Vol.2009-MPS-74 No.11
2009/7/13

# CellBroadbandEngine への 遺伝的プログラミングの最適化

白川有莉† 美坂千穂† 石川千里† 高田雅美† 城和貴†

本研究では，CellBroadbandEngine™を用いて，遺伝的プログラミング(GP)の高速化を目指す．GP は，構造的表現を用いるため，計算時間が膨大である．そこで，マルチコア化を行い，並列処理を可能とする．また，木構造で表現された遺伝子情報を配列に置き換え，DMA ダブルバッファリングを用いることによって，データ容量の問題を回避する．配列を用いることによって，データアクセスの時間の短縮にもつながる．GP の適応度計算では，各遺伝子に対しても同じ計算式が利用される．そこで，SIMD 化を用いて実行時間の短縮を試みる．

# Optimization of a GP Application for the Cell Processor

Yuri Shirakawa† Chiho Misaka† Chisato Ishikawa†
Masami Takata† and Kazuki Joe†

In this paper, we report some optimization techniques for genetic programming (GP) on Cell processors. GP is computationally intensive because of large variety of structural representations. We parallelize a GP program for multi core processors. The genetic information expressed by GP's tree structure is replaced with array data. We solve the problem of huge data capacity by DMA double buffering for Cell processors. In the fitness calculation of GP, the same calculations are used by gene. We transform the calculation into SIMD operations to get more speed up.

## 1. はじめに

デジタルデバイスの能力が向上するとともに，携帯電話やデジタルカメラ，プリンターなどのデバイスが高性能かつ比較的低価格で手に入るようになり，身近に画像処理技術の恩恵を得ることが増えている. 例えば, 顔認識や指紋認証などが挙げられる．しかしながら，このような画像処理は現在，人がアルゴリズムを 1 つずつ考案しているため，処理目的や対象が限定されてしまっている．そのため，ある目的のために作成された画像処理フィルタは，ほぼ同じ処理をおこなう場合でも目的が異なると使用できないことが多く，汎用性が低くなる．そこで，学習対象を変更することで、対象に合ったプログラムを自動的に生成することが出来る遺伝的プログラミング（GP）を用いて，画像処理フィルタを自動生成することが望まれる．

しかしながら，GP の特徴である構造的なデータ表現は，その特性ゆえに操作に時間が掛かる．また，学習用の画像が大きい場合や枚数が多い場合には，多大な計算時間が必要となるため，高速化が求められている．高速化の手法としては，ハイスペックな計算機を利用する方法があるが，これは非常に高価であり，容易に利用することはできない．

そこで本研究では, PLAYSTATION3(以下 PS3)のための GP アルゴリズムを開発することで, 高速化をはかる. 本研究で PS3 を用いる理由として, 2 つの点が挙げられる．まず 1 つは，Cell Broadband Engine(以下 Cell)を搭載している点である．このプロセッサは世界最速のスーパーコンピュータ Roadrunner にも使用されている．もう 1 つは，単体では安価であり，容易に手に入ることができるからである．

本稿は，2 章で GP について説明し，3 章では，Cell への実装の高速化手法についての提案を行い，4 章で実験，5 章でまとめと今後の課題を述べる．

## 2. 遺伝的プログラミング（GP）

### 2.1 GP の概要

本章では，GP の概要とその基本的なプログラムの流れについて述べる．

GP は，進化的計算法，あるいは進化的アルゴリズムとよばれる研究分野の 1 つであり, 進化論的な考え方に基づいてデータを操作し, 最適化問題を, 解く手法である．

同じ進化的計算のひとつに Genetic Algorithm(GA)がある．生物は，さまざまな環境に適応しながら進化していく．環境への適応が高い遺伝子ほど生き残り，その中で交叉し子孫を残すことを繰り返すことで，より環境に適した優秀な子孫を残すことができる．この進化の生物学的機構にヒントを得て，データ構造を変形，合成，生成する

†奈良女子大学大学院人間文化研究科
†Graduate School of Humanities and Sciences, Nara Women's University

