

ポータブル画像処理ソフトウェア・パッケージ SPIDER の開発†

田村 秀行^{††} 坂根 茂幸^{††} 富田 文明^{††}
横 矢 直和^{††} 金子 正秀^{†††} 坂上 勝彦^{††††}

画像処理アルゴリズムをプログラムとして収集・蓄積し、ポータブルなソフトウェアとして流通させるための仕様を検討し、ソフトウェア・パッケージ SPIDER を開発した。

従来の画像処理ソフトウェアのほとんどは、ハードウェア・システムに依存したプログラミングや、汎用性のない設計思想によって移植性を欠いている。この要因を分析し、現時点でのアルゴリズム収集と高い移植性を実現するプログラム仕様を採用した。画像処理プログラムはすべて入出力操作（ファイル・アクセス、画像用周辺機器駆動等）に独立な FORTRAN サブルーチンとする他、SPIDER 仕様では画像処理概念の整理、コーディング上の制約、コメント文やマニュアルの記法も規定した。

その結果、代表的な画像処理アルゴリズムのサブルーチン・ライブラリとその管理プログラム群、計442個のプログラムからなるパッケージを能率よく製作できた。このパッケージは、きわめて高いポータビリティを有し、すでに約120の研究グループで利用されている。

1. ま え が き

デジタル画像処理は、最近最も活発な研究開発が行われている分野の1つである。特に過去10年の間、ハードウェア技術の目ざましい進歩とともにさまざまな処理手法が発表されてきた。しかし、その多くは特定の応用分野の画像に依存したものであり、一般的な図形・画像を処理する技術体系の確立が望まれている。そのためには、これまでの各種アルゴリズムを比較検討し、体系的整理を行う必要がある。

アルゴリズムを比較する共通基盤としての標準的なデータは、本学会の標準化活動の一環として、データ交換用の標準フォーマットが起案され¹⁾、それに基づいて広汎なデータが収集されている²⁾。しかしながら、アルゴリズムをプログラムの形で収集・蓄積するのはデータの場合ほど容易でない。計算機間で移植可能 (transportable) なソフトウェア仕様を決めることが必須の条件であるが、たとえ標準仕様を決めてもこの仕様に基づいてプログラムを収集するには多大の労力を必要とする。

これまでに開発された各種画像処理ソフトウェア^{3), 4)}のうち、汎用性が高く比較的良好に用いられているものとして、Jet Propulsion Laboratory の VICAR (Video Information Communication And Retrieval)⁵⁾と名古屋大学の SLIP (Subroutine Library for Image Processing)⁶⁾がある。いずれも大型計算機向に作られたソフトウェアで、多くのアルゴリズムが収録されているが、実際の研究に即して蓄積されてきたため、分野的にはかなりの偏りがある。また、ポータブル・ソフトウェアの標準形式と考えるには、移植性はあまり高くない。

プログラム交換のための画像処理ソフトウェアの標準化は、米国⁷⁾や西欧⁸⁾でも検討されている。米国の方針は、システムに依存する部分をすべて標準化し、この上に共通ソフトウェアを構築する考え方で、すでにいくつかの OS とのインタフェースの核 (kernel) が試作されている⁹⁾。一方、西欧では、アルゴリズムの簡潔な記述と専用プロセッサとの整合に重点をおき、既存言語の適性検討と新言語の提案¹⁰⁾が進められている。しかし、いずれにせよ、多くのアルゴリズムの蓄積までには長い年月を要する。

これに対し、我々は、きわめて高い移植性と豊富なアルゴリズム収録をともに達成したソフトウェア・パッケージ SPIDER (Subroutine Package for Image Data Enhancement and Recognition) を開発した。これは、プログラムの蓄積と流通には、初期セットとして充分充実したパッケージが必要との考えに基づい

† Development of a Transportable Image Processing Software Package SPIDER by HIDEYUKI TAMURA, SHIGEYUKI SAKANE, FUMIAKI TOMITA, NAOKAZU YOKOYA (Electrotechnical Laboratory), MASAHIDE KANEKO and KATSUHIKO SAKAUE (Institute of Industrial Science, University of Tokyo).

