社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－CG－123 (8)
2006／7／11

GPUを用いた3次元モデルの検索の高速化

木下 豪規　安藤 英俊
山梨大学大学院医学工学総合教育部　山梨大学大学院医学工学総合研究部

概要

近年，多くの分野で3次元モデルが使用されてきている．膨大な数の3次元モデルが存在する今，既存の3次元モデルを効率的に管理，再利用したいという考え方がある．再利用する際に3次元モデルを検索しなければならないが，検索にかかる時間が増えてしまい，高速化ということが重要になってきている．本論文では，GPUを用いて3次元モデルを並列的に検索するアルゴリズムを考案し，高速化を試みる．そして，CPUで検索した際の時間と比較，検証をする．
キーワード：3次元モデル，形状類似検索，GPU

GPU Acceleration of Retrieval of 3D Models

Hideki Kinoshita　Hidetoshi Ando
Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi

abstract

Recently, 3D models have been used in various fields. A great number of 3D models exist. Therefore, there are demands for efficient managing and reusing of existing 3D models. At the time of reuse, it requires a searching of 3D models. However, the time of a searching becomes longer with in the number of 3D models. As a result, the demand for acceleration of searching has been growing. In this paper, we present an algorithm for GPU acceleration of retrieval by using parallel computation. Finally, we compare and verify the difference of runtime between CPU and GPU.

1 はじめに

近年，コンピュータグラフィックス（CG）の発展により，ゲーム，映画などのエンターテイメントの世界において3次元モデルが多く利用されるようになった．それに伴い，3次元モデルを効率よく管理，再利用することの重要性が増してきている．そのため，3次元モデルを対象とした形状類似検索の研究が注目されるようになっている．

また，グラフィックスハードウェアであるGraphics Processing Unit（GPU）は急速に性能を向上させ，従来固定機能であったパイプラインの一部を，現在ではプログラムで制御することが可能になり，自由なグラフィックス表現を可能にしている．性能においても3次元CGの必要性に伴い需要が増えてきていることから，以前では映画でしか表現できなかったグラフィックスがリアルタイムで表現できるようになってきている．

これらを背景として，本論文では中澤ら[1]による3次元モデルの外観に基づいた形状類似探索システムを取り上げ，当システムにおいて3次元モデルの形状類似度として用いたMultiple-Orientation Depth Fourier Descriptor (MODFD)の比較をGPU上で計算することにより，検索の高速化を試みる．

2 関連研究

3 次元モデルの形状類似検索システムの標準的な流れを図 1 に示す．システムがユーザから検索要求として与えられた 3 次元モデルと検索対象となるデータベース中のモデルとの類似検索を行い，検索結果として類似度の高いモデルをユーザに表示するというものである．3 次元モデルの形状特徴を抽出した特徴量を求め，特徴量間の何らかの距離をモデル間の相違度とし，相違度が小さいモデルを類似度の高いモデルとする．

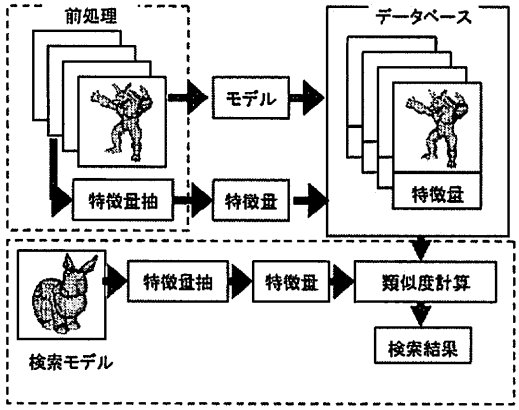

図 1 ：モデルの検索システムの流れ

特徴量の一つに MODFD と言う特徴量がある．これはモデルの周囲に配置された多数の視点から得られるモデルの深度画像を基にしたものである．深度画像が描画できれば計算可能であるということから，ポリゴンスープやボクセル表現など，ほとんどのモデルデータ形式に適用可能であるという特徴を持っている．

