J-034

GPUによる計算機合成ホログラムの高速化

Computer Generated Hologram using Graphics Processing Unit

増田信之† 田中喬† 白木厚司† 伊藤智義†*

Nobuyuki Masuda Takashi Tanaka Atsushi Shiraki Tomoyoshi Ito Takashige Sugie

1. まえがき

近年，３次元動画像の表示技術の研究開発がさかんに行われており，数値シミュレーション結果の可視化や医療分野における人体情報の可視化など様々な分野での応用が期待されている．本研究では物体の光の干渉と回折を利用して３次元動画像を記録再生するホログラフィ技術に焦点をあてる．

ホログラフィでは３次元の情報をホログラムに記憶するが，そのホログラムは光の干渉をシミュレートすることにより計算機で作成することができる．そのようにして作成されたホログラムは計算機合成ホログラム(CGH：Computer Generated Hologram)と呼ばれる．しかしCGHの作成にはその膨大な演算量のために相当な時間がかかってしまう．これがホログラフィを用いた３次元動画像システムの大きな問題点の一つとなっている．

現在までに我々の研究室では，この問題を解消するためにFPGA(Field Programmable Logic Array)を用いた専用計算機(HORN(HOolographic ReconstructioN))を開発してきた．

本研究では近年，CPU以上に高速化のスピードが上がっているGPU（Graphics Processing Unit)に着目し，CGH計算の高速化を目的としてCGHの計算をGPUを用いて行い，CPUとその計算速度を比較することでGPUによるCGH計算の有効性を検証することを目的としている．

2. 計算機合成ホログラムの作成

記録する３次元物体がN点で構成されているとすると，式(1)を計算することによってCGHを作成することができる．

$$I(x_\alpha, y_\alpha) = \sum_j^N \cos(2\pi\theta) \qquad (1)$$

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

Iはホログラム面上(x_α, y_α)での光の強度，pはホログラムの画素間隔，λは参照光の波長，(x_j, y_j, z_j)は物体点の座標である．また，物体のz座標がx, y座標に比べて十分大きくなるように配置することで式(2)を式(3)のように近似することができ，こうすることでルート計算をなくすことが出来，計算負荷を軽減できる．

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

ここで式(3)の第１項を省略してもホログラムを作ることができる．本研究では式(1)と式(3)の第１項を除いた式を用いてCGH計算を行った．

3. GPU

コンピュータグラフィックス(CG)の分野では非常に多くの浮動小数点演算が求められることが多く，CPUではリアルタイム処理に追いつくことが不可能である．そこでGPUのようなグラフィックス処理に特化した専用の演算装置が用意されている．

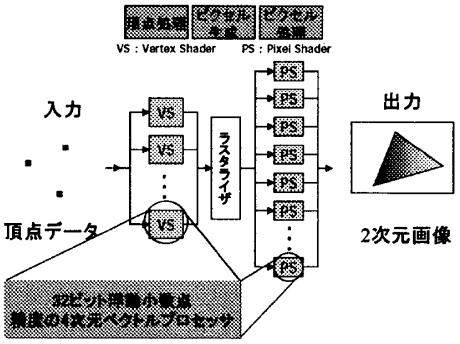


Fig.1 ：GPUパイプライン概略図

GPUは固定された内部処理のパイプラインを持ち，複数のベクトル演算装置で並列処理を行うことでCPUを上まわる高い演算能力を得ている．Fig.1にそのパイプラインの概略を示す．GPUのパイプラインに入力された頂点のデータは頂点処理を行う「Vertex Shader」に送られ、その後、ピクセル生成を行う「ラスタライザ」に送られ、さらにピクセル処理を行う「Pixel Shader」で処理され、2次元画像としてモニタなどの画面に出力される。そして近年，パイプラインの一部を自由にプログラム可能にしたGPUが登場した．パイプライン中で自由にプログラム可能になったのは、Vertex ShaderとPixel Shaderである．