†† 電子技術総合研究所

††† 東京大学生産技術研究所 (現在, KDD 研究所)

†††† 同上 (現在, 電総研)

ている。SPIDER 開発は計画的に進められ、現実的な移植性を高める方式を採用することにより、目標とするパッケージが実現できたので、その概要を報告する。

以下の章では、まず画像処理ソフトウェアの構成と移植の障害となる要因の分析から始め、SPIDER 設計方針に至るまでを述べる。次いで、決定された SPIDER 仕様の概略を記し、第 1 期開発分の構成と内容を紹介する。

2. 画像処理ソフトウェアの移植性

2.1 画像処理システムのソフトウェア

画像処理ソフトウェア特有の問題として、大量データの操作と画像入出力機器の利用がある。この二大要素は、画像処理システムの中心となる計算機の規模に、矛盾した要求を出す。このため、計算機複合体の構成や独自の画像データ管理方式等、画像処理システムごとに工夫がなされているが、その分だけソフトウェアの負担が大きい。現存の研究実験用のシステムの状況は表 1* に示すように、独自のシステム構成が大半を占めている。

この種の画像処理システムにおけるソフトウェアの担うべき機能は、次のように大別できる。

- (1) ユーザコマンドの解釈
- (2) 画像入出力機器の制御
- (3) 画像処理アルゴリズムの実行
- (4) 画像データ・ファイルの管理

以上の機能がまとめて画像処理ソフトウェア・システムに組み込まれるとき、その管理プログラムには、さらに a) 共通パラメータの管理、b) エラー処理、c) メモリ・オーバレイの管理、等の機能が含まれることが多い。このため、ハードウェア構成やシステムの設計理念に依存するところがきわめて大きい。

表 1 画像処理実験設備についてのアンケート調査

Table 1 Current state of image processing facilities.
(発送 40 名, 回答 33 名)

Q 1: 画像処理実験設備	Q 2: 画像処理実験のプログラミング言語
中・小型機を中心とした 研究室独自のシステム...20名	アセンブラ..... 1名
センタの大型計算機のみ... 2名	FORTRAN15名
両方の併用.....11名	FORTRAN 主体 一部アセンブラ..... 9名
	両方の併用..... 7名
	上記に加えて PL/I 1名

* SPIDER 開発に先立ち、国内の主要研究グループ 40 に対して行ったアンケート調査の一部である。

各研究グループごとに画像処理システムがあり、その数だけ画像処理ソフトウェアがあるという過言でない。

2.2 プログラムの移植を妨げる要因

画像処理アルゴリズムの蓄積という観点から考えると、画像処理ソフトウェアは画像処理プログラムのライブラリとその他のシステム・プログラムに分けられる。ここで、ポータブル・ソフトウェアを純然たる画像処理サブルーチンのライブラリに限っても、移植性の達成は容易でない。多くの場合、前項の(3)の機能が(1), (2), (4)と明確に分離してプログラムされていないからである。

我々は、画像処理ソフトウェアの現状を調査し、プログラムの移植を妨げている要因を、次のように分析した。

(i) プログラミング言語

画像入出力機器の駆動だけにとどまらず、システムのかなりの部分をアセンブラに頼っている場合も少なくない。旧式のミニコンの場合、DOS の下で高級言語を使っても、大きな配列を扱えないためアセンブラ・ベースでデータ管理を行うためである。

FORTRAN 等を用いる場合でも、通常の処理系の違いによる移植の問題点^{11), 12)}は残る。

(ii) 画像ファイルのアクセス

画像に対する演算とデータ・アクセスを混在させたプログラムでは、ファイルの形式の不統一が移植の障害となる。画像処理プログラム内での画像入出力手続き標準化の提案¹³⁾もあるが、各々の画像処理システム設計の要となっている部分だけに、統一は難しい。

(iii) 画像データの内部表現形式

主記憶内の画像配列についても、データ構造が多様である。PAX¹⁴⁾のプレーン構造、ピラミッド構造の表現、メモリ節約のための数画素分のバック表現等が、障害のもととなっている。画像を単純な 2 次元配列として表現する場合ですら、添字と画像の対応は一意的でなく、不一致の原因となる(後述)。

(iv) 画像領域のプログラム間での受け渡し

各画像処理サブルーチンへの入出力画像のリネージは、引数による結合や COMMON 文による共通ブロックを利用する結合などがあって、一定していない。特に作業用領域などは管理プログラムの動的制御下にある場合も少なくない。

(v) エラー処理

実行時のエラー処理に、指定箇所へのジャンプ、

スーパーバイザ・コール, 実行停止等の様々な形態があり, これも移植時の障害となる。

(vi) 画像の大きさと濃淡レベル

システム設計の都合上, 扱える画像の大きさや濃淡レベル数が制限されていることが多い。計算機の語長による制限や入出力機器の精度から, これらのパラメータの上限が決められ, 画像処理プログラムにまでも影響を及ぼしていることがある。極端な場合, 全く固定されていることすらある。

