

85-86

論 説

C言語によるシステムLSIのアーキテクチャ最適化手法と設計事例による
評価について 神戸 尚志・山口 雅之 1

ベジエ曲線によるダイコンの形状記述とその応用
..... 淡 誠一郎・淡 裕美子・吉田 康子 27

実 践 報 告

発達障害グレーゾーンの大学生を観察でアセスメントする方法の一試案
～大学教員ができる発達アセスメント～ 山田 章 59

彙 報 87

掲載論文要旨 89

大阪学院大学
人文自然学会

2023

C 言語によるシステム LSI のアーキテクチャ 最適化手法と設計事例による評価について

神 戸 尚 志・山 口 雅 之

C Based System LSI Architecture Design Methodologies and its Evaluation Using Several Applications

Takashi Kambe・Masayuki Yamaguchi

あらまし

システム LSI は携帯電話, 自動車, AV 機器などの各種電子機器に広く利用され, 年々複雑化・高度化が進んでいる. システム LSI の設計法として, C 言語による回路設計手法が注目されている. 本文では, C 言語設計における, データアクセス法, 並列化・パイプライン化, アプリケーションに特化した演算回路設計, モジュール構成の変更等のアーキテクチャ最適化の適用について, 粒子追跡技術における相関値計算のハードウェア化を例に説明し, その後, 幾つかの設計事例を用いて, 回路の高速化に対する面積増加の比率 (Performance/Area Ratio, 以下, PAR) を用いて, アーキテクチャ最適化手法の効果について適用例ごと及び手法ごとに評価する.

キーワード: C 言語設計, アーキテクチャ最適化, 粒子追跡技術, データアクセス法,
並列化・パイプライン化, 専用演算回路化

1. はじめに

近年の集積回路技術の急速な進展によってデジタル LSI の集積度は年々急増し、今や数千万ゲートを超える大規模なシステムを 1 つの LSI の中に実現するシステム LSI が携帯電話、自動車、AV 機器などのさまざまな電子機器に広く利用され、複雑化・高度化がますます進んでいる。各種アプリケーションを高速に処理するためには、アーキテクチャ設計の最適化が重要となる。この問題を解決するために、3 種類のアーキテクチャ設計手法がシステム LSI 設計のために提案されている。第 1 はプロセッサに基づいたアーキテクチャ設計である。これは、DSP[1]、ASIP[2]、MPSoC[3]、GPGPU[4]、Network on chip[5]など、特定のアーキテクチャに基づいている。この手法は大規模設計には向いているが、プロセッサタイプを選び、かつ/または、それを変更することが困難である。第 2 は、LISA[6]、EXPRESSION[7]などのようなアーキテクチャ記述言語に基づいたアーキテクチャ設計法である。アーキテクチャ記述言語を使用し、アーキテクチャ設計空間の探索とリターゲット可能なコンパイラ/シミュレータツールキットの自動生成を支援することができる。しかし、あらかじめ準備したアーキテクチャテンプレートや、言語セマンティックスによって制限されている。第 3 は本文で提案する C 言語に基づいたアーキテクチャ設計である。C 言語によるアーキテクチャ設計では、オリジナルの C 言語記述に対して、ビット精度指定、メモリ宣言やアクセス記述、並列動作記述、関数構造に対する変更、及びプラグマによるパラメータ指定などにより、様々なアーキテクチャを実現できる。

本研究では、第 3 のアプローチのひとつである Bach システム [8,9]を用い、幾つかの適用事例をもとに、C 言語によるアーキテクチャ最適化手法の評価を行う。

本論文では、第 2 章で C 言語設計システム Bach の概要を述べ、第 3 章で 4 種類のアーキテクチャ最適化手法を提案し、第 4 章で粒子追跡における相関値計算へのアーキテクチャ最適化手法の適用とその効果を述べ、第 5 章で回路の高速化に対する面積増加の比率 PAR を用いてアーキテクチャ最適化手法の効果について適用例ごと及び手法ごとに評価し、第 6 章で、結論を述べる。

2. C 言語設計システム Bach の概要

現在のシステム LSI の設計法は、いまだにクロックに同期したレジスタ間のデータ転送動作に着目した RT (Register Transfer, レジスタ転送) レベルでの設計が主流であり、設計にはハードウェア記述言語 (Hardware Description Language, HDL) が用いられる。しかし、RT レベルの設計では動作とクロック毎のタイミングを同時に設計するため、設計が詳細化され、アーキテクチャを短期間に設計することは困難である。また

設計段階での仕様変更による影響が大きいため、事前のアーキテクチャ検討に多大な時間を必要とする。これに比べ、C 言語等の動作記述から動作合成を用いてシステム LSI を設計する手法は、動作タイミングを動作合成ツールによって自動的に決定するので、設計者はアーキテクチャ設計に集中できる。動作合成とは、動作記述における各演算を実行する時間の決定（スケジューリング）や、各演算の演算器への割当て、各変数のレジスタへの割当て、演算器やレジスタの共有（アロケーション）を自動で行う技術である。

本章では、C 言語によるシステム LSI 設計について述べ、それを実現するために筆者らが研究に使用した Bach システムを紹介する。

2.1 C 言語によるシステム LSI 設計

一般に、C 言語によるシステム LSI 設計は以下の手順で行われる。

- (1) 要求仕様決定後、実現すべき機能全体を C 言語でプログラミングを行ない、これをコンパイル・実行する事により、アルゴリズム検証を行う。すでに記述されたプログラムがあればそれを流用する。
- (2) プログラムの構造、各関数の処理時間などを解析し、さまざまな実現方法を検討するためのアーキテクチャ探索を行う。設計対象のうち、どの部分をソフトウェアとして実現し、どの部分をハードウェアで実現するかについて、その分割を決定する。
- (3) ハードウェアで実現すべき部分について、処理に適切なサイズのモジュールに分割し、個々のモジュールを動作合成により、RT レベルに変換する。
- (4) 性能と回路規模についての要求仕様を満たすまで、必要ビット数の決定、データアクセス法の改良、並列化・パイプライン化、アプリケーションに特化した演算器回路設計、モジュール構成の変更などのため、C 言語記述を関数単位に変更し、C レベルシミュレーション・動作合成を行い、その効果を確認する。
- (5) 最後にソフト・ハード協調検証により、システム動作を確認する。

2.2 Bach システムを用いた回路設計

アルゴリズム設計によく用いられる C 言語に基づく“SystemC”[10]と呼ばれるシステムレベル設計言語と、それらを扱うシステムの標準化が提案され、最近、幾つかの市販システム[11]がこの言語をサポートしている。しかし、SystemC では、回路のアルゴリズム以外にクロックを記述中に明示せねばならないため、大規模な LSI を設計するためには多大な手間が必要となる。

ANSI C 言語を拡張したハードウェアアルゴリズム設計言語・ツール群として、シャープ株式会社が開発した Bach システムがある。Bach システムで使用される設計言語である Bach C 言語では、回路のアルゴリズムのみ記述すればよく、回路構造の詳細まで考慮する必要がない。このため、システム LSI のアルゴリズムを、C 言語などのソフトウェア

言語で記述するのと同様に自然に記述することができ、ハードウェア設計を効率化できる。また従来の C 言語で記述されていたアルゴリズム設計資産の再利用も容易である。

また Bach C 言語用の動作合成ツールである Bach コンパイラは、Bach C 言語で記述したクロックや回路構造の記述を含まないアルゴリズムのみの記述を動作合成して、短時間で回路化することができる。Bach コンパイラにより、アルゴリズム設計段階で様々な回路構成を設計・評価し、それらのアルゴリズムを回路化した場合の回路規模や処理時間を求めることができる。これにより、回路規模と処理時間のトレードオフを考慮したアルゴリズム設計ができ、アーキテクチャ設計を最適化することが可能になる。

2.2.1 Bach システム

Bach システムは厳格に仕様決定された言語と、記述から実際の回路に変換する動作合成ツール、検証・デバッグを補助するツールなどにより実用的な LSI 設計が可能であり、C 言語ベース設計を実現するためのもっとも有力な設計システムのひとつであるといえる。現在、大規模集積システム設計教育研究センター (VDEC) [12]を介して日本の大学へその利用が提供されている。

Bach システムは以下のツールで構成される。

- ・ Bach C 言語の並列記述を実行するインタプリタ型のソースレベルデバッガ (Bach デバッガ)
- ・ Bach C 言語の記述を、並列処理のスケジューラ内蔵の ANSI C 記述に変換し、コンパイルした後実行するコンパイラ型のシミュレータ (Bach シミュレータ)
- ・ Bach C 言語による機能記述から自動的に RT レベルの回路を合成するコンパイラ (Bach コンパイラ)
- ・ Bach C 言語の動作合成結果 (スケジューリング、アロケーション結果) のビューア、合成結果の RT レベルのネットリストのビューアなどのツール群

Bach デバッガや Bach シミュレータにより、Bach C 言語で記述したアルゴリズムの動作確認と評価ができる。また、そのアルゴリズム記述をハードウェア化したときの回路規模や処理時間などの性能は、Bach コンパイラと各種ビューアを用いて評価することができる。

また、Bach システムはアーキテクチャ最適化を実現するための各種プラグマをサポートしている。プラグマを Bach C 記述中に指定することにより Bach コンパイラによるハードウェア合成を制御することができる。プラグマには、ループパイプラインを自動生成し、そのスループットを指定する `throughput` プラグマ、専用演算回路化を実現する `preserve` プラグマ、ループを展開し自動並列化する `unroll` プラグマ、モジュール構成方法を指定する `internal_par` プラグマなどがある。

2.2.2 Bach C 言語

ハードウェアは基本的には並列動作する複数の部品によって構成される。ハードウェア化に最適なハードウェアアルゴリズムは必ずしも逐次動作を想定したソフトウェアアルゴリズムと同じものとは限らない。Bach C 言語はソフトウェアのアルゴリズム設計で一般に使用されている ANSI-C 言語をもとに、ハードウェアアルゴリズムの設計に最低限必要とされる機能を備えるため、以下に示す拡張を行っている。

(1) ビット幅

変数宣言時に変数のビット幅を指定することができる。ビット幅は#を前置した数字を型名と変数名の間に記述して指定する。また、型変換演算子も同様の書式で指定する。

(2) 並列記述

「par 構文」により、並列動作を明示的に記述できる。処理が par 構文に移ると、par 構文内の各ブロックまたは文を並列に実行し、全ブロックが終了した後、その par 構文の次の処理に移る。図2.1(a)では、 $a=b*c$ と $d=e*f$ が並列に実行される。

(3) データ通信

同期チャンネルまたは非同期チャンネルを用いて並列動作する処理間のデータ通信を明示的に記述できる。同期チャンネルと非同期チャンネルは par 構文の外で chan や achan というキーワードを用いて宣言する。同期チャンネルは、par 構文内の異なるブロックにある send 関数と receive 関数間でデータを転送する（図2.1(b)）。

同期チャンネルは1対1の片方向通信であり、データ転送が完了するまで次の処理に移らないのでデータの正当性を保障している。非同期チャンネルは、このような制限がなくいつでも読み書き可能であり、並列動作する複数のブロックから読み書き可能な共有変数としても使用する。

```
par {
    a = d * c ;
    d = e * f ;
}
```

(a) par 構文

```
chan int #8 ch ;
par {
    {
        int#8 send_data;
        ...
        send ( ch , send_data );
        ...
    }
    {
        int#8 receive_data;
        ...
        receive_data = receive ( ch );
        ...
    }
}
```

(b) 同期チャンネル

図2.1 Bach C 言語の並列記述

Figure2.1 A parallel process description using Bach C language

3. アーキテクチャ最適化手法

幾つかの C 言語ベース設計の結果をもとに、システム LSI のアーキテクチャを最適化する設計手法として、データアクセス法、並列化・パイプライン化、アプリケーションに特化した演算回路設計、モジュール構成の変更を提案する。この 4 種類の方法における性能と回路規模はトレードオフの関係にある。本章では、要求仕様の性能を確保しつつ、回路規模の増大を抑える最適化法を中心に考察する。

なお、回路規模の面からは、必要ビット幅の決定も重要である。ソフトウェア記述では、変数や配列は通常 16, 32, 64 ビット単位などで表現されているが、計算精度上必要となるビット数は一般にそれよりも細かいビット単位で規定できる。ビット幅指定により、演算回路、フリップフロップ、配線などを大幅に削減することができる。しかし、ビット数は演算中のオーバーフローやアンダーフローをも考慮する必要がある。Bach システムでは、ビットアキュレートな C レベルシミュレータを用い、実データによるビット指定との精度確認を行って必要最低限なビット数を決定することができる。本論文では、ビット幅指定後の回路についてそのアーキテクチャ最適化を考える。

3.1 データアクセス

システム LSI では、各種アプリケーションを高速に処理するためには、データ記憶及びアクセスについてのアーキテクチャ設計が重要である。近年のオンチップメモリの増大傾向に見られるように、各種データの記憶法及びアクセス法がコスト・性能などの点でシステム LSI 設計に大きな影響を与えるからである。

データアクセスアーキテクチャにおいて重要な点は、データアクセス時間がシステム LSI 性能のボトルネックとならないよう、オンチップにデータを保持しつつ、チップサイズを増大させないようオンチップのメモリサイズを最小化することである。この問題を解決するため、ハードウェア用のメモリ構成及びアクセスについて、データアクセス状況により次の 2 つの方法を適用する。

(1) 大規模データを複数回アクセスする場合

用途に特化した構成を持つ小規模オンチップメモリ（レジスタや SRAM）を設計し、頻繁にアクセスするデータを保持することで、外部メモリアクセスを削減し、処理を高速化する。

例えば、Bach C 記述中の配列は、Bach コンパイラによってレジスタとして実装するか SRAM として実装するか選択することができる。また、各演算回路が必要とするデータを複数のメモリに分割することによるメモリアクセス競合の解消や、同種類のデータの連結によって複数データの同時アクセスなども必要に応じて行う。

(2) データを順序的にアクセスする場合

メモリアクセスとそのデータに対する演算をパイプライン化することにより、現在のデータについて演算処理中に、次のデータをオンチップメモリに読み込む事で、データのアクセス時間を吸収でき、オンチップのデータ量を1回の演算に必要な量のみに削減できる。

また、対象プログラムが文字型の変数や複雑なポインタを用いている場合、整数型の変数やポインタ構造を配列型に変更することで高速化と回路規模の削減を図る。

3.2 並列化・パイプライン化

処理の並列化として、データ分割による並列化・パイプライン化について述べる。パイプライン化とは、処理を流れ作業で行う事によって、ハードウェアの性能を向上させるための手法である。

図3.1(a)の様に各工程が1単位時間かかる N 行程から成る処理を考える。単純に考えると、この処理を3回繰り返して終了させるためには $N \times 3$ 時間を要する。図3.1(b)の様に、これを N 人の人が流れ作業によって各行程を分担し、前の工程から受け継いだ製品に1単位時間で加工を施して、後の工程に引き継ぐようにする。この場合、元々の処理では N 時間に一つしか製品が完成しないが、流れ作業では見かけ上、1単位時間に一つの製品が完成することになる。つまり処理時間は約 N 倍に改善される。パイプラインでの各工程をステージという。 N 行程から構成されるパイプラインは N ステージパイプライン、または N 段パイプラインと呼ぶ。

C 言語ベース設計における並列化、パイプライン化は関数単位もしくは for 文単位に行う。まず、設計者はパイプライン化したい各モジュールの処理時間がほぼ同じになるようボトルネックとなる部分を高速化する。ボトルネックとなっている要因を解析し、メモリ

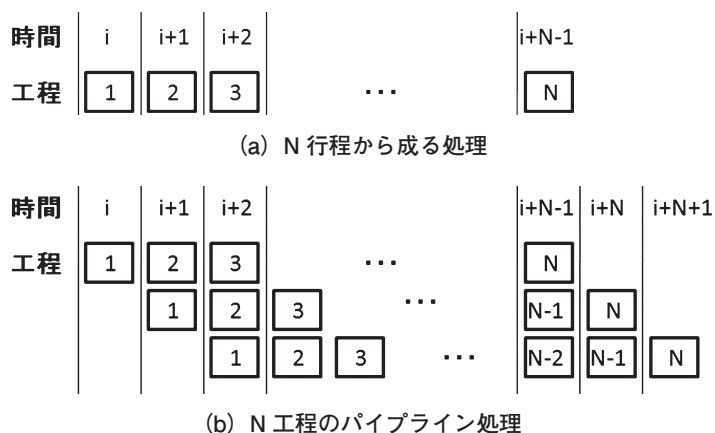


図3.1 パイプラインによる処理
Figure3.1 N stages pipelining

アクセスがネックの場合はデータアクセス法の改良、特定の演算がネックの場合は専用演算回路の設計、処理の手順や構成が問題であればモジュール構成の変更などを、BachC 記述の書き換えによって行い、高速化を図る。

データ分割による並列化は、大規模なデータに対して有効な場合があるが、並列度に応じて回路規模が増大する。また、分割境界線上で計算が正しく行われるようデータにオーバーラップを持たせるなど付加的な処理が必要になる場合がある。

パイプライン化は回路規模への影響が少なく、性能向上に有効であるが、各ステージの処理時間やパイプライン段数の決定が重要である。

Bach システムでは、for 文のパイプライン化は `throughout` プラグマを指定することにより、指定されたサイクル数のパイプライン回路が自動的に生成される。ただし、同期通信などサイクル数が不定の処理は除外する必要がある。

関数単位のパイプラインは、`send`, `receive` 関数を用いてモジュール間で開始・終了信号を送受信させることにより実現する。この場合、各パイプラインステージの処理時間に差があっても正しい動作を得る事ができる。

Bach システムでは通常は処理の各部で演算器を再利用し回路規模の増加を抑えるように合成される。演算器を再利用するようにスケジューリングされているループ処理などでパイプライン化を指定すると、処理時間を優先するため演算器が増加し、回路規模の増加の原因となる場合があるので注意が必要である。

3.3 アプリケーションに特化した演算回路設計

対象となるアプリケーションが限定されている場合には、ボトルネックとなる部分の処理をできるだけ高速化し、処理時間的に余裕がある処理の演算回路の規模をできる限り削減することによって全体の演算にかかる処理時間と回路規模のバランスを最適化することが可能である。このようなアプリケーションに特化した演算回路設計として、処理時間を重視した専用演算回路化と、回路規模を重視した演算回路個数の指定を行う。

数値演算などを繰り返し行う場合、その処理時間が全体のボトルネックとなることが多い。ハードウェアによる乗除算、各種関数、複合演算などが特に回路規模、処理時間共に影響が大きい。このような処理に対しては、演算回路化することで処理全体を演算器として実装可能となり、高速化の効果が大きい。

演算回路化では、処理が組み合わせ回路として実現されるため、演算器としての遅延が設計対象のクロック周期より長くなる場合がある。Bach システムではこの場合、自動的に演算回路を含むデータパスの遅延量を推定し、適切なサイクル数のマルチサイクルパスとしてスケジューリングを行う。これにより 1 サイクル同期の場合に比べ多くの処理を一度に行うことができ、また設計者は演算器の周辺回路の制御に時間を費やされることなく

動作設計に集中できる。

組み合わせ回路での実装のため処理内に不定回数のループなどを含むことはできないが、for 文などを用いた固定回数ループの記述に対しては、unroll プラグマを用いて自動展開を行うことで、ループを含む一連の処理を演算回路化できる。また、同様の理由でメモリアクセスや通信などの処理も含むことはできない。

C 言語ベース設計では、以上の手法を以下の手順で専用回路化を行う。

- (1) プログラムを解析し、多くの処理時間を要する演算部分を抽出し関数化する。このとき、専用回路化を行う上での制限事項に該当する処理が含まれていないか確認する。ループ処理に関しては固定回数であるか確認し、そうであれば unroll プラグマにて自動展開を指定する。
- (2) 対象処理内部の計算式の変形を行う。除算、平方根、平均値計算などを含む複雑な計算式に対しては、数学的に等価な計算に置き換え、単純化を行う。
- (3) 減算シフト法など各種ハードウェアアルゴリズムを Bach C 言語で関数記述し、これをコールする。また、演算回路を HDL で記述し、これをコールすることも可能である。
- (4) 演算回路化対象の関数に preserve プラグマを指定する事で他の部分との回路共有を無くし、高速化する。

また、この演算回路の数を指定することで回路規模を削減することが可能である。手法としては internal par プラグマの使用や、資源制約オプションなどがある。internal par プラグマを使用することで複数の回路モジュールの構成を変更することができる。これを利用して、演算器を共有化したいモジュールを統合することで、リソースの有効活用ができる。演算器指定オプションでは、演算器の個数を指定することができる。例えば、4bit 加算器 ADD4 の個数を 1 個にする場合、動作合成時に「-fc ADD4=1」とオプションを指定する。処理時間より回路規模を優先したい場合、このオプションを指定することで演算器の個数を制限でき、処理時間と回路規模のトレードオフを考慮した柔軟な回路設計ができる。

3.4 モジュール構成変更

逐次的に実行される処理間で大量のデータを転送する場合や、大容量のメモリを使用してデータを入出力する場合は、一度に処理するデータの量を限定したり、処理の順番を入れ替えるといった、処理モジュールの構成の変更を行うことで、入出力するデータ量を削減し高速化が可能となる場合がある。

また、データアクセス、並列化・パイプライン化、専用演算回路化などの高速化・回路規模の削減を進めると、ソフトウェアで実現する処理や、処理時間の少ないモジュールとの通信が、パイプライン化やメモリアクセス削減などの高速化に対するネックとなること

がある。問題のモジュールがソフトウェア部であれば、ハードウェア化による高速化を図る。ハードウェア部であれば、データ共有度を考慮し、他のモジュールと統合し、処理時間を吸収する。

このようなアーキテクチャの最適化は Bach C 言語によって記述されたハードウェアアルゴリズムに対し、並列性を考慮しながら修正し、Bach シミュレータなどで動作を確認しながら行うことになる。

4. 粒子抽出アルゴリズムへのアーキテクチャ最適化手法の適用とその効果

筆者らがこれまで Bach システムを適用した回路設計の一覧を表4.1に示す。ゲート規模及び高性能化は、元となる C 言語プログラムをもとに、コンパイル、シミュレーションした結果に対し、各種のアーキテクチャ最適化法を適用した後のゲート規模とシミュレーション時間を示し、括弧内には、ビット幅指定後の回路記述との倍率を示す。本研究では、Bach C 言語を用いて対象とする回路とそのテストベンチを記述し、Bach シミュレータにより動作確認・デバッグ後 Bach コンパイラにて動作合成を行う。その結果出力され

表4.1 Bach システムを適用した回路設計一覧

Table4.1 Summary of design results using Bach system

C 言語によるアーキテクチャ最適化手法とその評価

適用対象	ゲート規模 (gates)	高性能化 (ns)
DCT[17] (静止画圧縮)	71,315 (約0.08倍)	144,150 (約53.94倍)
相関値計算[15] (粒子抽出)	74,900 (約1.27倍)	21ms (約1,037倍)
音声認識[20] (第1パス部)(*)	195,587 (約4.05倍)	4,611 (約4.48倍)
リードソロモン[22] (16バイトの消失訂正)	11,704 (約0.60倍)	36,682 (約9.09倍)
SHA-1暗号化[28]	7,660 (約0.1倍)	345 (約8.8倍)
AES 暗号化[27] 鍵サイズ128ビット	167,283 (約1.69倍)	47.2 (約6.6倍)
RDOQ[26] (H264レート歪み最適化)	830,464 (約3.89倍)	4,038 (約24.9倍)
H264[24] (動き検出)	705,673 (約0.55倍)	37,328 (約4.81倍)

(*)第1パス部は、音素 HMM から状態遷移・出力確率値を、パイグラムから言語出現確率値を求め、文の候補を出力する。第2パス部は、トライグラムを用いて再探索・再評価し最終的な文の候補が出力する。ここでは、処理全体の約67%を占める第1パス部のハードウェア化を行った。

た RTL 記述 (VHDL 記述) を用いて, 処理速度の計測や回路動作のシミュレーションを業界標準ツール Modelsim を用いて行い, 回路規模測定と遅延計測のための論理合成を業界標準ツール Design Compiler を用いて行った. 回路の動作周波数は 100MHz であり, 論理合成に使用したライブラリは東京大学大規模集積システム設計教育研究センター (VLSI Design and Education Center, VDEC) 提供の HITACHI 0.18 μ m である.

本章では, 粒子抽出アルゴリズムへのアーキテクチャ最適化法の適用について述べる.

4.1 粒子抽出アルゴリズム

粒子追跡システムとは, 流れ場解析のための手法の一つである. 現在, 科学技術の分野で流れは様々な場面で出現する. 航空機・船舶・自動車などの交通機関の性能や安全性, あるいは高層ビルの耐風性や風害, 半導体チップの製造や冷却における流れの問題, あるいは医学における血流などの問題など, 流れ現象の解明と制御が科学技術の新たな展開のキーテクノロジーとなっている.

粒子追跡技術は, 流れ場を空間的に連続して測定する目的で開発されたものであり, 粒子マスク相関法 (PMC 法) により, 撮影画像から粒子を抽出する. 更に, PMC 法により抽出した粒子情報をもとに, カルマンフィルタ法と χ^2 検定を組み合わせた KC 法により, 粒子を追跡する.

図 4.1(a) は実際に水中を漂う粒子をカメラで撮影したものである. 実際の計測現場では水中の粒子画像を 1/15[s] 間隔で撮影する. 撮影画像は x 軸方向 1,008 画素, y 軸方向 1,018 画素の 8bit 輝度である. 図 4.1(b) はある時刻の粒子の流れを, 粒子追跡システムを用い, 追跡した様子である. 多点同時に粒子追跡されている様子が分かる.