一方，先に述べたように，GPU の性能は急速に発展している．そして，GPU は高速な並列計算機であり，パイプラインの一部をプログラムで制御できるようになった．最新のものでは 48 個のピクセルユニットを持つ GPU まで存在する．そのため，本来グラフィックス専用のハードウェアである GPU をソーティングや物理シミュレーションなどの描画以外の汎用的な目的で用いる General-Purpose computation on GPUs (GPGPU)と呼ばれる研究分野が近年確立されてきている．[2]

これらを背景に鳥越ら[3]は， MODFD を GPU 上で計算するアルゴリズムを提案し，特徴量抽出の高速化を行なった．しかし，MODFD の計算は事実，前計算であるため高速化による恩恵はそれほど大きくないと言える．そこで本論文では，検索計算の高速化を目指した．検索計算は，実行時に行なわれるタイムクリティカルなプロセスであり，高速化による恩恵は非常に大きいと言える．

3 研究概要

本節では， まず初めに MODFD についての概要を述べ，次に鳥越らが提案した GPU を用いた MODFD の計算方法を述べる．最後に，本手法による MODFD の比較計算を GPU で行う方法について述べる．

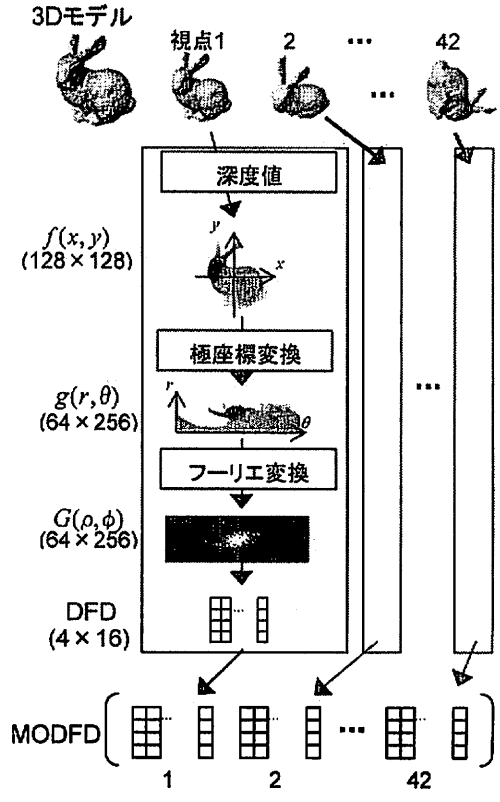

図 2 : MODFD の計算手順

3.1 MODFD

MODFDとは，中澤らが提案した3次元形状類似システムにおいて用いられた特徴量である．MODFDは，大きさや回転に不変な特徴量であり，値は図2の手順によって求める事が出来る．

モデルの周囲におかれた42個の視点について特徴量を計算し，これをDFDと呼ぶ．またMODFDは42個のDFDをまとめたものである．

DFDの計算は，まず初めにモデルの位置と大きさについて正規化を行う．そして，モデルの深度値を128×128画素のテクスチャに描画する．更に，このテクスチャに対して，極座標変換を行い，64×256画素のテクスチャを作成する．次に，この深度テクスチャに対してフーリエ変換を施し，64×256個の値からなる振幅スペクトルを得る．最後に，振幅スペクトルから大きな値の集中する低周波成分を4×16個だけ抽出し，それをDFDとする．

このMODFDを使ってモデル間の距離を計算し，相違度の小さいモデルを類似度の高いモデルとみなし検索を行う．モデルX,Yの相違度計算は以下の式で定義される．

$$d(X_i,Y)=\min_{1\le k\le p}\left(\sum_{j=1}^{q}\left|X_{i,j}-Y_{k,j}\right|\right) \qquad (1)$$

$$D(X,Y)=\frac{1}{p}\sum_{i=1}^{p}d(X_i,Y) \qquad (2)$$

(1)式は，モデルXの視点iのDFDとモデルYのp個の視点のDFDでそれぞれ距離を求め，その最小値を取った値である．pは視点数，qはDFDの要素数である．(2)式は，モデルXにおいてすべての視点からの最小の距離の平均を求めるための式である．