†千葉大学工学部 †*千葉大学工学部／科学技術振興機構

それぞれ、32bit 浮動小数点精度の 4 次元ベクトルプロセッサが並列に組み込まれている．本研究ではこれらのプログラム可能なプロセッサを用いて，GPU による CGH 計算を行い，その高速化を評価した．

4. 結果

70 点で構成された物体(星型)から 48 万画素(800×600)のホログラムを CPU および GPU を用いて計算したときの計算時間を Table1 に示す．また作成された CGH を用いて，ホログラム再生用の光学系で再生した再生像を Fig.2 に示す．

本研究で用いた GPU は，nVIDIA 社 GeForce 6600 で，Vertex Shader が 3 並列，Pixel Shader が 8 並列されている．前節で述べたように，GPU では，Vertex Shader と Pixel Shader の両方がプログラム可能であるが，この計算では，より並列度の高い Pixel Shader を用いて CGH 計算を行っている．また，GPU の計算においては，グラフィックス API に DirectX 9.0c を使用し，上位レベルシェーダ言語として，HLSL(High Level Shader Language)を使用した．

Table1：CGH 計算時間

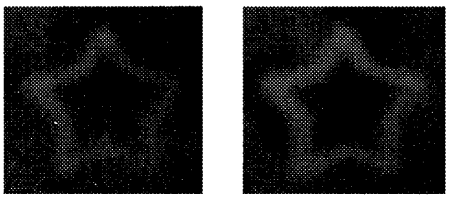
CPU	GPU
2980 [ms]	67 [ms]

また、計算に用いられた CPU，GPU は以下のとおりである．

Table 2 ：CPU 及び GPU

CPU	Intel Pentium 4 3.2GHz
Memory	2GB
OS	Linux
Compiler	Intel C++ Complier
GPU	nVIDIA GeForce 6600

CGH 作成の計算時間を比べると GPU を用いた場合，CPU を用いた場合より約 45 倍の計算速度が得られた．これは GPU の並列性と高い演算能力を CGH 計算に応用できたと結果といえる．



[i] CPU [ii] GPU

Fig.2 再生像

5. まとめ

本研究では GPU 上で CGH 計算（物体点数 70 点，ホログラム画素数 800×600 点）を実装することで CPU の約 45 倍の高速化を図ることに成功した．GPU は並列ベクトルプロセッサであり，その高い並列性を活かすことができるような計算には CPU を上回る高速な処理ができると言える．CGH 計算を考えると，各ピクセルにホログラムの座標データをセットすることでその並列性を活かすことができ，さらにすべてのピクセルで同じ計算を行うようになっている．つまり CGH 計算は GPU の構造に適していると言うことができる．しかし今回開発したシステムでは一度に GPU 内に保持できる物体点数が 100 に制限されてしまうという問題点を抱えている．これは GPU 内に搭載されているレジスタの数によるものであるが，このことに関しては，物体点をいくつかのグループに分け，その結果をグラフィックカード上の VRAM に一時的に保存することで，解決することができる．ただ，VRAM からのデータの読み込みには，約 2ms の時間がかかり(ホログラムの画素数が 512×512 の時)，このことが，速度向上の妨げになるが，さらに性能が向上した GPU を使用することで，計算の高速化が期待できる．現在のハイエンドな GPU には，Vertex Shader が 6 並列，Pixel Shader が 16 並列されている．また，物体点数やホログラムの画素数に制限はあるが，秒間 30 枚のホログラムの作成が可能であり(物体点 47 点，画素数 800×600)，実際に研究室にあるホログラム再生用の光学系を用いることで，3 次元動画のリアルタイム処理が可能になっている．

6. 文献

[1] 辻内順平："ホログラフィー"，裳華房(1997)

[2] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba and T. Sugie ``A special-purpose computer for electroholography HORN-5 to realize a real-time reconstruction,"
Optics Express, Vol.13, pp1923-1932(2005)

[3] Randima Fernando, (監訳)中本 浩：
"GPU Gems 日本語版"，ボーンデジタル(2004)

[4] 今給黎 隆："DirectX 9 シェーダプログラミングブック"，毎日コミュニケーションズ(2004)