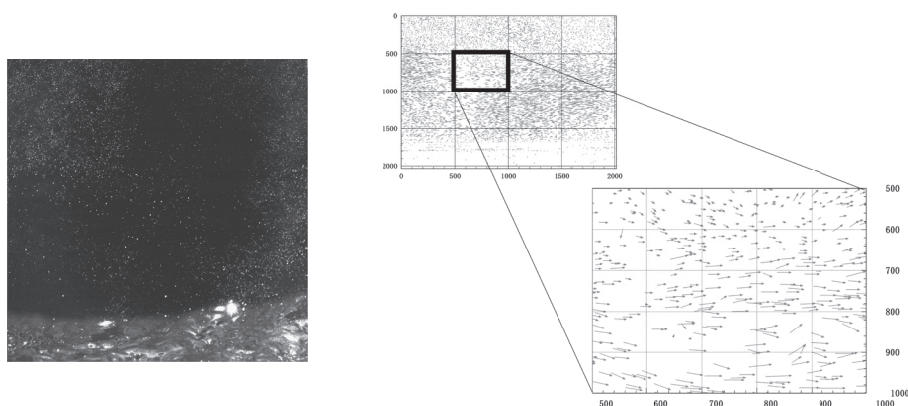


図 4.1 (a) 撮影画像

(b) 粒子追跡の様子

Figure 4.1 Photographic Image (a) and Particle tracking (b)

PMC 法は“テンプレートマッチング法”をもとにした粒子抽出法である。“テンプレートマッチング法”は画像から抽出しようとする対象物の形が分かっているとき、これをマスクとして撮影画像との間で相関係数を計算するものである。PMC 法は円形に近い粒子画像の抽出のために開発された手法であり、粒子画像は対称な二次元正規分布で近似できるものとし、これをマスク（テンプレート）として対象画像上に重ね、相関係数を計算する。マスクと粒子画像の形状が等しければ、輝度は違っても相関係数はほぼ 1 になる。マスクを撮影画像上で走査させ、相関係数を計算する。その値がある閾値以上になる部分に粒子が存在するとみなし、粒子位置を割り出す。相関値計算の式は(1)式で表される。相関値計算は、撮影画像の全画面に対して行う。マスクを画像左上から順に 1 画素ずつ移動しながら走査していき、各位置において相関値計算を行う。

$$r = \frac{\sum_{i,j=1,1}^{n \times m} (I_{i,j} - \bar{I})(M_{i,j} - \bar{M})}{\sqrt{\left[\sum_{i,j=1,1}^{n \times m} (I_{i,j} - \bar{I})^2 \right] \left[\sum_{i,j=1,1}^{n \times m} (M_{i,j} - \bar{M})^2 \right]}} \quad \text{式(1)}$$

n, m: 水平, 鉛直比較領域サイズ i, j: 水平, 鉛直位置

$I_{i,j}$: 元画像の輝度値

$\bar{I}$: 元画像の輝度値の平均値

$M_{i,j}$: マスク画像の輝度値

$\bar{M}$: マスク画像の平均値

PMC 法のアルゴリズムは、(1)画像の読み込み、(2)画像拡張、(3)マスク画像と撮影画像の相関値計算、(4)2 値化処理、(5)連結領域の抽出（ラベリング）、(6)粒子像中心計算、(7)粒子位置情報の出力の順に行われる。本節では、処理時間の大半を占め、相関値計算部をハードウェア化の対象とした。

4.2 相関値計算回路のアーキテクチャ最適化

リアルタイムで処理可能な相関値計算回路の設計を行うにあたり、2.1で述べた C 言語によるシステム LSI 設計手順に従い、オリジナルのソフトウェア C 記述を Bach C 言語を用いて書き換える。その際に、相関値計算処理の Bach C 記述（回路記述）中で使用される各変数に対して、とりうる最大の値を想定してビット幅を指定する。この段階で生成される回路を、「相関値計算逐次処理」とする。このままでも Bach コンパイラを用いて動作合成を行えば、論理合成可能な RTL 回路記述に変換されるが、この段階での回路は各処理を逐次的に実行するようスケジューリングされるため高速な回路にはならない。筆者らはこの回路に対して 3 章で提案した高速化・回路規模削減手法を 6 項目に分けて段階的に適用し、アーキテクチャを最適化した。ただし、各段階に 1 つの最適化手法を適用するのではなく、ある段階では複数の最適化手法を組み合わせることで同時に適用した。以下の表 4.2

に、適用した6項目の最適化とその効果（高速化・回路規模削減）を示す。「最適化手法の分類」の欄の●は、適用された手法を示し、「効果」の欄の○は最適化手法に適用により得られた効果が高速化・回路規模削減のいずれかもしくは両方であることを示す。

また、以下の手法をすべて実装した全体回路構成図を図4.2に示す。

4.2.1 1 マスク処理及び相関値判別式の変形

相関値計算処理は、対象領域の平均値・分散の計算、対象領域とマスクの比較（相関値前処理計算と呼ぶ）、相関値計算の順に行われる。相関値計算逐次処理回路では、各処理について全画面分計算を行い、相関値計算の演算途中結果をメモリに記憶するという方法をとっていた。一つの比較領域（1マスクと呼ぶ）ごとにすべての計算を行なう。本手法により演算途中結果格納用のメモリが不要となり、メモリ容量の削減やメモリアクセス数の削減ができ、処理を高速化することができる。

1 マスク処理における相関値判別式の変形は、相関値計算処理に含まれる算術演算の高速化・回路規模削減を行う手法である。式(1)の相関値判別式では平均値計算や除算が必要となり、処理時間や回路規模がともに増大する。そのため式(1)を以下の式(2)のように変形し、平均値計算、平方根計算、除算計算を削除する。

式(2)中の変数 M のみの項はマスク画像に対する要素しか含まないため、定数で表現できる。この変形により除算と平方根計算がなくなったため値を整数のみで表すことができる。これにより、処理速度に対する効果では、演算器中の遅延の削減や、演算プロセスの短縮による高速化が可能となる。また、Σの乗算部は従来の40ビットの固定小数点ではなく、オーバーフローを防ぐため unsigned int 型の16ビットで表現できるようになり、計算に使用する演算器の回路規模も削減が可能となる。

$$\begin{aligned}
 (0.7)^2 & \left[n \times m \sum_{i,j}^{n \times m} I_{i,j}^2 - \left(\sum_{i,j}^{n \times m} I_{i,j} \right)^2 \right] \left[n \times m \sum_{i,j}^{n \times m} M_{i,j}^2 - \left(\sum_{i,j}^{n \times m} M_{i,j} \right)^2 \right] \\
 & < \left[n \times m \sum_{i,j}^{n \times m} (I_{i,j} M_{i,j}) - \left(\sum_{i,j}^{n \times m} I_{i,j} \right) \left(\sum_{i,j}^{n \times m} M_{i,j} \right) \right]^2
 \end{aligned}
 \tag{2}$$

表4.2 各種高速化・回路規模削減手法の適用

Table4.2 Design improvements using architecture optimization methods

節	高速化・ 回路規模削減手法	最適化手法の分類				効果	
		データ アクセス	並列化・ パイプライン化	アプリケー ションに特 化した演算 回路設計	モジュール 構成変更	高速化	回路規模 削減
4.2.1	1 マスク処理及び 相関値判別式の変形	●		●	●	○	○
4.2.2	データ再利用によるメモリアクセスの削減	●				○	
4.2.3	相関値判別式の演算回路化			●		○	
4.2.4	画像拡張処理の相関値演算回路への内装	●			●	○	
4.2.5	メモリアクセスと相関値演算処理のパイプライン化	●	●			○	
4.2.6	連続動作パイプライン回路の複数同時処理化	●	●			○	

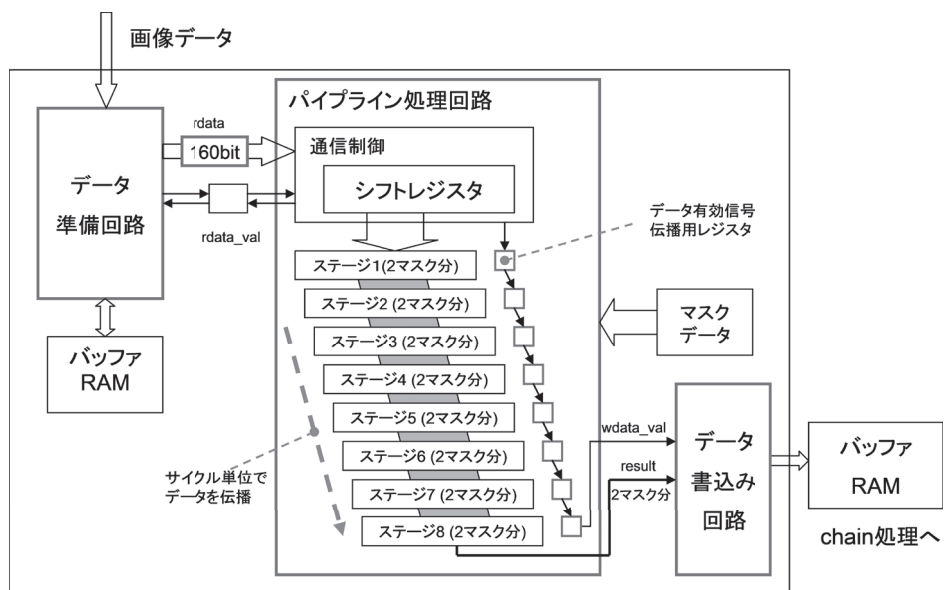


図4.2 マスク同時処理連続動作パイプラインのブロック図

Figure4.2 Block diagram of parallel processing using pipelining

4.2.2 データ再利用によるメモリアクセス削減

マスク画像のサイズが 7×7 画素であるため、相関値計算を行うには49画素の画像データが必要である。しかし i 回目と $i + 1$ 回目ではマスク画像を右に 1 画素ずらした領域に対して相関値計算を行うため、 i 回目に使用した画像データは左端 1 列以外 $i + 1$ 回目でも使用するよう、49画素をレジスタに保持しメモリアクセス回数を削減する。

画像データの再利用の様子を図4.3に示す。メモリから取得した画像データを直接計算に使用せず、1 回目は49画素の画像データをメモリから取得後レジスタに保持し相関値計算を行う。 $i + 1$ ($i \geq 1$) 回目は7列目のデータのみをメモリから読み込み、 $i + 1$ 回目の相関値計算を行う (図4.3(b))。このようにレジスタに保持した値を再利用することによって図4.3(c)に示す i 回目と $i + 1$ 回目の共通領域についてはメモリアクセスが不要になる。

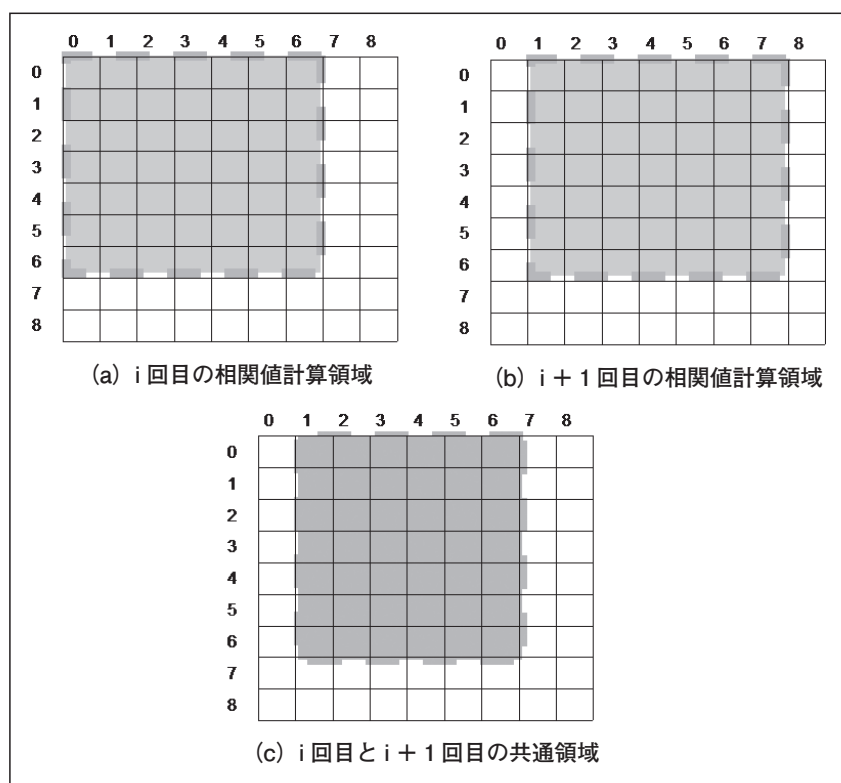


図4.3 画像データの再利用
Figure4.3 Image data reuse

4.2.3 相関値判別式の演算回路化

Bach システムでは関数はすべてインライン展開され、1つの演算器を共有して繰り返し使用することによって回路規模の削減を行っている。そのため各演算器単位でスケ

ジューリングが行われるが、演算器を専用化したい場合は `preserve` プラグマを指定することによって、関数を1つの演算器として扱うことが可能となる。演算回路化を行うことで、このプラグマを指定した一連の処理は組み合わせ回路で実現されるので処理内容にループや通信・メモリアクセスを含んではならない。3.3節で述べた手法によって、相関値計算処理はループや通信・メモリアクセスを含まずに処理を行うことが可能となったので、相関値判別式全体を演算器として実装可能となった。以下、相関値判別式の演算回路を `judge` 回路と呼ぶ。`judge` 回路は相関値計算全体が組み合わせ回路で実装され、その遅延量は24nsであり、システム全体を100MHzで動作させる場合、`judge` 回路の演算器としての処理時間は3サイクルになる。この場合 `Bach` コンパイラでは自動的にこの回路を3サイクルのマルチサイクルパスとして実装するため、設計者は周辺回路設計時のタイミング制御に大きな時間を消費することはない。これにより、相関値計算1マスク分を3サイクルで実行可能となった。

4.2.4 画像拡張処理の相関値演算回路への内装

サブピクセル精度を保障するため相関値計算の前処理として行われる画像拡張処理（以下 `expand` 処理と呼ぶ）を、マルチサイクル動作の `judge` 回路内に実装する。`judge` 回路に inputs する画像データ量は拡大前の輝度データとし、回路内で相関値計算1マスク分に必要な量のみの拡大を行うことで、外部メモリ容量とメモリアクセスを削減する。画像拡張の様子を図4.4に示す。(a)は拡張前の5×5画素の輝度データを示し、(b)を経て(c)の9×9画素に拡張される。

画像拡張処理の相関値演算回路への内装した結果、`expand` 処理を組み込んだ `judge` 回路を `ex_judge` 回路とする。`ex_judge` 回路に5×5画素分の拡張前の画像データを入力すると、内部で拡張処理を行い9×9画素となる。このデータを用いて4マスク分までの相関値計算を行う。4マスク分の相関値計算を複数同時に行う。相関値演算式の特徴から並

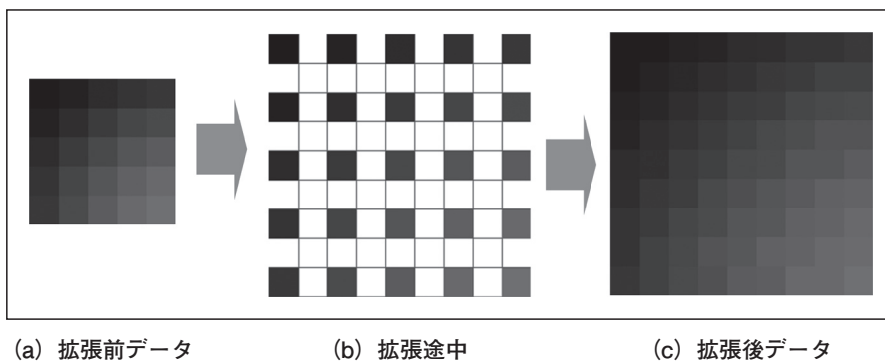


図4.4 画像拡張の様子
Figure4.4 Image data expansion

列マスク間で演算途中結果を共有することで回路規模増加を抑える。5回のメモリアクセスを行うことで拡大前データを3画素×5行分取得でき、これを用いて12マスク分の相関値計算が可能になる。

4.2.5 メモリアクセスと相関値演算処理のパイプライン化

4.2.4で述べたように、ex_judge 回路を備えた相関値計算回路では、12マスク分の処理を終えた後、メモリアクセスを行い新規データの格納を行う。メモリアクセスと相関値演算処理をパイプライン化し、メモリアクセス時間を吸収することで、高速化する。

メモリアクセス用回路と相関値演算回路を独立させ、並列に動作させる。メモリアクセス用回路を“データ準備回路”，相関値演算を担う回路を“相関値処理回路”と呼ぶ。各回路間には画像データ通信用の信号線を設け、データ準備回路が読み込んだデータを相関値処理回路に同期通信で渡す。これにより、データ準備回路が読み込んだデータを相関値処理回路が処理している間、次のデータを読み込む。

さらに、フラグを用いた通信によって、メモリアクセスと相関値計算との処理時間差を吸収したパイプライン化を適用する（図4.2のデータ準備部とパイプライン処理部のパイプライン）。この回路のタイミング図を図4.5に示す。

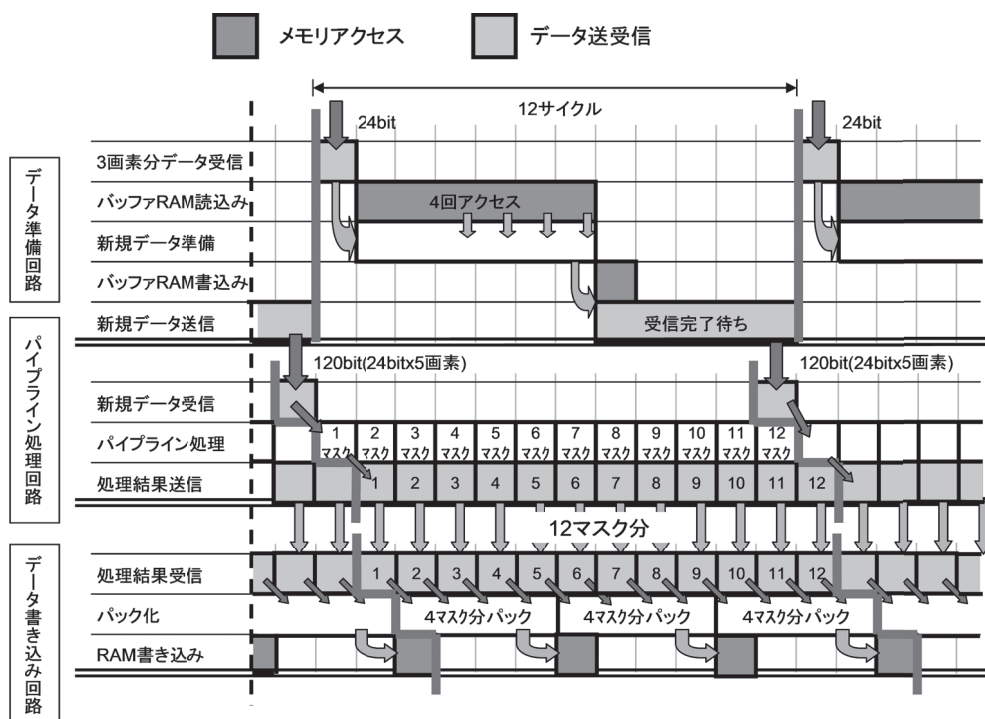


図4.5 メモリアクセスと相関値演算処理のパイプライン化回路のタイミング図

Figure4.5 Timing chart of pipelining among memory accesses and correlation value calculation

4.2.6 連続動作パイプライン回路の複数同時処理化

4.2.4で述べた ex_judge 回路を分割し、1 サイクル単位でパイプライン動作させ、パイプライン動作を行う各演算器を2マスク分同時に処理するよう拡張する。またそれに伴い、データ書込み回路に受け渡す処理結果も1サイクルにつき2マスク分となるため、通信線の拡張を行う。また、データ書込み回路は1サイクルに2マスク分の処理結果を受信し、4マスク分を連結してバッファRAMに書き込む動作となる。パイプライン動作を行う各演算器を拡張する際、4.2.4で述べた手法で内部の演算途中結果を共有することで、回路規模の増加を抑える（図4.2のパイプライン処理回路部）。

4.3 設計結果

表4.2で示した各種高速化・回路規模削減手法を実際に設計し回路実装を行った結果を示し、それぞれに対する評価を行う。

各々の手法の効果を確認するため、作成した回路とその設計結果を表4.3に示す。各回路についてPMC法のソフトウェア出力結果とハードウェアの出力結果を比較した結果、ソフトウェアと同等の精度を確認した。

これらの高速化・回路規模削減手法により、回路規模の増加を抑えながら高速化を実装できていることがわかる。全ての高速化・回路規模削減手法を適用した回路では、処理速度はソフトウェア処理に対して約1,000倍、リアルタイム処理における処理時間である66 [ms]の1/3となった。これにより、30枚/秒の処理に対応できる。

表4.3 各アーキテクチャ最適化手法の適用結果

Table4.3 Design results applied architecture optimization methods

	最適化の適用	処理時間 (ms)	回路規模 (ゲート)
	ソフトウェア	21,780	—
(A)	逐次処理	39,913	59,097
(B)	(A) + 1 マスク処理及び相関値判別式の変形	2,573	70,407
(C)	(B) + データ再利用によるメモリアクセス削減	820	68,754
(D)	(C) + 相関値判別式の演算回路化	491	82,978
(E)	(D) + 画像拡張処理の相関値演算回路への内装	143	92,978
(F)	(E) + メモリアクセスと相関値演算のパイプライン化	41	54,576
(G)	(F) + 連続動作パイプライン回路の複数同時処理化	21	74,900

5. アーキテクチャ最適化手法の評価

本章では、表4.1に示した8個の設計事例において、アーキテクチャ最適化手法を適用した結果をグラフ化し、データアクセス法、並列化・パイプライン化、アプリケーションに特化した演算回路設計、モジュール構成の変更等の各手法の効果を比較検討する。

粒子追跡の相関値計算以外の各事例における設計結果の詳細は、表4.1に記載した文献から引用する。設計結果の一覧表を付録に添付する。

一般に、性能向上と回路面積増加はトレードオフの関係にある。例えば、高速化のために並列処理化を行えば、その分、回路面積は増大する。逆に、面積削減のため、演算器などを共有すれば、演算処理が順序的になり、性能は劣化する。本文では、事例ごと、手法ごとに、適用前と適用後の高速化に対する面積増加の比率（PAR）について、Y 軸を高速化、X 軸を面積増としたグラフを作成した。このグラフにおいて、 $PAR = 1$ 以上であれば、面積増に対し、高速化の効果がより大きいと考えることができる。

5.1 アーキテクチャ最適化手法の適用結果の評価

表4.1の適用例ごとに、アーキテクチャ最適化手法を順次適用した場合の PAR の推移を矢印で図5.1(a,b,c)に示す。例えば、図5.1(a)における相関値計算では、表4.3について(A)から(G)の各々の結果について PAR の推移を矢印で示している。適用例ごとに目標仕様が異なり、PAR の変化の大きさや変化には差があるため、その変化を確認容易にするため、高速化の値によって、(a),(b),(c)の3つのグラフに分ける。図5.1(b),(c)のアーキテクチャ最適化は付録に記載している。どれも目標仕様を達成している。ここで、目標仕様とは、例えば、4章の相関値計算回路は、4.3節で述べたように、粒子追跡システムにおいてリアルタイム処理を行うための目標仕様が66ms 以下であり、これを満足している。他の回路も同様に本研究で設計した回路を搭載したシステムがリアルタイム処理を可能とすることを目標仕様としている。また、現在の微細化製造技術では、1チップに搭載可能なゲート数は数千万個になっており、適用例に示すどの回路もゲート数も100万ゲート以下であり、1チップ化可能である。

5.2 アーキテクチャ最適化手法の PAR 比較

アーキテクチャ最適化手法のうち、多く用いたパイプライン化(a)、並列化(b)、及びデータアクセス法(c)について、各アプリケーションへの適用時の個別の PAR 値をすべて求め、分布グラフ（図5.2）に示す。点線は、 $PAR = 1$ を示す。ただし、実際には、例えば、表4.2に示すように、一つの回路改良に複数の手法を用いるケースが多々あるので、ここでは主として効果がある手法に代表させている。このため、高速化は、パイプライン段数や並列数に単純に比例していないが、一定の傾向は出ているので、それについて

考察する。

パイプライン化手法（図5.2(a)）は、すべて $PAR = 1$ の線を超えており、パイプラインの段数が多いほど、バラツキがあるが、より高速化できている。このバラツキはパイプライン化を適用した箇所の大小や前後の回路やパイプラインの制御回路などのオーバーヘッドに起因すると考えられる。例えば、4.2.5節に示す図4.5のパイプライン化では、4.2.4節までの相関値計算回路の高速化により、前後のメモリアクセス回路の処理時間との処理時間がほぼ同じとなったため、パイプライン化の効果が出ていることを確認できる。

並列化（図5.2(b)）では、どの適用例も高速化を達成しているが、付記した並列数に比べて単純に高速化されていない。この理由は、パイプライン化と同様に並列化を適用した箇所の大小や前後の回路や並列化の制御回路などのオーバーヘッドに起因すると考えられる。また、 $PAR = 1$ を下回るケースも幾つかあるが、これは面積増加より高速化を優先させたためであり、これらは以降の段階で並列化された回路部の最適化が行われ、全体的な最適化が図られていることが図5.1で確認できる。

データアクセス法は、 PAR 値 = 1 の線の傾きから最も高速化を達成している手法と考えられる。 $PAR = 1$ を下回るケースも幾つかあるが、これは並列化と同様に面積増加より高速化を優先させたためであり、これらは以降の段階で回路部の最適化が行われ、全体的な最適化が図られていることが図5.1で確認出来る。設計事例によって扱うデータ量やそのアクセス法が異なるので、定量的な比較は今後の課題である。

このような比較結果から、最大の効果を得るためには、どのような場合にそれぞれのアーキテクチャ最適化手法を適用すべきかという適用基準を見つけ出していくことが今後の重要な課題と考えられる。

6. あとがき

本論文では、C 言語設計における、データアクセス法、並列化・パイプライン化、アプリケーションに特化した演算回路設計、モジュール構成の変更等のアーキテクチャ最適化手法を提案し、粒子追跡技術における相関値計算のハードウェア化を例に、アーキテクチャ最適化手法の適用結果について述べた。

さらに、提案手法をさまざまな設計事例に適用し、その設計を通じてアーキテクチャ最適化の効果を評価した。これらの設計事例では、適用例ごと及び手法ごとに最適化の効果を高速化/面積増加比（ PAR ）指標にもとづいて測定し、効果確認を行った。すべての適用例において、最終的な設計の目標仕様を実現することができ、提案手法の有効性を確認した。手法ごとの評価では、データアクセス法が最も大きな効果を上げる可能性を持ち、パイプライン化は安定的に、少ない面積増で高速化を実現する効果があると考えられる。並列化手法は、適用箇所によってバラツキがあるが、一定の面積増を前提として、高速化

を実現するには直接的で効果的な手法であることを確認した。

今後、さらに適用対象を拡大しつつ、新たなアーキテクチャ最適化手法の考案と適用基準の明確化について考察を進める。

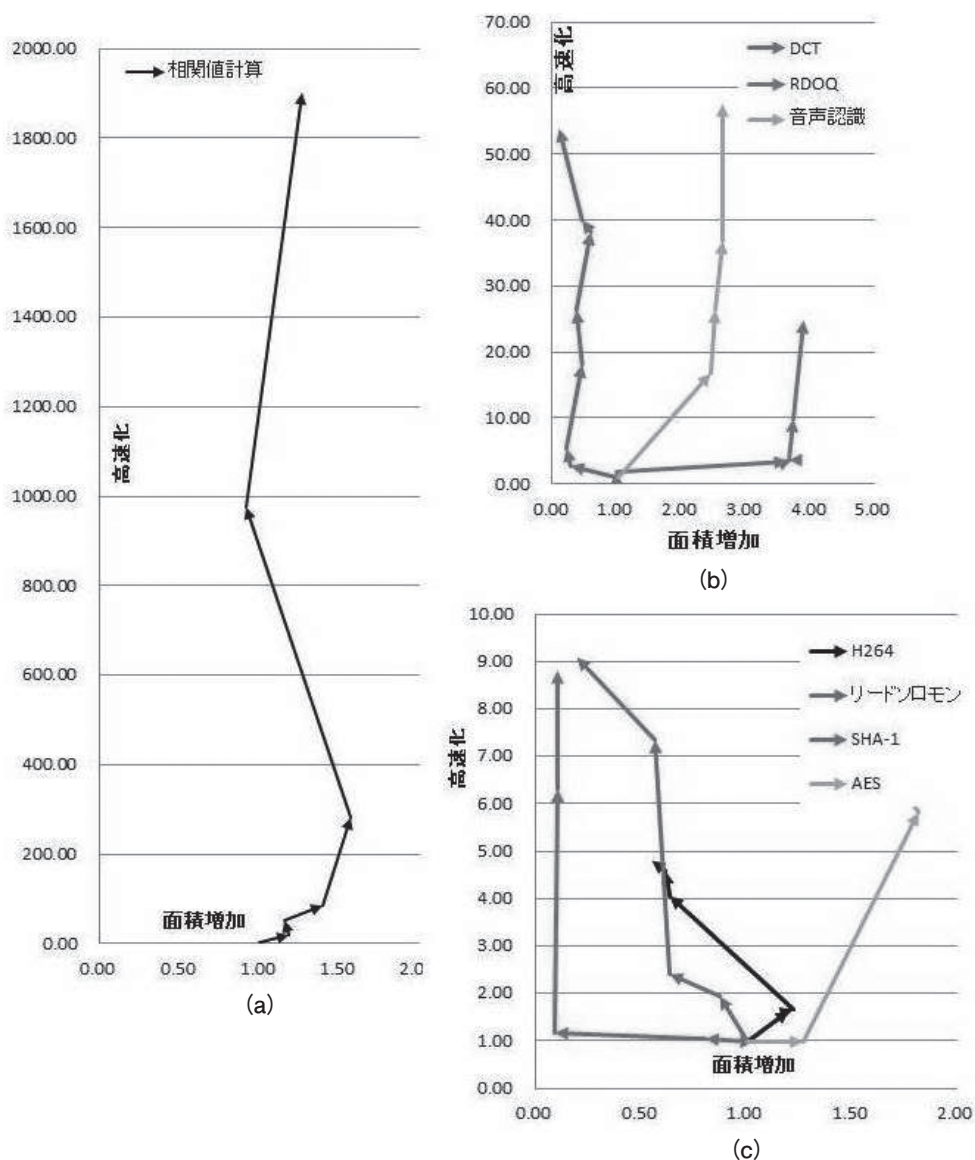


図5.1 適用例に対するアーキテクチャ最適化手法による PAR の推移

Figure5.1 Transition of PAR of architecture optimization methods for applications

