パーソナルマシン向き汎用対話型画像処理システムのソフトウェア設計†

福島重広†† 木村雄太郎††*

多様な機能をもつ対話型汎用画像処理システムが普及型のきわめて小規模な低廉機器環境下において実現可能なことをシステムソフトウェアの設計例によって示す．とくにシステムの枠組としてのプログラム管理と画像データ管理の方式，コマンドの制御，マンマシン対話方式について詳述する．本システムではプログラムも画像データもともにディスクに常駐するオーバレイ方式がとられている．開発効率を重視して，とくに画像処理プログラムはFORTRANで書くようにしている．画像処理プログラムやユーティリティはコマンド入力により起動され，制御パラメタ値は促進メッセージへの応答により与えるが，あらかじめ選択的に固定しておくこともできる．このためにパラメタファイルが用いられ，また，入力管理ルーチンが用意されている．自律的な処理が行えるように，マクロコマンドが定義でき，そのためのエディタが用意されている．パラメタ値固定を併用すれば，必要最小限の対話による操作が可能になるわけである．また，システムや各画像の処理経過が記録されるようになっており，これをエディタへ入力することによって，処理経験にもとづくマクロコマンド定義が容易にできる．このほかにも，利用の気楽さや応用システムの開発の便宜を考慮した柔軟な対話が可能になっている．画像データアクセスインタフェースルーチンが用意されているので利用者プログラムの作成も容易である．

1. まえがき

70年代以来，ミニコンピュータ主体の汎用の対話型画像処理システムがいくつか報告されてきた[1)~5)]．これらのシステムでは，知能端末として大・中型のホスト計算機の負荷軽減のために画像入出力機器の制御と低レベル処理を分担したり，専用のOSやアセンブリ言語を用いることにより実行効率の向上を図るといったことが目標とされてきた．したがって，システムに用意された機能を使用するかぎりにおいてはきわめて便利であり，今日の高性能の画像専用プロセッサの原型になってきたわけであるが，一方，一般の利用者が自らプログラムを開発して組み込んだり，システムを改良したりすることはほとんど不可能であり，また，システムと気軽に対話しながらマクロな手続きを組み上げていくとか，コマンドやパラメタ値の入力上の使いやすさに対するソフトウェア上の配慮もあまりなされていなかったように思われる．その後，OSの発展と画像処理アルゴリズムの多様化に対応して，このような配慮のもとにシステム開発が行われ，対話とプログラム開発上の便宜のためにかなりの工夫がなされた[6)]．しかし，このシステムは複数の利用者による同時使用をも目ざしていたため，複雑な構成になっていたようである．これは開発当時として比較的高級なミニコンや高機能のOSが選択されたという事情にもよるものと思われる．同様のシステムが米国でも開発されたが詳しくは報告されていない[7)]．われわれは一人の利用者による占有使用を目標として簡素かつ変更が容易で操作性も高いシステムを開発した[8)]．

ところで，最近，低廉な汎用計算機（いわゆるパソコンなど）が普及し，また，ディジタル画像処理技術の普及にともなって，低廉な画像入出力機器や画像メモリも出回るようになってきた．このような低廉機器は，高性能の画像専用プロセッサのような実行効率のよさは到底望めないにしても，個人的な利用が可能であるから，ソフトウェアとハードウェアの両面において，比較的自由が効くという利点がある．したがって，個人的規模で自由に画像処理をしてみたいという要求が利用者側にも強くなってきている．しかし，一方でまた，画像処理に関してはソフトウェアの支援体制が十分にできあがっていないという問題がある．最近，パソコンによる画像処理システムが発表されるようになってきてはいるが，上述のミニコン時代の後追いをしているだけのような感もある．しかし，個人用機器にはその占有性を生かしたシステム開発の方向があると考えられる．本稿はその意味での参考に供するためにわれわれのシステムについてまとめたものである（本システムは1978年から1979年にかけてミニコ

† Software Design of a General-Purpose Interactive Image Processing System for Personal Machines by SHIGEHIRO FUKUSHIMA and YUTARO KIMURA** (Department of Electrical Engineering, Kyoto Institute of Technology).

†† 京都工芸繊維大学工芸学部電気工学科
* 現在 (株)島津製作所
** Currently, with Shimadzu Corporation.

