

複数のGPUを用いた超高速進化的画像処理システム

安藤 淳^{†1} 長尾 智 晴^{†1}

筆者らは先に、数種類の画像処理フィルタの適用順序および総数を遺伝的プログラミング (GP) によって最適化することで目的の画像処理を自動構築する方式、木構造状画像変換自動生成システム (ACTIT) を提案した。しかし、画像のサイズや枚数が大きい場合には膨大な計算時間を要しており、その高速化が必要となっていた。一方、汎用グラフィックスボードに搭載されているGPUにおいて、処理の一部をプログラミングによって自由に書き換えることが可能となったことから、GPUを汎用計算に活用する研究が最近注目されている。そこで本論文ではACTITの画像処理フィルタをGPU上で実装し、処理の高速化を行なう。元来並列性が高い画像処理にGPUを適用することは非常に有効である。

Fast Evolutionary Image Processing using Multi-GPUs

JUN ANDO^{†1} and TOMOHARU NAGAO^{†1}

We have already proposed the system which automatically constructs image processing with Genetic Programming (GP), Automatic Construction of Tree-structural Image Transformation (ACTIT). However, these complex image processing systems require much computing time because the problem uses large and many images. On the other hand, the research which employs GPU for the general purpose of calculating presently becomes popular. In this paper, we propose a fast evolutionary image processing system on multiple GPUs. We employ GPU to ACTIT for the purpose of reducing optimization time. Besides, the system effectively calculates in parallel with multiple GPUs for the fast processing. The optimization speed of the proposed system is several hundred times faster than that of the ordinary ACTIT.

1. はじめに

近年、画像処理・認識は多くの問題で必要とされているが、取り扱い問題に強く依存していることが多く、対象ごとに逐一画像処理を構築するのは非常に困難で手間がかかる。そこで、筆者らは先に、あらかじめ数種類の画像処理フィルタを用意し、そのフィルタの適用順序および総数を遺伝的プログラミング (GP; Genetic Programming)¹⁾ によって最適化することで目的の処理を近似する方式を提案した。この方式は木構造状画像変換自動生成システム (ACTIT; Automatic Construction of Tree-structural Image Transformation)²⁾ と呼ばれ、これまでに様々な問題に適用され、その有効性が示されている。しかしながら、学習用の画像のサイズや枚数が大きい場合などは膨大な計算時間を要しており、その高速化が必要となっていた。

一般に処理の高速化手法として、高速なハードウェア上での実装や複数のプロセッサを用いた並列処理などが考えられるが、近年安価で取り扱いやすい汎用グ

ラフィックボードに搭載されているGPU (Graphics Processing Unit) を汎用計算に活用する研究³⁾⁻⁷⁾ が注目されている。そこで、本論文ではACTITにGPUを適用することで、処理の高速化を行なう。さらに複数のGPUを用いて効率的に高速化するための方法を提案し、試作した超高速進化的画像処理システムの実験結果について述べる。

2. 本研究に関連する従来研究

2.1 ACTIT

ACTITは遺伝的プログラミングを画像処理に適用した研究の1つである。この手法では、1つあるいは複数の入力画像から1つの出力画像への変換を、あらかじめ用意した数種類の画像処理フィルタを任意の形の木構造に組み合わせることによって実現する。

図1にACTITシステムの処理過程を示す。まず、処理対象となる原画像、それらに対する理想的な画像処理の結果である目標画像、必要に応じて、適応度評価の重みをその階調値の大きさと表現した重み画像を教師画像セットとして用意する。次に、GPによる木構造状画像変換の最適化を行なって、出力として最大適応度をもつ木構造状画像変換を得る。そして、生成

^{†1} 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University

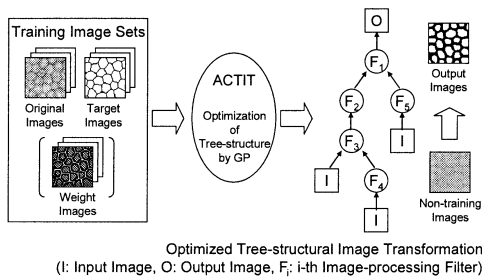


図 1 ACTIT システムの処理過程
Fig. 1 The processing flow of ACTIT system.