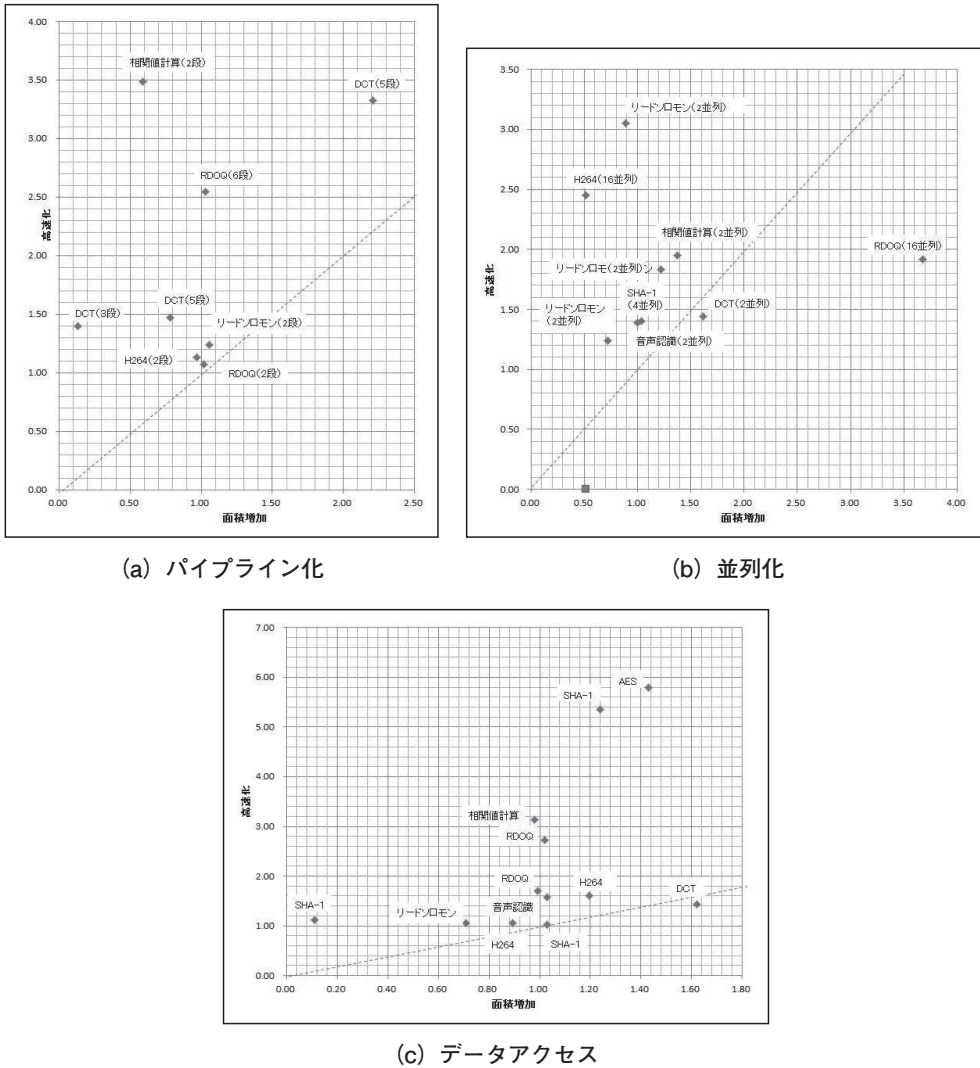


図5.2 アーキテクチャ最適化手法の PAR 比較
Figure5.2 PAR comparisons of some architecture optimization methods

参考文献

- [1] F. Catthoor and H. J. De Man: "Application-Specific Architectural Methodologies for High-Throughput Digital Signal and Image Processing," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 38, No. 2, 1990, pp. 339-349.
- [2] S. Kobayashi, K. Mita, Y. Takeuchi and M. Imai: "De-sign Space Exploration for DSP Applications Using the ASIP Development System PEAS-III," Proceedings of the Acoustics, Speech, and Signal Processing, Vol. 3, 13-17 May 2002, pp. 3168-3171.

- [3] S. Pasricha and N. Dutt, “COSMECA: Application Specific Co-Synthesis of Memory and Communication Architectures for MPSoC,” Proceedings of the Conference on Design, Automation and Test in Europe, 6-10 March 2006, pp. 700-705.
- [4] <https://www.nvidia.com/ja-jp/>
- [5] M. P. Vestias, H. C. Neto: “Co-Synthesis of a Configurable SoC Platform based on a Network on Chip Architecture,” Proceeding of Asia and South Pacific Design Automation, pp.48-53, 2006.
- [6] O. Schliebusch, A. Hoffmann, A. Nohl, G. Braun and H. Meyr, “Architecture Implementation Using the Machine Description Language LISA,” Proceeding of the Asia and South Pacific Design Automation, Bangalore, 7-11 January 2002, pp. 239-244. doi:10.1109/ASPDAC.2002.994928.
- [7] A. Halambi, P. Grun, V. Ganesh, A. Khare, N. Dutt and A. Nicolau: “EXPRESSION: A Language for Architecture Exploration through Compiler/Simulator Retargetability,” The Proceeding of Design, Automation and Test in Europe Conference and Exhibition, 9-12 March 1999, pp. 485-490.
- [8] T. Kambe, A. Yamada, K. Nishida, K. Okada, M. Ohnishi, A. Kay, P. Boca, V. Zammit, T. Nomura: “A C-based synthesis system, Bach, and its application,” Asia and South Pacific Design Automation Conference pp. 151-155, 2001年1月.
- [9] K. Okada, A. Yamada, T. Kambe: “Hardware Algorithm Optimization Using Bach C”, IEICE Trans. Fundamentals, Vol. E85-A, No. 4, pp. 835-841, 2002.
- [10] <https://www.accellera.org/downloads/standards/systemc>
- [11] https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis.html
- [12] <http://www.vdec.u-tokyo.ac.jp/>
- [13] 大窪啓太, 神戸尚志: “特定用途向け低ビット演算回路の一設計法,” 情報処理学会論文誌 第48巻 第5号 pp. 1918-1925, 2007年5月.
- [14] 上津寛和, 中川寛誉, 神戸尚志: “2次元ジグザグ走査による相関値計算回路の設計とその評価,” システム制御情報学会論文誌, 第21巻, 第1号 pp. 18-27, 2008年1月.
- [15] 上津寛和: “粒子抽出システムのアーキテクチャ最適化に関する研究,” 平成19年度近畿大学大学院理工学研究科, 博士前期課程論文, 2008年3月.
- [16] 酒井皓司: “粒子追跡システムのアーキテクチャ最適化に関する研究,” 平成19年度近畿大学大学院理工学研究科, 博士前期課程論文, 2008年3月.
- [17] 北村友一: “アーキテクチャレベルにおける回路高速化手法に関する研究,” 平成23年度近畿大学大学院理工学研究科, 博士前期課程論文, 2012年3月.
- [18] 澤野肇, 神戸尚志, 荒木統行: “動的再構成可能回路における JPEG エンコーダ設計とその評価,” 電気学会論文誌 C, Vol. 132, No. 11, pp. 1733-1740, 2012年11月.

- [19] T. Kambe, H. Matsuno, T. Miysazaki, A. Yamada: “C-based Design of a Real Time Speech Recognition System,” IEEE International Symposium on Circuits and Systems 2006 pp. 1751-1755, 2006年 5 月.
- [20] 才辻誠: “音声認識システムのアーキテクチャ最適化に関する研究,” 平成19年度近畿大学大学院理工学研究科, 博士前期課程論文, 2008年 3 月.
- [21] 江口彰彦, 岸田和也, 神戸尚志: “An Application Specific Circuits Design for a LVCSR System,” European Conference on Circuit Theory and Design, pp. 790-793, 2009年 8 月.
- [22] 小田島賢和: “リードソロモン符号復号化回路のアーキテクチャ最適化に関する研究,” 平成19年度近畿大学大学院理工学研究科, 博士前期課程論文, 2008年 3 月.
- [23] 吉田敦郎, 東祐司, 宮崎渉, 田中照人, 神戸尚志: “リコンフィギュラブルプロセッサを用いたリードソロモン符号復号化回路の設計,” 電子情報通信学会, VLSI 設計技術研究会技術報告, pp. 65-68, 2008年 3 月.
- [24] 永井翔太: “動画像圧縮技術のハードウェア化とその高速化に関する研究,” 平成26年度近畿大学大学院理工学研究科, 博士前期課程論文, 2015年 3 月.
- [25] 森口元気, 神戸尚志: “レート歪み最適化による量子化プロセスの高速化とそのハードウェア化,” 電気学会論文誌 C, Vol. 135, No. 11, pp. 1299-1306, 2015年11月.
- [26] 森口元気: “動画像圧縮技術のハードウェア化に関する研究,” 平成27年度近畿大学大学院理工学研究科, 博士前期課程論文, 2016年 3 月.
- [27] 印藤佑樹: “AES 暗号アルゴリズムのハードウェア化に関する研究,” 平成24年度近畿大学理工学部電気電子工学科, 卒業論文, 2013年 3 月.
- [28] 高橋香保里: “SHA-1アルゴリズムのハードウェア化に関する研究,” 平成24年度近畿大学理工学部電気電子工学科, 卒業論文, 2013年 3 月.

(2022年11月 2 日受理)

(付録)

図5.1, 5.2に示した図について, 相関値計算以外の設計結果を以下に添付する. 詳細は参考文献を参照ください.

H264[24] (動き検出)

適用内容	処理時間[ms]	回路面積[gates]	手法の分類
オリジナル	179.388	1,282,345	—
メモリアクセス削減	111.288	1,535,288	データアクセス
ループ融合	108.032	1,562,476	ループ変換
16並列化	44.184	816,998	並列化
2 段ループパイプライン化	39.27	790,866	パイプライン化
構造体解体	37.328	705,673	データアクセス

RDOQ[24]

(レート歪み最適化)

適用内容	処理時間[μs]	回路規模[gates]	手法の分類
オリジナル	100,419	213,706	
メモリのオンチップ化	58,918	211,772	データアクセス
16並列化	30,654	778,654	並列化
ループパイプライン化 (2 段)	28,556	794,006	パイプライン化
プライオリティエンコーダ	28,049	785,035	アルゴリズム変更
入力ビット連結	10,309	802,509	データアクセス
6 段パイプライン最適化	4,038	830,464	パイプライン化

DCT[17]

適用内容	処理時間[ns]	回路規模[gates]	手法の分類
Chen アルゴリズム	7,776,000	850,473	
冗長な演算削減 / 乗算・除算のシフト演算化	2,880,000	245,006	式変形
専用回路化	1,440,000	180,804	専用回路化
5 段パイプライン化	432,000	399,801	パイプライン化
6 ブロック連続 5 段パイプライン化	294,000	311,035	パイプライン化
x 方向・y 方向の並列化	204,000	502,729	並列化
三角関数の近似計算	194,400	391,650	式変形
データ入出力を含めた 3 段パイプライン化	144,150	71,315	パイプライン化

リードソロモン[22]

(16バイト消失訂正)

適用内容	処理時間[ns]	回路規模[gates]	手法の分類
オリジナル	106,414	61,284	
データアクセス	100,744	43,793	データアクセス
シンドローム計算の2並列化	54,978	53,451	並列化
多項式演算の2並列化	44,338	38,743	並列化
チェン探索の2並列化	14,532	34,624	並列化
チェン探索の2段パイプライン化	11,704	36,682	パイプライン化

SHA-1暗号化[28]

適用内容	処理時間[ns]	回路規模[gates]	分類
オリジナル	3,036	71,513	
メモリアccessの改良	2,936	73,791	データアクセス
ループ展開	2,889	58,142	ループ変換
関数の受け渡し改良	2,571	6,168	データアクセス
メモリアccess改良とループ融合	481	7,660	データアクセス
4並列化	345	7,660	並列化

AES 暗号化[27]

適用内容	処理時間[ns]	回路規模[gates]	分類
オリジナル	313.7	99,184	
除算器の削除	311.5	126,005	式変形
メモリアccess	53.7	179,952	データアクセス
ループ変換	47.2	167,283	ループ変換

音声認識[20]

第1パス部

適用内容	処理時間[ms]	回路規模[gates]	分類
オリジナル	380,979	190,136	
ガウス分布・指数対数計算専用回路化	22,772	463,420	専用回路化
出力キャッシュ	14,451	477,603	メモリアccess
並列探索（2並列）	10,340	496,817	並列化
ハーフカッティング	6,604	499,210	アルゴリズム変更

ベジエ曲線によるダイコンの形状記述とその応用

淡 誠 一 郎¹・淡 裕 美 子²・吉 田 康 子³

Shape Description of Radish by Bézier Curve and its Application

Seiichiro Dan¹・Yumiko Dan²・Yasuko Yoshida³

はじめに

果物や野菜の形状は人の嗜好や収穫・加工の効率などに大きく影響する。形状は土壌や気候、栽培過程などの環境要因にも影響されるが、遺伝的要因によるところが大きい。本研究で扱うダイコンの形状にも様々な遺伝子に関わることが予想され、ダイコンの効率的な育種を行ううえでは根形に関わる遺伝的メカニズムの解明が期待される。本来、量的形質である根形に関わる遺伝子の同定には、形状の量的な測定が必要である。しかし、従来研究では、楕円フーリエ記述子を用いた形状計測方法^[1,2]が量的な手法として挙げられるぐらいであり、ほとんどの研究では根形は質的にしか評価されてこなかった。

図形の形状記述法としてはさまざまな方法が考えられる。楕円フーリエ記述子を含む各種フーリエ記述子を使ったダイコンの形状記述とその問題点についてはすでに報告した^[2,3]。本報告では、コンピュータで滑らかな曲線を扱う際に用いられるパラメトリック曲線のうち、その代表であるベジエ曲線を用いた形状記述とその応用例について述べる。

フーリエ記述子は、図形のシルエット全体の特徴を少数のパラメータで捉えることができるため、形状の定量評価や分類のための強力なツールである（例えば SHAPE^[4]）が、曲がりを持った個体の長さや幅の計測、軸に沿った幅の変化の様子を捉えるなどといった

1 大阪学院大学 情報学部 Faculty of Informatics, Osaka Gakuin University

2 神戸大学大学院 農学研究科 Graduate School of Agricultural Science, Kobe University

3 神戸大学大学院 農学研究科附属食資源教育研究センター Food Resources Education and Research Center, Graduate School of Agricultural Science, Kobe University

目的には不向きである。そのような目的には、任意の曲線を少数の制御点で記述でき、曲線上の位置を1つのパラメータで連続的に表せるパラメトリック曲線の方が適する。本稿で解説するベジエ曲線を用いた形状記述法は、ダイコンの多様性評価のための形状解析や計測に利用しており、その成果については [5,6]ですでに報告済みである。これらの報告は農学的な視点での成果報告であり、利用している画像処理プログラムの詳細については説明されていないので、ここに報告する次第である。

2. 対象画像と前処理

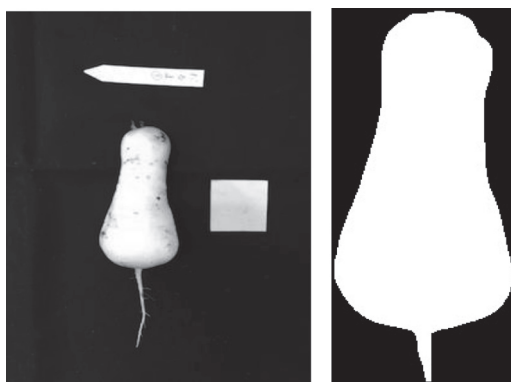


図1 入力画像と記述対象のシルエット画像の例

図1は本研究で対象とする実画像の一例と記述の対象となるシルエット画像である。実際に形状記述の対象とするのは右のシルエット画像の輪郭上に等間隔に取った標本点の列である。実画像からシルエット画像、シルエット画像から輪郭の標本点の列を得る手順は先の研究ノート[2]で述べた。

以下では標本点の列

$$[(x_1, y_1), (x_2, y_2), \dots, (x_M, y_M)]$$

を対象物の形状あるいはシルエットと同一視する。 $(x_k, y_k), k = 1, 2, \dots, M$ はシルエットの輪郭線から等間隔でサンプリングされた点の座標である。ここではデジタル画像を前提としているので、各座標成分は整数で表される。点列はほぼ等間隔に取られるが、デジタル画像であるのでサブピクセル単位の量子化誤差を含む。

3. ベジエ曲線によるシルエット記述についての基本考察

3.1 ベジエ曲線とは

ベジエ曲線はコンピュータグラフィックスにおいて滑らかな曲線を描くための代表的な

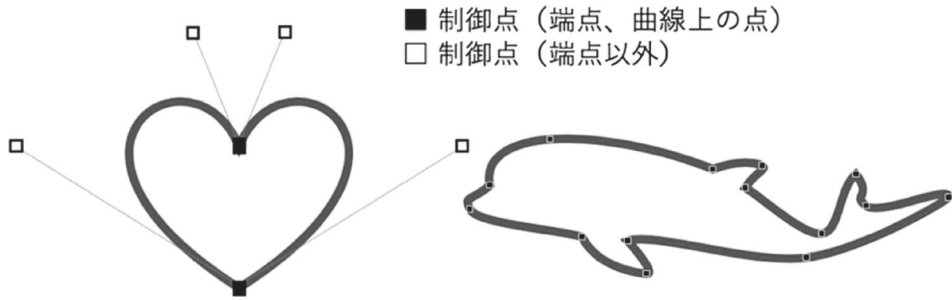


図2 ベジエ曲線で描かれた図形の例

手法である。図2にベジエ曲線で描かれた図形の例を示す。ベジエ曲線はいくつかの制御点によって区分的な曲線を定義する。複数の区分曲線をつなぎ合わせていくことで、いくらでも複雑な曲線を表現できる。

ベジエ曲線上の各点は1つのパラメータ変数の関数として表現される。そのような曲線表現はパラメトリック曲線とよばれる。先の研究ノート[2]で扱ったフーリエ記述子では経路長をパラメータとして形状を記述したが、それもパラメトリック曲線の一種である。ベジエ曲線を拡張したパラメトリック曲線として、B-Spline 曲線や有理ベジエ曲線、さらにそれらの拡張である NURBUS 曲線など、ベジエ曲線より表現力の高い手法もあるが、表現力が高いものほど曲線を定義するための項目が増える。イラストやCG制作が目的であるならば、表現力や自由度は高いほどよいので、ベジエ曲線よりも B-Spline、B-Spline よりも NURBUS 曲線の方が優れていると言えるが、解析や識別が目的であれば、表現能力が必要十分な範囲で、記述が簡潔な方が好ましい。そこで、ここでは制御点を用いたパラメトリック曲線の中でもっともシンプルな記述であるベジエ曲線を取り上げる。

ベジエ曲線の定義

有限個の制御点 $B_0, B_1, \dots, B_N$ と $[0,1]$ を定義域とするパラメータ t により、次式で定義される曲線を N 次のベジエ曲線とよぶ。

$$B(t) = \sum_{n=0}^N b_{N,n}(t) B_n \quad \dots(1)$$

ここで、

$$b_{N,n}(t) = {}_N C_n (1-t)^{N-n} t^n \quad \dots(2)$$

はバーンスタイン基底関数とよばれる。 ${}_N C_n$ は N 個から n 個を選ぶ組み合わせの数である。

バーンスタイン基底関数に関して、

$$\sum_{n=0}^N b_{N,n}(t) = \sum_{n=0}^N {}_N C_n (1-t)^{N-n} t^n = (t + (1-t))^N = 1 \quad \dots(3)$$

が成り立つ．すなわち，バーンスタイン基底関数 $b_{N,n}(t), n=0,1,\dots,N$ の合計は常に 1 であり，ベジエ曲線においては制御点の混合比率を表していると解釈できるので，ブレンディング関数とも呼ばれる．

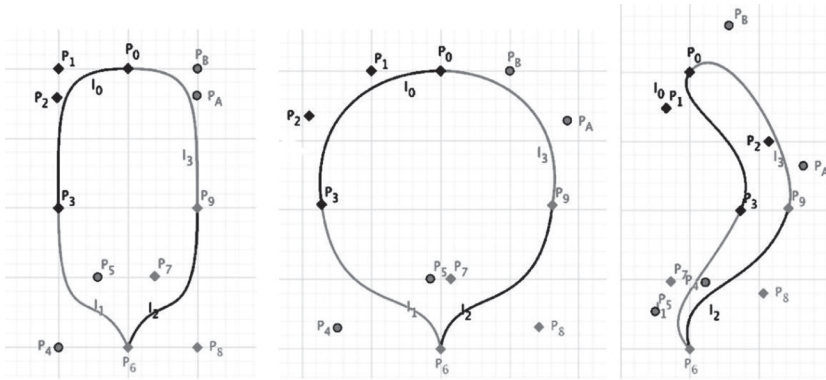


図3 ベジエ曲線をつないで描かれたダイコンふうの図形

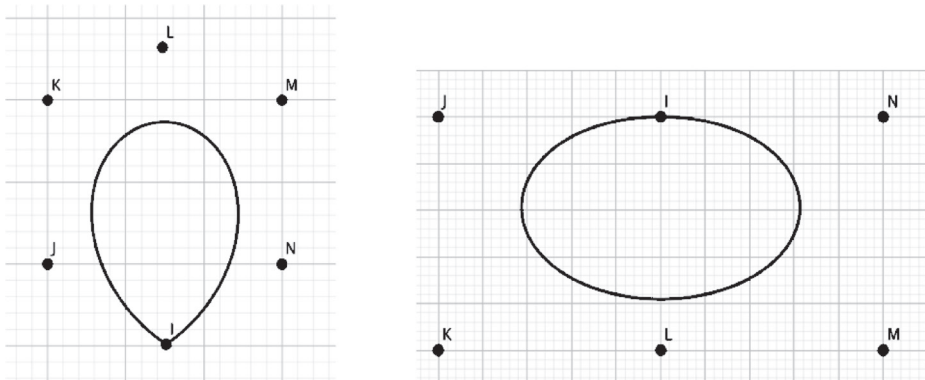


図4 6次ベジエ曲線で描かれた閉曲線

ベジエ曲線をつなぎ合わせることで，複雑な図形を形作ることができる．例えば，図3の3つのダイコンふうの図形は，それぞれが4つの3次ベジエ曲線をつなぎ合わせて描かれている．制御点はそれぞれ計12であり，この12点の配置だけで形状が定まる．次数を上げてうまく制御点を配置すれば，1つのベジエ曲線で閉曲線を形作ることもできる．図4の2つの図形は共に I, J, K, L, M, N, I を制御点とする6次ベジエ曲線である．

この様に，ベジエ曲線で形状を近似すれば少数の制御点だけで連続的に変化する輪郭形状を表現することができる．分割数あるいは制御点の数を増やせばいくらかでも複雑な形状を表現することができる．また，ベジエ曲線は曲線上に同じ x 座標を持つ点が複数存在するような曲線でも扱えるという点も重要である．そのような曲線には多項式近似は適用できないが，ベジエ曲線による近似は可能である．実際，本研究で扱うダイコンの輪郭の全

周はもちろんのこと、2分割した場合でも曲線上の座標値の成分同士が一意対応となるとは限らない。

ベジエ曲線の制御点の位置は、図形をスケーリング、回転、平行移動させると変化する。しかし、楕円フーリエ記述子などと同様に、楕円近似の長軸を基準として正規化すれば、ベジエ曲線の制御点は形状を規定する不変特徴とみなせるので、フーリエ記述子同様、形状の分類や識別に応用できる可能性もある。

3.2 点列へのベジエ曲線のあてはめ

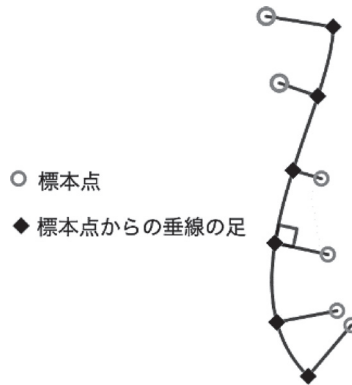


図5 曲線当てはめの誤差

輪郭線をベジエ曲線で近似するということは、標本点の集合 $\{P_k = (x_k, y_k): k = 1, 2, \dots, M\}$ にもっとも当てはまるベジエ曲線の制御点を見つけるということである。あるベジエ曲線と与えられた標本点とのずれを、次の式で定義し、この値が最小となる曲線を求めることを考える。

$$E = \frac{1}{2} \sum_{k=1}^M |P_k - B(t_k)|^2 \quad \dots(4)$$

ただし、 $B(t_k)$ はベジエ曲線上で標本点 $P_k = (x_k, y_k)$ に対応づけられる点で、 t_k はその点に対応するベジエ曲線のパラメータである。 $B(t_k)$ は理想的には標本点 P_k と一致すべきであるが、当然誤差は生じる。曲線の方を固定した場合、(4)式を最小化するためには、 $B(t_k)$ を P_k に最も近い点、すなわち P_k から曲線に下ろした垂線の足に一致させればよい(図5)。 t_k の値は求めたいベジエ曲線に依存して決まるものであるが、一旦それは忘れてこの値が既知であると仮定し、最小二乗法を用いて曲線あてはめを行うことを考える。具体的な計算方法は以下の通りである。

当てはめに用いるベジエ曲線の次数を N 、制御点を $\{B_n = (u_n, v_n): n = 0, 1, \dots, N\}$ とする。誤差関数 E は非負であり、制御点の座標の連続関数であるのだから、誤差が最小となる場合があるはずであり、そのとき、

$$\begin{cases} \frac{\partial E}{\partial u_n} = 0, & n = 0, 1, \dots, N \\ \frac{\partial E}{\partial v_n} = 0, & n = 0, 1, \dots, N \end{cases} \quad \dots(5)$$

が成立するはずである。そこで、この連立方程式を解くことで、誤差最小の制御点の組み合わせ、もっとも標本点にフィットするベジエ曲線を求める。

ベジエ曲線の式を成分に分解すると、

$$\mathbf{B}(t) = (B_x(t), B_y(t)) = \left(\sum_{n=0}^N b_{N,n}(t)u_n, \sum_{n=0}^N b_{N,n}(t)v_n \right) \quad \dots(6)$$

(5)の第一式に(4)式、(6)式を順に代入して式変形していくと、

$$\begin{aligned} \frac{\partial E}{\partial u_n} &= \sum_{k=1}^M \left((B_x(t_k) - x_k) \frac{\partial B_x(t_k)}{\partial u_n} \right) = \sum_{k=1}^M \left((B_x(t_k) - x_k) b_{N,n}(t_k) \right) \\ &= \sum_{k=1}^M \left(\left(\sum_{i=0}^N b_{N,i}(t_k)u_i - x_k \right) b_{N,n}(t_k) \right) \\ &= \sum_{i=0}^N \left(\sum_{k=1}^M b_{N,i}(t_k)b_{N,n}(t_k) \right) u_i - \sum_{k=1}^M x_k b_{N,n}(t_k) = 0 \end{aligned} \quad \dots(7)$$

が得られる。この式は

$$A_{n,0}u_0 + A_{n,1}u_1 + \dots + A_{n,N}u_N = B_n \quad \dots(8)$$

ただし、

$$A_{n,i} = \sum_{k=1}^M b_{N,i}(t_k)b_{N,n}(t_k), \quad B_n = \sum_{k=1}^M x_k b_{N,n}(t_k) \quad \dots(9)$$

と書ける。

パラメータ $t_k: k = 1, 2, \dots, M$ を固定した場合、 $A_{n,i}$, B_n は定数となるので、(8)式は $N+1$ 個の未知数 $u_0, u_1, \dots, u_N$ の線形方程式である。この方程式は $n = 0, 1, \dots, N$ について、計 $N+1$ 個作れるので、それらをすべて連立させて解けば、仮のパラメータ $\{t_k\}$ の下ではあるが、標本点への最適當ではめのベジエ曲線を得ることができる。(5)の第二式 (y 成分) についてもまったく同様である。

3.3 漸近的ベジエ近似アルゴリズム

上で述べた手順は最小二乗法による曲線回帰の常套的な枠組みをベジエ曲線に適用したにすぎない。そうするために各標本点に対応するパラメータ t_k を定数として扱ったが、実際にはあてはめ前に t_k はわかっていない。高次の N 次ベジエ曲線に対して、各標本点

から降ろした垂線の足の座標を制御点と標本点の座標により陽に表現するのは困難であるし、かといって、 t_k まで未知数として扱おうと、線形方程式に帰着できなくなる。

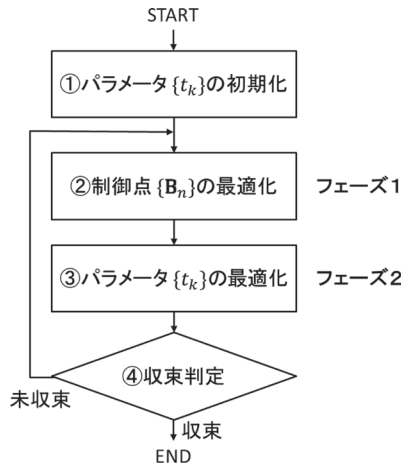


図6 漸近的ベジエ近似アルゴリズム

そこで、図6に示すように、

フェーズ1 暫定的なパラメータ $\{t_k\}$ の下で制御点 $B_n: n = 0, 1, \dots, N$ を最適化

フェーズ2 暫定的な制御点 $\{B_n\}$ の下でパラメータ $t_k: k = 1, 2, \dots, M$ を最適化

という2つのフェーズを残差が収束するまで繰り返すことで、漸近的に標本点に近似ベジエ曲線をフィッティングさせていくようなアルゴリズムを考える。

処理の流れをもう少し詳しく説明する。

処理の流れ

- ① パラメータの初期化。区間 $[0,1]$ を標本点の数で等分割した値を各標本点に対応付ける。すなわち、 $P_k: k = 1, 2, \dots, M$ に対するパラメータを $t_k = (k-1)/(M-1)$ とする。
- ② パラメータ $\{t_k\}$ を固定し、あてはめ誤差が最小となるように制御点の集合 $\{B_n: n = 0, 1, \dots, N\}$ を最適化する。(フェーズ1)
- ③ 制御点 $\{B_n\}$ できまるベジエ曲線に対し、各標本点に対して距離が最小となる点のパラメータを求め、パラメータ $t_k: k = 1, 2, \dots, M$ を更新する。(フェーズ2)
- ④ 収束条件を満たしていれば終了。さもなければ、②に戻り、処理を繰り返す。

3.4 アルゴリズムの実装

アルゴリズムを、表1に示す4種の方法 fit0, fit1, fitT0, fitT1 で実装してみた。違いは、上で説明した流れの②と③の実装法の違いである。②の制御点の決定に関して、fit0, fit1, fitT1 は最小二乗法を用い、fitT0 は勾配法を用いる。③のパラメータの決定に関して、fit0

表 1 漸近的ベジエ近似アルゴリズムの実装方法

実装法名称	②制御点 $\{B_n\}$ の最適化方法	③パラメータ $\{t_k\}$ の最適化方法
fit0	最小二乗法 (3.2で説明した計算法)	①で初期化した値のままで固定
fit1	最小二乗法 (3.2で説明した計算法)	パラメータ区間 $[0,1]$ を細かく区切って暫定的な近似曲線上に密に候補点を設定し*, 各標本点 $P_k; k=1, \dots, M$ に対して距離最小の候補点を見つけ, そのパラメータを t_k とする.
fitT0	勾配法 (Tensorflow, Adam) 暫定的な $\{t_k\}$ の下で, 前の繰り返しで求めた $\{B_n\}$ を初期値とし, 勾配法による $\{B_n\}$ の更新を mloop 回繰り返す. (default: mloop=3)	勾配法 (Tensorflow, Adam) 暫定的な $\{B_n\}$ の下で, 前の繰り返しで求めた $\{t_k\}$ を初期値とし, 勾配法で $\{t_k\}$ を更新する.
fitT1	最小二乗法 (3.2で説明した計算法)	上に同じ

*隣接する候補点間の距離が1画素以内に収まるような間隔で候補点を設定.