ン上で実現したが，ソフトウェア自体の考え方は，むしろ今日のパソコンに向いている)．

使いやすい画像処理システムのソフトウェアへの要求仕様としては，画像処理アルゴリズムの豊富さ，可制御パラメタの豊富さ，画像のデータ型やサイズの任意性，コマンド情報検索機能，システム利用経過報告機能，画像データの利用経過・利用状態報告および保護機能，制御パラメタ値の選択的固定機能，マクロコマンド定義機能，利用者プログラムの組み込みの容易性などがあげられよう．本稿はこのような要求を満たす汎用画像処理システムが小規模の低廉機器環境下においても実現できることを設計例によって具体的に示す．

このようなシステムでは，まず第一に，プログラムやデータの管理方式，コマンド制御，マンマシン対話方式などのシステムの枠組を明確に設計しておかなければならない．実際，画像処理のための個々のアルゴリズムはソフトウェアパッケージ[9]などを参考にして利用者が集積していくこともできる．したがって，以下では，まずシステムの概要を示し，その後，システム管理とユーティリティなど，システムソフトウェアの構成について詳しく述べる．なお，本ソフトウェアには高木らのシステム[6]の考え方がとり入れられているが，記述の煩雑さを避けるため，細かな比較対照は行わない．

2. システムの概要

本システムの仕様は大略つぎのようである．

1) 対話型システムであり，画像処理プログラムやユーティリティはコマンド入力により起動する．制御パラメタ値は促進メッセージへの応答として入力する．パラメタ値を選択的に固定することもできる．

2) プログラムの多くはFORTRANで書かれている．とくに画像処理プログラムをFORTRANで書くことにより開発効率を上げている．

3) 自律的処理が行えるように，マクロコマンド定義機能をもつ．パラメタ値固定機能を併用することにより，必要最小限の対話による操作が可能である．

4) 画像サイズ 512×512 までの正方形画像が処理できる．画像データの型は整数型，実数型，複素数型(このときは画像サイズ 256×256 まで) のいずれでもよい．

5) 基本的な画像処理アルゴリズムを多く備える．

6) 画像のデータ型やサイズが管理されているので，コマンドを処理機能だけで特定できる．データ保護機能もある．

7) 補助記憶上の画像データへのアクセスインタフェースルーチンやパラメタ値の入力管理ルーチンが用意されているので，画像処理プログラムの作成が容易である．

ハードウェア環境は，CPU として汎用ミニコン NOVA 01 (主記憶 32 キロ語，16 ビット)，補助記憶として 2.5 メガバイトのディスク 1 台を用い，このほかに，必要に応じて画像入出力装置や磁気テープ装置などを付加するものとした．

3. ソフトウェア構成

本システムの構成プログラムにはシステム管理，ユーティリティ，画像処理の 3 種類があり，それぞれ，必要に応じて主プログラムにより起動される．これらをシステムのセグメントプログラムと呼ぶ．

3.1 システム管理

本システムは開発効率を重視してメーカ提供の汎用オペレーティングシステム NOVA RDOS 上に実現した．このため，主記憶の利用者領域は約 20 キロ語しかなく，その効率的配分のため，プログラムも画像データも，ともにオーバレイ構造とした．主記憶はきわめて小さい主プログラム* の常駐領域のほか，画像データのオーバレイ領域として 8 キロ語，セグメントプログラムのオーバレイ領域として，1,536 語，システム制御データ領域として 2.5 キロ語，ランタイムライブラリ常駐領域として 6.5 キロ語をそれぞれ配分し，残りを FORTRAN ランタイムスタックに用いている．プログラムオーバレイ領域は普通の画像処理プログラムには十分なサイズであり，約 200 行の FORTRAN 文に対応している．画像データアクセスやパラメタ値入力はランタイムライブラリの中のサブルーチンを呼び出すだけであるから，オーバレイ領域はごくわずかしか占有しない．また，モジュラリティを高めるために画像機器のハンドラもセグメントプログラムに含めるようにしているので，機器の増設による主記憶配分への影響もほとんどない．