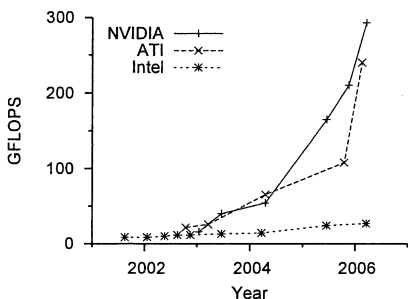


図 2 近年の CPU と GPU の性能向上
Fig. 2 The computational power of CPU and GPU in this several years.

された木構造状画像変換が教師画像セットと同様な特徴をもつ未知画像群に対しても有効であることを期待する。ACTIT はこれまでに 2 次元欠陥抽出画像処理や 3 次元医用画像処理など、様々な問題に適用され、その有効性が示されている。

2.2 General Purpose GPU

図 2 に示されるように、ストリームプロセッサの一種である GPU の処理能力は近年飛躍的に増大し、単純な演算性能において既に CPU (Central Processing Unit) を上回っている。また、従来の GPU は座標変換やレンダリングなどのグラフィックス処理を高速に実行する機能しかもっていなかったが、最近の GPU は、より複雑な処理を行なうために、グラフィックスパイプラインの一部をプログラミングによって自由に書き換えることが可能となっている。そこで現在、GPU をグラフィックス処理以外の汎用目的に活用する、GPGPU (General Purpose GPU)^{3),4)} が注目を集めている。

GPU は元来グラフィックス処理を行なうプロセッサであるため、条件分岐・メモリアクセス処理は苦手であるが、並列・多量のデータ処理は得意であるなどといった特徴をもつ。これらのことから比較的大きく、かつ分割しやすいデータを扱う画像処理は GPGPU に向いているのではないかと考えられる。

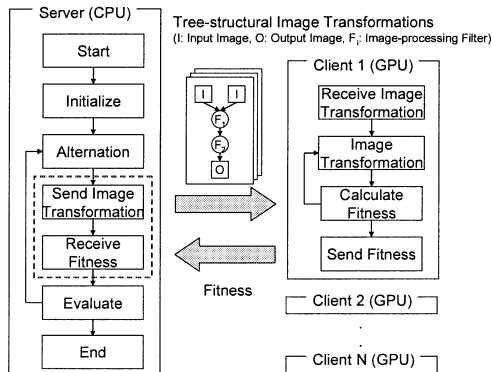


図 3 複数の GPU を用いた ACTIT
Fig. 3 ACTIT using multi-GPUs.

3. 超高速進化的画像処理システム

3.1 GPU の適用

GPU は大量の入力データそれぞれに対して、単純で高密度な演算を施して出力するような処理を得意としている。そこで本システムでは、GP の選択・交叉・突然変異などの世代交代・遺伝的操作は従来の ACTIT と同様に CPU 側で行ない、単調な繰り返し処理であるが計算コストが高い画像変換・適応度計算は GPU 側で行なう。

図 3 に GPU を用いた ACTIT の処理過程を示す。最初に、GPU の初期化処理として教師画像セットのロード、GPU 用に記述された画像処理フィルタのコンパイル、ロードを行なっておく。ここで、GPU では画像の 4 つの要素 (R:Red, G:Green, B:Blue, A:Alpha) に対して同時に演算処理を行なうことができるため、それぞれの要素に 1 セットの教師画像セット (合計 4 セット) を割り当てることとする。画像変換・適応度計算部では、GPU が CPU から指定された木構造のフィルタ列の並び順に従って原画像に画像処理フィルタを適用していく。そして、得られた出力画像と目標画像の差の絶対値に重み画像を乗じることで適応度を計算して、出力用に用意したテクスチャメモリに保存する。それらの処理を 1 世代の間で更新する個体数分繰り返した後、保存しておいた適応度だけをまとめて CPU に返す。結果として 1 世代の間における CPU と GPU の情報のやりとりは適用する画像処理フィルタの指定と適応度だけとなる。このように学習過程における CPU-GPU 間のデータの通信回数を極力少なくすることによって、更なる処理の高速化を図っている。

3.2 複数の GPU を用いた並列化

システムの更なる高速化のために複数の GPU を用いた ACTIT の並列化を行なう。本研究では汎用のグラフィックスボードが搭載された汎用の PC (Personal

