計算時間（1 ノードの場合）

1 GPU board / node，全画素数：1,638,400 pixels (LCD: 1 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	16.82	27.93	49.92	71.77	93.79
全処理時間 (ms)	18.67	29.65	51.64	73.49	95.47
全処理時間－CGH 計算時間 (ms)	1.85	1.72	1.72	1.72	1.68
2 GPU board / node，全画素数：3,276,800 pixels (LCD: 2 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	19.13	29.96	51.84	73.94	96.01
全処理時間 (ms)	21.88	32.64	54.55	76.66	98.76
全処理時間－CGH 計算時間 (ms)	2.75	2.68	2.71	2.72	2.75
3 GPU board / node，全画素数：4,915,200 pixels (LCD: 3 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	21.53	32.49	54.53	76.45	98.51
全処理時間 (ms)	24.63	35.62	57.67	79.64	101.74
全処理時間－CGH 計算時間 (ms)	3.10	3.13	3.14	3.19	3.23

表 5　マルチ GPU クラスタシステムによる計算時間（2 ノードの場合）

1 GPU board / node，全画素数：3,276,800 pixels (LCD: 2 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	16.94	27.90	49.51	71.66	93.58
全処理時間 (ms)	18.67	29.65	51.19	73.48	95.24
全処理時間－CGH 計算時間 (ms)	1.73	1.75	1.68	1.82	1.66
2 GPU board / node，全画素数：6,553,600 pixels (LCD: 4 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	19.25	30.04	52.11	74.01	95.98
全処理時間 (ms)	22.04	32.83	54.96	76.78	98.87
全処理時間－CGH 計算時間 (ms)	2.79	2.79	2.85	2.77	2.89
3 GPU board / node，全画素数：9,830,400 pixels (LCD: 6 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	21.59	32.60	54.61	76.52	98.54
全処理時間 (ms)	24.90	35.93	58.01	79.81	101.84
全処理時間－CGH 計算時間 (ms)	3.31	3.33	3.40	3.29	3.30

4. 性能評価

4.1 マルチ GPU クラスタシステムの評価

マルチ GPU クラスタシステムにおいて，オペレーティングシステム(OS)として Linux (CentOS 5.5)を使用した．GPU プログラム開発環境として NVIDIA 社が提供している CUDA 3.2 Toolkit を用いた．なお，64KB のコンフィグラブルな L1 キャッシュ/共有メモリにおいて，48KB の共有メモリと 16KB の書き込み可能な L1 キャッシュの設定とした． MPI ライブラリとして MPICH2-1.3.2pl を使用し，グラフィックス API として OpenGL を使用した．

ノード数と 1 ノードに搭載している GPU ボード数に対して，CGH の計算時間，CGH 計算から LCD 出力するまでの全処理時間を比較検討した．なお，物体点数 512 ~ 4,096 点まで調査した．表 4 は，1 ノードにおいて GPU ボードを 1 ~ 3 枚まで使用したときの結果を示す．同様に，表 5 は 2 ノードの場合，表 6 は 3 ノードの場合，表 7 は 4 ノードの場合の結果を示す．

表 4 ~ 7 において，1 ノードに搭載した GPU ボード数が同じ場合，ノード数の増加による CGH 計算時間及び全処理時間の遅延は 0.5ms より小さい．一方，全てのノード数の場合において，CGH 計算時間は GPU ボード数が 1 枚増加するにつれ，約 2.5ms 遅延している．CGH 計算において，GPU で計算する前に各ノードのメインメモリから GPU ボード上のデバイスメモリへ物体点の位置座標データを転送する．その転送回数は GPU ボード数に比例し，これが遅延する原因となっている．また，表 4 ~ 7 において“全処理時間－CGH 計算時間”は，CGH 計算処理以外の処理時間を示す．主に，各 MPI プロセスによる NFS からメインメモリへの物体点位置座標データのロード，MPI プロセス間の同期処理，計算により得られた CGH の LCD 出力を行う．“全処理時間－CGH 計算時間”においても，GPU ボード数が 1 枚増加するにつれて遅延している．遅延の原因の一つは MPI プロセス間の同期処理である．さらに，1 つの MPI プロセスに GPU ボード 1 枚を割り当てており，各 MPI プ