システム管理はプログラム管理，データ管理，コマンド制御，対話管理に大別される．

3.1.1 プログラム管理

本システムの制御はすべてセグメントプログラムによって行われ，主プログラムは指定されたセグメント

* FORTRAN 文 10 行とアセンブラ文 24 行から記述されている．

プログラムをオーバレイ領域へロードし，起動するだけである．コマンド入力の促進や実行すべきセグメントプログラムの決定も専用のセグメントプログラムが行う．このプログラムはシステム起動時またはセグメントプログラム実行終了時に自動的にロードされる．一方，一般のセグメントプログラムがその次に実行すべきセグメントプログラムを指定することもできるようになっているので，オーバレイ領域に一度にロードできないような大きなプログラムもチェイニングにより実行可能である．

3.1.2 データ管理

任意サイズの画像を取り扱うために，画像データ常駐領域はディスク上にとられており，アクセスしたい画素を含むデータブロックを主記憶上のデータオーバレイ領域へ逐次ロードするようになっている．ディスク上の領域は二つの作業ファイルから構成されている．一方は2,048ブロック（1ブロック＝256語）を占める．画像サイズが指定されると，この領域は分割可能な数（最大50）の画面に分割され，画面番号が1から順につけられる．整数画像に対しては1画面，実数画像に対しては連続した2画面，複素画像に対しては連続した4画面が割り当てられる．他の作業ファイルは514ブロックから成り，可変サイズの画像（画像番号0）を割り当てる．これは，画像処理では通常は画像サイズが不変であるが，画像サイズ低減や類似検出のような異サイズ画像間操作もあるという事情を考慮したものである．

さて，画素へのアクセスは主記憶上のデータオーバレイ領域をバッファとして行われるわけであるが，そのインタフェースのために仮想配列方式が用いられている．このためのサブルーチンが三つ用意されており，それぞれ，バッファの割り当て，画素へのアクセス，終了宣言を行う．これは，ちょうど，ファイルアクセスがオープン，読み書き，クローズの3種類の命令によって行われるのに似ている．これらのサブルーチンの機能を呼び出し手続きによって説明しよう．

1) バッファの割り当て．主記憶にとられた8キロ語のデータ領域またはその一部をバッファとして画像データに割り当てる手続きは

CALL VADD (IMAGE, ISIZE, IPICT, NPS, NPD, IER)

である．配列 IMAGE がバッファとして用いられ，そのサイズは ISIZE で与える．バッファには画面番号 NPS から始まる画像領域のデータが読み込まれ，

表1 仮想配列制御ブロックの内容

Table 1 Contents of a virtual array control block.

Element	Content
1	Number of columns per page* of the image.
2	Column length (or row length) of the image.
3	Displacement of the first record of the source image on the disk file.
4	Displacement of the first record of the destination image on the disk file.
5	Channel number of the file including the source image.
6	Channel number of the file including the destination image.
7	Page number of the page in the first partition of the buffer.
8	Page number of the page in the second partition of the buffer.
9	Page number of the page in the third partition of the buffer.
10	Image data type: integer=1, real=2, and complex=4.
11	Number of records per page, equal to the content of the first element multiplied by that of the tenth.
12	Address of the first location of the first partition of the buffer.
13	Address of the first location of the second partition of the buffer.
14	Address of the first location of the third partition of the buffer.

* A page is a segment of an image, loaded in one partition of an image buffer at one time.

処理結果は画面番号 NPD から始まる画像領域に書き出される．したがって，NPS と NPD に異なった値を与えることにより，原画像データを保存することもできる．バッファは3分割され，それぞれ，画像の連続した列に対応する連続した論理レコードを単位としてロードされる．この単位を頁と呼ぶ．3分割法は頁境界付近の画素に近傍操作が適用される際に頻繁なディスクアクセスが生ずるのを防ぐ目的で導入された方法である[6]．配列 IPICT は14語からなる制御ブロックであり，呼び出しプログラムへ戻るときに，画像の一列の長さ，データ型，頁当り列数および論理レコード数，原画像と処理結果画像をそれぞれ含む作業ファイルのチャンネル番号，作業ファイル上における両画像の先頭レコード番号，バッファの各分割部分上の頁の頁番号，各分割部分の先頭番地がセットされる（表1）．