は固定して更新しない, fit1は原始的に最寄り点を探して更新する, fitT0, fitT1は勾配法を用いて徐々に更新するという違いである. なお, ベジエパラメータ $\{t_k\}$ の初期値は区間 $[0,1]$ を均等分割した値とし, 制御点の初期値は fit0 を1度実行して求めた値とする. fitT0は②③ともに勾配法を用いるが, ②の更新 mloop 回につき, ③の更新を1回実行する. 本稿の実験では, 経験的に mloop=3とした.

実装に用いたプログラミング言語は python であり, numpy, sympy, Tensorflow などのライブラリを用いた. 最小二乗法の実装には数値解析ライブラリ numpy の線形連立方程式のソルバー linalg.solve() 関数を用いた. fitT0, fitT1 で用いる勾配法は機械学習ライブラリ Tensorflow を用いて実装し, 勾配法の最適化アルゴリズムとしては, さまざまな機械学習問題で広く使われ, 定評を得ている Adam 法を採用した.

Tensorflow には Adam 以外にも多くのオプティマイザが用意されている. 本質ではないので詳細は割愛するが, そのうち SGD, RMSprop, Adam, Adamax, Adagrad, Adadelta, Nadam, Ftrl を用い, さまざまな条件で比較実験した結果, この問題には Adam オプティマイザが安定的に優秀であることがわかっている. 比較実験の結果の一例を図7に示す.

Adam オプティマイザにはハイパーパラメータがたくさんあり, 処理時間を左右する. 本研究ではもっとも重要なハイパーパラメータである学習率 learning_rate のみをカスタマイズし, 他のパラメータはデフォルト値で実験を行う. 学習率は, ベジエパラメータの最適化では0.001, 制御点の最適化では30と定めた. この値は, ハイパーパラメータの最適化ツール optuna を用い, 実装法 fitT0 で, 後述する実験対象画像のいずれにおいても解が得られ, かつ, 平均的に処理ステップ数が少なくなるような値を試行錯誤的に探して決めた値である. あくまでも実験してみた範囲であるが, 収束可能な学習率の範囲は fitT0 よ

り fitT1 の方が広く、fitT0 が収束する値なら fitT1 も収束することを確認している。

実験環境は以下に示すとおりである。

CPU : ThinkCentre M75q Gen 2 (タイプ 11JJ)

AMD Ryzen 7 PRO 4750GE 3.10 GHz (8 core)

RAM 16GB オンボード GPU

Python 環境 : Python 3.9 (Anaconda) + Tensorflow 2.6, numpy 1.21.2, sympy 1.9

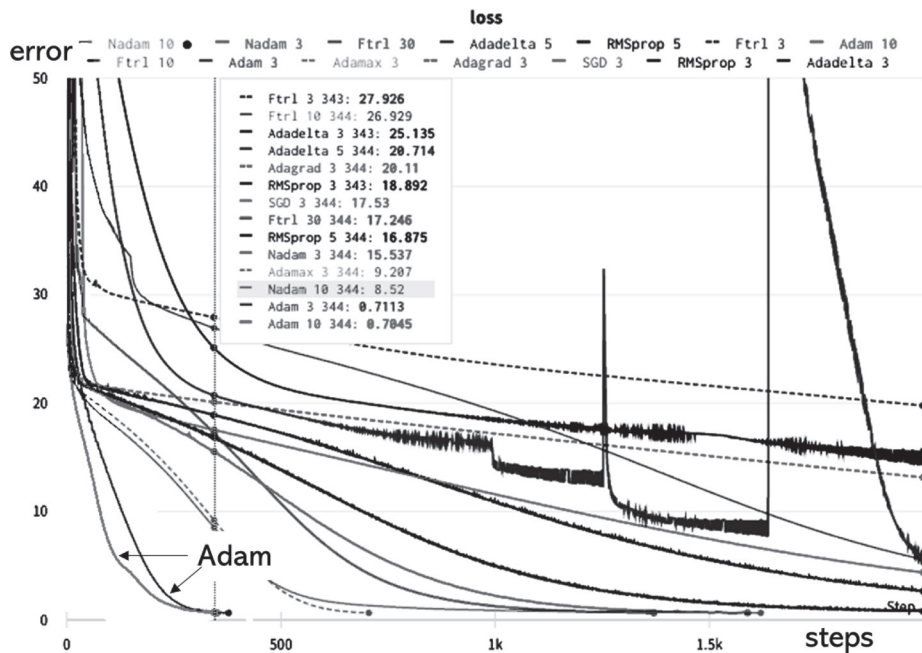


図7 最適化アルゴリズムの比較結果例
(横軸はステップ数、縦軸は残差平方平均)

3.5 オーバフィッティング対策とオーバフィッティングの判定

① オーバフィッティング対策

一般に曲線の自由度が必要以上に高いとオーバフィッティングが発生しやすい。点列への曲線あてはめでは標本点間に対しては何も制約がないため、標本点にはフィットするが標本点間で曲線とシルエットが大きくずれるという可能性は常にある。オーバフィッティングの例を図8に示す。

オーバフィッティング対策としては、標本点を増やす方法と曲線の自由度を下げるようなペナルティ項を目的関数に追加する方法が考えられる。標本点を増やすのは単純で効果的であるが、計算量とのトレードオフとなる。経験上もっともずれの発生しやすいのは両端部であることがわかっている。アドホックではあるが両端とその隣接標本点の中間点を

フィッティング対象に加えるとオーバーフィッティングはかなり抑制できる。

標本点間での無用な湾曲を抑制するために、標本点における2次微係数の平方や接線方向と標本点間を結ぶ方向の不一致度などをペナルティ項として追加するのも一定の効果がある。ただし評価項が増えると項の重みの最適化という新しい問題が発生する。

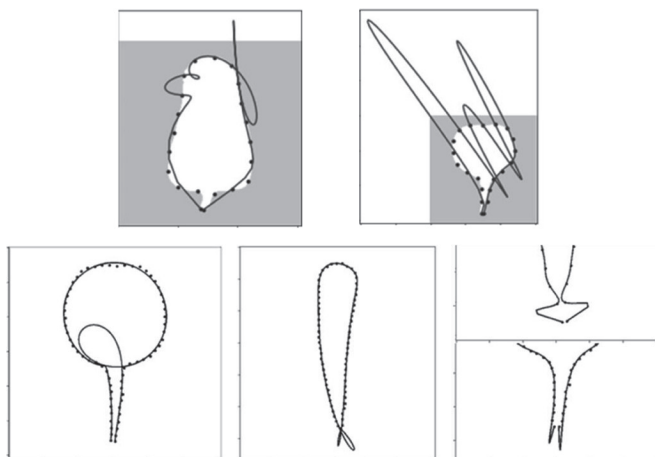


図8 オーバフィッティングの例

② オーバフィッティング判定

オーバーフィッティングかどうかは程度の問題であり、判定するとすれば何らかの尺度を設けてしきい値で判定せざるをえない。基本的には近似曲線と輪郭との距離を尺度とすればよいが、どこを測るかが問題である。図9(a)に示すように、すべての輪郭点の近くを通っていても輪郭から逸脱する曲線はありえ、このようなケースは輪郭点列側からの最短距離だけを測っても見逃されてしまう。このケースは、近似曲線の方を十分細かく刻み、経路全体にわたって輪郭点までの距離を測ればよいが、距離だけの判定では図9(b)のケースのような順序関係の矛盾を見逃してしまう。このように厳密に考えても、しきい値の設定や近似曲線の刻み間隔次第で判定結果は左右される。

重要なのは、個人差の生じる人の主観的な判断ではなく、何らかの客観的基準を設けることである。本稿の以下の実験では次のような簡易的な判定方法を採用することにした。

- i) 輪郭点列を標本点の位置で分割し、区間ごとの点列を長さ基準で4等分するような位置を求め、5点(区間の始点、終点+3つの4分位点)で代表させる。
- ii) 近似曲線を i) で決めた区間に対応する区分曲線に分割し、i) 同様に4等分し、5点で代表させる。
- iii) 区間ごとに、輪郭点側と近似曲線側の代表点同士の距離を求め、5つの距離の標準偏差があらかじめ設定したしきい値を超える区間が一か所でもあればオーバーフィッ

ティングと判定する。

標準偏差では代表点5点すべてが同程度にずれているとオーバフィッティングと判定されないことになるが、5点のうち2点は標本点であるのでフィッティング条件成立下では距離は小さいはずであり、5点の標準偏差が大きいということは残りの3点でのずれが大きいということになる。判定基準を平均値（誤差の合計としても同じ）や最大値とすることもできる。図8に示したような、目視であきらかにオーバフィッティングが生じていると感じるようなケースはいずれの指標でも検出できる。

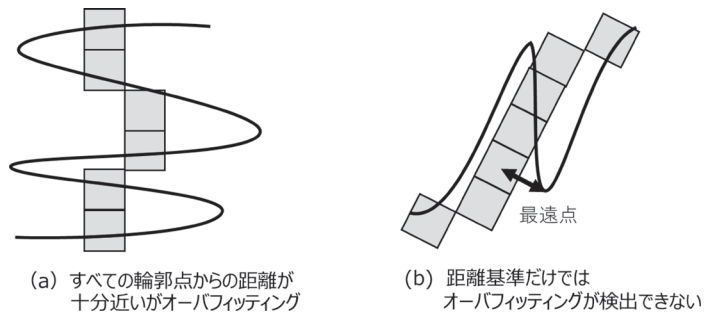


図9 オーバフィッティング判定の難しさ

4. ベジエ曲線によるダイコンのシルエット形状記述

実際のダイコンのシルエット全体、あるいは、左右に分割された輪郭線のそれぞれの記述に必要なベジエ曲線の次数や標本点の数、実装法の優劣などを実験的に調べていく。

4.1 フィッティングの評価尺度

近似曲線の次数を上げていけば画素レベルの微細な凹凸にまでフィットさせることができるが、ひげ根や不自然な切断による凸凹などのノイズにまでフィットさせるのは意味がない。欲しいのは、ノイズは無視して、個体のシルエットに自然な形で沿う滑らかな曲線である。ただそのような抽象的な要求を評価するのは困難であるので、ここでは3章で説明した曲線あてはめ法におけるあてはめ誤差、すなわち標本点と近似曲線上の対応点の距離の平方の平均値をフィッティングの尺度とする。

曲線あてはめの結果に影響を与える項目としては、

- 対象画像（画像サイズとシルエットのサイズ）
- 勾配法の収束判定基準と最適化アルゴリズム
- 標本点の数
- ベジエ曲線の次数
- 許容誤差

●実装法とその設定（最小二乗法／勾配法，オーバーフィッティング対策）

などが挙げられる．本報告では，対象とするシルエット，画像のサイズと個体のサイズ，勾配法の収束判定基準，実装法の設定は以下で述べるように固定し，標本点の数，ベジェ曲線の次数，許容誤差，実装法の違いで結果がどう変化するかを調査することにする．

① 対象とするシルエット

図10に示す9個のシルエットを実験対象とする．これらは2017年に兵庫県加西市にある神戸大学大学院農学研究科附属食資源教育研究センターの圃場にて栽培し，それぞれの適期に収穫後に撮影した実写真をもとに，雑音除去・平滑化などの画像処理を施して得られたシルエットである．供試された品種は，「だるま大根」（中原採種場），「平安すしらす聖護院」（タキイ種苗），「方領」（トーホク），「カザフ辛味大根」（そ生研），「マコトちゃん」（トーホク），「ねずみ」（日本タネセンター），「おふくろ」（タキイ種苗），「夏つかさ」（トーホク）である．「ねずみ」については，当センター内の粘質の土壌の影響から品種本来の形に栽培できていない可能性があったため，栃木県宇都宮市にある株式会社トーホクの清原育種農場で栽培，収穫したものも使用した．これらのダイコンを9つのカテゴリー：逆三角形，横楕円形，角（つの）形，三角形，丸形，円筒形，だるま形，楕円形，先流れに分類した．カテゴリーは，ダイコンの根形を網羅するように，日本国内で使用されている「農林水産省 農林水産植物種類別審査基準」^[7]，「農業生物資源遺伝バンク 特性評価マニュアル」^[8]，および国際的な基準である「IBPGR descriptors for Brassica

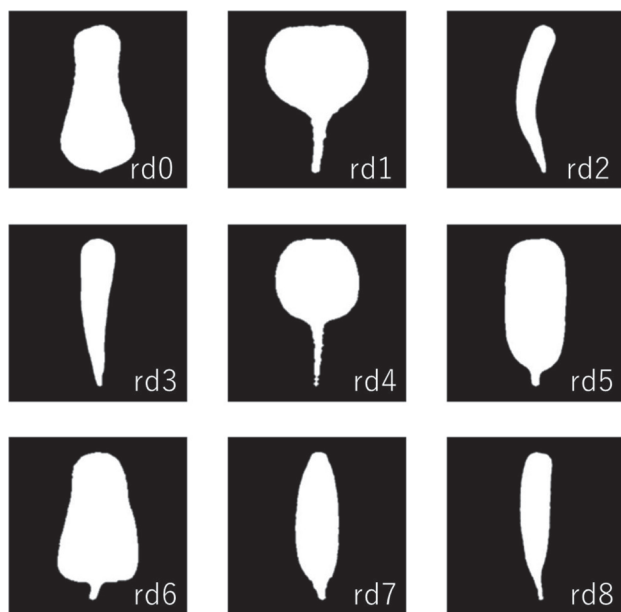


図10 実験対象とするダイコンのシルエット画像

and Raphanus」^[9]，「UPOV Guidelines for the conduct of tests for distinctness, uniformity and stability. Radish; Black radish」^[10,11]に記載されている根形のカテゴリーを参考に設定した。

② 画像のサイズと個体のサイズ

シルエットのバウンディングボックスの長径が256となるようにリサイズし， 308×308 画素のキャンバス中央に配置した白黒画像を処理対象画像とする．この画像から画像処理ライブラリ OpenCV の `findContour()` 関数を用いて得られる8連結の輪郭点列が曲線あてはめの対象である．この輪郭を末端部の曲率最大点で開いて開曲線化したものを全周輪郭，それをさらに上端部で2分割したものを左右分割輪郭とよぶことにする．

③ 収束判定基準

- i) あてはめ誤差の平方平均（以下「あてはめ誤差」と省略する）がしきい値 `err_th` 未満
- ii) あてはめ誤差の減少量の移動平均（直近3回分）が，あてはめ誤差としきい値 `err_th` の差の1/100,000未満
- iii) 1,000回連続で誤差の最小値が更新されない
- iv) 繰り返し回数が3,000回を超えた

のいずれかの基準が満たされた場合に繰り返しを終了する．いずれのケースでも処理過程であてはめ誤差がもっとも小さかった時点の曲線をあてはめ結果とする．

④ 実装法の設定

実装法のうち，勾配法の最適化アルゴリズムは上述したように Adam 法を用い，学習率は統一する．オーバフィッティング対策としては，両端部の標本点の中間点計2点を対象に加える方法を `fit0` に適用する．どの実装法でも制御点の初期値の決定には `fit0` を使っているので，結果的に全実装法にオーバフィッティング対策を施したことになる．

4.2 全周輪郭のベジエ近似

シルエット全周を1つのベジエ曲線で近似することを考える．ダイコンのシルエットの輪郭は1つの閉曲線となるが，ベジエ曲線は開曲線の記述であるので，輪郭をどこかで切断して開曲線化する必要がある．ここでは根の尖端部を切断点として採用することにする．主題から外れるので説明は省略するが，ほとんどの個体で，下部尖端部は曲率に基づき自動抽出可能である．

ここでは，まず，近似次数ごとのあてはめ結果の違いと誤差の程度を調べるための基礎実験を行った後，各シルエットの全周を近似するのに必要な次数，標本点数の違いによる近似結果の違い，オーバフィッティング対策の効果について調べる．

① 実験 A 基礎実験

4つの実装法それぞれで次数をさまざまに変え、図10に示した9つのシルエットの全周輪郭のベジエ近似を試みた。4つの実装はいずれも正しく機能し、近似曲線が得られることが確認できた。標本点数65で次数を変化させていった場合のあてはめ結果の違いの例を図11に示す。図11に示した結果は実装法 fitT0 による結果であるが、fit1, fitT1 を用いても目に見える大きな違いはない。

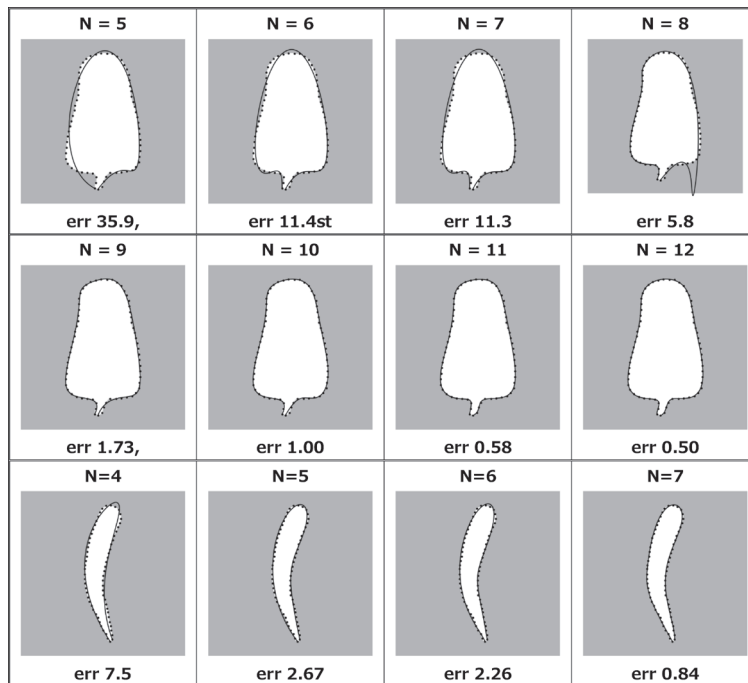


図11 近似次数による近似結果の違い (N: 近似次数)

② 実験 B 全周近似に必要な次数

ダイコンのシルエットをベジエ曲線で近似するのに必要な次数と実装法の優劣を調査するため、標本点の数を65、許容誤差（あてはめ誤差の許容値）を1.0, 0.65, 0.5として、実装法 fit0, fit1, fitT0, fitT1 のそれぞれで、許容誤差内にかつオーバーフィッティングとならない最低の近似次数を調べてみた。

なお、オーバーフィッティングの判定基準は上述のとおりであり、オーバーフィッティング判定のしきい値は、「標本点間の平均経路長×あてはめ誤差のしきい値」とした。

i) 必要次数の比較

実験結果を表2に示す。網掛けは同じ対象に対してもっとも次数の低いものである。近

表2 全周近似に必要な次数

err_th = 1.0				
	fit0	fit1	fitT0	fitT1
rd0	17	7	7	7
rd1	18	10	10	9
rd2	20	8	9	7
rd3	16	8	5	5
rd4	21	10	12<11>	10
rd5	15	8	9	8
rd6	17	10	12<11>	10
rd7	19	6	6	6
rd8	18	8	9	9<8>

err_th = 0.65				
	fit0	fit1	fitT0	fitT1
rd0	23	7	7	7
rd1	24	10	12	10
rd2	24	10	9	8
rd3	18	9	6	10<9>
rd4	27	11	12	10
rd5	19	9	9	9
rd6	21	11	12	11(10)
rd7	26	8	7	6
rd8	22	9	10	9<8>

err_th = 0.5				
	fit0	fit1	fitT0	fitT1
rd0	28	9	8	8
rd1	>30[0.51]	15	15	10
rd2	26	11	9	11
rd3	19	10	7	10<9>
rd4	>30[0.60]	13	12	10
rd5	24	11	9	9
rd6	27	14	12	12
rd7	>30[0.54]	11	8	7
rd8	25	12	10	9

似次数の上限は30までとし、30次でもあてはめ誤差が許容誤差以下にできなかった場合は“>30 [数値]”と記してある。数値は30次で収束した時点のあてはめ誤差である。<次数>と（次数）はオーバフィッティング判定を省いた場合の最低次数である。<次数>と（次数）の違いは、目視によりオーバフィッティングが顕著であると感じるかそうでない

かの違いである。これは個人の主観に左右されるのであくまでも参考情報と考えられたい。

fit0は他の実装と比べて近似に必要な次数が2倍程度以上となる。この表の数値には現れないことであるが、標本点数65に対して次数が30にもなると、標本点のぶれに忠実にフィットしすぎて曲線全体の滑らかさが失われてくる。

許容誤差内でより少ない次数でフィッティングできることを実装法の優劣の尺度とするならば、fit1, fitT0, fitT1のうちのいずれかの実装法が常に他よりすぐれているということではなかった。処理対象と許容誤差により優劣は入れ替わる。fit1の場合、対象 rd0～rd8の全周近似に必要な次数の範囲は、許容誤差1.0, 0.65, 0.5に対して、6～10次、7～11次、9～15次であった。同じくfitT0の場合は、5～12次、6～12次、7～15次、fitT1の場合は、5～10次、6～11次、7～12次であった。許容誤差1.0では3つの実装に大きな違いはないが、許容誤差が0.5になるとfit1は他の2つの実装より1ないし2次多い次数が必要となっている。

なお、同じ次数同じ実装法であれば、許容誤差1.0, 0.65, 0.5の結果の違いはわずかであり、目視ではほとんど違いがわからない。近似次数や実装法が違っててもこの許容誤差の範囲では違いはわずかであり、変曲点や端点付近をよく見比べると違いがわかる、という程度である。

最後に、オーバフィッティングに関してコメントしておく。この実験の範囲では、fit0はオーバフィッティングが生じにくく、もっともオーバフィッティングが多く生じたのはfitT0であった。fitT0はベジエパラメータと制御点の両方に勾配法を適用するため、ローカルミナにトラップされる可能性が組み合わせて増えるためであろう。また、今回の実験の範囲内でひどいオーバフィッティングが生じたのは、許容誤差以下で近似できる最低次数ちょうどの場合であった。もちろん標本数もオーバフィッティングの発生に大きく影響する。今回の実験では標本数を65と多めに設定したが、標本数をもっと減らすとオーバフィッティングが発生しやすくなる。

ii) 処理時間の比較

次に処理時間に焦点を当てるため、許容誤差を1.0として、近似に必要なベジエ曲線の次数とその結果を得るまでの繰り返しのステップ数および処理時間を調べた。標本点数は先と同じ65とした。結果を表3に示す。なお、処理時間は処理環境や使用しているライブラリ関数に大きく左右される。実験対象の4つの実装法はそれぞれ依存しているライブラリ関数が異なるので、実装法間での処理時間の絶対値の比較は意味がない。また、学習率などの設定を変えると結果は大きく変わる。

数値以外の記号の意味はつぎのとおりである。

UF あてはめ誤差が1.0未満とならずに処理打ち切りとなった。カッコ内は誤差の最小値。

OF あてはめ誤差1.0未満となって停止したがオーバフィッティングが生じている。

表3 あてはめ誤差が1.0未満となる次数と処理時間およびステップ数

	fit0			fit1	fitT0	fitT1
	次数	処理時間	次数	処理時間 / ステップ数	処理時間 / ステップ数	処理時間 / ステップ数
rd0	17	2.9s	7	210.0s / 236	59.9s / 455	82.1s / 161
rd1	18	3.3s	9	UF (5.60)	UF (3.03)	615.7s / 739
			10	521.3s / 334	327.8s / 1621	167.1s / 162
rd2	20	3.9s	7	UF (1.11)	UF (1.33)	343.1s / 675
			8	1867.7s / 1722	UF (1.16)	832.2s / 1279
			9	226.6s / 166	226.8s / 1260	45.0s / 50
rd3	16	2.5s	5	UF (1.46)	50.9s / 555	137.1s / 525
			6	UF (1.18)	92.5s / 818	97.6s / 261
			7	UF (1.18)	111.3s / 826	175.3s / 347
			8	371.0s / 337	OF (138.0s/895)	OF (49.0s/44)
rd4	21	4.4s	10	2202.1s / 1395	UF (2.05)	674.1s / 658
			11	1504.4s / 843	OF (653.6s/2373)	618.4s / 498
			12	122.4s / 57	152.8s / 573	30.7s / 18
rd5	15	2.3s	8	629.5s / 581	UF (2.88)	160.2s / 243
			9	281.0s / 213	340.0s / 1875	87.2s / 101
rd6	17	2.9s	10	3254.5s / 2080	UF (1.14)	2808.3 / 2781
			11	333.2s / 183	UF (1.02)	72.9s / 56
			12	181.3s / 85	72.0s / 268	61.0s / 39
rd7	19	3.6s	6	111.0s / 160	30.0s / 259	31.1s / 79
rd8	18	3.2s	8	670.8s / 613	OF (137.0s/894)	OF (1511.0s/2313)
			9	613.0s / 463	102s / 559	184.9s / 220