3. SPIDER の設計方針

ソフトウェアの完全な移植性を求めるならば, 何らかの標準案を決めて, 以後すべてこの仕様に従ったプログラムを作成すればよい。この意味では, 米国方式の OS とのインタフェース標準化は興味深い試みである。しかし, 利用されている計算機が多様なわが国では, インタフェースの核部分の開発が利用者側の大きな負担となる。

この場合, 核パッケージが提供されたとしても, 従来の研究実験システムからの転換が強いられる。また, 標準案の不備を改善し, その上に画像処理アルゴリズムをインプリメントしてライブラリを作っていくには, かなりの時間を要すると考えられる。

我々は, 実際にアルゴリズムを蓄積しプログラムを交換することを目的として, より現実的な達成方式を採った。その基本方針は,

- 画像処理アルゴリズムのサブルーチン・パッケージのみを開発し, 画像処理システムの管理プログラムに独立なものとする。

- プログラムは実行効率より汎用性を重視する。完全な移植性は追求しないが, 中型程度の計算機で障害なく画像処理実験ができるよう配慮する。

- 特殊な画像処理システムへの移植や高速化のための改良はユーザの責任とするが, プログラムは解読しやすい明快なものとする。

- パッケージの流通を促進するため, 初期セットとしてかなりの数のアルゴリズムを収録して公開する。

- プログラム蓄積のモデル・ケースとなるよう, ソースプログラム中の注釈行やマニュアルの書き方についても仕様を定める。等である。

システムに独立な画像処理サブルーチン形式は, 大画像の分割処理方式で我々が提案した素ルーチンの 4

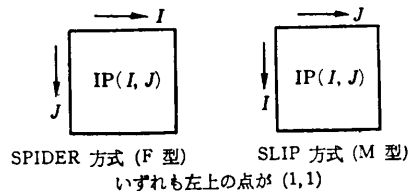


図 1 2次元配列の添字と画像の対応

Fig. 1 Correspondences between 2-D array and image.

表 2 添字—画像対応方式の比較

Table 2 Comparison of two methods for corresponding parameters and image.

SPIDER 方式 (F 型)	SLIP 方式 (M 型)
<ul style="list-style-type: none"> • FORTRAN の記憶領域の割り当て (第 1 添字から変化する) が, 最も自然な水平ラスタ走査に対応する。仮想記憶方式のとき好都合。 • 第 1 添字: 水平方向, 第 2 添字: 垂直方向という関係が, $x-y$ 座標系と同じで理解しやすい。反面, 垂直方向の向きは違うので, 鏡像関係にあり回転による $x-y$ 表現との一致は不可能。 	<ul style="list-style-type: none"> • 添字の順序が行列要素の表現と一致する。行列を扱う演算をプログラムしやすい。 • 添字の順序は, $x-y$ 座標系の水平・垂直方向の概念とは合わないが, 画像を 90° 回転すれば両者は一致する。

条件¹⁵⁾がある。前述の SLIP のほとんどのサブルーチンが, この条件を満足することが確認されている¹⁶⁾。

我々はこれらを念頭において, SPIDER ではさらに高い移植性をもつよう, 各サブルーチンに対して多くの制約を課した仕様を検討した。この結果, 2.2 にあげた移植を妨げる要因に対しては, 各々次のような解決策を得た (番号が対応している)。

(i) 現時点では FORTRAN を使わざるを得ない (表 1 の Q 2 参照)。PASCAL が広まりつつあるが, 整合配列やライブラリの概念がないので, 画像処理には適さない。

(ii) 画像データの準備と格納は, ユーザプログラムの責任とし, SPIDER サブルーチンは主記憶内 (もしくは同等に扱える仮想記憶内) の 2次元配列のみを扱う。原則として他の入出力文も禁止する。

(iii) 1画素=1配列要素として扱う。2次元配列の添字と画像の向きとの対応は, 図 1 のように SLIP とは異なった解釈をする*。(表 2 に両方式の比較を示した)。

(iv) 画像はすべて配列名を引数として受け渡す。

* 一般にこの 2つの表現のいずれかが用いられていて, 無意識のうちに思考が片方に固定されている人が多い。両方式のプログラムを混用すると, 方向に依存する演算 (回転, 追跡等) で不都合が生じるので注意を要する。

原則として、他の場合も共通ブロックの使用は禁止する。

(v) エラー発生に対してはエラーコードを引数として返す。SPIDER サブルーチン中での、STOP 文、PAUSE 文、スーパーバイザ・コールは、いかなる場合も禁止する。

(vi) 画像配列の仮引数は整合配列 (adjustable array) とし、画像の大きさは必ず引数で与える。濃度レベル数に関する演算についても同様に、必要な配列名と大きさ (レベル数) を引数で与えられるよう設計する。