表 6　マルチ GPU クラスタシステムによる計算時間（3 ノードの場合）

1 GPU board / node，全画素数：4,915,200 pixels(LCD: 3 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	16.40	27.79	49.78	71.61	93.37
全処理時間 (ms)	18.36	29.57	51.60	73.42	95.13
全処理時間－CGH 計算時間 (ms)	1.96	1.78	1.82	1.81	1.76
2 GPU board / node，全画素数：9,830,400 pixels(LCD: 6 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	19.13	30.06	52.03	74.01	96.03
全処理時間 (ms)	21.94	32.88	54.88	76.76	98.85
全処理時間－CGH 計算時間 (ms)	2.81	2.82	2.85	2.75	2.82
3 GPU board / node，全画素数：14,745,600 pixels(LCD: 9 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	21.65	32.65	54.58	76.55	98.52
全処理時間 (ms)	25.04	35.89	57.89	79.86	101.88
全処理時間－CGH 計算時間 (ms)	3.39	3.24	3.31	3.31	3.36

表 7　マルチ GPU クラスタシステムによる計算時間（ 全 4 ノードの場合）

1 GPU board / node，全画素数：6,553,600 pixels (LCD: 4 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	16.79	27.58	49.87	71.41	93.75
全処理時間 (ms)	18.67	29.39	51.60	73.25	95.49
全処理時間－計算時間 (ms)	1.88	1.81	1.73	1.84	1.74
2 GPU board / node，全画素数：13,107,200 pixels (LCD: 8 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	19.18	30.03	52.01	74.06	95.98
全処理時間 (ms)	22.00	32.82	54.87	76.85	98.75
全処理時間－計算時間 (ms)	2.82	2.79	2.86	2.79	2.77
3 GPU board / node，全画素数：19,660,800 pixels (LCD: 12 台)					
物体点数	512	1,024	2,048	3,072	4,096
CGH 計算時間 (ms)	21.64	32.65	54.59	76.61	98.56
全処理時間 (ms)	25.11	35.97	58.07	79.96	102.02
全処理時間－計算時間 (ms)	3.47	3.32	3.48	3.35	3.46

ロセスによる物体点位置座標データのロード回数は GPU ボードの枚数に比例する．これも遅延の原因となっている．

表 7 より，本マルチ GPU クラスタシステムのリアルタイム処理により，2,048 点で構成された三次元物体による約 20M pixel の CGH を約 20 fps(frame per second)で表示することができる．つまり，12 枚の電子ホログラフィ用 LCD パネルを用いて，2,048 点で構成された三次元物体の動画像を約 20 fps で再生できることを示している．

4.2　計算高速化の検討

CPU(Intel Core i7 930)1 個による CGH の計算時間と本システム(4 ノード，12 GPU)による CGH の計算時間を比較し，本システムによる計算高速化について検討する．CPU で用いる CGH 計算プログラムは，マルチスレッド・プログラムである．cos 関数の計算をテーブルに置き換え，CPU の性能を十分発揮できるように最適化した．また，ハイパースレッディングを使用し，CPU で使用可能な 全 8 スレッドを用いて CGH 計算を行った．なお，コンパイラとして，gcc 4.1.2 を使用し，コンパイラオプションを“-pthread -O3 –lm”とした．

CPU 及びマルチ GPU クラスタシステムによる CGH(19,660,800 pixels)の計算時間と，CPU に対する本システムの高速化率を表 8 に示す．表 8 において，“12 GPU”は本システムによる CGH の計算時間を示している．また，高速化率は，本システムによる計算時間に対する CPU による計算時間の比の値である．

表 8 において，CPU による計算時間は物体点数に対して比例している．これは，物体点数が 512 点以上のとき CPU の計算速度がほぼ一定となっていることを示している．

一方，マルチ GPU クラスタシステムにおいては，物体点数が増えるにつれて，CPU に対する本システムの高速化率は向上している．物体点数が 1,024 点以上のとき， CPU による計算に対して 1,000 倍以上の計算高速化を実現している．特に，物体点数が 4,096 点のときには，CPU に対して 1,600 倍の計算高速化を達成している．