ことを目指し考案されたのが GA である．GP は，これを拡張したもので，GA が遺伝子型を配列で表現するのに対し，木構造のデータを用いることが特徴である．

GP はさまざまな分野に応用されており，その有効性が確かめられている．適用例として，パターン認識やデータベース，分子生物学，ロボット制御，政治・経済など，多岐に渡っている．

### 2.2 GP の処理手順

GP の基本的な処理手順は GA と同様で，図 1 のようになる．まず，初期個体群を生成し，個体それぞれの適応度を求める．適応度は最初に目標となる教師用データを与え，その教師用データと比較することで計算可能である．評価は，適応度が一定の値を超えた場合，または，世代数が一定の値を超えた場合に終了となる．終了条件を満たさない場合，適応度に基づき次世代を残すことの出来る親個体を選択し，選択された個体の中で交叉・突然変異を起こす．交叉・突然変異にはそれぞれ確率が設定してあり，個体が選ばれても次世代となる子孫を残すことができるとは限らない．また，突然変異確率は，自然界と同様に交叉確率と比べ非常に小さい．次世代決定後，再び適応度計算と評価に戻る．これらの操作を繰り返すことにより，教師用データに近い解答を探すことが可能となる．そのため，GP は最適化問題の解法として有効である．

GP では，木構造を用いることで GA では表現することができなかった数式やプログラムのコードを扱うことができる．生成したいプログラムにより木構造の終端要素，非終端要素は異なるが，これらをランダムに組み合わせることにより，未知の関数の同定や，処理アルゴリズムが未知の問題に対する有効なプログラムを自動生成することができる．この組み合わせを決定するために，GP では，複数の部分木に対して交叉や突然変異を起こすことにより，部分木の組み合わせを変更する．交叉は，2 つの親のそれぞれに対してランダムに選択した部分木を交換する．突然変異は，ある節をランダムに別の節や部分木に交換することをいう．交叉や突然変異によって生成された部分木のうち，いくつかを選択して次世代に残す．その選択法として，トーナメント選択法，ランキング選択法，エリート選択法，ルーレット選択法がある．トーナメント選択法は，いくつかの個体をランダムに選択し，その中の最良個体を選択する．これを個体数だけ繰り返すことで個体群を選択する方法である．ランキング選択法は，あらかじめ順位により選ばれる確立を決めておく方法であり，各個体を適応度によりランク付けし，それによって個体の選ばれる確率を決定し選択する．エリート選択法は，現在の最良個体を次世代に必ず残す戦略．これにより，個体群全体の最大適応度は単調増加する．ルーレット選択法は，適応度が大きいほど選ばれやすく，小さいほど選ばれにくい選択法であるが，適応度が低くても選択される可能性もある．これにより，個体群が多様化し局所解に陥りにくい．

### 2.3 画像処理フィルタ作成のための GP

本研究で高速化する GP は，画像処理フィルタを自動生成するものである．プログラムの流れは，次のようになる．初めに，初期個体数，世代数をパラメータとして与え，さらに，原画像と教師用データを与え実行する．初期個体をパラメータで与えた個数分生成し，全ての個体の適応度を求める．適応度は，原画像を自動生成されたプログラムを用いて処理を行い，その結果と教師用データとの値の比較により行う．適応度を求める式は(1)である．ここで $W_x$，$W_y$，$V_{\max}$ は画像の縦幅，横幅，輝度値の最大値であり，$O$，$T$ がそれぞれ出力画像と目標画像である．出力画像と目標画像の各画素での差の和が小さいほど，出力画像が目標画像に近いことになる．この式から，適応度の最大値は 1 となる．

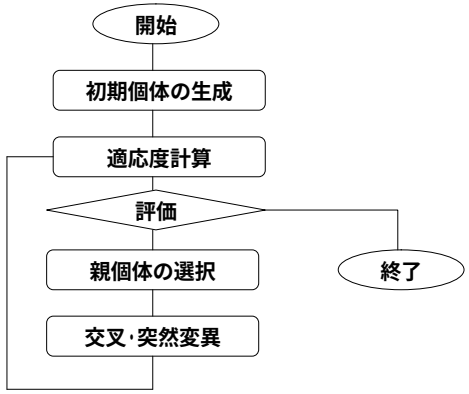


図 1　GP の基本的な流れ

$$fitness = 1 - \frac{\sum_{x=1}^{W_x}\sum_{y=1}^{W_y} |O(x,y) - T(x,y)|}{W_x * W_y * V_{\max}} \quad (1)$$