4. SPIDER サブルーチン仕様

SPIDER に登録されるサブルーチンが満たすべき仕様は、3. の基本方針のもとに、プログラム作成中にも検討を加えて決定した。紙数の制限上、詳細と注意事項は利用者用マニュアル¹⁷⁾に譲り、本論文ではその概略のみを列記する。

4.1 FORTRAN 言語のレベル

JIS で規定された最高水準 7000 は現在ではかなり低く、ほとんどのコンパイラが付加機能を競っている。これを標準案というだけで JIS 7000 に制限するのは、ほとんどのユーザの慣用に負担をかけ、新規プログラムの登録促進を妨げる。また、JIS 7000 に正しく一致するコンパイラというのも見当らない。

SPIDER 仕様では、現行の国産計算機の M シリーズ、ACOS シリーズ、COSMO シリーズの中位機種程度の FORTRAN レベルを認めることにした。JIS 7000 を超える機能、規定されていない項目については、上記 3 シリーズの共通部分集合を許し、ANSI FORTRAN 77 で縮小される機能* は禁止することを原則とした。ただし、概念は同じだが記法が一致しない項目については、共通でないからといって排除することはなく、いずれかの記法を採用した。

以上の条件だけでは、かなり多くの項目で JIS 7000 を超える機能が見られるが、入出力文や停止文等を禁止した SPIDER サブルーチン設計方針の制限により、大部分が考慮対象外となる (詳細は略す)。

4.2 サブルーチン設計方針

3. で述べた (i)~(vi) の対策以外で、画像処理サブルーチンの設計方針として主要なもののみを記す。

■ SPIDER に登録する画像処理プログラムは、

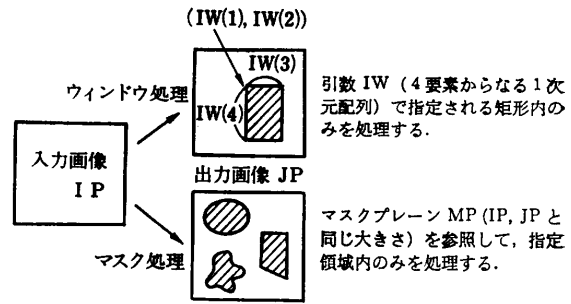


図 2 ウィンドウ処理とマスク処理

Fig. 2 Concept of windowing and masking operations.

原則としてサブルーチンとし、副プログラム間の引用関係 (階層性) を許す。(ユーザが直接使用しない内部ルーチンは、関数副プログラムでもよい。)

■ 標準的な画像データ形式は、単精度整数型の 2 次元配列とする。同機能で実数型配列を扱うサブルーチンは必要に応じて別に作成する。本質的に実数型や複素数型を必要とする処理 (例、FFT) についてはこの限りではない。

■ 画像に対する演算は、引数で与えられた配列全画面を処理対象とするものを基本とする。画面内の一部分を処理するウィンドウ処理や参照プレーンによるマスク処理 (図 2 参照) を行うものは、基本型のサブルーチンとは別に付加的に作成する。

■ 原則として入力画像と出力画像は別の仮引数とし、入力を保存する。入力と出力で配列を共有してメモリを節約したい場合には、(許されている演算についてのみ) 実引数を同じにする。

■ その他、画像処理演算の性格に応じて、サブルーチンへのモジュール化に対して配慮する (5.3 参照)。

4.3 コーディング上の取り決め

これも主要なもののみを記す。

■ サブルーチン名の形式、引数名の頭文字、よく使う引数の種類に対する代表的な変数名 (または配列名) は指定の様式に従う (詳細は略す)。

■ 入出力画像配列用の引数は、他の引数より前に、エラーコード用の引数は一番後に置く。

■ 10 進数以外の定数 (8 進数、16 進数、文字等) は、DATA 文で指定し、実行文には書かない。

■ 文番号はなるべく FORMAT 文と CONTINUE 文のみにとどめる。

■ DO の入れ子や IF 文による分岐先はなるべく

* たとえば、DO 拡張範囲やホラリス型文字定数などがある。

indentation を行ってプログラムを見やすくする。

■ 反復には原則として、DO ループを用い、IF 文と GOTO 文の組合せは避ける。また、多重 DO 文の順序は、第 1 添字から順に内側の入れ子とする。

■ 処理系の語長に依存する演算は極力避ける。やむを得ぬ場合は、特記事項として注釈行並びにマニュアルに明記する。

プログラムは、SUBROUTINE (または FUNCTION) 文、主要注釈行 (CF, CM は除く)、宣言文、DATA 文、実行文の順とする。

4.4 注釈行についての規定

コメント (注釈行) は、テキストエディタやサービスプログラムで検索区分できるよう性格によってグループ化する。下記のように、行の先頭文字 C に続く第 2 文字を識別記号とする。