ディスクアクセスと転送の回数を減らすために，処理プログラムが画像データの読み込みだけ，または，書き出しだけを行う場合も考慮してある．すなわち，

NPD＝−1とすれば，バッファのデータは他の頁が読み込まれることによりオーバライトされるときでもディスクへは書き出されない．これは，たとえば濃淡値の統計量を求めるときのように，読み込みだけの処理を考慮したものである．また，たとえば画像生成や並列処理（例：非再帰的フィルタリング）のように処理結果をディスクへ書き出すだけでよい場合は，NPS を負の値にすることにより，ディスクからの読み込みが省略される．引数 IER はエラーフラグであって，NPS により指定された原画像領域が未定義な場合，NPS の値が実数画像や複素画像に割り当てられている連続した複数の画面の先頭に対応していない場合，または，NPD により指定された結果画像領域の中に書き込み禁止画面が含まれている場合にエラー状態がセットされる．

2) 画素へのアクセス．バッファが割り当てられると，その画像の各画素は手続き

CALL VARW (I, J, IVAR, IRW, IPICT)

によりアクセスできる．ここで IPICT は前述の制御ブロックである．つまり，このブロック名は画像識別子になっている．整数の対 (*I*, *J*) は行列的に各画素を指す．IRW＝1 のときは画素値が IVAR に読み込まれ，IRW＝2 のときは IVAR の値が画素に書き出される．画像のデータ型が IPICT により判別できるので，このサブルーチンは整数・実数・複素数のいずれも処理することができるようになっている．また，画素ごとに頻繁に呼び出されるサブルーチンであるから，時間短縮のために，アセンブリ言語で書かれている．たとえば，サイズ 64×64 の整数画像の全画素をアクセスするのにかかる時間は1秒以下である．

3) 終了宣言．仮想配列のすべての操作が終了すると，バッファに残っている頁をディスクへ書き出すために，終了宣言が必要である．これは手続き

CALL STBF (IPICT)

によって行われる．IPICT は前と同様である．

以上のように，本システムの仮想配列サブルーチンは三つしかなく，また，引数の数も少ない．したがって，プログラミング上は主記憶の配列要素をアクセスするのとほとんど同じ感覚で用いることができる．

ところで，システム利用時にはデータ型に応じて異なる数の画面から成る画像がいくつも使われるので，その保守が問題となる．このために，各画面はそのヘッダ部に状態記録を保存している．すなわち，ヘッダ部にはその画面を含む画像のデータ型（未定義，整数，実数，複素数），実数・複素画像の場合はその先頭画面からの相対位置，書き込み禁止フラグ，その画像に適用されてきたコマンドの数，および，それらのコマンド名が記録される．これらのヘッダ情報は，たとえば仮想配列サブルーチン VADD でつぎのように用いられる．結果画像として書き込み禁止フラグがセットされていない画面だけを含む領域が宣言されると，それらの画面ヘッダは図1のようにすべて初期状態にクリアされる．逆に，いずれかの画面が書き込み禁止になっていれば，サブルーチン VADD はエラーリターンを生ずる．また，各セグメントプログラムは整数画像，実数画像，そしてもし可能なら複素画像も処理するように書かれており，その手続きの選択はヘッダ部の画像データ型記録を仮想配列制御ブロック IPICT を介して参照することにより行われる．したがって，画像のデータ型にかかわらず同一のコマンドを用いることができる．さらに，各画像の履歴を利用者がいちいち覚えなくてもよいので，画像数が多いときでも気楽に使える．

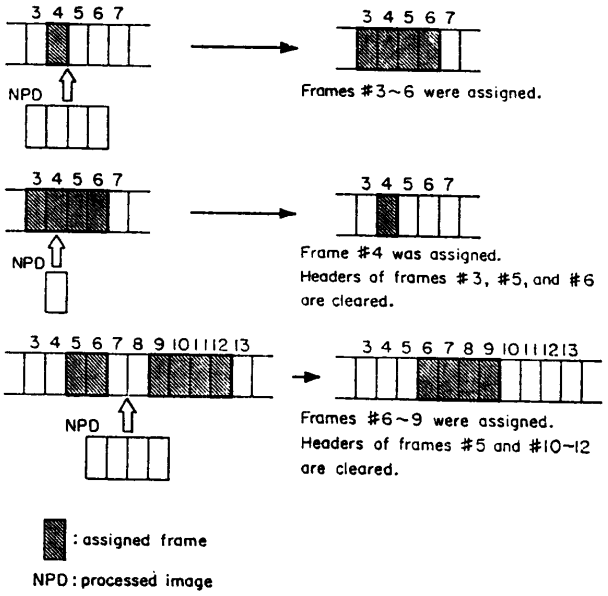

図1 画面の管理（左：割り当て前，右：割り当て後）

Fig. 1 Management of the frames. One frame is assigned to an integer image; two and four consecutive frames are assigned to real and complex images, respectively. Left: before assignment, Right: after assignment.