UF：許容誤差内で収束せず。カッコ内は誤差の最小値，OF：オーバフィッティング

fit0は近似に高い次数を必要とするが直接解法であるため非常に高速である。fit0以外の3実装法は近似曲線の初期値を決めるのにfit0を用いているためfit0より処理時間がかかるのは当然なのであるが、勾配法の反復計算の計算量は非常に大きく、また、時間幅が広い（本実験の環境で30秒～1時間弱）。反復計算1回分の計算量は次数が高いほど大きくなるはずであるが、同じ誤差に到達するまでに必要な反復回数は次数が高いほど少なくて済む傾向がある。結果として次数が高い方が処理時間は短傾される傾向が見て取れる。

fit1とfitT1の結果を比較すると、両者でステップ数当たりの処理時間に大きな違いはないが、同対象、同次数で比較すると必要なステップ数がfit1よりfitT1の方が少なくて済んでいる。fit1とfitT1は制御点の最適化に最小二乗法を用いるという点で共通しており、違いはベジエパラメータの最適化法である。fit1のベジエパラメータの最適化フェーズでは、その時点での近似曲線に対して個々の標本点の最寄り点を求め、標本点に対応付けるパラメータ値を最寄り点のそれに書き換える。この方法はいきなり最適解に到達できる可能性

はあるが、時間軸にそって見たときに値の連続性がないため振動を伴いやすい。一方 fitT1 の勾配法は、パラメータの値をもとの値を基準として最適と思われる方向に徐々に変化させていく。そうしても多かれ少なかれ振動は発生するが、fitT1 で採用している Adam オプティマイザには振動を抑える仕組みが組み込まれており、とても優秀である。結果として fitT1 の方が fit1 より少ないステップ数で収束できているものと推測される。

fitT0 と他の 2 つの実装法を比べると、fitT0 のステップ数は総じて大きな値となっており、処理時間は fitT1 と同程度、fit1 と比べると短めである。fitT0 と fitT1 はベジエパラメータの最適化法は同じであり、違いは制御点の求め方である。fitT1 が最小二乗法で制御点を求めるのに対し、fitT0 は勾配法で徐々に制御点を変化させていく。この構図は先のベジエパラメータの最適化における fit1 と fitT1 の関係と同じであるが結果は逆で、勾配法を使う fitT0 の方がステップ数は多くなっている。勾配法における収束までのステップ数は学習率に左右されるので学習率の設定を変えれば fitT0 のステップ数を抑え、処理時間を短縮できる可能性はある。

以上の結果と、fitT1 に特化した値を使えば、fitT1 の結果はさらに改善される（先に述べたが、この実験では比較のため fitT0 に合わせた同一の学習率を用いた）ことを考えると、今回比較した実装の中では fitT1 が総合的に優れていると言えよう。

許容誤差 $\text{err_th}=1.0$ 、近似次数 11、実装法 fitT1 の場合の近似結果を図 12 に示す。他の実装法による結果との違いはわずかである。

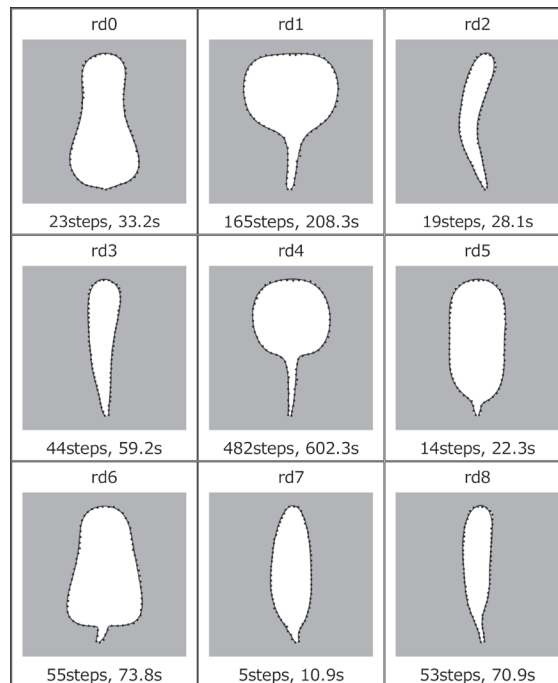


図 12 全周近似結果 (fitT1, 標本数 65, 11 次近似, $\text{err_th}=1.0$)

③ 実験 C 標本点数の違いによる近似結果の違い

ここまでの実験では標本数を65と多めに設定したが、標本数を変化させた場合のあてはめ結果の違いを調べる。近似次数を11, 許容誤差 err_th を1.0に統一し、標本数を21, 25, 33, 41と変化させて、fitT1で曲線あてはめを試みた。結果を図13～15に示す。

輪郭の曲率変化が緩やかな個体であれば標本数が21でもフィッティング結果は良好である。比較的複雑な形状の個体 rd1, rd4, rd6 は標本数21では変曲点付近でずれが目立つ。rd1では輪郭としてありえない曲線になってしまっており、標本数は21では少なすぎといえる。標本数25では rd4, rd6 の末端部で少し不自然な箇所が見られる。標本数33以上では、輪郭と近似曲線のずれはわずかであり、ダイコンの形状記述としては必要十分であると考えられる。

標本数の増加は当然計算量の増加につながる。標本数を横軸、処理時間を縦軸に取ったグラフを図16に示す。図13～図15には示していないが、図16には標本数29, 51, 65の結果も含まれる。

繰り返し最適化というアルゴリズムの性質上、計算量は標本数の単純な関数ではない。処理時間はおおむね繰り返し回数と標本数の積に比例するはずである。グラフを見る限り、標本数21の結果を除けば、グラフはおおむね緩やかな右肩上がりとなっている。上述のように標本数をむやみに大きくしても近似結果にはあまり差はないので全周近似の標本数は30～40程度が適当だといえる。

④ 実験 D オーバフィッティング対策の効果

ここまでの実験結果はすべて、両端の標本間に1点ずつ、計2点標本点を追加するという簡易的なオーバフィッティング対策を施した上での結果である。これは対策なしでは端部付近の曲線が不自然になるという現象が頻発することが経験上わかっているからであるが、実際対策がどの程度効果があるのかを確認するために、対策を施さなかった場合と施した場合を比較しておく。

同じ個体に対してオーバフィッティングが生じるかどうかは標本数と近似次数に左右される。組み合わせをすべて試すのは実験に時間がかかりすぎるので、標本数は65とし、10, 11, 12次近似。許容誤差1.0という条件でオーバフィッティング対策なしの場合とありの場合を比べてみた。

結果を表4に示す。数字は3.5の②で述べた、簡易的なオーバフィッティングの判定基準により、オーバフィッティングと判定された区間の番号である。標本数が65であるので区間は64区間、区間番号は0～63である。x印は当該次数では許容誤差内のフィッティングができなかったことを意味する。その場合は、許容誤差の設定がその結果の最小誤差と等しかったと仮定して判定している。

この実験により次のことがわかった。

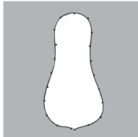
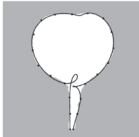
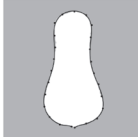
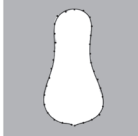
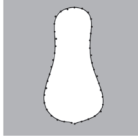
標本数	rd0	rd1	rd2
21	 25steps, 14.8s	 >3000steps, 1509.2s	 9steps, 6.7s
25	 24steps, 16.0s	 233steps, 134.7s	 16steps, 11.5s
33	 26steps, 21.3s	 236steps, 167.8s	 20steps, 16.8s
41	 30steps, 28.6s	 156steps, 133.1s	 22steps, 21.8s

図13 標本数によるフィッティング結果の違い (rd0, rd1, rd2)

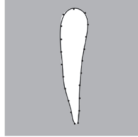
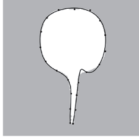
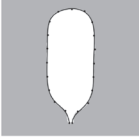
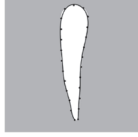
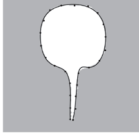
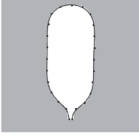
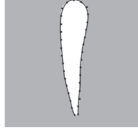
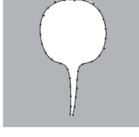
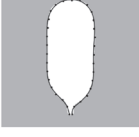
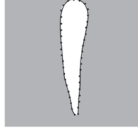
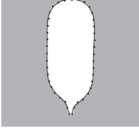
標本数	rd3	rd4	rd5
21	 26steps, 15.4s	 1381steps, 709.5s	 45steps, 24.8s
25	 30steps, 19.5s	 672steps, 383.0s	 29steps, 18.9s
33	 41steps, 31.9s	 540steps, 379.8s	 15steps, 13.9s
41	 46steps, 41.8s	 575steps, 480.0s	 25steps, 24.5s

図14 標本数によるフィッティング結果の違い (rd3, rd4, rd5)

ベジエ曲線によるダイコンの形状記述とその応用

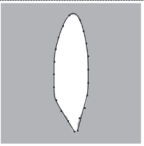
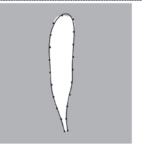
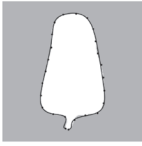
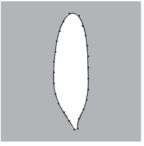
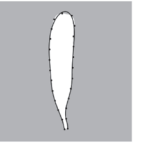
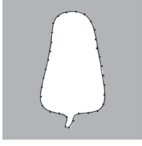
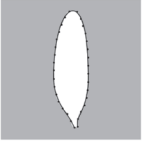
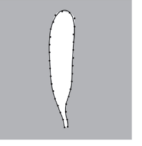
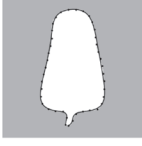
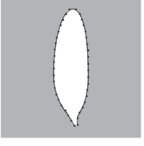
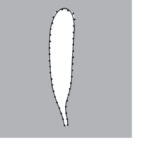
標本数	rd6	rd7	rd8
21	 92steps, 48.3s	 11steps, 7.7s	 14steps, 9.5s
25	 94steps, 55.8s	 3steps, 4.2s	 29steps, 19.0s
33	 57steps, 42.6s	 5steps, 6.5s	 33steps, 26.4s
41	 86steps, 74.8s	 4steps, 7.1s	 75steps, 66.3s

図15 標本数によるフィッティング結果の違い (rd6, rd7, rd8)

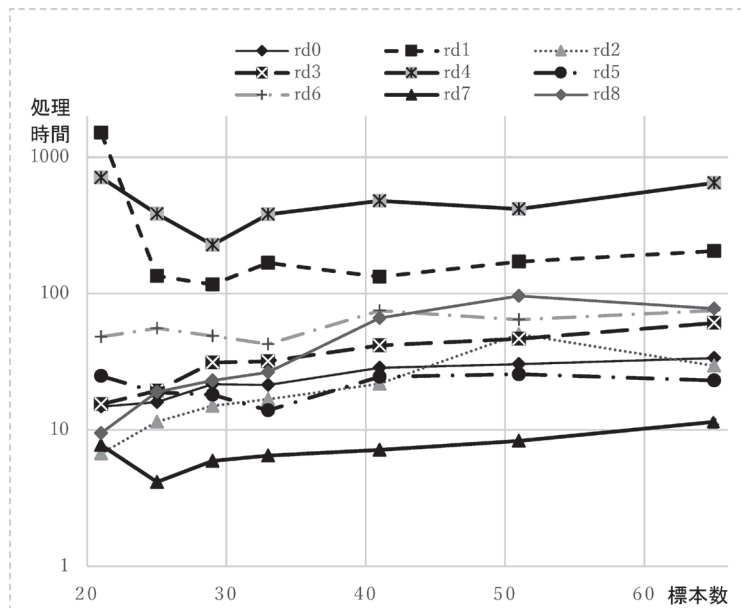


図16 標本数と処理時間の関係

- rd2, rd5, rd7はオーバフィッティングを起こしにくい。対策を施さなくても、10～12次の範囲でいずれの実装でも自動で検出されるような大きなオーバフィッティングは発生していない。
- fit1, fitT1は12次近似ならばオーバフィッティング対策を施さなくても、自動検出されるような大きなオーバフィッティングは発生していない。
- オーバフィッティングは端点部と変曲点付近に発生することが多い。端点部のオーバフィッティングはfd0, rd1, rd3, rd4, rd8で、変曲部のオーバフィッティングはrd4, rd6で発生することがある。
- 端点部のオーバフィッティング対策は効果的である。fit1, fitT1では端点部のオーバフィッティングが見られなくなり、fitT0では程度を抑えることができています。
- 端点部のオーバフィッティング対策が間接的に変曲点付近のオーバフィッティングを抑制することもある。

表4 端部オーバフィッティング対策の効果（許容誤差1.0, 標本数65）

オーバフィッティング対策なし（数字はずれの大きい区間の番号, xは誤差1.0以上）									
	fit1			fitT0			fitT1		
対象	10	11	12	10	11	12	10	11	12
rd0	0, 63						0, 63		
rd1	0, 63	0, 63					0, 63	0, 63	
rd2									
rd3	63			x 0, 63	0	0	63		
rd4	x 0, 63	0		x 52	52		x 0, 63	0, 63	
rd5									
rd6				x	x 2				
rd7									
rd8	63						63		

オーバフィッティング対策あり									
	fit1			fitT0			fitT1		
対象	10	11	12	10	11	12	10	11	12
rd0									
rd1									
rd2									
rd3				0, 63	0	0			
rd4				x	x 52				
rd5									
rd6				x	x 2				
rd7									
rd8									

図17にオーバフィッティングが抑制された結果の例を示す。左二つは端点部にオーバフィッティングが発生する例、右2つは変曲点付近にオーバフィッティングが発生する例である。

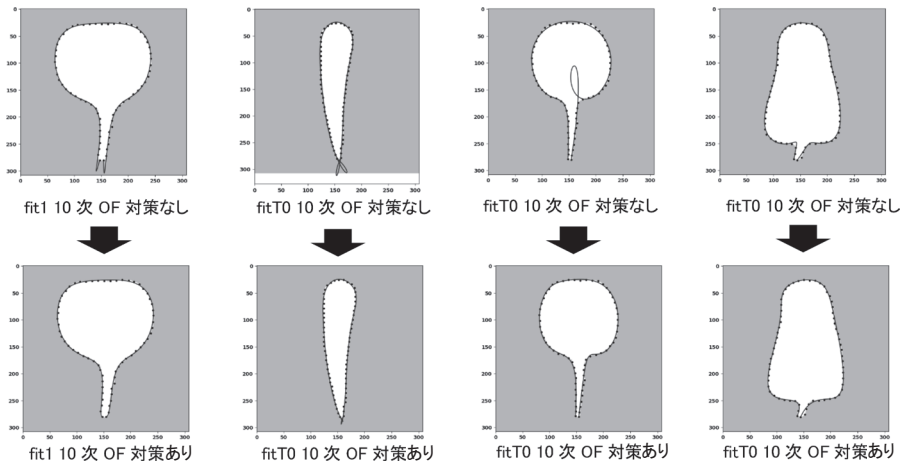


図17 オーバフィッティング対策の効果

4.3 輪郭の分割近似

図3では一つのシルエットを4分割し、4本の3次ベジエ曲線を接続することで1つの輪郭を表現するという例を示した。単に形状を記述すればよいだけであればどう分割しても記述は可能であるが、記述同士の類似性や差異を解析したり、その結果を分類に利用したりするには、自動であれ手動であれ、一意性のある分割法が必要になる。

ダイコンの形状を記述すると同時に計測を可能にするという観点からは、図18に示すような中心軸を基準とするような分割が理想的である。

論点が発散するし個体によって成否が左右されるので詳細説明は割愛するが、下端の分割点としては、先の全周近似で述べたように曲率最大点を候補点とし、上端の分割点としては局所的な対称性が極大化される点を候補として算出するプログラムを作成した。本稿で実験対象としている9つの個体では、妥当な分割位置が自動判定できている。また候補点が人の判断する分割位置とずれている場合に手動で分割位置を修正指定するようなツールも作成してある。以下の実験では、上部下部ともに自動判定された分割候補点から半径3画素以内の輪郭点を取り除き、左右2本の輪郭線点列とした。

3.1でダイコンの輪郭は2分割したとしても曲線上の座標値の成分同士が一意対応となるとは限らないと述べたが、図18の2つの形状はその例である。これらの輪郭は中心軸の上下端が水平になるように回転させたとしても、元の図形の上部位置で曲線がオーバハングし、同じ x 座標を持つ点が複数できてしまう。そのような曲線は多項式近似できない。

また，下部で同様のことが起こる個体もある．ベジエ曲線による近似はそのようなケースにも適用可能である．

全周輪郭を対象とした実験により，必要な標本点数，オーバフィッティング対策，実装法の優劣については一定の知見が得られたので，左右分割した輪郭の近似については近似に必要な次数のみを調べる．

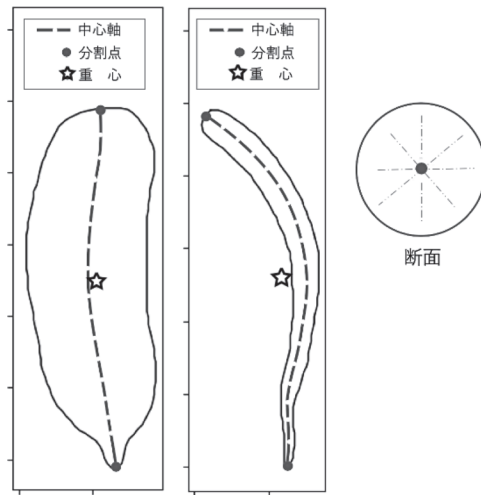


図18 中心軸を基準とした輪郭の分割

実験 E 左右分割輪郭の近似に必要な次数

標本点の数を22とし，実装法は fitT1 を用いて，あてはめ誤差1.0未満を達成できる近似次数を調べてみた．結果を表5に示す．表からわかるようにシルエットを左右に分割して別々に扱い，あてはめ誤差の許容基準1.0で近似する場合，4～6次のベジエ曲線で近似できる．

ダイコンの形状の中でもっとも複雑なのは rd0 や rd6 のように中くびれのある形状であろう．直感的にはそれらは4次の曲線で近似できそうであるが，結果は rd0 が予想通り4次であるのに対し，rd6 は6次となった．違いは rd0 には尻尾状の部分がなく，rd6 にはあるという違いである．他の個体についても，尻尾状の部分を近似対象から省いてよいのであれば次数を1ないし2減らすことができると思われる．しかしながら，形状記述前で中心軸も不明な時点で，不要部分を自動判定するのは困難である．また，先行研究[5,6]では，肥大部下部の幅の減衰曲線を品種の特徴の一つとして挙げており，むやみに先端部をカットするわけにもいかない．

近似次数を6次に統一し， $err_th = 1.0$ とした場合の近似の結果を図19に示す．

rd5 と rd6 の下部の先端に輪郭とのずれが少し見られるが，下部はひげ根の処理の仕方で形状がさまざまに変化するため，それらの部分を忠実にトレースするのは無意味であ

る。また上部も葉部との切断によって形状が不連続となりがちである。本稿の実験では上下の除外対象を小さめに設定したが、形状解析を目的とする場合は切断面や下部尖端部で形状が不規則な部位は対象から除外すべきである。

表5 左右分割近似に必要な最低次数

	左側	右側
rd0	4	4
rd1	4	5
rd2	4	4
rd3	4	4
rd4	5	6
rd5	5	5
rd6	6	6
rd7	4	4
rd8	4	5

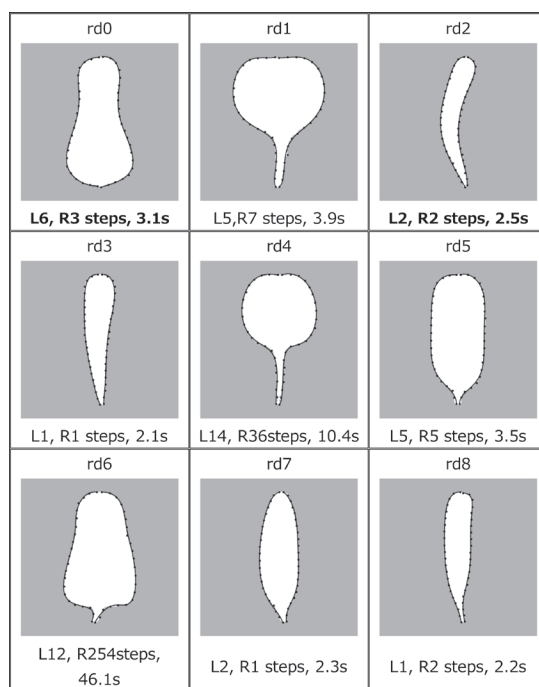


図19 左右分割輪郭の近似結果 (fitT1, 標本数22, 6次近似, err_th=1.0)

5. 自動計測への応用

本稿で提案したベジエ曲線による形状記述を形状分析や自動計測に応用することを考える。

5.1 中心軸の推定

砲弾型の典型的なダイコンをイメージすると、「長さ」や「幅（径）」という属性は自明で疑問の余地がないように思われるが、個々の個体は多かれ少なかれ曲がりがあり、角型のように曲がっていることが特徴となっている品種さえ存在する。何をもって「長さ」と定義するのは自明ではない。「幅（径）」についても同様である。「長さ」や「幅」という概念は中心軸の存在を暗黙の裡に前提としており、「長さ」とは中心軸の長さ、「幅」とは中心軸に対する垂直断面の径と考えるのが自然である。人手で計測する際には、頭の中でイメージする中心軸に沿った長さや幅を計測することになるが、それでは個人差が避けられない。

ここでは4.3で説明した左右分割のベジエ近似曲線を用いて中心軸を推定する方法を示す。

自動計測が目的であるので、得られる軸がダイコンの真の中心軸に一致するかどうかはあまり重要ではない。人がイメージするのに近い軸が機械的に得られるということ、よって計測結果に個人差がでない、ということに意味がある。

中心軸の推定は、①まず、左右の輪郭の平均曲線を求め、②その曲線上に等間隔に取った標本点に低い次数のベジエ曲線を当てはめて中心軸と見なす、という手順で行う。

① 左右輪郭の平均曲線

左右の近似ベジエ関数をそれぞれ $B_L(t)$, $B_R(t)$ とし, $B_c(t) = (B_L(t) + B_R(t))/2$ と定義する。rd0～rd8 について $B_L(t)$, $B_R(t)$, $B_c(t)$ を描いたものを図20に示す。 $B_L(t)$, $B_R(t)$ のパラメータ t は互いに独立であり, $B_c(t)$ は単に $B_L(t)$, $B_R(t)$ の同一パラメータ値の点同

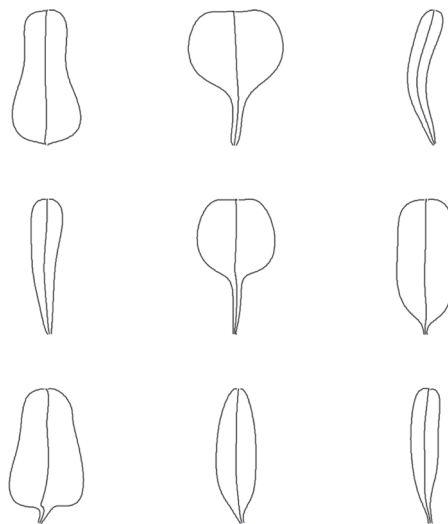


図20 左右の輪郭の平均曲線

土の midpoint の軌跡を表すに過ぎないが、 $B_c(t)$ はおおむね中心軸として違和感のない位置を通ることがわかる。ただし、例えば rd6（左下）の下部のように、左右の輪郭形状の曲率が大きく変化する場所では不自然になりがちである。

② 再近次による中心軸の推定

$B_c(t)$ の次数は $B_L(t)$, $B_R(t)$ の次数と同一になるが、中心軸が複雑な凹凸をもつのは不自然である。S 字状や弓状に湾曲した個体まで考慮するとしても高々 4 次程度の曲線ととらえる方が自然である。そこで、曲線 $B_c(t)$ 上に等間隔になるように標本点を設定し、4 次のベジエ曲線をあてはめる。あてはめのアルゴリズムは輪郭線と同一である。結果を図 21 に示す。

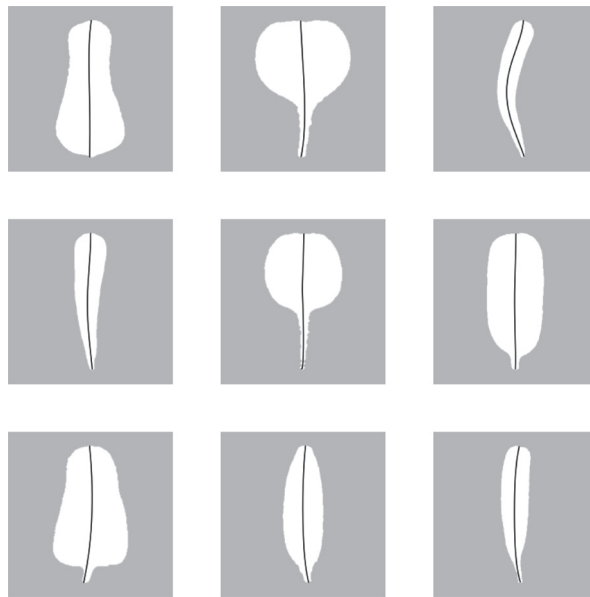


図21 中心軸の推定

5.2 曲がり補正と自動計測

中心軸上の各点における垂線断面がシルエットから切り取る線分の長さを、その点における「幅（径）」、中心軸に沿って測った中心軸の長さを「個体長」と定義する。中心軸の一方の端点を始点として中心軸に沿って測った距離を縦軸、幅を横軸にとってプロットすれば、曲がりがなかった場合の形状を合成することができる。

① 長さの自動計測

原画像と同じサイズの画像キャンバスに中心軸を点列として描き、OpenCV の

arcLength() 関数でその点列の長さを求めることで、中心軸の長さを近似計算することにする。中心軸がパラメトリック関数として表現されているので、数値積分で長さを求めることも可能である。

② 幅の計測

ベジエ曲線で表現した中心軸が数式として表現されているので、中心軸上の任意の点での垂線の方程式も数式处理的に得られる。理論的には、この直線の方程式と両側のベジエ曲線の方程式を連立させて解けばよい。両側が6次で表現されているならば、6次方程式となり、6つの複素解が得られるが、パラメータ範囲が $[0,1]$ の実数と決まっているので、その範囲の解を取り出せばよい。実際それが可能であることは確かめたが、次数が高いため大変計算時間がかかる。別の解法として、実際に輪郭線と垂線を画像として描くことで交点を求めるという方法が考えられる。これは厳密解とはならないが、原理的に誤差はサブピクセル程度であり、数式処理より圧倒的に高速に計算できる。

③ 曲がり補正形状の合成

中心軸に沿って幅をサンプリングし、幅の半値を座標、中心軸に沿った長さを座標とみなしてベジエ曲線をあてはめることで、曲がりのない場合の輪郭線形状を求めることができる。この際、中心軸上で等間隔になるように幅の標本を採るのは容易であるが、そうすると、図22の(a)のように、輪郭の方向が垂直に近い箇所では標本点は密に、水平に近い箇所ではまばらになってしまう。逆に(b)のように左右の輪郭上に等間隔に標本点を設定して対応する点を結んで幅とするという考え方だと対応点同士を結ぶ直線が中心軸と垂直になるとは限らず、幅の値としてはふさわしくなくなる。ここでは次善の策として、次のようにして幅の標本データを作成することにした。