CS (Sequence)	コーリング・シーケンス
CP (Purpose)	目的・機能
CA (Argument)	各引数の説明
CV (Variable)	その他の変数の説明
CN (Note)	引数の条件や使用上の注意
CE (Error)	エラーコードの説明
CM (Maintenance)	修正・保守用の特記事項説明
CD (Date)	作業年月日と作業者名 (作成・虫取り・改良の都度)

CK (Keyword) キーワード (区切りは「,」)
 以上は SUBROUTINE 文の後、宣言文の前に入れる。ただし、CM は修正すべき文の直前に入れてもよい。以下のコメントはプログラム途中で随時入れる。

CF (Flow) アルゴリズムの大略のフローをたどれるような注釈

C* 雑コメント

C } 等は、好みによって単なる区
C----- } 切りとして使ってよい。

4.5 マニュアル記入事項

利用者用マニュアルに記入すべき項目は、サブルーチン名につづく ① 目的・機能, ② Calling Sequence, ③ 引数の説明, ④ 引数についての条件, ⑤ 使用上の注意, ⑥ アルゴリズム, ⑦ エラー処理, ⑧ 特記事項, ⑨ 実験例, の 9 項目である。特に、移植時の問題点は ⑧ に記して注意する。

5. SPIDER の構成と内容

5.1 パッケージの構成

これまで述べてきた方針と仕様に基づいて、各種画像処理プログラムを収録したパッケージ SPIDER を開発した。第 1 期開発分の SPIDER は、次の 3 種のプログラム群から構成されている。

(a) 画像処理用プログラム

各種画像処理アルゴリズムのプログラム群であり、SPIDER の中心となる部分である。総数 411 個のサブルーチンまたは関数副プログラムであるが、マニュアルに記載されユーザが直接利用できるのは 350 個*のサブルーチン (表 3) である。

(b) リスト処理用パッケージ LPN

(a) のプログラムのあるものは、数値計算やリスト処理といった画像処理以外の演算も必要とする。このうち数値計算については、いずれの処理系においても何らかのライブラリが用意されていると考え、準備はユーザ側の責任とした。SPIDER 開発に主として用いた ACOS-6 についてのサブルーチンが引用されているので、それ以外の処理系ではこの部分を書き換える必要がある。一方、リスト処理については用意されていないことが多いので、FORTRAN で利用可能なリスト処理プログラムとして発表されている LPN¹⁸⁾ を追加・修正して、SPIDER に付録として収録した。これは、総数 28 個の副プログラム群である。

(c) サービス・プログラム

(a) 及び (b) は、個々の処理系でオブジェクト・ライブラリとして使用されることを前提としている。パッケージとしてこれらの付帯情報を管理するためのプログラムが 3 種類用意されていて、表 4 に示す機能をもつ。いずれも主プログラムであり、カード形式 (または同等のファイル中) のソース・プログラムを入力とするよう作られている。

以上の (a), (b), (c) 計 442 個のプログラムが、SPIDER の第 1 期開発分であり、カードで約 42,000 枚、マニュアルは約 700 ページである。

5.2 アルゴリズムの採用方針

中心となる画像処理用プログラムについては、汎用パッケージとして流通するよう、なるべく広範囲のアルゴリズムを収集の対象とした。まず、「画像処理」

* このうち、名古屋大学の協力により SLIP から変換移植したものが 70 個ある。両ライブラリの併用を考慮して、混乱を避けるためサブルーチン名はすべて変更してある。

表 3 SPIDER 第1期開発分画像処理プログラム
Table 3 Image processing programs in SPIDER 1st edition.