3.1.3 コマンド列の制御

本システムではコマンド列をマクロコマンドとして定義することができる．マクロコマンドは一般のコマ

ンドと同様にして用いられる．マクロコマンドを別のマクロコマンドに含めることもできる．

マクロコマンドによる処理に柔軟性をもたせるために補助コマンドを設けている．？？というコマンドはマクロコマンドの実行を一時的に停止させる．この状態において，利用者はコマンドを任意に与えて実行させることができる．これは特定の種類の画像を対象とした自律的な応用システムを組み上げていくのに有用である．すなわち，一連の処理手順のうち，確立している部分はそのままマクロコマンドにする；もし，適切な処理が未確定の部分があれば，そこにはコマンド？？を一時的に与えておく；開発の進展につれて処理が確定してくると，？？の部分を置き換えていくというようにである．一時停止状態で処理の流れを変えることもできる．これについてはつぎの小節で述べる．繰り返しのための LOOP，NEXT という補助コマンドもある．繰り返しの回数は実行前に指定するだけでなく，実行時に毎回繰り返し続行か中止かを利用者に指示させることもできる．したがって，利用者が中間結果を見ながら，繰り返しの中止時点を決定するということができる．また，繰り返しループの入れ子のプログラムは，制御パラメタ値として，繰り返し変数値またはその一次関数値を受け取ることもできるようになっている．したがって，制御パラメタ値をつぎつぎに変更しながら処理し，その結果を異なった結果画像として残していくというような操作も自律的に行わせることができる．

3.1.4 マンマシン対話管理

利用者がシステムを気楽に使えるために対話方式の多様性は重要な要素である．本システムではコマンドの制御パラメタ値やコマンド列の指定などのために表2のシステムコマンドが設けられている．

各コマンドの制御パラメタ値は促進メッセージへの応答として入力するので，利用者は入力項目やコマンドシンタクスを覚えなくてもよい．処理の柔軟性のために制御パラメタはなるべく多く設けるのを原則としている．この原則の副作用は，同じコマンドやマクロコマンドを何度も同じパラメタ値を与えて実行させる場合に，入力が煩わしく，時間がかかり，間違いやすくなることである．この頻繁な入力操作をなくすために，実行前に，選択的に制御パラメタ値の固定ができるようになっている．すなわち，実行時には，固定されていないパラメタ値だけを入力すればよい．さらに，固定されたパラメタの入力促進メッセージと値をコンソールに表示させることもできる．これは処理の進行状況の把握に役立つ．

表2 システムコマンド
Table 2 System commands.

Name	Meaning
ABORT	Abort the macro command sequence.
SKIP	Skip to another command.
LOOP-NEXT	Define an iteration loop (simple loop only).
RELEAS	Release the holding state of the macro command.
??	Hold execution of the macro command sequence.
DELETE	Delete a macro command.
EDITOR	Edit a macro command.
MACRO	Retrieve the macro command information.
INIT	Initialize the system: image size, console, and headers.
MAP	Report on the defined images: type and protection.
PICT	Report on the image: type, protection, and history. Change the protection flag state.
REPORT	Report the system state: image size and working file allocation.
0-SIZE	Change the size of the image #0, the variable size image.
MENU	Retrieve the command information.
MODE	Change the system mode.
STOP	Stop the system. Save the current state on the file.
SYSTEM	Show or change the system control data.
SWAP	Swap with the user program.
TYPE	Type the system log.
TIME	Show the time upon completion of each command.