- i) 左右の輪郭の近似曲線上で等間隔になるように標本点を設定する。

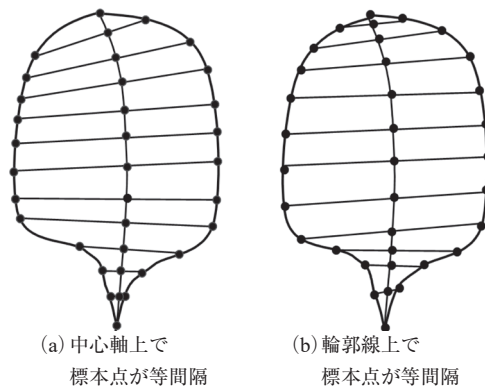


図22 幅の標本化

ii) 左右の対応する標本点を結んでできる直線と中心軸の交点を中心軸上の標本点とする。

iii) 中心軸上の各標本点において中心軸に対する垂線を引き、左右の近似曲線との交点をそれぞれ求め、それらを結ぶ距離を幅の標本値とする。

この手順で定まる中心軸上の各標本点について、中心軸上端から軸に沿って測った距離を y 座標、幅の半値を x 座標として座標列をつくり、これまでと同じようにベジエ曲線をあてはめてやれば、曲がりがない場合の形状が得られる。

標本点数を22、近似の次数を6とした場合の結果を図23に示す。

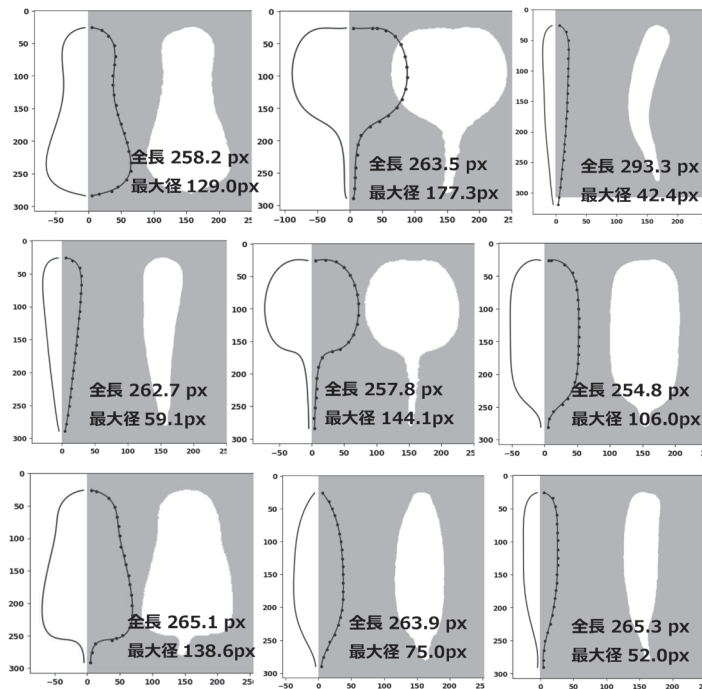


図23 補正形状の生成と自動計測結果

6. おわりに

われわれがイメージする典型的なダイコンのシルエットは左右対称であるので、形状を記述したいなら輪郭の左右どちらかの形状を記述すれば十分のように思われる。また、典型的なダイコンだけをイメージすると、その形状は2次関数か、多めに考えても高々4次関数で近似すれば十分のように思える。しかし、実際の個体の中心軸は直線とは限らず、中心軸の曲がりが特徴となっている品種もある。また輪郭の形状は品種によって多種多様で、ダルマ型のような複雑な輪郭を持つ品種もあるし、個体それぞれは完全な左右対称ではない。本研究では、そのような個性を含めて、個体の形状を少ないパラメータでなるべ

く忠実に記述することを目的としており、今回はベジエ曲線をあてはめるという方法について検討した。

まず、シルエットにベジエ曲線をあてはめるアルゴリズムを示し、輪郭全体の近似に必要な次数と標本数、輪郭を2分割した場合の近似に必要な次数を実験的に示した。もちろん、これらの値はどこまで忠実に形状の凹凸を再現したいかによって変わるが、形状特徴を大まかにとらえて品種の識別を行ったり、個体同士の形状の差異を比較したりすることが目的であるならば、本稿で示した程度（全周で10次、片側で6次）の近似で十分目的を果たせるであろう。また、得られた形状記述により機械的に計測を行う方法も示した。個体の曲がりにも対応でき、計測者ごとの計測値のばらつきをなくすることができるようになる。

本研究を通して、輪郭をベジエ曲線で記述することで様々な応用の可能性が拓けることが分かった。本稿では輪郭線の近似をスタートとして、中心軸、そして補正形状の記述にベジエ近似を利用した。説明を省略したが実はシルエットの分割点を求めるための曲率最大点を見つける際にも内部的にベジエ近似を利用している。その他、接線や垂線を求めたり、交点を計算したりといった、数值的に行うと大変面倒な処理が数式処理として簡単に記述できる点もメリットである。

本稿では曲がりに沿った全長と最大径の自動計測の例を示したが、本手法では中心軸と径の関数表現が得られるので、任意の部位の径や径の変化のパターン、中心軸の曲率やその変化のパターンなどをとらえることが可能である。条件に合う部分の取り出しや削除も容易であり、全体だけでなく部分形状に基づいた解析や品種の分類、識別などに有用である。最初に述べたように、文献[5,6]における評価方式では本報告の形状記述法を利用している。ベジエ近似した記述に基づいて、長幅比、最大径の位置と程度、曲がり特徴量を自動計測し、また、先端部のみの形状を再記述、根形評価に利用している。

参考文献・参考 URL

- [1] Iwata, H., S. Niikura, S. Matsuura, Y. Takano and Y. Ukai : “Evaluation of variation of root shape of Japanese radish (*Raphanus sativus* L.) based on image analysis using elliptic Fourier descriptors”, *Euphytica* 102, pp.143-149 (1998).
- [2] 淡誠一郎 : “根菜の画像解析のための形状記述についての研究ノート—フーリエ記述子による形状記述—”, 大阪学院大学 人文自然論叢, 79-80, pp.55-67 (2020).
- [3] 淡誠一郎 : “デジタル画像におけるフーリエ記述子の回転不変性の検証”, 大阪学院大学 人文自然論叢, 79-80, pp.1-20 (2020).
- [4] Iwata, H. and Y. Ukai : “SHAPE: a computer program package for quantitative evaluation of biological shapes based on elliptic Fourier descriptors”, *J. Hered.* 93, pp.384-385 (2002).

- [5] 淡裕美子, 淡誠一郎, 吉田康子: “ダイコンの根形を特徴づける形質の選定とその評価方法の確立”, 育種学研究19 (日本育種学会 第132回講演会要旨集), 別2, p.77 (2017).
- [6] 淡裕美子, 淡誠一郎, 吉田康子: “画像解析を用いたダイコンの根形を特徴づける形質による多様性評価”, 育種学研究21 (日本育種学会 第135回講演会要旨集), 別1, p.101 (2019).
- [7] “農林水産省 農林水産植物種類別審査基準－ダイコン”, <http://www.hinshu2.maff.go.jp/info/sinsakijun/kijun/1528.pdf> (2017).
- [8] “農業生物資源ジェンバンク 特性評価マニュアル”, https://www.gene.affrc.go.jp/manuals-plant_characterization.php (2023).
- [9] IBPGR: “Descriptors for Brassica and Raphanus”, Bioversity International (1990).
- [10] “UPOV Guidelines for the conduct of tests for distinctness, uniformity and stability, Radish”, https://www.upov.int/en/publications/tg-rom/tg064/tg_64_6.pdf (1999).
- [11] “UPOV Guidelines for the conduct of tests for distinctness, uniformity and stability. Black radish”, https://www.upov.int/en/publications/tg-rom/tg063/tg_63_6.pdf (1999).

プログラムの公開

本研究で説明したシルエットのベジエ曲線あてはめを含む形状解析用プログラムをGitHubに公開している. → RadishLab <https://github.com/iciromaco/RadiShLab>

(2023年2月9日受理)

発達障害グレーゾーンの大学生を観察でアセスメントする方法の一試案 ～大学教員ができる発達アセスメント～

山 田 章

A Proposed Method for Observational Assessment of College Students with Developmental Disabilities Gray Zone

Akira Yamada

I はじめに ～なぜ、観察によるアセスメントか～

1. 大学におけるグレーゾーンの大学生の増加

2007年の特別支援教育の実施以降15年が経ち、発達障害（診断あり）の大学生が増えてきている。

独立行政法人日本学生支援機構が2022年に公開した「2021年度）大学、短期大学及び高等専門学校における障害のある学生の修学支援に関する実態調査分析報告」によると、発達障害（診断書有）の人数は、8,698人である。調査が始まった2006年は127人だから、かなり増加していると言える。

上記は、診断書のある人数だが、2020年1月博報堂 Pechat 開発チーム・博報堂こそだて家族研究所・LITALICO 発達ナビが行った「ASD と子育て実態調査」第一弾調査では「13,262人の保護者の回答のうち、0～22歳の子で『ASD（自閉症スペクトラム）と診断された子』は2.3%、診断は受けていないがASDの疑いがある『グレーゾーンの子』は5.4%だった。」とある。

つまり、診断書をもった発達障害の学生以外にも「発達障害グレーゾーン」（以下グレーゾーン）と呼ばれる学生が、大学には一定数いることになる。

2. 私立大学のグレーゾーンの学生に対する合理的配慮の義務化

『障害を理由とする差別の解消の推進に関する法律』（以後『障害者差別解消法』）によると「障害者の法的範囲」を「(前略) これは、障害者が日常生活又は社会生活において

受ける制限は、身体障害、知的障害、精神障害（発達障害を含む）その他の心身の機能の障害（難病に起因する障害を含む）のみに起因するものではなく、社会における様々な障壁と相対することによって生ずるものとのいわゆる『社会モデル』の考え方を踏まえている。したがって、法が対象とする障害者は、いわゆる障害者手帳の所持者に限られない。」としている。（下線部は筆者）

つまり、この『障害者差別解消法』は、最終的にはいわゆる障害者手帳を所持していないグレーゾーンをも支援対象にしているのである。この『障害者差別解消法』が、2021年にさらに改正された（2021年6月4日発布3年以内に施行）。社会的障壁に対する「合理的配慮」が、今までは民間事業者（私立大学を含む）には努力義務となっていたが、今回の改正によって、今後は義務として配慮提供が求められるようになったのである。

つまり、私立大学においても、在籍するグレーゾーンの大学生に対して合理配慮（支援）をしていく義務があるのである。

3. 観察によるグレーゾーンのアセスメント

では、どのようにすればそのグレーゾーンの学生に合理的配慮、つまり支援ができるのだろうか。

2011年公開の「特別支援教育の在り方に関する特別委員会（第12回及び第13回）における教職員の確保及び専門性の向上に関する主な意見の総論」には、「（前略）その子どもにとって必要な支援は何かをアセスメントし、支援計画、指導計画に反映する技術は大切な専門性である。この二つが機能しないと、子どもに対する支援は充実しない。」と述べられている。

つまり支援教育においては、発達的アセスメント（以後アセスメント）を行い、その後合理的配慮のある支援計画を立てる必要があるのである。しかし、大学に来るグレーゾーンの学生の殆どは、自ら合理的配慮を申し出てこないのである。その理由は、自分の脳の特性に気づいていない、気づいていてもなんとなくきた自信がある、気づいているが知られたくない、気づいているがどこに相談していいか分からないなど、様々である。だから、アセスメントが必要だと感じたときには、大学の教員自らが何らかの方法でアセスメントするスキルを獲得する必要があるのである。

そこで、学生側から申し出てこなくても、教員の観察のだけでグレーゾーンの学生を発見しアセスメントできるツール『観察だけでできる発達アセスメントツール』を開発した。開発にあたっては、筆者が多数の WISC 検査を実施しアブダクションを駆使して開発した「小中学校版 観察だけでできる発達障害の実践的分類とその支援の理論と方法」（参考文献『教室で使える発達の知識』参照）を基盤にした。

Ⅱ 大学の教員が『観察だけでできる発達のアセスメントツール』の使い方の解説

大学の教員が『観察だけでできる発達のアセスメントツール』の使い方を、筆者による「発達障害の実践的分類」別（参考文献『教室で使える発達の知識』参照）に解説していく。この方法は、教員の観察だけで行えるので、コンプライアンスを気にすることなく発達のアセスメントできるのが特徴である。

1. 《見たことをまとめるのが苦手》なタイプのアセスメント

(1)脳機能のどこに偏在（脳機能の凸凹）があるのか？

このタイプは、視覚情報を取り入れて一時的に記憶しておく《視覚的短期記憶》か、取り入れた視覚情報を更に保持しながらまとめ上げ理解しそれに対して思考する《視空間的ワーキングメモリー》か、それらまとめあげて理解し思考を加えた情報を使って《身体・手指》を使うところのどこか、または全部に脳の機能偏在がある。

よって、このタイプを《見たことをまとめるのが苦手》なタイプと呼んでいる。

(2)観察によるアセスメントの方法

このタイプの《視覚的短期記憶》、《視空間的ワーキングメモリー》《身体・手指》の機能偏在を見つけるために、表1の10項目を観察してチェックする。この10項目のうち5項目以上チェックが付いたら《見たことをまとめるのが苦手》なタイプということになる。

表1 見たことをまとめるのが苦手なタイプのチェック表

① 手書きのレポートの字が、ひどくて読めない
② 文章を読むとき、読むところを指で押さえて読んでいる
③ 内容がまとまらないので、レポートが書けない
④ 映像を見るときに、寝てしまったり退屈してしまったりする
⑤ 板書をノートに写すとき、他の学生よりかなり時間がかかる
⑥ 机の上が片付かないので、ものが机からよく落ちる
⑦ 話合いの時に、状況把握がうまくできない（空気が読めない）
⑧ 相手の表情から、相手の気持ちを推し量ることができない
⑨ 運動が、苦手 よく人にぶつかる 机等でよく足を打つ
⑩ 図形やグラフが、書けない 絵や観察画などが大変下手

各項目について、観察するスキルを身につけるための参考として、視覚認知の観点から若干の解説を加える。

① 手書きのレポートの文字が、ひどくて読めない

人に読ませるためのレポートの文字が、乱雑なのはきれいに書こうとしてもうまくいかないからである。それは、視覚的に文字の構成を見て記憶したり、頭の中で再構成して手指で再現したりする力が弱いことに原因がある。ただし、きれいに書くように指示したら書ける場合は、この項目に該当しない。

この項目は、特にレポートの漢字を観察して、線が傾いていたり、1本足らなかつたり多かつたり、口が丸かつたり、編と旁がずれていても判断できる。白紙に書いた文字列が傾いたり、ノートに書いた横文字がぎっしりと上に固まっていたりしても、この項目にチェックをつける。

② 文章を読むとき、読むところを指で押さえて読んでいる

文章を読む場面で、文が跳んでしまったり読んでいるところが分からなくなったりするのを防ぐために、読んでいるところを指でたどりながら読んでいる学生がいたら、それはたくさん並んでいる文字を視覚的に正確に追っていけないからである。

筆記用具で線を引きながら読んだり、文字列にものさしを当てて読んだりしている学生もこの項目にチェックをつける。

③ 内容がまとまらないので、レポートが書けない

「本や論文を読み、インターネットで情報を沢山集めたが、レポートをどこから書いていいか全然分からない」と訴える学生がいたりする。様々な情報を理解するためには言語的ワーキングメモリーを使うが、分類して整理し図式化してレポートにまとめるためには、視空間的ワーキングメモリーを使う。だから、そこが弱いから、「うまく情報をまとめることができない」と言うのである。

④ 映像を見るときに、寝てしまったり退屈してしまったりする

授業中に映像を見る場面があると、机にうつ伏せになって寝てしまったり、退屈そうにしている学生が見られたら、それは映像という大量の視覚情報を受け取るために大変疲れてしまうためか、その視覚情報がうまく取れないために退屈してしまうからだ。

本当に寝ているのではなく寝ているフリだと分かっても、その他、映像を見るたびに何かの文具やスマホ等で遊んでいる学生もこの項目をチェックする。

⑤ 板書をノートに写すとき、他の学生よりかなり時間がかかる

板書をノートに写すとき、他の学生よりかなり時間がかかる学生がいたら、それは、黒板を写すためには黒板の文章を覚えて書く必要があるのに、視覚的短期記憶が弱いと文章がなかなか覚えられないからだ。つまり、覚えられないので黒板を頻繁に見ることになり、写すのに時間がかかるのだ。

⑥ 机の上が片付かないので、ものが机からよく落ちる

授業中の机の上に物があふれ返っていて、ものを床に落して大きな音を立てる学生がいれば、それは、片付けができないからだ。片付けるためには、ある一定の空間にもものを重ねて隙間のないようにはめ込んでいく視空間認知が必要なのだが、そこが弱いからで片付

けができないのである。だから、自然と机からものが落ちて音を立てるのである。

⑦ 話合いの時に、状況把握がうまくできない（空気が読めない）

授業のときに、発問に対してトンチンカンなことを言い出す学生や少人数で話し合っているときに、場違いなことを発言する学生がいる。俗にいう「空気が読めない」状態の学生である。

今、周囲で何が行われているのか、どんな状況なのかを把握するためには、たくさんの視覚情報を瞬時に取り入れて、まとめあげて現状がどうなっているのか理解しなければならない。そのためには、視覚的短期記憶と視空間的ワーキングメモリーを働かせ続けなければならないのだが、空気の読めない学生はそこが弱いのである。

近いが、ちょっと違う場合がある。視覚情報は正確に取り込んでいるが、まとめ方が自分勝手な考え方（自分ルール）のために「空気が読めないかような状態」になっている場合だ。この場合は、この項目にチェックはしない。

⑧ 相手の表情から、相手の気持ちを推し量ることができない

人の顔から気持ちを推測するためには、顔の表情を読み取る必要がある。人の顔の表情には、たくさんの視覚情報が含まれている上に時々刻々と変化していく。それを随時読み取って相手の気持ちを推し量るためには、視覚認知を最大限に使わないといけないが、視覚認知が弱いとこれがうまくできないのである。

よく似ているが、最初から人の目（顔）を見ないために、人の気持ちが分からない学生もいる。俗にいう「目が合わない」学生だ。そのケースは、そもそも視覚情報を相手から取っていないので、この項目にチェックは入れない。

⑨ 運動が、苦手 よく人にぶつかる。机等で足を打つ

運動は、身体の完成（参考文献『感覚統合 Q & A』参照）とも大きく関係するが、基本的に他者の動きを視覚的に捉えて再現することで学んでいくものである。だから、視覚認知が弱いと、運動が、特に鉄棒・跳び箱・マット・縄跳びが苦手になる。

自分と物との関係を把握するのに、視空間認知を使わないといけない。だから、授業の始まりや終わりなど移動するときに、人とよくぶつかったり机やイスに足をぶつけたりしている学生がいたら、その学生は視空間認知が弱いことになる。つまり、自分と他人や机、イスとの距離の把握に失敗しているのである。

⑩ 図形・グラフが、うまく書けない。絵や観察画などが大変下手

大学には美術の授業がないので絵を描く機会が少ないが、授業中に図形やグラフを書いたり実験の観察画を書いたりする。図形やグラフや観察画には視覚認知を駆使するので、うまく書けない学生がいれば視覚認知が弱いことが分かる。

どうしてもチェックできないときは、何かの絵（キャラクター）を見て写させるか、フリーハンドで菱形を書かせてチェックしてもよい。キャラクターが全然似ていなかったり、菱形の角が丸くなっていたりしたら、この項目にチェックを入れる。

2. 《聞いたことをまとめるのが苦手》なタイプのアセスメント

(1) 脳機能のどこに偏在（脳機能の凸凹）があるのか？

このタイプは、聴覚情報を取り入れて一時的に記憶しておく《聴覚的短期記憶》か、取り入れた聴覚情報を更に保持しながらまとめ上げ理解し、それに対して思考する《言語的ワーキングメモリー》か、それらまとめあげて理解し思考を加えた情報を使って《コミュニケーション》するところのどこか、または全部に脳の機能偏在がある。

よって、このタイプを《聞いたことをまとめるのが苦手》なタイプと呼んでいる。

(2) 観察によるアセスメントの方法

このタイプの《聴覚的短期記憶》、《言語的ワーキングメモリー》《コミュニケーション》の機能偏在を見つけるために、表2の10項目を観察してチェックする。この10項目のうち5項目以上チェックが付いたら《聞いたことをまとめるのが苦手》なタイプということになる。

表2 聞いたことをまとめるのが苦手なタイプのチェック表

① 長い説明で、寝てしまったり退屈してしまったりする
② 先生が指示すると「もう一度言ってください」と聞いてくる
③ 先生が指示すると、周りを見てから行動するので少し遅れる
④ 発問に対して指名を受けても、固まってしまって発言しない
⑤ 少人数での話し合いで、聞いているだけ参加しようとししない
⑥ 本人の話がダラダラと長く、何を話しているのか分からない
⑦ トラブル時に、暴言を吐いたり暴力を振るったりする
⑧ 九九を覚えるのに苦労した 大学生になっても九九は苦手だ
⑨ 文章を読ませると、語尾が勝手読みになる
⑩ 漢字は書けるが、読みや意味を聞くと「知らない」と答える

各項目について、観察するスキルを身につけるための参考として、聴覚認知の観点から若干の解説を加える。

① 長い説明で、寝てしまったり退屈してしまったりする

長い説明を聞き取るためには、聴覚的短期記憶を継続的に使用しながら、取り入れた聴覚情報を言語的ワーキングメモリーでどんどんまとめ上げていく必要がある。うまくできないと、今の説明の内容が理解できる前に、更に次の聴覚情報が入ってくる状態になる。すると、説明内容が混迷してきて最後には説明を聞くことを諦めてしまう。だから、疲れてしまって寝てしまったり、寝た振りで誤魔化したりするのである。

長い説明が始まるたびに、退屈してしまって近くにある文具やスマホで遊んでいる学

生も、この項目をチェックする。

② 先生が指示すると「もう一度言ってください」と聞いてくる

先生の指示を聞いて実行するためには、聴覚短期記憶と言語的ワーキングメモリーを使う。つまり、先生の指示を聞いていたのに「もう一度言ってください」という学生は、指示が聞き取れなかったか、聞いていたのに理解できなかったことになる。つまり、聴覚的短期記憶か言語的ワーキングメモリーが弱いことになる。

③ 先生が指示すると、周りを見てから行動するので少し遅れる

聴覚情報がうまく聞き取れなかったり理解できなかったりすると、人は視覚情報で聴覚情報を補う。だから、何か行動する指示を出されたときに、周りを見て行動が遅れてしまう学生は、聴覚認知が弱いことになる。

④ 発問に対して指名を受けても、固まってしまって発言しない

先生が発問してそれに応えるためには、聴覚的短期記憶を使って先生の発問情報を聞き取り、言語的ワーキングメモリーを使って素早く理解しそれに対する答えを考える必要がある。指名を受けて発言を求められたのに、ただ立つだけで全然発言しようとしなない学生は、聴覚的短期記憶か言語的ワーキングメモリーが弱いので、立ったまま黙って固まってしまうしか他に方法がないのである。

⑤ 少人数での話し合いで、聞いているだけ参加しようとしなない

少人数で話し合うためには、聴覚認知を頻繁に使わなければならない。先生の課題指示内容や司会の言葉、他のメンバーの意見など、すべて聴覚情報である。これらの情報を聴覚的短期記憶で聞き取って、それに対して頭に浮かんだ自分の意見をまとめるのにも言語的ワーキングメモリーを使う。これらが全てできてやっと自分の意見が言えるのだが、これができないと思った学生は、前もって「話し合いには、参加しないでおう」という態度を取るのである。つまり、聞くことだけに専念するのである。

⑥ 本人の話がガラガラと長く、何を話しているのか分からない

自分の考えを他人に伝える場合、前もって言語的ワーキングメモリーでまとめておく必要がある。ここが弱いと、聞いている方はまとまっていない情報を長々と提供されるので「この話し手は、どんな考えを持っているのか」と全く理解できなくなるのである。だから、ガラガラと長く話す学生は、言語的ワーキングメモリーが弱いと言えるのである。

⑦ トラブル時に、暴言を吐いたり暴力を振るったりする

コミュニケーションするためには、相手の意見を聴覚短期記憶で聞き取り、言語的ワーキングメモリーを使ってしっかり理解し自分の意見を考える必要がある。平常時にそこがしっかりできていれば、トラブルになったときにもコミュニケーションをして解決しようとする。そうせずに、トラブル時に暴言・暴力が出るのは、普段から聴覚認知が弱くうまくコミュニケーションしてないために、短絡的に頭に浮んだ言葉や言葉を介さない行動に出るから暴言や暴力になるのである。

⑧ 九九を覚えるのに苦労した。大学生になっても九九は苦手だ

九九は小学校のときに聴覚認知を使って覚える。つまり、口で唱えて音声情報として覚えるのである。だから、聴覚的短期記憶が弱いと、九九を覚えるに大変苦労したはずである。そのため、大学生になっても苦手意識が残っていることが多い。

苦手がどうか試してみるならば、特に聴覚認知が弱いと覚えにくい七の段と四の段が最適である。

⑨ 文章を読ませると、語尾が勝手読みになる

文字を目で読むときは視覚認知を使っているが、声を出して読むときには、その情報を言語的ワーキングメモリーで聴覚情報に置き換えてから読む必要がある。そのときに、言語的ワーキングメモリーが弱いと、よく知っている単語や語尾が出現したときに「また出てきた」と連想して、自分の読みやすいように読んでしまう現象が起こる。それが「勝手読み」である。つまり、「勝手読み」が出る学生は、言語的ワーキングメモリーが弱いのである。

⑩ 漢字は書けるが、読みや意味を聞くと「知らない」と答える

漢字を学習するときには形は視覚認知を使って覚えるが、読み方や意味については聴覚認知を使う。それをワーキングメモリーで統合してから長期記憶に送って覚えていくのだが、聴覚認知が弱いと読み方や意味がうまく学習できないので、長期記憶には視覚情報だけが送られたり、間違った読みや意味が統合して送られたりする。だから、板書を写すときなどに漢字が書けるのに意味の知らない学生がいれば、聴覚認知のどこかが弱いのである。

3. 《注意集中に課題がある》タイプのアセスメント

(1)脳機能のどこに偏在（脳機能の凸凹）があるのか？

人は情報を得ようとしたときには、興味のある対象に注意を向け集中したのち視覚認知や聴覚認知を使い始めるのだが、視覚認知や聴覚認知を使い始める前に、その対象から注意集中が本人の意思とは関係なく転動してしまう人がある。つまり、その人は「注意集中」を自動的にコントロールするところから脳の偏在があるのである。そのため、行動が多動になったり、衝動的な行動が増えたりする。

よって、このタイプを《注意集中に課題がある》タイプと呼んでいる。

(2)アセスメントの方法

このタイプの「注意集中」をコントロールすることの脳の機能偏在を見つけるために、表3の10項目を観察してチェックする。この10項目のうち5項目以上チェックが付いたら《注意集中に課題がある》タイプということになる。

表3 注意集中に課題があるタイプのチェック表

① 授業中、席で絶えずゴソゴソして落ち着きがない
② 足をイスの座面に折り曲げて座っている
③ 周りの音や周りの目につく物で、気が散っている
④ 絶えず友達としゃべっている 注意してもやめられない
⑤ 授業中に指示を出すと、すぐさま聞き返す
⑥ 授業中に指名されないのに、発言することがある
⑦ きれいに書けるのに、面倒くさいので適当に殴り書きしまう
⑧ 計算等の簡単なことで、ケアレスミスをよくする
⑨ がんばって課題をしているのに、ふと気が散る
⑩ 漢字で、首尾一貫しない間違い方をする