3.2 GPUを用いたMODFDの計算

鳥越らは，図3で示すように前項で述べたMODFDの計算を全てGPUで行った．

まず42個の視点について128×128画素のグレースケールテクスチャに深度値を描画する．次に作成された42枚の深度テクスチャを4枚ずつ参照し極座標変換を行い，64×256画素のRGBAテクスチャの各カラーチャンネルにそれぞれの結果を格納する．これによって，64×256画素の極座標系深度テクスチャが計[42/4]=11枚生成される．そして，これらの深度テクスチャに対してMorelandら[4]の手法を拡張し，高速フーリエ変換を行う．最後にフーリエ変換された結果が格納されたテクスチャを参照し，物体の形状特徴を表わすエネルギーが集中している低周波成分であるDFDを抽出する．抽出したDFDは4×16×11=4×176画素のRGBテクスチャ上に並べて格納し，これをあるモデルの形状特徴量を示すMODFDテクスチャとした．その結果，鳥越らは大幅な高速化に成功した．

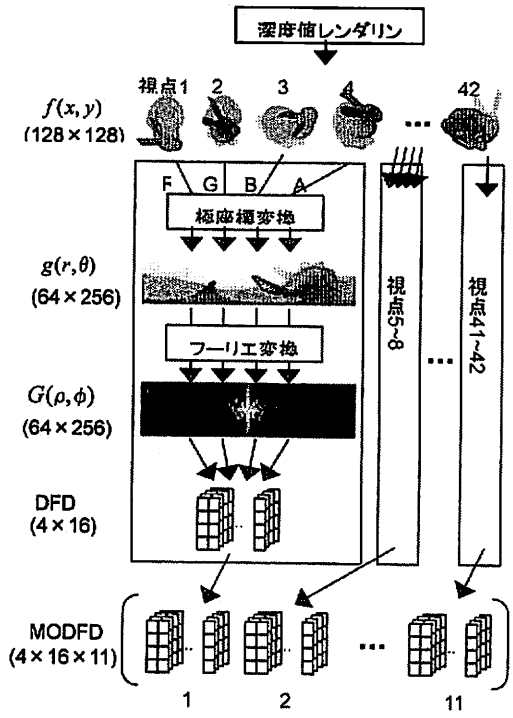

図3：GPUを用いたMODFDの計算手順

3.3 MODFDの格納方法

GPUで並列計算を行うには，すべてのモデルのMODFDテクスチャを一枚のテクスチャとして扱う必要があると言える．検索モデルとデータベースにある各モデルの比較計算を一個ずつ行うのではなく，並列に行うためである．