パラメタ値の固定と入力促進の抑制はパラメタファイルによって制御される．これは各パラメタの値と状態の表である．パラメタの状態は「未固定」，「固定」，「実行時に繰り返し制御変数の値により決定」の3通りある．実行時のパラメタ値入力は未固定パラメタに対してのみ行う．

一つのコマンドに対して9通りまでのパラメタセットの指定ができ（パラメタファイルセグメントと呼ぶ），それらはコマンド名の後にピリオドで区切って数字を付加することにより識別される．一つのパラメタファイルセグメントは16個までのパラメタを制御できる．パラメタ値の指定をプログラム中で記述するための入力管理サブルーチンが設けられており，その呼び出し手続きは，整数パラメタの場合，

CALL ICHK (N, IVAR, LL, UL, "MESSAGE")

また，実数パラメタの場合，

CALL RCHK (N, VAR, LL, UL, "MESSAGE")

である．引数Nはパラメタ識別番号である．実行時にはまず各パラメタの状態が調べられる．固定状態であればパラメタファイルから読み込んだ値が引数IVAR または VAR にセットされる．繰り返し制御変数によって与えられるべきであれば，システム制御データ領域からコピーされる．未固定であれば促進メッセージ "MESSAGE" をコンソールへ出力し，そこから入力値を読み込む．このとき，許容入力値の下限LL と上限 UL の間にはいっているかどうかが調べられ，許容範囲外ならば再入力が促進される．これらのサブルーチンを用いることにより，プログラム中のパラメタ値入力手続きの記述が著しく簡略化される．

処理が無限ループに陥ったり，暴走したりしないように，実行時のパラメタ値のチェックはかなり念入りに行われる．パラメタ値がすでに固定されていても，その妥当性がチェックされ，不適当な値ならば入力が促進される．固定パラメタ値が一度参照されると，システム制御データ領域に読み込まれているパラメタ状態は未固定に変更される（パラメタファイル自体の内容は変化しない）．パラメタ値から導かれる他の制御変数についても，その妥当性をチェックし，値が不適当ならばコンソール入力をするようにセグメントプログラムを書くことにしている．以上のような措置は，つぎのトラブル——パラメタ値固定後，初期化コマンドINIT によって画像サイズを変更したため，値が不適当になる；マクロコマンドの繰り返し制御変数値より決定した値を他のパラメタの許容上限・下限値として用いる場合に，固定値が不適当になる；個々のパラメタの値が妥当であっても，それらから計算される他の制御変数の値が不適当である——が生じたとき，パラメタ値を変更し，正常に処理を続けるために必要である．パラメタ値の固定は通常の実行モードでは行われず，コマンド MODE によってパラメタ記憶モードに変更してから行う．セグメントプログラムは，まずパラメタ値を決め，つぎにシステム制御ブロックのモード情報を調べ，もし実行モードならば処理を実行するというように，一定の形式で書くようにしているので，パラメタ記憶モードならば処理は省略され，対話時間が短縮される．

マクロコマンドの組み立てはコマンド EDITOR により起動されるセグメントプログラムが管理する．各コマンドはステートメントと見なされ，ステートメント番号が付けられる．コマンド列はステートメント番号の順序になっているものと解釈される．コマンドの挿入，削除，訂正は BASIC 言語と同様に行われる．EDITOR のサブコマンドを**表3**に示す．

表3 マクロコマンドエディタのサブコマンド

Table 3 Subcommands of the macro command editor.

Name	Meaning
H	Quit editing.
K	Clear the editing buffer.
P	Register the command sequence as a macro command.
R	Renumber the statements.
T	Type out the commands in an increasing order of the statement number.
Tn	Type out the command with the statement number n.
Tmn	Type out the commands between the statement numbers m and n.
Y	Read into the editing buffer the command sequence of a defined macro command, the system log, or the history of an image.

マクロコマンドを定義する簡単な方法は BASIC によるプログラミングと同様である．一方，処理の履歴からマクロコマンドを組み立てることも，つぎのようにできる．システムは256ステップ前までの適用されたコマンド名を記録しているので，これを EDITOR のバッファへ読み込んで編集することにより，マクロコマンドを定義することができる；また，作業ファイル上の各画像は適用されてきたコマンドの履歴をヘッダ部に記録しているので，これを EDITOR のバッファへ読み込み，上と同様に定義することもできる．これらは，実際の処理経験にもとづいて適切な処理手順を発見していく場合に有用な方法である．

マクロコマンドの中にコマンド？？があれば，そこで実行が一時停止され，**図2**のように，いろいろな制御が可能になる．ここで，マクロコマンドの実行レベルをつぎのように定義しよう：利用者が発したマクロコマンドはレベル1で実行される（利用者レベルは0）；また，レベル n で実行された他のマクロコマンドにより起動されたマクロコマンドの実行レベルは $n+1$ である．どのレベルにおいてもコマンド？？が与えられると，制御はレベル0に戻る．このとき，利用者は自由にコマンドを入力することができる．この一時停止状態はコマンド RELEAS によって解除され，制御は元のレベル（図2ではレベル2）へ戻り，マクロコマンドの実行が続けられる．コマンド SKIP によって任意のマクロコマンドへ制御を移したり，コ