表 8　CPU とマルチ GPU クラスタシステム(4 ノード，12 GPU)による CGH 計算時間の比較(CGH の画素数：19,660,800 pixels)

物体点数	CGH 計算時間 [ms]		高速化率
	A CPU (8 threads)	12 GPU	
512	19,764	21.64	913
1,024	39,534	32.65	1,211
2,048	78,982	54.59	1,447
3,072	118,455	76.61	1,546
4,096	157,935	98.56	1,602

本システムによる CGH 計算は，物体点数が少ないときに高速化率が低くなる．これは，4.1 節でも述べたようにノードあたりの GPU ボード数の増加によって生じた遅延時間が原因である．物体点数が少ないと CGH の計算時間も短くなるため，CGH の計算時間に対する遅延時間の占める割合は大きくなり，計算速度は低下することになる．しかし，物体点数が 512 点の場合において，表 7 より全処理時間は 25.11ms であることから，約 40 fps で CGH を表示することができる．よって，電子ホログラフィによるリアルタイム再生が可能である．物体点数が少ない場合においても十分な計算速度が得られていることがわかる．

5. まとめ

本論文では，最大 12 枚の LCD パネルの利用を想定し，1 ノードに 3 枚の GPU ボード(NVIDIA Geforce GTX 480)を搭載した 4 ノード構成のマルチ GPU クラスタシステムに CGH 計算を実装した．

本システムによる計算高速化を検討した結果，2,048 点の三次元物体による約 20M pixel の CGH を約 20 fps で表示することができることが示された．さらに，4,096 点で構成された三次元物体による約 20M pixel の CGH 計算において，CPU(Intel Core i7 930, 8 スレッド使用)に対し，1,600 倍の計算高速化を実現した．

謝辞

本研究は，科学研究費補助金若手研究(Ｂ) (課題番号 22700060)，及び，総務省・戦略的情報通信研究開発推進制度(SCOPE) (課題番号 09150542)による補助のもとで行われました．深く感謝の意を表します．

参考文献

[1] D. Gabor, “A new microscope principle,” Nature 161, 777-778, 1948.

[2] G. Tricoles, “Computer generated holograms: an historical review,” Appl. Opt. 26, pp.4351–4360, 1987.

[3] P. S. Hilaire, S. A. Benton, M. Lucente, M. L. Jepsen, J. Kollin, H. Yoshikawa, and J. Underkoffler, “Electronic display system for computational holography,” Proc. SPIE 1212-20, pp.174–182, 1990.

[4] N. Hashimoto, S. Morokawa and K. Kitamura, “Real-time holography using the high-resolution LCTV-SLM,” Proc. SPIE 1461, pp. 291-302, 1991.

[5] K. Sato, K. Higuchi and H. Katsuma, “Holographic television by liquid crystal devices,” Proc. SPIE 1667, pp.19–31, 1992.

[6] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, “Computer generated holography using a graphics processing unit,” Opt. Express 14, pp.603-608, 2006.

[7] NVIDIA, GPU Gems 3, Addison-Wesley,2007.

[8] T. Hamada, T. Narumi, R. Yokota, K. Yasuoka, K. Nitadori, and M. Taiji, “42 TFlops hierarchical *N*-body simulations on GPUs with applications in both astrophysics and turbulence,” Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009.

[9] N. Takada, T. Shimobaba, N. Masuda, and T. Ito, “Improved Performance of FDTD Computation Using a Thread Block Constructed as a Two-Dimensional Array with CUDA,” ACES Journal, vol. 25, no. 12, pp. 1061-1069, 2010.

[10] 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," Opt. Express,17, pp.16038-16045, 2009.

[11] 白木厚司，伊藤智義，増田信之，下馬場朋禄，〝複数の液晶ディスプレイパネルを用いた電子ホログラフィ再生像の拡大〟，情報技術レターズ，Vol. 5, pp. 247-248, 2006.

[12] 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.

[13] NVIDIA, NVIDIA CUDA C Programming Guide ver. 3 .2, NVIDIA, 2010.

[14] M. Lucente, “Interactive computation of holograms using a look-up table,” J. Electron. Imaging **2**, pp.28–34, 1993.