Computer) 複数台を用いて並列システムを構築する。

並列化によってどれだけ高速化することができるのかを決める要因は計算ノード数とシステム全体における並列化可能部分の割合である。ここで ACTIT では並列処理が可能な画像変換・適応度計算部が処理時間の大半を占める。このことから ACTIT を並列化することは非常に有効であると考えられる。

並列計算を行なうにあたり、並列性を妨げる要因として考えられるのはノード間の同期・通信時間である。GA (Genetic Algorithm)^{8),9)}・GP の代表的な並列モデルである Master-Slave モデルでは個体を計算ノードへ送った後、全ての個体の適応度計算が終了するまで待つため、同期待ち時間が発生してしまう。そこで、同期時間を削減するために並列モデルとして Parallel-MGG¹⁰⁾ を用いる。Parallel-MGG では MGG (Minimal Generation Gap)¹¹⁾ による世代交代で各世代に更新される個体群をそれぞれ複数のノードに割り当てて、並列に、非同期に適応度計算を行なうモデルであるため、これによって同期時間を削減することができる。従来の ACTIT では、全体の処理時間の中で並列化可能部分が占める割合が非常に大きいため通信時間はほぼ無視できる。しかし、GPU を用いた ACTIT では並列化可能部分である画像変換・適応度計算部が高速化されたため、相対的に通信時間の影響が増大する。そこで本研究では Parallel-MGG を改良して通信時間の削減を図った。

従来の Parallel-MGG モデルでは計算ノードでの計算が終わった後、管理ノードに個体を返し、その後次に次の世代の個体を計算ノードに送っている。改良した Parallel-MGG モデルでは各ノードに待合室としてバッファを設けて先に個体を待たせておき、計算が終わった後、すぐに次の個体の計算を始められるようにすることで通信時間を削減し、効率的に並列処理を行なう。

4. 実験

4.1 実験環境

構築したシステムは PC 複数台を LAN ネットワーク経由でつないだものである。グラフィックスボードは NVIDIA GeForce 7900 GS を用いた。また、GPU のプログラミングを行なうためにグラフィックス API として OpenGL を、シェーダ言語として Cg (C for graphics) を用いた。

画像処理フィルタは、注目画素、及びその近傍画素の積和演算などを中心とした、GPU 上で実装可能な単純なフィルタを 1 入力フィルタ、2 入力フィルタ合わせて 37 種類用いた。教師画像セットとして横 × 縦が $64 \times 64 \sim 1024 \times 1024[\text{pixel}^2]$ のサイズをもつ細胞画像を用いた。

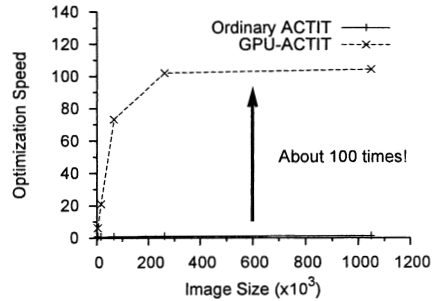


図 4 従来の ACTIT と GPU を用いた ACTIT の比較実験結果
Fig. 4 Experimental results of comparison of ordinary ACTIT and ACTIT using one GPU.

4.2 従来の ACTIT と GPU を用いた ACTIT の比較実験

始めに、従来の ACTIT と 1 基の GPU を用いた ACTIT との比較実験を行なった。横軸を画像サイズ、縦軸を処理速度としたグラフを図 4 に示す。なお、図中の処理速度とは、従来の ACTIT で画像サイズが $64 \times 64[\text{pixel}^2]$ の場合を 1 としたときの単位画像サイズあたりの処理速度である。

実験結果から、GPU を用いた ACTIT では従来の ACTIT と比べて、画像サイズが小さい場合 ($64 \times 64[\text{pixel}^2]$) では 5 ~ 10 倍、画像サイズがある程度大きい場合 ($256 \times 256[\text{pixel}^2]$ 以上) では 80 ~ 100 倍高速となっている。GPU は大量の入力データそれぞれに対して、単純で高密度な演算を施して出力するような処理を得意としているため、GPU を用いた ACTIT では画像サイズがある程度大きい場合に特に有効であることが分かる。進化的画像処理を実問題に応用する場合、扱う画像のサイズや枚数は大きいことが予想される。このため、GPU を用いた ACTIT の高速化は非常に有効であると考えられる。