求めた適応度を基準とし，世代交代の際に，適応度の低いものを新しい個体と置換していく．これを繰り返していくことで，目的の処理に近い画像処理フィルタを生成することができる．ただし，適応度が 1 となるか，または，パラメータで与えた世代数を超えた場合は，終了とする．

## 3. Cell における最適化手法

本章では，遺伝的プログラミングの問題点を挙げ，これを解決するための高速化手法について述べる．

3.1 節では，高速化に使用する Cell プロセッサについて述べ，3.2 節で，GP の問題点とその解決方法について述べる．3.3 節以降では，解決に用いた高速化手法について述べる．3.3 節では，マルチコア化について，3.4 節では，DMA ダブルバッファリングについて，3.5 節では，SIMD 化について述べる．

### 3.1 Cell

Cell は，マルチコア方式のプロセッサである．PS3 を初め，一部のサーバーやワークステーションなど，様々な製品に採用されている．Cell は，1 基の制御系プロセッサコア（PPE）と 8 基の演算系プロセッサコア群（SPE）で構成される．これらのプロセッサコアは，ElementInterconnect Bus(EIB)と呼ばれる高速なバスで接続されている．また，EIB はメインメモリや外部入出力デバイスとも接続されており， 各プロセッサは EIB を経由してデータアクセスをおこなう．また，SPE はメインメモリと独立しており，プログラムやデータは一旦，メインメモリから SPE のローカルメモリである LS に転送しなければならない．

本研究で使用する Cell プロセッサは PS3 に搭載されているものである. ただし, PS3 に搭載されている Cell プロセッサは，SPE が 7 基となっている．しかし，この 7 基のうち 1 基は故障した場合のためであり，実際に使用できるのは 6 基となっている．このため，本研究で使用できる最大 SPE 数は 6 基である．

### 3.2 問題点と解決方法

GP には，多大な計算時間を要するという問題点がある．特に必要とする部分は 2 箇所あり，1 つは，交叉部分で，もう 1 つは，適応度計算部分である．今回高速化の対象とする GP は画像処理を目的としているため，特に適応度計算には多くの計算が必要になると予測される．そこで，本研究ではマルチコアプロセッサである Cell プロセッサの SPE を用いて，適応度計算部分をマルチコア化し，適応度計算を並列処理することで，この問題を解決する．マルチコア化について詳しくは，3.4 節で述べる．

SPE のローカルメモリサイズは 256KB ととても小さく, 適応度計算に必要となるデータ全てを 1 度にローカルメモリに保持することができない．特に，画像データはデータ量が多い．そこで，DMA ダブルバッファリングで用いるバッファ領域を活用する．計算に必要な画像データのみを，DMA 最大転送サイズと同サイズのバッファに DMA 転送することにより，SPE でのローカルメモリに占める画像データの割合は少なくなり，メモリの使用量も削減できる．また，データをダブルバッファリングで転送することで，プログラムの待機状態をなくし，処理時間の短縮を図る．DMA ダブルバッファリングについては，3.4 節で詳しく述べる．