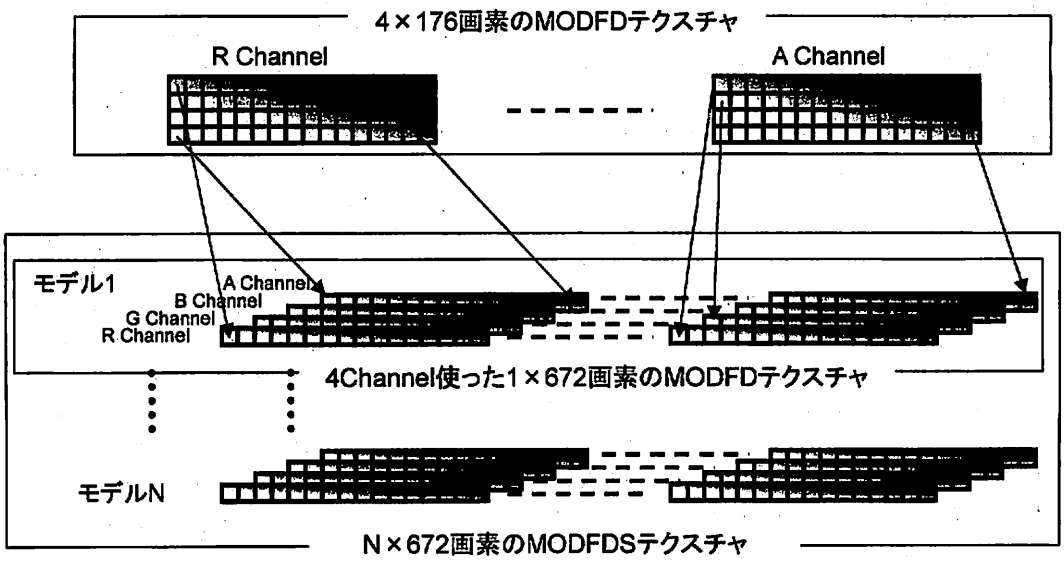

図 4：MODFD データの格納方法

本手法では鳥越らが作成した MODFD テクスチャを使い，全モデルの MODFD が格納された一枚のテクスチャを作成する．これを MODFDS テクスチャとし，作成方法を図 4 に示す．あるモデルの MODFD テクスチャを 1×672 画素のテクスチャに変換し，それを縦に並べて N×672 画素のテクスチャを作成する．N はデータベース内のモデル数である．

具体的に，まず鳥越らが作成した MODFD テクスチャの 4×176 画素の R Channel について，1×176 画素の RGBA 画像に変換を行う．4×176 画素のテクスチャの 1 列目を 1×176 画素の RGBA テクスチャの R Channel に格納，2 列目を G Channel，3 列目を B Channel，4 列目を A Channel に格納する．MODFD テクスチャの 4×176 画素の G Channel，B Channel，A Channel についても同様に変換し，最後に横に画像をつなぎ合わせ，1×672 画素の MODFD テクスチャを作成する．

すべてのデータベースモデルについて，変換を行い，縦につなぎ合わせ，N×672 画素の MODFDS テクスチャを作成する．

3.4 検索方法

検索モデルの MODFD テクスチャを作成し，MODFDS テクスチャと特徴量の比較を行う．図 5 は，検索モデルと MODFDS テクスチャの一列，つまりあるモデル A との比較を行う際の計算例である．

GPU では 2PASS に分けて計算する．まず 1PASS 目は(1)式について計算する．検索モデルの MODFD テクスチャの端から最初の視点の要素に相当する 16 画素に注目し，この 16 画素とモデル A の端から 16 画素ずつ，値の差の絶対値を取って相違度を計算し，その中で最小値を探す．この値をレンダリングターゲットに出力する．レンダリングターゲットは，N×42 画素のテクスチャである．同様に検索モデルの次の 16 画素について注目し，モデル A の端から 16 画素ずつ，値の差の絶対値を取って，最小値を探す．検索モデルの 42 個すべての視点について最小値を計算する．検索モデルと他のモデルについても同様に計算を行なうと N×42 ピクセルのテクスチャにすべてのモデルの 42 視点についての最小の相違度を得ることが出来る．

2PASS 目は，N×42 画素のテクスチャについて，N×1 画素のテクスチャをレンダリングターゲットとし(2)式について計算する．(2)式により検索モデルとデータベース内のモデルについての相違度を求める事が出来る．