<p>直交変換 [53] FFT (基数: 2, 混合, 任意), WHT, Haar 変換, Slant 変換, 離散的 Cosine 変換 (パタフライ形式, FFT 形式). 関連ルーチン: フーリエスペクトル計算, WHT スペクトル計算 (2種), たたみこみ (直接法, FFT 法). 相関 (直接法, FFT 法), パワースペクトル面での分布計算 (半径方向, 角度方向) と窓処理 (矩形, ガウス, ハミング, ハニング), 周波数領域での微分フィルタ (gradient, Laplacian), その他.</p> <p>位置合わせ [26] 相関法 (粗サーチ, 精サーチ), SSDA 法 (固定または傾斜しきい値, しきい値の指定または自動決定, 粗サーチまたは精サーチ), アフィン変換 (4点線形補間, 9点2次補間, 補間なし) とパラメータ決定 (回転角, 参照点), 一般2次変換 (アフィン変換と同じ3種).</p> <p>強調と平滑化 [16] ヒストグラム変換 (平坦化, 双曲線化, 一般), 反復による強調とノイズ除去 (2種), ヒステリシス平滑化 (標準型, 対称型), E-フィルタ, 高速メディアノ・フィルタ, エッジ保存平滑化.</p> <p>復元 [15] 逆フィルタ (4種), ウィーナ・フィルタ, 制限つき最小2乗フィルタ, SVD による擬似逆変換 (3種).</p> <p>エッジと線の検出 [25] 差分型 (Laplacian, Roberts, Sobel, Prewitt), テンプレートマッチング型 (Prewitt, Kirsch, Robinson), Frei & Chen アプローチ, Hueckel オペレータ (Hueckel, M��r�� & Vassy), 反復型 (Kasvand), Hough 変換 (Duda & Hart, O'Gorman & Clowes), ヒューリスティック探索法, その他.</p>	<p>弛緩法の応用 [13] 基本ルーチン群, 線の強調 (係数の指定または自動決定2種), エッジの強調, ノイズ除去.</p> <p>テキスト解析 [23] Cooccurrence 行列, 差分統計量, 局所的極値, ランレンジス, 自己回帰モデル, 自己相関, フーリエ特徴, テキスチャーエッジ検出, テキスチャーエッジ保存平滑化.</p> <p>領域分割 [6] KS 検定法, ヒューリスティック法, 反復合併法, 相対的類似度法, Split and Merge 法, その他.</p> <p>幾何学的特徴の処理 [81] 連結成分の処理 (ラベル付け, ラベルの通し番号化, 取捨選択), 境界線抽出, 境界線の記述 (チェーン符号, 傾き, 曲率, フーリエ記述), 形状の特徴 (始点, 重心, 外接長方形, 面積, 周囲長, サイズ, 伸長度, モーメント, フーリエ記述), 膨張と収縮, 細線化 (4種), 縮退化 (5種), WPM 法, 距離変換とスケルトン, 断面, 投影, その他.</p> <p>基本的処理 [83] 基本統計量, ヒストグラム作成, しきい値処理 (各種) としきい値選択 (4種), 線形フィルタ, 濃度階調の変換 (正規化, シフト, 他), 再量子化, 画像定数間・画像間四則演算, 転置, 型変換, 代入と転写 (各種).</p> <p>その他 [9] 図形 (テストパターン) の発生, LP への画像出力.</p>
--	---

[] 内はユーザが直接利用できるサブルーチン数を示す.

表 4 サービス・プログラムの機能
Table 4 Contents of service programs for handling secondary information on source programs.

プログラム名	指定事項	機能
SP 1	副プログラム名 コメントのレベル区分	指定の (もしくはすべて) の副プログラムについて, 4.4 節で規定したコメント情報を, いくつかのレベルに分けて出力する.
SP 2	キーワード	CK で始まるコメント文を検索し, 指定したキーワードのいずれか1つを含む副プログラムの Calling Sequence を出力する.
SP 3	なし	副プログラム間の引用関係を表にして出力する.

の範囲として, 画像を入力とした処理のうち, 劣化した画質を強調・復元して変換画像を出力する狭義の画像処理技術と, 認識・理解のための特徴抽出を行う画像解析技術の両分野を扱っている. 第1期開発分の SPIDER サブルーチンを機能別に分類したものを表に示す. ここでは, データ圧縮のための符号化法*, 投影像からの画像再構成法は含んでいないが, その他の画像処理分野全般について, 従来発表されていると

* 直交変換も符号化の重要な手法であるが, ここでは周波数領域での処理や特徴抽出のための一般的な手法として入れた. その他の符号化特有の手法 (例, 予測符号化, フレーム間相関) は含んでいない.

のライブラリよりも広汎にアルゴリズムを収集しているといえよう.

画像処理の各分野については代表的な手法を網らそうよう努力したが, アルゴリズム・ライブラリとして特に留意した点は次の通りである.

(1) 類似した手法が数多く提案されているが, 評価が定まっていなないもの (差分オペレータ, しきい値選択法等) や, 用途に応じて使い分ける必要があるもの (細線化, 縮退化等, 表5参照) は, なるべく多くの種類を集める.

(2) 基本的に同一のアルゴリズムでも, 用法によって著しく計算効率の違うもの (FFT, SSDA 法, アフィン変換等) は, 各種変形版もサブルーチンとして独立させる.

(3) よく知られている手法でも, やや複雑なため個人ではなかなか揃えられなかったもの (弛緩法, Hueckel オペレータ, 発見的探索法, Split & Merge 法等) を積極的にプログラム化して収集する.