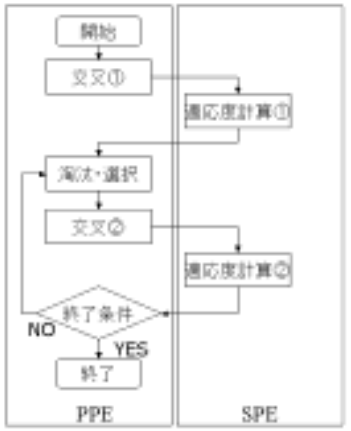


図 2　Cell 用プログラムの流れ

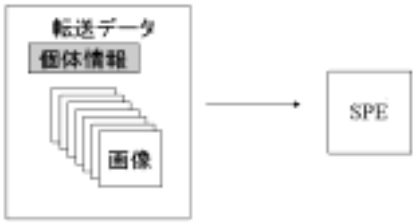


図 3　1 個体・SPE 1 基での転送データ

第 2 章でも述べたように，GP は各個体の持つ遺伝子情報等を木構造として表現しており，木構造は深くなるにつれ，使用メモリは多くなる．また，木構造にアクセスする場合, ポインタアクセスとなるため, データアクセスに時間がかかることがある．このように，木構造にはデータアクセス上の欠点があると言える．前述したように，本研究では，適応度計算を SPE 上で行うため，木構造で表現された遺伝子情報を含む各個体の情報を SPE へ DMA 転送する必要がある．しかしながら，画像データ同様，SPE 上で使用できるメモリサイズは限られているため，データ構造を木構造のまま使用するとメモリ不足が起こると考えられる．そこで，本研究では，適応度計算に必要な遺伝子情報のみを一旦 PPE 上で取り出し，配列にその情報を代入する．この配列データを PPE から SPE へ DMA 転送し，適応度計算に使用する．このとき，配列で表現

されている遺伝子情報に，木構造と同様にアクセスできる仕組みを作ることで，SPE 上での使用メモリを削減する．さらに，木構造から配列へとデータ構造が変更されたため，ポインタアクセスによるデータ参照を行わないため，遺伝子情報へのアクセスが若干速くなることが期待できる．加えて，遺伝子情報を PPE から SPE へ転送を行う場合にも，配列を使用することから，何度も転送を行う必要がなくなる．

適応度を求める際に行われる計算は，全画素に対して行われる．このため，用いる画像サイズが大きくなるにつれ，計算回数も多くなる．そこで，計算式に SIMD 化を行い，同時に計算を実行していくことで，さらに高速化を目指す．SIMD 化については，3.5 節で詳しく述べる．

### 3.3 マルチコア化

本研究で用いる画像処理フィルタを作成する GP のプログラムの流れは，図 1 のよう
になる．これをマルチコア化するにあたり，元の GP のプログラムを Cell 用に移植しなければならない．そこで，GP のプログラムから適応度計算を行う部分のみを SPE 用に，それ以外の部分を PPE 用としてプログラムを開発する．

Cell 用プログラムの流れは図 2 のようになる．まず，PPE で交叉①が行われ，個体が生成される．ここで生成される個体数はパラメータによって指定された数である．次にこれら個体全てに対して，SPE で適応度を求める．適応度計算①においては，存在するすべての個体に対して行う．このため，適応度計算①は，SPE1 基で 1 個体の計算をすることにする．このときの割り振り方は次のようになる．

$$\text{各SPEで計算する個体数} = \frac{\text{全個体数}}{\text{使用SPE数}} \qquad (2)$$

次に，データの転送について説明する．前述したように，SPE から直接メインメモリは参照できないため，一度 SPE の LS へと転送する必要がある．そこで，PPE から各 SPE へ，計算に必要となる遺伝子情報や，原画像と教師用データの転送を行う．

適応度計算①では，1 個体の適応度計算をすべて同じ SPE で行うため，図 3 のように，各 SPE に対して，ある 1 個体の遺伝子情報とすべての画像データ計 6 枚分（原画像 3 枚・教師用データ 3 枚）を転送することになる．

SPE 上で適応度計算②を求めた後，その値を使用し，PPE 上で淘汰・選択が行われる．ここで，出来た個体の中から適応度の低いものを 2 つ選出し削除し，その削除された個体に代わるものを，交叉②で生成する．そして，新しくできた 2 つの個体に対して，適応度計算②を行う．この計算もまた前述したように，SPE 上で行う．ここでは，適応度を求める個体数は 2 つであるから，SPE3 基を使用し，1 つの個体について計算を行う．この場合，データ転送は，図 4 のようになる．各 SPE には，同一の個体