各項目について、観察するスキルを身につけるための参考として、注意集中の観点から若干の解説を加える。

このタイプは、他のタイプと混同しやすいので判定に注意する必要がある。《見たことをまとめるのが苦手》《聞いたことをまとめるのが苦手》なタイプだと判定した場合は、その判定を優先させるべきである。なぜなら《見たことをまとめるのが苦手》は視覚情報で《聞いたことをまとめるのが苦手》は聴覚情報で、注意集中が切れやすいからである。つまり、まず《見たことをまとめるのが苦手》《聞いたことをまとめるのが苦手》のタイプをチェックし、《見たことをまとめるのが苦手》《聞いたことをまとめるのが苦手》のタイプだと判断したなら、《注意集中に課題がある》タイプではないということになる。

① 授業中、席で絶えずゴソゴソして落ち着きがない

落ち着きのなさは、脳機能の偏在のせいで注意集中が次から次へと転動し、体がそれに反応してしまうことによって起こる。つまり、注意集中できていないと、本人の意思とは関係なく体が動いてしまうのである。それが「ゴソゴソする」「じっとしているのが難しい」の正体である。

② 足をイスの座面に折り曲げて座っている

足をイスの座面にあげて足を折り曲げて座っているのは、勝手に動いてしまう体を制御したり、体が傾いてしまったりするのを防ごうとしている行動である。だから、イスの座面に足を折り曲げて座っている学生は、自分の弱い注意集中を高めようと頑張っているのであり、つまり、注意集中が弱いからそういう座り方をするのである。

貧乏ゆすりをしたりペン回しをしたりするのも、この座り方と同じ意味があるので、貧乏ゆすりをしたりペン回しをしたりしていたりしている学生もこの項目をチェックする。

③ 周囲の音や周りの目につく物などで気が散る

このタイプは何の原因がなくても注意集中が転動していくので、ちょっと気になる音やちょっと目立つものがあると、余計にその音や目立つものに注意集中が向いてしまう。つまり、重要な聴覚情報と雑音の区別が付きにくのである。同じく重要な視覚情報と関係ない目立つものとの区別もつかないのだ。だから、学生がつまらない時計の音やエアコンの音が気になっているとか、どうでもいい黒板周囲の目立つ掲示物や窓からの日光に目を向けてしまうなら、注意集中が弱いと言えるのである。

④ 絶えず友達としゃべっている。注意してもやめられない

小学校時代に注意集中が転動して多様な場所に向かって走り回っていた子どもも、大学生の頃にはそれを自分なりにコントロールして落ち着いてくる場合がある。しかしその場合、その代替作用として多弁になっていることが多い。この多弁は脳機能の偏在から来ているので、注意されても自分ではなかなか止められないのである。

⑤ 授業中に指示を出すと、すぐさま聞き返す

注意集中が多方面に転動している学生でも、ある程度授業を受けている意識（授業にも注意集中の一部が向いている）はある。だから、あらぬ方向に注意集中が向いていても、先生が何か指示を出したことはすぐに分かる。だから、課題を喪失する危機感を抱いて、指示の内容をすかさず聞き返すのである。つまり、指示をすぐに聞きかえす学生は、注意集中が弱いと言える。

⑥ 授業中に指名されないのに、発言することがある

大学生になると「授業中に挙手して、指名されない限り発言してはいけない」というルールは理解している。それにもかかわらず発言してしまうのは、脳の機能に注意集中の弱さがあり「この発問に答えたい」という衝動を抑えきれなくなるからである。このタイプは注意すると「すみません」とすぐあやまるが、注意しても「どうしてよい意見を言っているのに叱られるのですか」というケースは、実行機能が弱いタイプである。

⑦ きれいに書けるのに、面倒くさいので適当に殴り書きしまう

授業中に書いたノートは後で試験勉強などに使うことが分かっているし、自分はきれいに書けることも知っている。それでも、早く書きたいとか面倒くさいという思いに負けて、板書をノートに写すときに乱雑に書いたりするのは、注意集中のコントロールに弱さがあるからである。

このタイプは、「丁寧に、書きなさい」と指導すると丁寧に書くことができるが、同じように乱雑な文字を書く《見たことをまとめるのが苦手》なタイプは、指導されてもきれいな文字はかけないという違いがある。

⑧ 計算等の簡単なことで、ケアレスミスをよくする

作業的な行為である計算を間違える、違う教室に間違って行って遅刻する、時間割を間違えて教科書等を持っていない、レポートの提出日を勘違いしたりするなどの簡単なミスを犯すのは、脳の注意集中する機能に弱さがあるからである。

⑨ がんばって課題をしているのに、ふと気が散る

人は、やる気になって何らかの課題に向かうと注意集中のコントロールが上がって、なかなか注意集中は切れなくなるようにできている。やる気になって頑張っていたのにふと気が散ってしまうのは、脳の注意集中する機能に転動してしまう弱さがあるからである。

このタイプは、できる能力もあるし意欲もあるので、先生に指導されるとすぐに課題に戻って頑張ることができるが、他のタイプは、指導されてもそれぞれ脳機能の弱さが原因で課題に戻れないという違いがある。

⑩ 漢字で、首尾一貫しない間違い方をする

同じ漢字を毎回同じように間違えたり、様々な漢字を同じ傾向で間違えたりするのであれば、それは視覚認知等に問題があり覚え間違いをしているのである。しかし、様々な漢字を様々な間違い方をするのなら、それは脳機能の注意集中の弱さあるからである。つまり、これもケアレスミス的一种と言える。

4.《実行機能が弱い》タイプのアセスメント

(1)脳機能のどこに偏在（脳機能の凸凹）があるのか？

このタイプは、実行機能の主な4つの機能である「注意集中・視覚認知と聴覚認知をコントロールする」「感情・思考のコントロールする（抑制）」「思考を柔軟にコントロールする（こだわり）」「何かを計画して手順よく実行していくためにプランニングする」のどれか、または全部に脳の機能偏在があるのです。

よって、このタイプを《実行機能が弱い》タイプと呼んでいる。

(2)観察によるアセスメントの方法

このタイプの注意集中と認知のコントロール、抑制のコントロール、思考の柔軟性のコントロール、プランニングの機能偏在を見つけるために、表4の10項目を観察してチェックする。この10項目のうち5項目以上チェックが付いたら《実行機能が弱い》タイプということになる。

ただし、この《実行機能が弱い》は《見たことをまとめるのが苦手》《聞いたことをまとめるのが苦手》、《注意集中に課題がある》と重なっていることがある。経験的暗黙知であるが、かなりの割合で重なっている。だから、《見たことをまとめるのが苦手》《聞いたことをまとめるのが苦手》、《注意集中に課題がある》の3タイプに該当していても、この項目をチェックすることが必要である。

表4 実行機能が弱いタイプのチェック表

① 授業の始まりのあいさつをしない タメ口で話しかてくる
② 自分は常に正しい 間違いを絶対認めない 謝らない
③ 本人から一方的に話すコミュニケーションをする
④ 行動や「状態をあらわす言葉」で、要求を言う
⑤ 興味ある勉強は全力でやるが、興味ないものは全然やらない
⑥ 自分のやり方や勝ち負けにこだわる
⑦ 授業中に、大事にしているものを取り上げると激怒する
⑧ スケジュール管理して、計画的に行動できない
⑨ 顔の表情が乏しい 困ると能面のようになる
⑩ 他人のルール違反を見つけると、必ず強く指摘（注意）する

各項目について、観察するスキルを身につけるための参考として、実行機能の観点から若干の解説を加える。

① 授業の始まりのあいさつをしない タメ口で話しかてくる

このタイプは注意集中・認知を自分の興味のある方にしか向けていないので、入ってくる情報が限定される。そのため「自分勝手なルール・常識・マナー」（以後、自分ルール）を身につけていたり「自分勝手なコミュニケーション」（以後、自分言葉）をしたりする。

だから「始まりのあいさつは、何故するのかが分からないからやらない」ルールだとか「だれにでも同じ話し方をするルール（先生であろうと友達と同じでよいルール）」を勝手に作ったりするのである。

② 自分は常に正しい間違いを絶対認めない 謝らない

このタイプは、思考の柔軟性のコントロールが弱いので、一度思い込んだことはなかなか変えようとしない。つまり考え方が固いのである。だから「自分の〇〇についての考えは正しい」と一度思い込んだら、なかなか曲げようとしない。

誰らからのどんな説得にも耳を貸さず、間違いを認めようとはしない。逆に、相手を自分の論理でやり込めて正しさを証明しようと強弁さえする。そして、結果として絶対に謝らないのである。

③ 本人から一方的に話すコミュニケーションをする

このタイプは、①でも書いたように、自分ルールを決めていたり自分言葉でコミュニケーションをしたりしている。この場合のコミュニケーションのルールは「自分が知っていることや興味あることを、人に伝えたい。しかし、相手の意見などの反応は望んでいない」である。だから、言いたいことを言い終えたり、相手からの質問があったりするとすぐさま立ち去るのである。

④ 行動や「状態をあらわす言葉」で、要求を言う

①で書いたように、このタイプは、自分言葉でコミュニケーションをしている。この場合のコミュニケーションのルールは「自分の思いを行動や現在の状態で伝えるので、私の思いを察知して支援する方法はそちらで考えてくれ」という傲慢なものだ。

例えば、腹が立つと机を蹴る、教科書を貸して欲しいときは「教科書を忘れました」早退したいときは「頭が痛い」という自分言葉のコミュニケーションをするのである。だから、結果として人に「～して欲しい」と決して頼まないのである。

⑤ 興味ある勉強は全力でやるが、興味ないものは全然やらない

思考の柔軟性が弱く抑制のコントロールができないと、好きなことは好きなだけやり、嫌いなことは全くやらないという行動パターンが続き、最終的に「0か100かの考え方」になる。

これが勉強場面では「興味関がある学問は、注意集中と視覚認知と聴覚認知を最大限に使って学ぼうとする」が、「興味関心が無いものは、そもそも注意集中や視覚認知、聴覚認知を向けようとししない」という行動になる。その科目は卒業必修単位だとか、それを習得しないと進学できないとかということ、別の問題として考えてしまうのだ。

⑥ 自分のやり方や勝ち負けにこだわる

思考の柔軟性が弱いので、一度やって成功するとそのやり方にこだわり、勝った喜びを一度味わうと勝ちにこだわるようになる。これも「0か100かの考え方」からくるものだ。だから、先生からよりよい方法を教授しようとしても「大丈夫です」「できますから」などその教えを拒否するし、ゲームや議論、その他、どんなくだらない勝負事にも勝ち負けにこだわるのである。

⑦ 授業中に、大事にしているものを取り上げると激怒する

思考の柔軟性が弱く抑制のコントロールが悪いと、何かものにこだわることもある。それに、独自のルールがついていることもある。「これを持っていると、大勢の中にいる緊張が緩み勉強が進む。しかし、これがないと何もできなくなってしまう」などだ。これも、ある意味、「0か100かの考え方」と言える。

大学生になってもそれが残っていて、授業中に大事なものを机の上に置いていたり、常に触っていたり、カバンに入れていて時々眺めていたりする。例えば、気に入りの飲み物・好きな本・小さい人形・小さいタオル・お気に入りのペンなどだ。

それを片付けさせようとしたり、取り上げたりしようすると激怒する。最近では、それが、スマホになっている学生が多くいる。

⑧ スケジュール管理して計画的に行動できない

実行機能に弱さがあると、プランニング（段取り）がうまくできないためにスケジュール管理が苦手になる。それにも関わらず、頭の中だけで考えてスケジュール管理しようとしたり、自分の好きなことを優先させたりしてスケジュール管理に失敗するのである。つ

まり、友達との約束を破ったり、レポートの期限が守れなかったり、明日のテストの準備をせず10日先のテスト勉強をしたりするのである

その他、履修届がうまく作れないのもプランニングが弱いからである。

⑨ 顔の表情が乏しい 困ると能面のようになる

表情は、後天的に話し相手の表情を見て覚えていくものである。しかしこのタイプは、注意集中・認知を自分の興味のある方にしか向けないので、入ってくる情報が限定される。つまり、普段相手の顔の表情を見ながらコミュニケーションをしないので、表情を学んでいく機会が少なく表情が乏しくなってしまうのである。

だから、困ったときにどんな表情をしたらいいのか分からず、表情のない能面のよう顔をしたりするのである。

⑩ 他人のルール違反を見つけると、必ず強く指摘（注意）する。

①でも書いたように、自分ルールを持ちやすい上に思考の柔軟性や抑制のコントロールが弱いと「正しいルールは、絶対に守らないといけない」とか「自分はルールを守っているのに、他人が守らないのはずい」などのルールを勝手に作ってしまう。

そして、先生や知らない学生やその他誰であろうと、ルールを守っていない人を見つけると、絶対に守らせようと指摘（注意）するのである。

5. 《熟考》タイプのアセスメント

(1)脳機能のどこに偏在（脳機能の凸凹）があるのか？

このタイプは、情報が脳の中を伝達していくスピードが遅いのが特性である。つまり、情報伝達スピードのコントロールに機能の偏在があるのである。

視覚的短期記憶と聴覚短期記憶を使って情報を取るのに時間がかかり、視空間的ワーキングメモリーや言語的ワーキングメモリーを使って情報を理解しまとめ上げるのに時間がかかる。そのため、コミュニケーションに時間がかかったり、運動がうまくできなかったり、不器用になったりする。

時間はかかるが、考えることは得意なので、このタイプを《熟考》タイプと呼んでいる。

(2)観察によるアセスメントの方法

このタイプの情報の伝達の遅さを見つけるために、表5の10項目を観察してチェックする。この10項目のうち5項目以上チェックが付いたら《熟考》タイプということになる。

このタイプは知的レベルが平均かやや高いが、現在の公教育では気づかれないことが多い。さらに、学習の定着が遅いために学習に遅れが出てくるので、境界線児（ボーダー）として扱われてしまうことも多い。だから、丁寧に支援されないと、大学まで進学して行くのは難しい。つまり、大学にはこのタイプは少ないと言える。

表5 熟考タイプのチェック表

① 普段の会話が、ゆっくりで声が小さい
② 簡単な質問でも、答えるまでに時間がかかる
③ 早く話すと聞き取れない、聞き間違いをする
④ 板書を写すのに時間がかかる ゆっくりしか書けない
⑤ 理解はするが、定着までに時間（回数）がかかる
⑥ 単純な反復作業を嫌がる 他者より時間がかかる
⑦ 一見、幼いように見えるが、話合ってみると賢く感じる
⑧ 行動するスピードが遅いため、全体行動が遅れる
⑨ 運動が、あまり得意ではない 走り方がぎこちない
⑩ なんとなく、いじめられやすい雰囲気がある

各項目について、観察するスキルを身につけるための参考として、情報の伝達が遅い観点から若干の解説を加える。

① 普段の会話が、ゆっくりで声が小さい。

このタイプは、情報がゆっくり入ってきてじっくり考えながら話すので、コミュニケーションが遅いのが特性である。そのせいで、対人面で様々な失敗を重ねて自信をなくし、声が小さくなっていることが多い。

② 簡単な質問でも、答えるまでに時間がかかる

このタイプは、どんな簡単な情報でもワーキングメモリでじっくり検討するので情報処理に時間がかかる。だから、例えば「果物で、何が1番好きですか？」というような簡単な質問でも、答えるのに時間がかかるのである。

③ 早く話すと聞き取れない、聞き間違いをする

聴覚短期記憶機能の情報処理が遅いので、早い大量の聴覚情報は処理しきれなくなる。だから、早口で話された聴覚情報は、聞き取れなかったり聞き間違えたりしてしまうのである。

④ 板書を写すのに時間がかかる ゆっくりしか書けない

視覚的短期記憶の情報処理が遅いので、《見たことをまとめるのが苦手》なタイプと同じように、板書の文章を覚えて書くノートテイクに時間がかかってしまう。《見たことをまとめるのが苦手》なタイプとの違いは、手指の動きが遅いのでゆっくり書くことと、丁寧な字を書くことである。

注意することは、単に丁寧に書いている学生と見分ける必要がある。単に丁寧に書いている学生は、必要があれば早く書くこともできるという違いがある。

⑤ 理解はするが、定着までに時間（回数）がかかる

ワーキングメモリーでまとめ上げられた情報は、経験と結びついて知識となり長期記憶に蓄えられる。その結びつくまでの時間とできあがった知識を長期記憶に伝達するスピードの遅さが原因で、知識が定着するためには時間が必要となる。だから、同じ練習問題をしても、一般的な学生より回数が必要になる。

⑥ 単純な反復作業を嫌がる 作業が他者より時間がかかる

単純な作業は、素早く情報を取り入れワーキングメモリーで熟考することなく、口で素早く唱えたりや手指を素早く動かしたりするものが多い。苦手な情報の処理の素早さを使い、得意な情報の熟考を必要としないので、このタイプは単純作業を嫌がる傾向にある。もともと遅い上に意欲もなくなるので、さらに作業は遅くなるのである。

⑦ 一見、幼いように見えるが、話し合ってみると賢く感じる。

①で書いたようにゆっくり小さな声で話すので、表面上は幼く見えたり、境界線児に見えたりすることがある。しかし、一旦時間をかけて会話してみると、難しい言葉を使ったり思慮深い話をしたりして賢さが伝わってくることもある。

⑧ 行動するスピードが遅いため、全体行動が遅れる

情報の処理の遅さは、行動のスピードに響く。つまり、行動が他者より遅くなるのである。これは、集団行動するとき一人だけ遅れる結果になる。特に、集団で歩く場面で観察しやすい。

⑨ 運動が、あまり得意ではない

運動では、視覚情報等を素早く処理して素早く身体・手指を動かす必要がある。だから、情報処理の遅いこのタイプは、運動全般が不得意になる。更に、本人も意識的に運動を避けるので、身体能力も伸びず更に運動が苦手になるという悪循環を引き起こす。

⑩ なんとなく、いじめられやすい雰囲気がある

①②③⑧⑨の特徴のために、集団から異質に見られることが多く、集団の質が悪化するといじめのターゲットになりやすい雰囲気を持っている。

6. 《作業》タイプのアセスメント

(1) 脳機能のどこに偏在（脳機能の凸凹）があるのか？

このタイプは、注意集中、ワーキングメモリー、実行機能、長期記憶の機能が平均を下回る、つまりそれら全部に機能偏在がある。しかし、視覚的短期記憶、聴覚的短期記憶、コミュニケーション、身体・手指活動には機能偏在が少くないので、日常生活には問題がなく簡単な作業をこなすこともできる。

高等な脳機能を使う学習には向かないが、簡単な作業は得意なのでこのタイプを《作業》タイプと呼んでいる。

(2) 観察によるアセスメントの方法

それを見つけるためには、《見たことをまとめるのが苦手》なタイプの表1と《聞いた

ことをまとめるのが苦手》なタイプの表2の10項目を観察してチェックする。その両方に5項目以上チェックが付いたら「作業タイプ」ということになる。

作業タイプは、学習に向かないために大学まで来る人数はかなり限られる。もし、進学してくるとしたら、入試を経ない推薦入学などその他の入学方法が考えられる。

Ⅲ 大学の教員が授業中にできるグレーゾーンのへの支援の方法

大学の教員が授業中にできるグレーゾーンのへの支援の方法を、筆者による「発達障害の実践的分類」別に解説していく。支援には、予防的に実施しておく取り組みと発見してから開始する支援の2種類がある。

1. 《見たことをまとめるのが苦手》なタイプの支援

(1)機能的にどんな困り感があるのか？

支援を考えるための資料として、脳の機能が持つ本質的な困り感をあげておくと、表6のようになる。

表6 《見たことをまとめるのが苦手》なタイプの機能的な困り感

・ 視覚情報が平面的であったり、表面的であったり、部分的であったりする
・ ものともものを見比べるのが苦手なため、量把握が苦手になる
・ 物の形を正確に把握しにくい
・ 複雑に交差した線の構成など入り組んだものが、分かりにくい
・ 取り込んだたくさんの情報を、図式化してまとめげるのが苦手

(2)実際にどんな困り感があるか？

では、実際の場面では、どんなことに困るかということをあげておくと、表7のようになる。

表7 《見たことをまとめるのが苦手》なタイプの実際の困り感

・ 漢字を覚えるのが苦手なため、レポートに平仮名が多くなる
・ 書字が苦手なため、たくさん文字を書くと大変疲れる
・ 調べ学習の報告であるレポートが、なかなかまとまらない
・ 数概念の未発達のために、数学全般が苦手になりやすい
・ 顔の表情がうまく読めないために、人の気持ちが分かりにくい そのため、トラブルになりやすい
・ 大人数の教室、グループワーク、模擬授業などで、周りの状況 がうまくつかめない（空気が読めない）

(3)授業中にできる支援

支援の基本は、視覚認知を支援し、聴覚認知を活用することである。①と②、③は予防的にやっておくとよい取り組みで、④から⑦は、アセスメントして発見次第実施するとよい支援である。

- ① 授業は、必ず資料を使って行う。パワーポイント使ったときは、それも印刷して渡すかスマホやパソコン等でダウンロードできるようにする。できれば事前に渡して、予習してくることを勧める。
- ② 授業を聞いて、大事なことや興味があり面白いと感じた内容は、授業の資料に関連付けてメモすることを勧める。
- ③ 図やグラフのなどの視覚情報は、「見て分かるように」と言わず「右肩がありで、増加しています」など、必ず分析結果を言語化する。
- ④ どうしても、板書を大量に写すような授業の場合、スマホで写真を撮ることや、タブレット・パソコンで打ち込みすることを認める。
- ⑤ 授業内容が分からないという申し出があった場合は、参考文献か参考になるインターネットサイトを紹介し声を出して読むように勧める。
- ⑥ レポートの提出率が悪いときは「口述筆記（録音したものを清書する）で、書いてみたら」とアドバイスする。または、レポートのまとめ方の見本として、レポートのテンプレートを配布する。
- ⑦ 話し合いやグループ活動などで、対人関係のトラブルが発生したときには「分からないことは、推測しないで相手の気持ちを含めてなんでも相手に聞いてしまいなさい」とアドバイスする。

2.《聞いたことをまとめるのが苦手》なタイプの支援

(1)機能的にどんな困り感があるのか？

支援を考えるための資料として、脳の機能が持つ本質的な困り感をあげておくと、表8のようになる。

表8 《聞いたことをまとめるのが苦手》なタイプの機能的な困り感

・複雑な話や長い話が聞き取れない 覚えていることができない
・聴覚情報をうまく聞き取れても、理解できないことがある
・聞かれた内容をすぐ理解し、素早く対応を考えるのが苦手
・知っている語彙が減ってしまう 語彙の意味を間違えて覚えてしまう

(2)実際にどんな困り感があるか？

では、実際の場面では、どんなことに困るかということをあげておくと、表9のようになる。

表9 《聞いたことをまとめるのが苦手》なタイプの実際の困り感

・一斉授業形式（講義中心）の勉強が苦手
・言葉の意味理解が苦手なために、文章を読むのが苦手になり本をあまり読まない
・臨機応変の会話が苦手ため、コミュニケーションを避ける傾向にある
・コミュニケーションが苦手で、友達ができにくい
・コミュニケーションに困ると、暴言を吐いたり暴力をふるったりする場合と、逆に、黙って固まったり、泣いたりして逃げようとする場合がある
・数学の計算はできるようになるが、文章題ができない

(3)授業中にできる支援

支援の基本は、聴覚認知を支援し、視覚認知を活用することである。①から④は予防的にやっておくとよい取り組みで、⑤から⑨は、アセスメントして発見次第実施するとよい支援である。

- ① 授業は、おおよその話の流れやポイントを板書・資料・パワーポイントを使って図で示すことから始める。
- ② 指名して何か発表させる場合は、発表内容を書く時間を取りそれを見ながら発表してもよい方式にする。
- ③ 授業内容は、箇条書きでいいのでメモすること勧める。更に、帰宅後、そのメモと資

料を使って授業内容をまとめたノートを作ることを勧める。

- ④ 授業内の指示内容は、必ず板書する。提出期限や試験の範囲など重要な内容は、大学のネット情報で常に確認できるようにする。
- ⑤ 授業内容がよく分からないと申し出があった場合は、分かりやすい参考文献やネットサイトを紹介して、読むようアドバイスする。
- ⑥ 授業内容が聞き取りにくい場合は、授業をスマホ等で録音するのを認め、家に帰ってから、聞きそびれた箇所や分からない所を繰り返し聞くことを勧める。
- ⑦ レポートの提出率が悪いときは、語彙を増やす活動や、好きな先生や作家の文章を視写することを勧める。
- ⑧ 話し合いやグループ活動などで、相手の話が聞き取れていないと判断した場合は「うまく聞き取れなかったら『もう1回言ってくれませんか』と頼みなさい」とアドバイスする。
- ⑨ 話し合いやグループ活動などで、相手の質問にうまく答えられていないと判断したら「うまく答えられなかったら『少し、考えさせてください』と頼みなさい」とアドバイスする。

3. 《注意集中に課題がある》タイプの支援

(1)機能的にどんな困り感があるのか？

支援を考えるための資料として、脳の機能が持つ本質的な困り感をあげておくと、表10のようになる。

表10 《注意集中に課題がある》タイプの機能的な困り感

・ 注意集中が、勝手にあっちこちに転動してしまう
・ そのため、ケアレスミスを起こしてしまう
・ そのため、大事な聴覚情報を取っていない状態になってしまう
・ そのため、大事な視覚情報を取っていない状態になってしまう
・ やりたいと思ったことは、衝動的に全部やってしまう

(2)実際にどんな困り感があるか？

では、実際の場面では、どんなことに困るかということをあげておくと、表11のようになる。

表11 《注意集中に課題がある》タイプの実際の困り感

・ 今やるべき課題が、分からなくなる
・ 授業中の会話、スマホの扱い、試験での注意など、授業での大事なルールを知らないことがある そのため、よく叱責される
・ 履修届け、出席回数、課題提出日、試験日、試験、計算などのケアレスミスが多い
・ 忘れ物が多い 何を持ってくるのかを忘れるし、準備していても、持ってくるのを忘れる
・ 退屈すると、授業中でも自分のやりたいことをやってしまう

(3)授業中にできる支援

支援の基本は、教員側から情報を出すときに、相手の注意を喚起して情報を確実に伝えることである。

①から④は予防的にやっておくとよい取り組みで、⑤から⑨は、アセスメントして発見次第実施するとよい支援である。

- ① 「今、何をすべきか」という課題を常に明確にする。そのためには、必ず事前に注意喚起して伝えたい情報を確実に聞かせたり見せたりする。
- ② 教室内や授業中のルールや暗黙の了解は、必ず言語化して正確に伝える。例えば、「授業中のおしゃべり」「授業中のスマホの扱い」「試験時のルール」など。できれば、復唱させて、覚えたことを確認する。書面にしたものを渡すと、さらによい。
- ③ 忘れ物は大目に見て、頼ませてから貸し出したり、再配布したりする。
- ④ 試験日や試験範囲、出席回数の確認、レポートの提出日など重要事項は、大学のネット情報で何度でも確認できるようにする。
- ⑤ 課題が分からないようだと判断したなら、近づいて頼ませてから一度だけ課題内容を繰り返す。
- ⑥ 忘れるか我慢できずにルール違反したときは、絶対に叱らず再度ルールの説明か確認をしてから、復唱させる。
- ⑦ 試験のときなど重要な場面で、落ち着きがなくケアレスミスが起こりそうだと判断したときは、近づいて個人的に注意喚起する。その後、必ずお礼を言わせる。
- ⑧ レポートを急いで出そうとして短い粗雑な内容のものを提出したら、提出期限を確認させてまだ余裕があることを知らせ、文字数やコペジ禁止などルールを守って落ち着いて書くようにアドバイスする。
- ⑨ 小グループで行動するとき、集合の日を間違ったり、作業分担などを忘れたりするなど、約束違反しないようにメモやスケジュール帳の活用をアドバイスする。

4. 《実行機能が弱い》タイプの支援

(1)機能的にどんな困り感があるのか？

支援を考えるための資料として、脳の機能が持つ本質的な困り感をあげておくと、表12のようになる。

表12 《実行機能が弱い》タイプの機能的な困り感

・興味あるところだけに、注意集中がいく
・いま何を見て、いま何を聞くべきなのかが分かりにくい
・思いや欲求をコントロールするのが苦手なため、自分の思いを全部実現したいと考えてしまう
・行動や考え方の切り替えが苦手なため、それまでの行動や考え方にこだわる　なかなか切り替えられない
・何か一つのことを確実にやり遂げたり、たくさんのことを一度にやり遂げたりするための段取り（計画）がうまくできない