表3に示したものは, この方針の下に時間と労力の制約の許す限りの収集であって, ライブラリの性格上今後新しい機能のプログラムを追加して行くことは何ら差しつかえない.

表 5 連結成分縮退化用のサブルーチン

Table 5 Subroutines for shrinking connected components.

サブルーチン名	処理方式	縮退点の位置	多重連結成分の縮退	備考
SRNK 1	並列型	外接長方形の左上	可	Leviardi のアルゴリズム
SRNK 2	並列型	ほぼ中心	可	Rao のアルゴリズム
SRNK 3	逐次型	塊状成分各々のほぼ中心	不可	田村のアルゴリズム
SRNK 4	逐次型	おおむね右下隅	不可	横井のアルゴリズム, 逐次 (I-1) 型
SRNK 5	逐次型	同上	可	横井のアルゴリズム, II 種逐次型

5.3 サブルーチン化のための配慮

画像処理アルゴリズムを SPIDER 仕様のサブルーチンにインプリメントするに当って、画像処理演算のモジュール化にいくつかの配慮がなされている。主なものは次の通りであるが、いずれも SLIP では考慮されていない点である。

(1) 階層化と下位ルーチンの一般化: SPIDER 仕様ではサブルーチン間の階層性を許しているので、下位レベルのサブルーチンを一般化し、なるべくユーザからも利用できるようにした。例として、ヒストグラム平坦化による画像強調の場合を図 3 に示す。ここで、HGTR 1 は任意のヒストグラム変換を、GTRN 1 はあらゆる濃度階調の変換を実行できるようサブルーチン化されている。この階層的モジュール化により、SPIDER 仕様では、最上位ルーチン HGEQ 1 は下位レベルで必要な変換表や入出力画像のヒストグラム領域とその大きさを、すべて引数として確保する面倒さがある。しかし、HGEQ 1L を HGHY 1L に変更するだけでヒストグラム双曲化が実現できる。また、下位サブルーチンを適宜組み合わせることにより、新しい画像強調法の開発も容易である。

(2) 入力画像の保存: この原則に対して、近傍処理演算は逐次処理と並列処理で対処の仕方が異なる。そこで、多入出力、反復演算、多段操作などの要因を含めて、図 4 のようなモデル・チャートを作成し、プログラム設計に用いる。この図式の採用により、作業

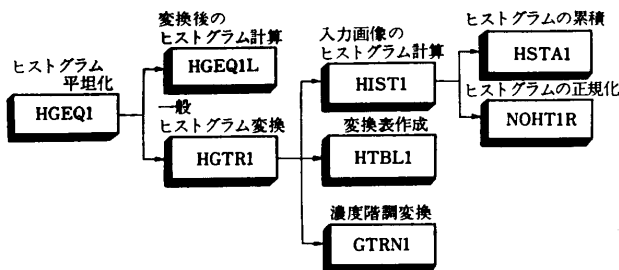
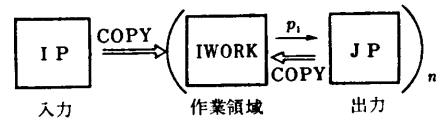


図 3 サブルーチンの階層構造例

Fig. 3 An example of hierarchical structure of SPIDER subroutines.



実引数

IP=IWORK or IP=JP が可能
ただし IWORK ≠ JP

図 4 並列型反復処理の図式

Fig. 4 Diagram of parallel iterative operation.

表 6 分割処理についての対応

Table 6 Considerations for partitioned processing

項目	対策
近傍処理	画面の端処理の明確化
反復演算	1 反復ごとの制御と収束状態の監視
位置に依存した処理	部分画面の位置を指定
累積型演算	部分画面内の結果を累積
追跡処理	画面端に達したとき追跡終了

領域と転写の必要性が容易に判断できる。

(3) 分割処理についての配慮: 主記憶内で一度に処理できない大画像については、いくつかの画面に分けて処理する分割処理方式¹⁵⁾を考える。画像処理アルゴリズムの分割処理可能性¹⁹⁾について、SPIDER サブルーチンはなるべくこれを達成するよう対処した。詳細は複雑になるので別論文に譲り、概略のみを表 6 に記した。図 3 中の HSTA 1 は、この対策を施した累積型ルーチンの一例である。これが、HIST 1 より下位にあることに注意されたい。

この他、対話型指示による画像強調や知識を導入した領域分割などモジュール化しにくい手法も若干残ったが、ポータブル・ソフトウェアとしては実現が難しいため今回は除外した。