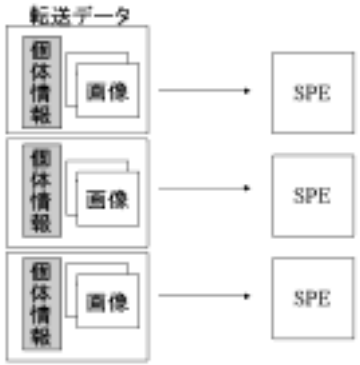


図 4　1 個体・SPE3 基でのデータ転送

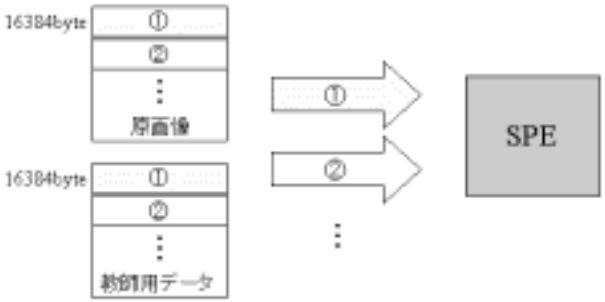


図 5　PPE から SPE への画像データ転送の様子

情報と，適応度計算に必要となる画像データ 2 枚を転送する．画像データは，原画像と教師用データそれぞれ 1 枚ずつである．ただし，適応度計算②では，1 個体に対し，SPE3 基を使用しているため，各 SPE で求められた値を PPE に転送し，最終的に PPE 上で，適応度を求めることになる．

最後に終了条件を満たしているかを判定する．満たしていれば終了，満たしていなければ，再び淘汰・選択へ戻る．

以上のように，適応度計算部分をマルチコア化し，高速化を目指す．

### 3.4 DMA ダブルバッファリング

効率的に画像データを DMA 転送するために DMA ダブルバッファリングを用いる．

3.4 節で述べたように，本研究では，適応度計算をマルチコア化し，各 SPE で行う．
このため，計算に必要な個体情報や画像データをメインメモリから各 SPE の LS へ DMA 転送する必要がある．しかし，前述したように，DMA 転送は 1 度に転送できる最大サイズが決まっている．そのため，画像データを分割し，転送する必要がある．

図 2 での適応度計算①において，PPE から SPE へ転送する画像データは，すべての原画像と教師用データである. データ転送の様子は図 5 のようになる. 転送は原画像，教師用データの 2 つを共に行う．また，このときの転送サイズは常に DMA 転送の最大値である 16384byte とする．よって，このデータサイズで画像データを図 5 のように区切り，① から順番に PPE から SPE へ，データの転送を行う．

適応度計算①においては，転送する画像はすべてであることから，それぞれ画像は 3 枚ずつ，計 6 枚転送を行う．このときの転送回数は(3)の式で求めることができる．

$$\text{転送回数} = (\text{Width}*\text{Height}/16384)*(\text{画像の種類}) \quad (3)$$

この画像の種類というのは，各原画像と教師用データの種類のことを指す．GP でデータとして扱う画像データは，原画像と教師用データなのだが，教師用データというのは，原画像に，ある処理を行ったものであり，元は現画像と同じ画像である．つまり，原画像と教師用データは 1 枚ずつ 1 組みになっている．

次に，図 2 での適応度計算②の転送手順について説明する．PPE から SPE への画像データの転送の流れは，図 5 と同様であるが，この場合，3.3 節でも述べたように，1 個体に対して SPE3 基を使用し，計算を行う．このため，転送回数は上記と異なり，次のように求めることができる．

つまり，適応度計算②では，各 SPE へ転送される画像データは，原画像，教師用データ，それぞれ 1 枚ずつである．

$$\text{転送回数} = (\text{Width}*\text{Height}/16384) \quad (4)$$

本研究では，SPE から PPE へは適応度計算部分で求められた値のみを転送する．画像データを転送し終われば，このデータに対して適応度計算を行う．これと同時に，次の画像データが転送される．そして，計算が終了後，SPE で求められた値を，SPE から PPE へ転送を行う．以後，これの繰り返しである．本研究では，このようにデータ転送に対してダブルバッファリングを行い，高速化を図る．