(2)実際にどんな困り感があるか？

では、実際の場面では、どんなことに困るかということをあげておくと、表13のようになる。

表13 《実行機能が弱い》タイプの実際の困り感

・自分の思いを抑制しないので、学びたいことだけを学び、学びたくないことを避ける
・上記のために、語彙や知識が偏ってしまう
・上記のために「ルール、常識、マナー」などが、他の人と少し違って集団生活がしにくい（自分ルール）
・上記のために、コミュニケーションが他の人と少しずれて、うまく通じないことが出てくる（自分言葉）
・0か100かの考え方の極端な考え方になる
・自分のことは、どんなことでも全部自分で決めようとする（自己中心的　自己決定に対するこだわり）
・困ったときでも、情報を収集したり人に頼ったりしないで、自分だけで解決しようとして、トラブルになったり問題行動になったりする（自分のことは、自分でやるというこだわり）
・あることを実現するための計画・予定を立てるのが苦手

(3)授業中にできる支援

命令や指示など力ずくや圧力では絶対に行動を変えないので、支援の基本は、下記の1の関わり方、つまりこのタイプに合ったコミュニケーションの取り方やコミュニケーションできる良好な関係（愛着）の構築にある。

①と②は予防的にやっておくといふ取り組みで、③から⑤は、アセスメントして発見次第実施するといふ支援である。

- ① 授業中に、当該学生が問題行動と取れるようなルール・常識・マナー違反の行動をしたときに、次の表14の4つの方法でコミュニケーションをして、正しいルール・常識・マナーを説明する。説明したあとは、納得してもらって、自己決定で正しい行動に切り替えてもらうよう要請する。もちろん、普段のコミュニケーションから、この4つの方法を使っておくと更によい。

表14 《実行機能が弱い》タイプとの4つのコミュニケーション方法

i)「想像」 相手の行動・言葉から、要求や考えていることを想像し相手の要求を言語化する 「君は、～して欲しいのではないかな?」「君は、～と考えているのではないかな?」
ii)「共感」 言語化された相手の要求やその元になった感情に共感して、それを言語化する 「君は、～して欲しいんだね」「君は、～と考えているんだね。」
iii)「覚えて」 大学のルール・常識・マナーを説明して覚えてもらい、覚えたかどうか確認するために復唱してもらう
iv)「すみません」 相手の要求が大学のルール・常識・マナーを超えるときには「すみません」をつけて頼んでももらう

- ② 上記の4つのコミュニケーションで、次の表15の6つを前もって「これは、うちの大学のルールだ」と説明しておく、問題行動を減らす効果がある。

表15 《実行機能が弱い》タイプに説明する6つのこと

・授業中は、先生や発表者に聴覚認知を向けておく
・授業中は、先生や黒板、スクリーンに視覚認知を向けておく
・大学や授業中には、どんなルール・常識・マナーがあるか説明して知らせておく
・興味ない授業でも、卒業に必要なので勉強することと、その授業のレポートは乗り気しなくても必ず出す
・自分の意見を聞いてもらったのだから、相手の意見も聞かなくてはいけない
・集団活動で問題が生じたときは、自分だけで解決するのではなく、情報をたくさん集めて他の解決方法を探究したり、トラブルに対応できる人や先生に相談したりする

- ③ 日常的に遅刻の多い学生には、日常生活のルーティーンの見直しを提案する。できれば、当該学生と相談しながら、朝起きてから学校に来るまでのルーティーンの組み立てを支援するとなおよい。
- ④ スケジュール管理できなくて、レポートの提出が悪い、演習の準備ができない、実習に遅刻するなどの学生には、スケジュール帳を紹介し、「to do list」の優先順位のつけかたや絶対忘れてはいけないことの記しのつけ方など、一週間のスケジュールのつけ方を説明し採用するように提案する。
- 手帳を見るのを忘れるタイプには、スマホのスケジュール機能やアラーム機能を活用するよう勧める。
- ⑤ 履修届けがうまく作れなくて失敗する学生には、当該学生に一度作ってもらい、履修できない科目がないか、時間が被っていないかなど問題点を点検する支援をする。その後、教務課に行かせて、最終確認をするように勧めるとなおよい。

5. 《熟考》タイプの支援

(1)機能的にどんな困り感があるのか？

支援を考えるための資料として、脳の機能が持つ本質的な困り感をあげておくと、表16のようになる

表16 《熟考》タイプの実際の困り感

・ 情報を取り込むのに時間がかかる
・ 情報をまとめるのに時間がかかる上に、深く考えてしまう
・ 情報の定着に、試行錯誤の回数や時間がかかる
・ コミュニケーションのスピードを上げるのが難しい
・ 素早く体を動かせない
・ 人とリズム感やテンポが違う

(2)実際にどんな困り感があるか？

では、実際の場面では、どんなことに困るかということをあげておくと、表17のようになる。

表17 《熟考》タイプの実際の困り感

・ 学習の定着に時間がかかるため、人より長い学習時間が必要となる
・ 先生や友達と臨機応変の会話が、うまくできない
・ 行事など集団行動で、みんなと合わせるのが難しい
・ 運動全般が苦手 特に、跳び箱、鉄棒、マット、縄跳び
・ 作業に時間がかかるので、単純作業を嫌がる
・ 全体的に、いじめの標的になりやすい雰囲気が出る

(3)授業中にできる支援

支援の基本は、知識の理解と定着のためやアウトプットのために時間を確保することである。

①から⑤は予防的にやっておくとい取り組みで⑥と⑦は、アセスメントして発見次第実施するとよい支援である。

- ① 自分で予習できる環境を整えてあげる必要がある。そのためには、事前に資料の配布をする。
- ② ノートテイクに時間が取らなくても済むよう、資料に板書内容も全部書き込む。それが難しいようなら、板書の写真を撮ることを許可する。
- ③ 自分で、家に帰ってから復習するように勧める。そのために、授業の録音を許可する。
- ④ 可能であれば、数回補講を実施する。無理な時は学習支援室を勧める。
- ⑤ 行動やコミュニケーションが遅いことで、からかわれたりすることないように配慮する。例えば、発言を指名しときには話し始めるまでの時間を十分に取る、グループを

作るときメンバーを配慮する、からかいが見られたら厳しく叱責するなど。

- ⑥ 当該学生とのコミュニケーションは、ゆっくり話すように心掛ける。
- ⑦ レポートや試験の答えを書くのに時間がかかるという申し出があれば、期限や時間を伸ばすなどの配慮をする。

6. 《作業》タイプの支援

基本的には、大学に進学してくるタイプではないので、この実践報告では簡単に述べる。

(1)機能的にどんな困り感があるのか？

脳の機能が困ることは、すべての機能が平均を下回ることから起こる全てのことである、

(2)実際にどんな困り感があるか？

実際の場面では、大学の学修の全てに渡って困る。

(3)授業中にできる支援

支援の基本は、最低限の基本的なことだけを学んで、単位を取れるように支援することである。

- ① 科目内容を単位が取れる最低範囲に限定する。可能であれば、やや課題をやや簡易にしたものに差し替える。
- ② 単位修得にあたっては、ある程度の配慮が必要となる。
- ③ 学習支援室など、学習の支援してもらえるところに常に通うように勧める。

Ⅳ まとめ ～今後の課題について～

今後の課題としては、「大学教員による個別のアセスメントと支援」で個別にグレーゾーンを支援するだけでなく「大学によるアセスメントと支援」へのバージョンアップである。つまり、「大学によるグレーゾーンの積極的発見方法」と「グレーゾーンの大学生を支援へ誘導する方法」を確立することである。

具体的には、新1年生全員のスクリーニングを行って、1年生全員の発達の状況を掴んでおき、それに現在大学が行っている4月の基礎学力テストと1年生前期の単位取得状況を合わせてグレーゾーンを確定し、大学側から積極的に当該学生に働きかけて、1年生後期から支援を開始することを意味する。つまり、「観察だけでできる『発達のアセスメントツール』を使って、どのようにして1年生のスクリーニングをするか」の研究が、今後の課題である。

今回は、個々の教員が、個人情報のコンプライアンスを考えずにグレーゾーンのアセスメントが可能になるように、観察だけでできるアセスメントの試案とアセスメント後のパターン別支援について述べた。各教員がこのアセスメントに習熟し、グレーゾーンの学生

の発見・支援を行い、グレーゾーンの学生のためによりよい大学環境が整う一助になれば幸いである。

参考文献

- 日本学生支援機構 「2021年度大学、短期大学及び高等専門学校における障害のある学生の修学支援に関する実態調査分析報告」 2022年 8 月26日 https://www.jasso.go.jp/statistics/gakusei_shogai_syugaku/index.html
- 内閣府 「障害を理由とする差別の解消の推進に関する法律」 2013年 6 月発布 2016年 4 月 1 日から施行 https://www8.cao.go.jp/shougai/suishin/law_h25-65.html
- 博報堂 Pechat 開発チーム・博報堂こそだて家族研究所・LITALICO 発達ナビ「ASD と子育て実態調査」第一弾「ASD への理解ギャップ」編 2020年 1 月 <https://www.hakuhodo.co.jp/news/newsrelease/79655/>
- 文部科学省 「特別支援教育の在り方に関する特別委員会（第12回及び第13回）における教職員の確保及び専門性の向上に関する主な意見の総論」 2011年 9 月15日 https://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/044/attach/1312978.htm
- 佐藤 剛 監修 永井洋一・浜田昌義 編集 1998年初版 『子どもの理解と援助のために 感覚統合 Q & A』 協同医書出版社
- 高山佳子 1998年 『LD 児の認知発達と教育 つまづきの理解から指導。援助の手だて』 川島書店
- 苦野一徳 2022年 『学問としての教育学』 日本評論社
- 鳥居深雪 2009年初版 『脳からわかる発達障害』 中央法規出版
- 森口佑介 2012年 『わたしを律するわたし 子どもの抑制機能の発達』 京都大学学術出版会
- 山田 章 2017年 『教室で使える発達の知識』 クリエイトかもがわ
- 湯澤正通・湯澤美紀 編著 2014年 『ワーキングメモリと教育』 北大路書房
- S.E. ギャザコール・T.P. アロウェイ 2009年 『ワーキングメモリと学習指導』 北大路書房

(2023年 2 月17日受理)

2022年度 人文自然学会研究会
2023年2月25日(土) 午後1時～

2022年度の人文自然学会研究会が、2023年2月25日(土)午後1時より、廣藤千代子副会長を司会者として Zoom により開催された。発表者と題目および発表要旨は以下のとおりである。

中山 麻耶

主婦イメージの変容

本報告は女性の社会進出が期待されるようになったこの20年において、「主婦」がどのように記述されているか、その変容を雑誌の分析から明らかにするものである。日本における主婦に関する研究は、1980年代からの主婦の社会進出の是非を問う議論が主であり、働く女性が増加して以降の主婦についての研究があまりすすんでいない。

本報告では主婦雑誌を資料として、男女共同参画社会基本法が制定され主婦の社会進出がすすんだ90年代後半から近年にかけて、読者である主婦たちに提示される「主婦像」がどのように変化したかを明らかにする。ここでは主婦を対象とした雑誌の中でも専業主婦向けとしての色が強い主婦向けファッション雑誌『VERY』(1995年創刊、光文社)を用いて、創刊から20年分の記事のタイトルをデータ化し、タイトルと内容に基づいた分類を行った。創刊当初は高階層の専業主婦を対象とした雑誌であったが、専業主婦の減少及び働く主婦の増加に伴い、現在では結婚や出産を機に離職した主婦の再就職を扱った特集も組むなど社会の変化を反映している。

分析の結果、2000年代後半から主婦イメージが大きく変化してきていることが分かった。誌面では役割ごとに服装が異なる記述が多くあり「母」「妻」「嫁」としてのおしゃれが描かれ、中でも「母」としての服装が他役割よりも重視され、かつ増加傾向にある。創刊当初は「主婦だけ」という言葉に表されるように、主婦であることが否定的なニュアンスで描かれる一方、近年では「主婦だからおしゃれ」のように主婦であることを肯定的に描く言説があらわれる。

主婦と労働についての記事も近年に表れるが、そこでは仕事とは自己実現や経済的利点のためのものではなく、「家庭で輝くため」のものであり、家族が一番で仕事が二番であるという家庭重視の働く主婦像が強調されている。

母役割の強調や主婦イメージのポジティブな変化、「家族のための仕事」という表象の変化から、主婦向けファッション雑誌において「家族」やその絆が押し出されている。家庭重視の専業主婦向け雑誌が、従来のな家族観の枠組みにおいて解放された女性の欲望を語るには、逆説的に家族主義がより強化されたといえる。

2022年度 人文自然学会総会
2023年2月25日(土) 午前9時～

2022年度の人文自然学会総会が、2023年2月25日(土)午前9時より、秋田亨会長を議長として OGU-Caddie により開催され、以下の案件について審議・報告があった。

- (1) 新役員の承認（審議）
 - 会 長 淡 誠一郎（情報学部教授）
 - 副 会 長 佐野 学（情報学部教授）
 - 庶務委員 中山 麻耶（経営実務科講師）
 - 編集委員 横山 輝樹（法学部講師）
- (2) 2022年度事業報告と2023年度事業計画の説明（報告）

神戸 尚志・山口 雅之

C 言語によるシステム LSI のアーキテクチャ最適化手法と 設計事例による評価について

システム LSI は携帯電話、自動車、AV 機器などの各種電子機器に広く利用され、年々複雑化・高度化が進んでいる。システム LSI の設計法として、C 言語による回路設計手法が注目されている。本文では、C 言語設計における、データアクセス法、並列化・パイプライン化、アプリケーションに特化した演算回路設計、モジュール構成の変更等のアーキテクチャ最適化の適用について、粒子追跡技術における相関値計算のハードウェア化を例に説明し、アーキテクチャ最適化手法の適用結果について述べる。

その後、幾つかの設計事例を用いて、回路の高速化に対する面積増加の比率（Performance/Area Ratio, 以下、PAR）を用いて、アーキテクチャ最適化手法の効果について適用例ごと及び手法ごとに評価する。これらの設計事例では、適用例ごと及び手法ごとに最適化の効果を高速化/面積増加比（PAR）指標にもとづいて測定し、効果確認を行なう。すべての適用例において、最終的な設計の目標仕様を実現することができ、提案手法の有効性を確認した。手法ごとの評価では、データアクセス法が最も大きな効果を上げる可能性を持ち、パイプライン化は安定的に、少ない面積増で高速化を実現する効果があると考えられる。並列化手法は、適用箇所によってバラツキがあるが、一定の面積増を前提として、高速化を実現するには直接的で効果的な手法であることを確認した。

淡 誠一郎・淡 裕美子・吉田 康子

ベジエ曲線によるダイコンの形状記述とその応用

果物や野菜の形状は人の嗜好や収穫・加工の効率などに大きく影響する。形状は土壌や気候、栽培過程などの環境要因にも影響されるが、遺伝的要因の影響が大きい。本研究で扱うダイコンの形状にも様々な遺伝子が関わることが予想され、ダイコンの効率的な育種を行う上で根形に関わる遺伝的メカニズムの解明が期待される。量的形質である根形に関わる遺伝子の同定には、本来ならば定量的な形状解析が必要であるが、従来研究のほとんどは質的な評価しかしておらず、定量的な分析手法としては楕円フーリエ記述子を用いた手法の提案ぐらいしか見いだせない。

フーリエ記述子はモノのシルエット全体の形状を任意の忠実度で記述・再現できる一般性のある手法である。ダイコンであれば根形全体の特徴をおおまかにとらえ、個体や品種の特徴を分析したり、分類したりする上では強力なツールとなりうる。しかし、形状全体を捉える特徴量であるため、部分的な形状を取り出してその特徴を記述したり、それを分析したりするのには向かない。また、曲がりを持った個体の長さや幅の計測、軸に沿った幅の変化の様子を捉えるといった目的には不向きである。そのような目的には、任意の曲線を記述でき、曲線上の位置を1つのパラメータで連続的に表せるベジエ曲線などのパラメトリック曲線の方が適する。

筆者らはこれまでにダイコンの多様性評価のための形状解析や計測にベジエ曲線による形状記述を利用しており、農学的な視点での成果についてはすでに報告済みであるが、形状記述とそれに基づく計測方法のアルゴリズムの詳細については未報告であった。本稿はそれらについて解説すると共に、ダイコンの全周および左右分割輪郭の近似に必要な標本数やベジエ近似の次数などについて、経験的に得られた知見について報告するものである。

山田 章

発達障害グレーゾーンの大学生を観察でアセスメントする方法の一試案 ～大学教員ができる発達アセスメント～

大学に発達障害を抱えながら入学してくる学生の数が、年々増えてきている。それ以外にも、発達障害グレーゾーン（以下グレーゾーン）と呼ばれる、発達障害的な特性を持った学生も増えていることも報告されている。

文部科学省は、それらの学生に『障害を理由とする差別の解消の推進に関する法律』（以下障害者差別解消法）などを基に合理的配慮（支援）することを求めている。2022年度の障害者差別解消法の改正で、私学でもその合理的配慮が義務化された。

発達障害の診断のある学生や、これまで支援を受けてきた経験のあるグレーゾーンの学生は、自ら合理的配慮を申し出てくるが、自分がグレーゾーンだと気づかず支援された経験のない学生は、自らは申し出てくることはない。それに関わらず、大学生活で履修届が作れない、レポートが書けない、ノートが早くとれない、先生の講義がうまく聞きとれない、つい遅刻してしまう、単位が取れないなどと悩んでいる現状がある。

改正された障害者差別解消法はこのようなグレーゾーンの学生も見つけ出し、積極的に支援することを求めている。一方、個人の権利や個人情報の保護の観点から、このグレーゾーンの学生を見つけ出すことは、大変難しい上にコンプライアンスを犯す危険を多大に含んでいる。

そこで、多数のWISC検査を実施したアブダクションを駆使して、グレーゾーンだと気づいていない学生のアセスメントを教員が観察だけでできるように、アセスメントツールを試案した。このツールは、観察だけで行うアセスメントのためコンプライアンスに触れる事なくグレーゾーンの発見・支援が可能なのである。

本実践報告は、『見たことをまとめるのが苦手』『聞いたことをまとめるのが苦手』『注意集中に課題がある』『実行機能が弱い』『熟考タイプ』『作業タイプ』の6つのタイプのグレーゾーンが発見できる観察ポイントとその解説、そして授業中に実施すべき基本的な合理的配慮で構成されている。

大阪学院大学人文自然学会会則

- 第1条 本会は「大阪学院大学人文自然学会」と称する。
- 第2条 本会の事務所は大阪学院大学図書館内におく。
- 第3条 本会は本学の設立の趣旨にもとづいて人文および自然科学の発展に寄与し、各分野にわたる会員相互の学術研究の交流を目的とする。
- 第4条 本会は次の事業を行う。
1. 機関誌『大阪学院大学人文自然論叢』の発行
 2. 研究会、講演会および討論会の開催
 3. その他、本会の目的を達成するために必要な事業
- 第5条 本会の会員は、大阪学院大学・大阪学院大学短期大学部の専任教員で、人文もしくは自然科学を専門とする者および本会の趣旨に賛同し、役員会の承認を得た者とする。
- 第6条 会員は本会の機関誌に投稿し、かつ機関誌その他の本会の刊行物の配布を受けることができる。
- 第7条 本会には次の役員をおく。任期は原則2年とする。ただし、連続2期を超えてはならない。
1. 会 長 1名
 2. 副 会 長 1名
 3. 庶務委員 2名
 4. 編集委員 2名
- 第8条 会長は会員（教授に限る）の中から投票により選出し、総長がこれを委嘱する。副会長および委員は、会長が会員の中から委嘱する。会長の選出において、得票数が同数の場合は、年長者を選出する。
- 第9条 会長は本会を代表し、会務を統轄する。副会長は会長を補佐する。
- 第10条 会長は役員会を招集して、その議長となる。
- 第11条 会長は会務執行に必要なとき、会員の中から実行委員を委嘱することがある。
- 第12条 総会は年1回これを開く。ただし、必要あるときは、会長が臨時に招集することができる。
- 第13条 本会の経費は大阪学院大学からの交付金の外に、有志からの寄付金、その他の収入をもってあてる。
- 第14条 各学会の相互の連絡調整をはかるため「大阪学院大学学会連合」をおく。本連合に関する規程は別に定める。
- 第15条 会計は毎年4月1日に始まり、翌年3月31日に終る。
- 第16条 本会会則の改正は総会の議を経て総長の承認をうるものとする。

附 則

この会則は、昭和49年10月1日から施行する。

昭和51年9月9日改正

昭和56年1月22日改正

昭和61年6月5日改正

昭和62年6月4日改正

平成15年6月12日改正

平成23年4月18日改正

平成25年4月1日改正

平成30年4月1日改正（ただし、第7条の編集委員の人数は、平成31年3月31日まで4名とする）。

2020年4月1日改正。

以 上

大阪学院大学人文自然論叢投稿規程

1. 投稿論文は、未発表のものに限る。
2. 投稿論文は、編集委員会が依頼した査読者の査読を経た後、編集委員会で掲載を決定する。
3. 投稿資格
 - イ. 投稿者は、原則として本会の会員に限る。
 - ロ. 会員外の投稿は役員会の承認を必要とする。
4. 提出原稿の分量の限度は、原則として本誌刷り上がりページ数12ページとする。

これは和文原稿の場合、200字詰原稿用紙80枚（16,000字）に相当し、欧文原稿の場合、A 4判・12ポイント・ダブルスペースで20枚に相当する。（提出原稿がデータの場合は、Word ファイル形式とする。または、その他のファイル形式の場合テキストファイル形式とする。なお、提出原稿は、正本とコピーを2部提出することとする。）

ただし、制限分量を超えるものについては編集委員会の承認を必要とする。
5. 本誌はB 5判とする。発行は、前期・後期の2回とし、年間総ページ数は220ページを基準とする。
6. 原稿は、論説、研究ノート、資料、その他（レビュー、紹介、書評、学内消息など）に区分し、この順序にしたがって編集する。原稿の掲載順序については、編集委員会で検討のうえ決定する。
7. 別刷（抜刷）40部 無償とする。
8. 要旨集発行のため、要旨を添えること。（200字詰原稿用紙3～4枚。邦文ヨコ書き）

なお、この要旨は独立行政法人 科学技術振興機構（JST）の「科学技術文献データベース」にも掲載される。ただし、和文1,000字、英文2,000字を超える場合は（...）と表示される。
9. 投稿され掲載された成果物の著作権は、著作者が保持する。

なお、出版権、頒布権は大学が保持するため、論文転載を希望する場合は、学会宛に転載許可願を提出願うこととする。
10. 投稿された論文の著作者は、当該論文を電子化により公開することについて、複製権および公衆送信権を大学に許諾したものとみなす。大学が、複製権および公衆送信権を第三者に委託した場合も同様とする。

この規程は、平成26年4月1日から適用する。

以 上

大阪学院大学人文自然論叢執筆要領

原稿について

1. 原稿は確定稿であることを原則とする。校正段階での加筆訂正は極力さけること。
2. 原稿は正本を提出し、かならず原稿提出票をつけること。原稿提出票は学会連合事務局（図書館内）にある。
3. 原稿は黒または青のペン書き、またはワープロ原稿とすること。（欧文原稿はタイプライト）。
4. 原則として目次は使用しないこと。
5. 原則として古典・地名または学術用語などの特殊な例を除き現代かなづかい、および当用漢字を使用し、難字は欄外に朱筆大書すること。
6. 原稿中の数字は原則としてアラビア数字を使用すること。
7. 和文原稿中の欧語はタイプライトまたは活字体とすること。

注と文献表について

8. 注は原則として脚注とする。その他の場合は、注の付け方を明示すること。
9. 原稿での注は一括して本文の文末にまとめること。印刷に際して脚注（あるいは指定した書式）となる。
10. 文献表を必要とする方は、原稿執筆に際して注の後に一括して書くこと。論文末に一括して掲載する。
11. 文献表の配列及び形式は各自が所属する学会で行われているものとする。

図・表について

12. 図・表は1表・1図ごとに別紙に書き、図表ごとに通し番号・図表名・説明文を記入して、本文原稿の右欄外に挿入箇所を指定すること。
13. 凸版印刷にかける図・表は原則としてトレーシングペーパーに墨入れし、厚手の台紙に貼付すること。
ただし、図表中に入れる文字は写植をするため鉛筆書きすること。
14. 写真は鮮明なものに限り、特別な場合を除いて白黒のものを使用し、取り扱い図・表に準ずること。
図・表・写真などのカラー印刷については、役員会の承認を必要とする。
15. 図・表・写真などはかならず縮尺または仕上り寸法を明示すること。

印刷・校正

16. 原則として、執筆者校正は2校までとし、2校までに校正の必要がなくなれば校了または責了と書くこと。
17. 執筆者校正は、校正刷到着の日より3日以内とする。
18. 執筆者校正の段階で加筆・訂正を加え、印刷組替えを生じた場合、組替料の実費を必要とする。また特殊な印刷によって通常の印刷費を大幅に上回る場合も同様とする。
19. 特別に指定される活字の大きさや字体については、執筆者が直接原稿に明示すること。それ以外の印刷上のスタイル、活字号数などは編集委員会において決定する。

以 上

執筆者紹介（原稿受理順）

神戸 尚志（情報学部）教授
山口 雅之（経営実務科）教授
淡 誠一郎（情報学部）教授
淡 裕美子（神戸大学大学院 農学研究科）
吉田 康子（神戸大学大学院 農学研究科附属食資源教育研究センター）
山田 章（教育開発支援センター）講師

大阪学院大学人文自然学会会員（2023年3月31日現在）

秋田 亨	情報学部	岩岡 真弘	経済学部	上地 宏	商学部
上原 邦昭	経営学部	梅田 享	外国語学部	大谷 朗	情報学部
小野寺秀俊	情報学部	川本 芳久	情報学部	神戸 尚志	情報学部
菊野 亨	情報学部	木村 正徳	商学部	佐野 学	情報学部
杉山 伸一	商学部	角田 聡	経済学部	竹井恵美子	国際学部
田中 豊	情報学部	谷口 高士	情報学部	淡 誠一郎	情報学部
寺井 正幸	情報学部	寺脇 久人	経済学部	豊永 昌彦	情報学部
中西 朗	商学部	中山 麻耶	経営実務科	西田 知博	情報学部
野口 隆	経済学部	廣藤千代子	経営実務科	藤原 哲郎	経営学部
増田 ゆか	経営実務科	松尾信之介	経済学部	松田 潤	情報学部
森井 紳	商学部	山内 武	経済学部	山口 雅之	経営実務科
山下 博志	情報学部	山田 章	教育開発支援センター	横山 輝樹	法学部
吉山 昭	情報学部	渡辺千香子	国際学部		

(50音順)

大阪学院大学人文自然学会役員

会長 秋田 亨（情報学部）
副会長 廣藤 千代子（経営実務科）
庶務委員 川本 芳久（情報学部） 増田 ゆか（経営実務科）
編集委員 田中 豊（情報学部） 山下 博志（情報学部）

2023年3月25日 印刷
2023年3月31日 発行

編集兼 大阪学院大学
発行者 人文自然学会
〒564-8511 大阪府吹田市岸部南二丁目36番1号
TEL 06 (6381) 8434(代)
印刷所 大枝印刷株式会社
〒564-0031 大阪府吹田市元町28番7号
TEL 06 (6381) 3395(代)

**THE BULLETIN
OF
THE CULTURAL AND NATURAL SCIENCES
IN
OSAKA GAKUIN UNIVERSITY**

Nos. 85-86

March 2023

CONTENTS

Articles

Takashi Kambe, Masayuki Yamaguchi

C Based System LSI Architecture Design Methodologies and its Evaluation Using
Several Applications 1

Seiichiro Dan, Yumiko Dan, Yasuko Yoshida

Shape Description of Radish by Bézier Curve and its Application 27

Practice Report

Akira Yamada

A Proposed Method for Observational Assessment of College Students with
Developmental Disabilities Gray Zone 59

News 87

Summaries 89

THE ASSOCIATION OF THE CULTURAL AND NATURAL
SCIENCES IN OSAKA GAKUIN UNIVERSITY
OSAKA JAPAN