6. む す び

画像処理システムのソフトウェアの移植性について検討し、現実的なプログラムの蓄積・流通を目的とした仕様によるサブルーチン・パッケージ SPIDER を開発した。SPIDER 仕様は単にプロ

グラム設計に関する制限事項にとどまらず、コメント文やマニュアル書式までも規定している。これは、ユーザレベルでの大規模な応用プログラムの開発・蓄積にとって、興味深い実例となるであろう。

蓄積と流通という目標に対して、第1期開発分のSPIDERはまず期待通り、あるいはそれ以上の実績を示したといえよう。画像処理アルゴリズム・ライブラリとしての質及び量は、これまでに公表されたいかなる画像処理ソフトウェアをも凌いでいる。これは従来の画像処理システム・ソフトウェアにあった無駄を省き、綿密で現実的な仕様設計に基づいて画像処理アルゴリズムのみをプログラム化したためである。

本稿は、ポータブル・ソフトウェア開発の一例として、また画像処理研究者のプログラム作成のガイドラインとしての資料を提供するものである。実際、SPIDERはこれまでに、国内だけで120以上の研究グループに配布され、ほぼ完全な移植実績を示している。また、これらユーザ・グループ内では、SPIDER仕様が定着しつつある。

謝辞

SPIDER開発には、アンケート調査、プログラムの収集と新規作成、コンパイル作業等について、数多くの方々のご協力を得ました。特に、東大生研の尾上守夫・高木幹雄両教授にはこの計画全般にわたるご指導を、名大の福村晃夫教授及び鳥脇純一郎助教授（現、豊橋技科大教授）にはSLIPの一部を変換・提供して頂くとご協力を得ました。また、前電総研パターン情報部長西野博二氏（現、筑波大学教授）のご理解とご支援によって本計画が実行できました。その他、石井治ソフトウェア部長を初めとする電総研関連研究部・室長及び研究室の皆様のご協力に感謝いたします。

参 考 文 献

- 1) 尾上他：イメージプロセッシングの振興と標準化，情報処理，Vol. 21, No. 6, pp. 645-659 (1980).
- 2) 尾上，坂内，稲本：SIDBA, MIPC-Report 79-1, 東大生研 (1979).
- 3) Borriello, L. et al.: An image processing support software study, CSATA Report, Bari,

Italy (1978).

- 4) Preston, K. Jr.: Image processing software—A survey, in Progress in Pattern Recognition, Vol. 1, North Holland, pp. 123-148 (1981).
- 5) Frieden, H.: Image processing system VICAR—Guide to system use, Jet Propulsion Lab., Rept. 71-135 (1971).
- 6) 鳥脇，福村：画像処理のためのサブルーチンライブラリSLIPについて，情報論，Vol. 22, No. 4, pp. 353-359 (1981).
- 7) Hamlet, R. G. and Rosenfeld, A.: Transportable image processing software, Proc. NCC, Vol. 48, AFIPS Press, pp. 262-272 (1979).
- 8) Workshop on “High-level Languages for Image Processing,” Windsor, Britain (June 4-8, 1979).
- 9) Guerrieri, E.: Software/O. S. interface kernel user manual, Rensselaer Polytech. Inst., IPL-TR-31-008 (1981).
- 10) Levialdi, S. et al.: Programming in PIXAL, Proc. 2nd Workshop on Image Data Description and Management, pp. 74-79 (1980).
- 11) 松下，山崎，丹下：プログラム・トランスフェラビリティ，情報処理，Vol. 16, No. 10, pp. 871-879 (1975).
- 12) 藤田，葛山，川原：プログラムの移植について，同上，Vol. 21, No. 11, pp. 1128-1135 (1980).
- 13) Haralick, R. M.: Image access protocol for image processing software, IEEE Trans. Soft. Eng., Vol. SE-3, No. 2, pp. 190-192 (1977).
- 14) Johnston, E. G.: PAX II picture processing system, in Picture Processing and Psychopics, Academic Press, pp. 427-512 (1970).
- 15) 田村，森：画像処理基本プログラムの互換性と大画像の分割処理方式，昭53信学総全大，No. 1010 (1978).
- 16) イメージプロセッシング研究連絡会資料（非公開）。
- 17) 画像処理サブルーチン・パッケージ SPIDER User's Manual, 電子技術総合研究所 (1980).
- 18) Bray, D. W.: Dynamic storage routines for Fortran programs, SIGPLAN Notices, Vol. 9, No. 9, pp. 2-13 (1974).
- 19) 鳥脇，横井：画像処理のアルゴリズム，情報処理，Vol. 21, No. 6, pp. 613-619 (1980).
(昭和56年1月19日受付)
(昭和56年12月17日採録)