### 3.5 SIMD 化

SIMD 化を行った箇所は，適応度を求めるプログラムの一部分であり，この部分は大きく 2 つに分けられる．

1 つは, 個体の遺伝子情報から実際に出来上がった画像処理を, 原画像に対して行い，各画素値を求める部分である．この部分は，遺伝子情報を元に，再帰呼び出しを何度も行い，各画素値に対して処理を行う．再帰呼び出しは，コーディングを容易に行うことができるが，処理速度が落ち，スタックのオーバーフローが発生するという点がある．そこで，この再帰呼び出し部分を，非再帰呼び出しに変更する．この際，再帰呼び出しを行っていた部分の画像データを参照する計算式に SIMD 化を行う．

もう 1 つは，適応度を導き出す直前の計算部分である．この部分では，まず，前述した各画素値を求める部分で導き出された値と, 教師用データの画素値との差を求め, 対象画像の輝度値との差を求める. 本研究では, この部分に関しても SIMD 化を行う．

## 4. 実験と考察

本章では，第 3 章で述べた手法の有効性を実験により確認する．使用する原画像と教師用画像データとして，128×128，256×256，512×512 のものをそれぞれ 3 枚用意する．個体の選択法は，ルーレット選択法とする．交叉率は，0.9，突然変異率は，0.1 である．本実験では，高速化手法を組み合わせて実行時間の計測を行う．ただし，全ての組み合わせに対して実験するのではなく，有効だと考えられる下記のものについて行う．

(ⅰ)マルチコア化
(ⅱ)マルチコア化とダブルバッファリング
(ⅲ)マルチコア化と SIMD 化
(ⅳ)マルチコア化と SIMD 化とダブルバッファリング

高速化率を確認するため，PPE のみで実行した場合の時間計測を行う．表 1 は，計測された実行時間の秒を表で表したものである．

4.1 節では，画像データ 128×128 を用いて実験を行い，考察する．同様に 4.2 節では，画像データ 256×256 を用いて，4.3 節では，画像データ 512×512 を用いる．

### 4.1 実験データ 128×128

PPE のみで実行した場合，約 1559 秒である．これに対して，(ⅰ)の場合は，約 109 秒である．これは，マルチコア化することにより，PPE 単体に比べ約 14 倍高速化されている．

(ⅰ)と(ⅱ)とを比較する．結果はそれぞれ，約 109 秒，約 111 秒となっており，実行時間にほとんど差はない．これは，PPE から SPE へ転送するデータサイズが 128×128 であるため, DMA ダブルバッファリングを必要としないからである．また, (ⅰ)と(ⅳ)とを比較した場合でも，ほとんど差がないことから，DMA ダブルバッファリングは効果を発揮していないことがわかる．

SIMD 化による影響について考える．(ⅰ)と(ⅲ)の実行時間はそれぞれ，約 109 秒，約 92 秒となっている．これは，SIMD 化したことで高速化されたと考えられる．

以上より，データサイズが 128×128 であるとき，高速化に有効な手段は，(ⅲ)である．このとき，PPE のみを使用としたプログラムの実行時間と比較して，約 17 倍の高速化に成功していることがいえる．さらに，ダブルバッファリングを用いることによって，若干ではあるが，より高速な実行が可能である．

### 4.2 実験データ 256×256

データサイズ 256×256 について考察を行う．PPE のみでの実行時間は，約 7533 秒となる．SPE6 つを使用して行ったマルチコア化したものは約 390 秒となり，約 20 倍高速化に成功している．

DMA ダブルバッファリングに着目する．(ⅰ)と(ⅱ), (ⅲ), (ⅳ)を比較すると，若干，実行時間は減少している．しかしながら，はっきりと DMA ダブルバッファリングの効果を見ることはできない．そこで，(ⅰ)と(ⅲ)とを比較する．適応度計算の一部を SIMD 化することにより，実行時間が減少していることがわかる．また，(ⅱ)と(ⅲ)を比較してもわかるように，DMA ダブルバッファリングではなく，SIMD 化の影響により実行時間が減少していることがわかる．