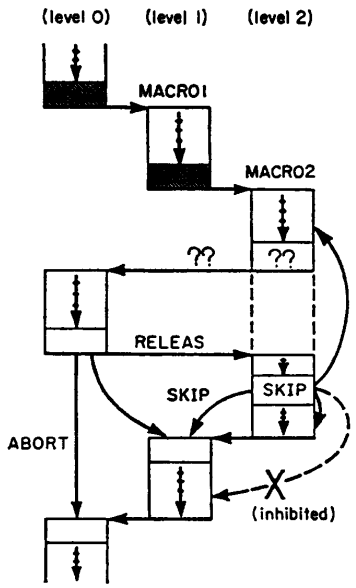

図 2 マクロコマンドの制御
Fig. 2 Control of macro commands.

マンド ABORT によってマクロコマンドの実行を中止することもできる．コマンド SKIP はマクロコマンド作成の補助コマンドとしても用いられる．

マクロコマンド自体にはパラメタファイルセグメントは対応付けられていない．マクロコマンド中の基本コマンドのパラメタファイルセグメントは，パラメタ記憶モードで各コマンドやマクロコマンドを入力することにより定義される．

3.2 ユーティリティ

プログラム開発やファイル管理はオペレーティングシステムの支援に頼っているが，そのほかに，本システムでは以下のサービスが提供される．

1) マクロコマンドの編集．これはコマンド EDITOR により行われる．機能については前述した．

2) コマンド情報の検索．コマンドに関する情報はメニューファイルと呼ばれる 169 ブロックから成るファイルに書き込まれている．このファイルの内容は，機能表(項目例：画像入出力，画像生成，変換，強調，復元，分割，統計など)，操作表（項目例：点操作，近傍操作，並列的，逐次的，論理的，算術的など)，整列表，コマンド台帳，コマンドごとのコメントなどである．整列表というのはコマンド台帳の各コマンドへのポインタの表で，各ポインタはコマンド名が辞書式順序になるように並べられている．コマンド台帳には，コマンド名，機能（複数項目可)，操作（複数項目可)，コメント記述部へのポインタが含まれている．利用者は機能や操作やコマンド名の頭文字の範囲を指定してコマンド名を一覧したり，特定のコマンドを指定してコメントを検索することができる．

このファイルの作成と更新のための補助プログラムがシステムとは別に用意されている．作成時には，ファイル領域の確保，機能表や操作表の項目の登録が行われる．一般に画像処理アルゴリズムの分類には多義性があるので，各表の項目は変更可能，また，各項目も択一ではなく，該当するか否かを記入するようになっている．更新時には，コマンドの追加，削除ができる．登録されているコマンドの記述，名前の表示も行う．このような特徴は利用者が本システムを自分用に作り変えていく場合の助けになる．

定義済みのマクロコマンドについても，コマンド MACRO により，名前やコメントを検索できる．

3) 標準フォーマット画像への対応．情報処理学会の標準フォーマット[10]で書かれた磁気テープデータのヘッダ情報検索，読み込み，および，このフォーマットによる磁気テープへの書き出しが可能である．ただし，書き出しフォーマットは 1 種類に限定した．また，読み込みも，フレームの一辺とサブフレームの一辺の長さが一致するものに限定した．これでも，標準画像データベース SIDBA[11]のデータは大体処理できる．

4. むすび

本稿では，利用者の多様な要求を満足する汎用対話型画像処理システムが，きわめて小規模のハードウェア環境のもとでも可能なことを，設計例によって示した．画像処理用アルゴリズムとしては，開発時に汎用のもの約 90 個を組み込んだが（付録 A. 1)，増設は容易である．本システムの利用経験から，使い勝手はかなり良く，とくに，画像処理の入門者にも取り付きやすく，また，画像処理実験を計算機が自律的に実行できること（付録 A. 2) や，利用者プログラムの作成と組み込みが容易なことなどが，利用した学生などには好評であった．ただ，ハードウェア資源をかなり無理して使っているので，たとえばソースプログラムから実行モジュールにいたるまで，および作業ファイル等等をディスクに一度に共存させられないとか，大きな表を主記憶上に作る余裕がないので画像処理アルゴリズムや実行効率が制限されるという問題もあった．これについては，最近のハードウェア環境が格段によくなっているので，主記憶・ディスク容量ともにもっと余裕のある使い方をすべきであると考えている．