4.3 GPU を用いた ACTIT の並列化実験

複数の GPU を用いた ACTIT と、1 基の GPU を用いた ACTIT との比較実験を行なった。並列モデルとして Master-Slave モデル、Parallel-MGG モデル、そして各ノードに待合室を設けることによって、通信時間が削減されるように改良した Parallel-MGG モデルを用いて実験を行ない、それぞれを比較した。GPU の数は 1 ~ 4 基とした。横軸をノード数、縦軸を処理速度とするグラフを図 5 に示す。なお、図中の処理速度とは従来の ACTIT の処理速度を 1 としたときの単位画像サイズあたりの処理速度である。

図 5 から、Master-Slave モデル、Parallel-MGG モデルでは同期・通信時間の影響からノード数を増やしてもあまり高速化することができていないことが分かる。しかし、同期・通信時間が削減されるように改良した Parallel-MGG モデルではノード数 2 で約 2.0 倍、ノード数 4 で約 3.8 倍になり、効率的に並列化されて

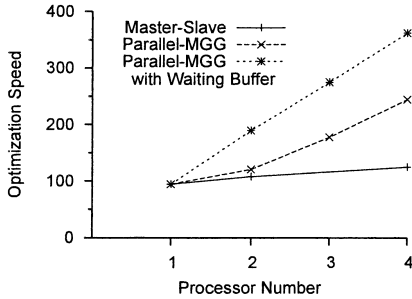


図 5 GPU を用いた ACTIT の並列化実験結果

Fig. 5 Experimental results of ACTIT using multiple GPUs.

いることがわかる。その結果、4基のGPUを用いたACTITによって、従来のACTITの約360倍まで高速化することができた。

5. おわりに

本論文では、ACTITの画像処理フィルタをグラフィックスボードのGPU上で実装し、処理の高速化を行なった。提案手法では負荷の高い処理のほとんどを高速なGPU上に移すことに成功した。加えて、そのシステムを並列化することによって更なる処理の高速化を図った。そして、比較実験を行なったところ、従来のACTITよりも数百倍高速となり、複数のGPUを用いた画像処理の有効性が示された。

本研究を発展させるためには、さらに考慮すべき要因がいくつか存在する。今回、実験するにあたり、GPU上で実行可能な単純なフィルタだけを用いたが、複雑なフィルタへの対応は不可欠である。また、複数のGPUを用いた新しい並列モデルの提案なども行ない、さらなる高速化が必要である。加えて、処理が高速になったことで、余った時間をより広い範囲の探索に割り当てることによって、解の高品質化も行なうことができる。以上のことを考慮し、超高速進化的画像処理システムの構築を目指す。

参 考 文 献

- 1) Koza, J.R.: *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
- 2) 青木紳也, 長尾智晴: 木構造状画像変換の自動構築法 ACTIT, 映像情報メディア学会誌, Vol.53, No.6, pp.888-894 (1999).
- 3) Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A. E. and Purcell, T. J.: A Survey of General-Purpose Computation on Graphics Hardware, *EURO-GRAPHICS 2005* (2005).
- 4) Fung, J., Mann, S. and Aimone, C.: Open-

VIDIA: Parallel GPU Computer Vision, *Proceedings of the ACM Multimedia 2005*, Singapore, pp.849-852 (2005).

- 5) Fok, K.L., Wong, T.T. and Wong, M.L.: Evolutionary Computing on Consumer Graphics Hardware, *IEEE Intelligent Systems*, Vol. 22, No.2, pp.69-78 (2007).
- 6) Harding, S. and Banzhaf, W.: Fast Genetic Programming on GPUs, *Proceedings of the 10th European Conference on Genetic Programming*, Valencia, Spain (2007).
- 7) Chitty, D. M.: A Data Parallel Approach to Genetic Programming Using Programmable Graphics Hardware, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, London, England, pp.1566-1573 (2007).
- 8) Holland, J. H.: *Adaptation in Natural and Artificial Systems*, The Univ. Michigan Press (1975), MIT Press (1992).
- 9) Goldberg, D.E.: *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison Wesley (1989).
- 10) 村 英敏, 安藤 淳, 長尾智晴: PC クラスタを用いた木構造状画像処理の高速化に関する研究, 映像情報メディア学会技術報告, Vol.30, No.17, pp.87-88 (2006).
- 11) 佐藤 浩, 小野 功, 小林重信: 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会誌, Vol.12, No.5, pp.734-744 (1997).