最後に求めた値についてソーティングを行ない，類似度高いモデルを探す．

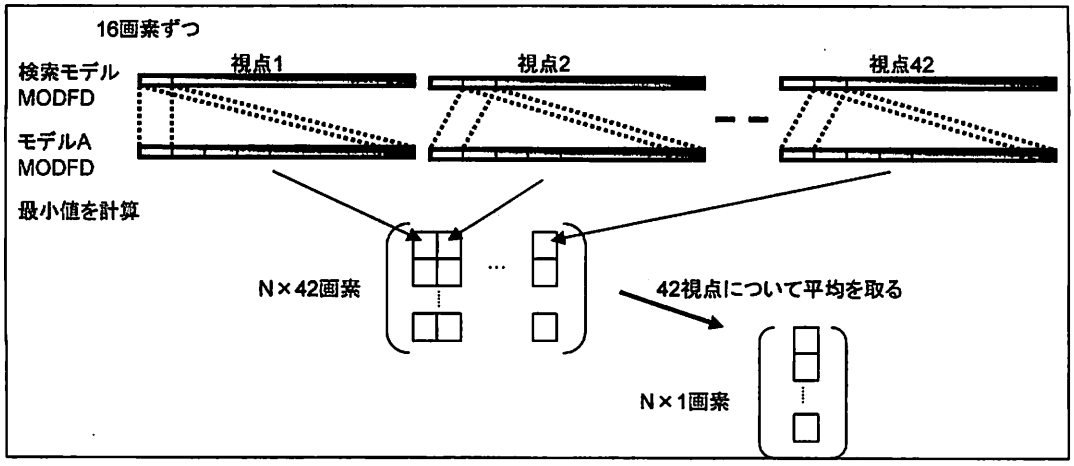

図 5：MODFDS テクスチャを使った相違度の計算方法

4 パフォーマンス検証

GPU 上で MODFDS テクスチャを作成し，類似度の高いモデルを検索するために(1)式，(2)式について GPU 上で DirectX9.0 のグラフィックスライブラリを用いて実装した．シェーダを記述する言語には Microsoft の上位レベルシェーダ言語である HLSL を使用し，コンパイルターゲットは，PS3.0 を使用した．本実装では，画素単位の比較計算を行なうため，頂点シェーダは使用していない．また本実装では，鳥越らが実装した MODFD のプログラムを拡張し，モデルの検索の高速化を行なった．3D モデルには，Princeton 大学が提供している Princeton Shape Benchmark(PSB) [5] から，989 個のモデルを選んで使用した．ただし，パフォーマンス検証の際には同じモデルを最大 4 回まで読み込み視覚的にモデルを増やして検索時間を測定した．最大 4 回というのは GPU で扱えるテクスチャのサイズが最大 4096×4096 画素であるためである．図 6 は，左上にあるモデルを検索要求モデルとし，類似度の高いモデルを探索し，類似度の高いモデルを左上，から順に 9 個出力していった画像である．左上のモデルは一番類似度が高く，二番に高いのは上段真ん中にあるモデルである．以下順に類似度が下がる．

表 1 は，CPU と GPU で検索計算を行なった場合の時間とモデル数を表している．GPU，CPU はそれぞれの検索時間で単位は[msec]である．また Time Ratio は，GPU の検索時間を CPU での検索時間で割った値である．

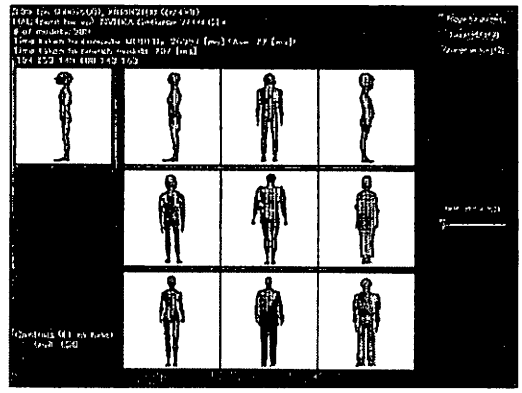

図 6：検索結果

表 1：モデル数における検索時間と比率

モデル数	GPU[msec]	CPU[msec]	Time ratio
10	34	7	0.205
50	35	34	0.971
100	36	70	1.94
200	37	138	3.72
500	41	354	8.63
989	51	706	13.8
1978	64	1431	22.3
2967	77	2179	28.2
3956	120	2877	23.9