以上のことから，データサイズ 256×256 で，有効な高速化手法の組み合わせは，(ⅲ)が挙げられる．この組み合わせにより，実行時間は約 25 倍高速化されている．本実験では，データサイズが LS のサイズを超えている．そのため，DMA ダブルバッファリングの効果が得られるはずである．しかしながら，実験結果からは，優位性がみられない．この点については，別の論文で議論する．

### 4.3 データサイズ 512×512

データサイズ 512×512 の場合について考察を行う．PPE のみを使用した場合は約 37500 秒となる．SPE6 基を用いてマルチコア化したものは約 1475 秒となり，実行時間を比較すると約 25 倍高速化されている．

まず，DMA ダブルバッファリングに着目する．(ⅰ)と(ⅱ)とを比較すると，データサイズ 256×256 の場合と同様に，DMA ダブルバッファリングの効果は見られない．しかしながら，(ⅱ)と(ⅳ)とを比較した場合では，実行時間に大きな変化が見られる．

表 1　実験結果

| | PPE only | (ⅰ) | (ⅱ) | (ⅲ) | (ⅳ) |
|---|---|---|---|---|---|
| 128×128 | 1559 | 109 | 111 | 92 | 93 |
| 256×256 | 7533 | 390 | 387 | 303 | 300 |
| 512×512 | 37500 | 1475 | 1480 | 1404 | 1016 |

(ⅲ)と(ⅳ)とを比較しても，実行時間に大幅な減少が見られる．これは DMA ダブルバッファリングの効果なのではないかと考えられる．ところが，前述したように(ⅰ)と(ⅱ)との比較からは優位性は認められない．この要因として考えられるのは，ダブルバッファリングと SIMD の関係性である．この 2 つの組み合わせを，データサイズ 512×512 に用いることで，最も効果がでるものと考えられる．ただし，その原因の詳細については，別論文で議論する．

以上より，データサイズ 512×512 の場合において，高速化手法の組み合わせは，(ⅳ)が最も有効であるといえ，この組み合わせにより約 36 倍高速化に成功している．

## 5. まとめと今後の課題

本研究では，画像処理を目的とした GP をマルチコアプロセッサである Cell に実装することにより，高速化を行った．高速化手法として，マルチコア化，DMA ダブルバッファリング，SIMD 化を用いる．これらのそれぞれに対する有効性を検証するために，各手法を組み合わせて実験を行った．実験に使用したデータは 128×128，256×256，512×512 である．データサイズ 128×128 に有効であった手法は，(ⅲ)と(ⅳ)の 2 つである．この手法により，約 17 倍高速化に成功している．データサイズ 256×256 に有効であった手法は，(ⅲ)と(ⅳ)の 2 つである．この手法により，約 25 倍高速化に成功している．データサイズ 512×512 に有効であった手法は，(ⅳ)である．この手法により，約 36 倍高速化に成功している．

今後の課題として，DMA ダブルバッファリングの効果をえられていない原因の追求があげられる．さらに，4.3 節でのデータサイズ 512×512 に対しての実験で指摘した，DMA ダブルバッファリングと SIMD の関係性についても追求する必要がある．

また，SIMD 化をさらに進めていくことが重要である．現在は，1 個体に対して適応度を求めているが，今後は，複数の個体に対して，同時に適応度を求める手法を考え，実装を行うことで，さらに高速化を実現できるのではないかと考えている．

## 参考文献

1) 長尾智晴　進化的画像処理　昭晃堂　2002 年
2) 伊庭斉志　遺伝的プログラミング　東京電機大学出版局　1996 年
3) 伊庭斉志　遺伝的プログラミング入門　東京大学出版会　2001 年
4) PLAYSTATION®3 Linux Information Site http://cell.fixstars.com/ps3linux/index.php/メインページ
5) Young risk taker http://rakuto.blogspot.com/2006/12/cell-dma_10.html
6) PS3 と Linux，電子工作も http://todotani.cocolog-nifty.com/blog/2007/06/celldma_b09f.html
7) PS3Linux で Cell プログラミング! http://plaza.rakuten.co.jp/miyazblog/