実行環境は，CPU Pentium4 3.6GHz 2GB RAM，GPU Geforce7800GTX 256MB VRAM である．

表 1 よりモデル数が少ない時は，CPU での検索の方が速く，モデル数が多くなると GPU での検索のほうが速くなっていることがわかる．また CPU の検索時間は線形にあがっているのに対し，GPU での検索時間は，線形ではない．さらに GPU での検索時間で，モデル数が 10 から 100 までに対して，値があまりかわっていない．これらの要因として，CPU と GPU 間における通信のオーバーヘッドが原因であると考えられる．そのためモデル数が少ない時は，このオーバーヘッドが処理時間に大きく影響してしまい，Time Ratio が小さな値になってしまっている．モデル数が増えると全体の処理時間に比べて，オーバーヘッドの割合が減るため，Time Ratio が大きな値になっている．

表 1 よりモデル数 1978 以上において GPU での検索時間が CPU での検索時間が 20 倍以上速くなっていることがわかる．特にモデル数 2967 個の場合においては 28 倍速くなっている．

以上の実験結果より，GPU 上で MODFD の比較計算を行った場合，CPU の最大で 28 倍の速度パフォーマンスを得ることが出来た．この事より，3 次元モデルの形状類似検索において，課題であった検索コストの削減について成功したと言える．また本手法を用いることにより，他の類似検索における特徴量の比較計算においても検索概念はほぼ同様であると考えられ，GPU で計算を行う事が可能であると言える．

5 まとめと今後の展望

本論文では，3 次元モデルの形状類似探索手法として中澤らの手法である MODFD と，鳥越らによる GPU を用いた MODFD の計算の高速化を取り上げ，MODFD の比較による検索を GPU で行う事で高速化を試み，パフォーマンス検証を行った．パフォーマンス検証においては表 1 より高速化に成功したといえる．

今後の課題として，GPU での検索においてモデル数を増やす事が上げられる．GPU で扱えるテクスチャは 4096×4096 画素が最大であり，すべての画素を有効に使えれば，MODFD を使ったモデル探索において 24,000 個以上のモデルを扱う事が可能である．そのためには，MODFDS テクスチャにおいて MODFD を効率的に配置する方法を考えなければならない．

また本手法で MODFD を使った場合では約 24,000 個が限界であり，さらにモデル数を増やして検索するには何らかの方法を考えなければならない．例えば，事前に類似度の高いモデルごとにクラスタリングしておく方法が考えられる．クラスタリングされた集合において MODFD を計算する事により，より多くのモデルを検索できると考えられる．

参考文献

[1] ZOHBUCHI,R., NAKAZAWA, M., AND TAKEI, T. 2003. Retrieving 3D Shapes Based On Their Appearance. *Proc. 5th ACM SIGMM Workshop on Multimedia Information Retrieval (MIR 2003)*, pp. 39-46.

[2] HARRIS, M. (NVIDIA Corporation) 2005. GPGPU: General-Purpose Computation on GPUs. *Presentation on Game Developers Conference 2005.*

[3] 鳥越信孝，池ヶ谷和博，鳥山孝司，大渕竜太郎，安藤英俊，"GPU を用いた外観に基づく 3 次元形状特徴量抽出の高速化"，Visual Computing グラフィクスと CAD 合同シンポジウム 2005,pp. 245-248.

[4] MORELAND,K., AND ANGEL, E. 2003. The FFT on a GPU. *Proc. SIGGRAPH / Eurographics Workshop on Graphics Hardware 2003*, pp. 112-119.

[5] PRINCETON SHAPE RETRIEVAL AND ANALYSIS GROUP. Princeton Shape Benchmark. http://shape.cs.princeton.edu/benchmark/.