画像処理技術の発達により，従来のように多額の設備投資が必要な専用高速プロセッサー辺倒ではなく，たとえば，研究の規模に応じ，実行効率をある程度度外視してそれなりに画像処理をしたいという要求に対しても顧慮すべき時期になった．本システムの経験がそのようなシステム設計の一助になれば幸いである．

参考文献

1) 長尾 真：画像処理用インテリジェントターミナルについて，ミニコンピュータの画像処理応用セミナーテキスト，日本自動制御協会，pp. 35-52 (1974).
2) 坂井利之他：ディジタル画像情報の会話型処理システム，情報処理，Vol. 15, No. 12, pp. 940-947 (1974).
3) 尾上守夫，柴田義文：SYstem64，信学技報，IE 74-60 (1974).
4) 麻田治男他：会話型画像処理システム—TOSPICS—，信学技報，PRL 75-51 (1975).
5) 谷内田正彦他：ミニコンを用いた対話形動画像処理システム，信学論，Vol. J61-D, No. 10, pp. 775-782 (1978).
6) 高木幹雄，坂上勝彦：ミニコンピュータによる対話形画像処理ソフトウェアシステム，信学技報，IE 77-63 (1977).
7) Haralick, R. M. and Mindon, G.: KANDIDA-TS: An Interactive Image Processing System, *Comput. Gr. Image Process.*, Vol. 8, No. 1, pp. 1-15 (1978).
8) 木村雄太郎他：小規模ハードウェアで実現する会話形汎用画像処理システム，信学技報，PRL 78-84 (1979).
9) 田村秀行他：SPIDER—移植性の高い画像処理ソフトウェア・パッケージ，電総研彙報，Vol. 44, Nos. 7 & 8, pp. 413-433 (1980).
10) 尾上守夫他：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).
11) Onoe, M. Sakauchi, M. and Inamoto, Y.: SIDBA Standard Image Data Base, MIPC Report, 79-1, Univ. Tokyo (1979).
12) 福島重広，相馬敬司：胃X線画像解析システムの一構成法，医用電子と生体工学，Vol. 16, No. 3, pp. 198-204 (1978).

付録

A.1 画像処理プログラム

システム開発時には表 A.1 に示すような算術変換や論理変換などの汎用の画像処理プログラムを用意した．特定の対象画像向けにシステムを加工する場合，

表A.1 画像処理プログラム

Table A. 1 Image processing programs.

Class	Programs
Display:	photographic, perspective, profile, projection, …
Transfer:	input from ITV, transfer between image areas, translation, sampling, transposition, transfer from/to standard format MT, …
Generation:	uniform noise, constant, …
Orthogonal transform:	Fourier, Walsh, Haar, and slant.
Statistics:	moments, minimum, maximum, and histogram.
Enhancement:	gray-level normalization (linear, histogram-modification); logarithm, exponential, absolute, square root, complex conjugate; filters (ideal, Butterworth, exponential, trapezoidal); windows (Hanning, Hamming, Gaussian); thresholding, local average, local variance, smoothing, template matching, SSDA, correlation, differentials, …
Segmentation:	chain coding, skeletoning, thinning, labeling, distance transformation, segment extraction, …

```
COMMAND = D-THRS
 * DYNAMIC THRESHOLD *
PROCESS(1) OR ABORT(2) ? 1
SOURCE PICTURE = 1
DESTI. PICTURE = 2
IMAGE SEGMENT SIZE = 9
THRESHOLD AVERAGE = 60.
THRESHOLD COEFF. OF VARIAT. = .2
AVERAGE REPORT(1) OR NOT(2) ? 1
AVERAGE

 ...printing of list of average density
    for each subregion...

ST. DEVIAT. REPORT(1) OR NOT(2) ? 1
STANDARD DEVIATION

 ...printing of list of standard deviation
    of density for each subregion...

                    :
                    :
                    :
# THRESHOLD
COMMAND =
```

(a) Operation without parameter fixing.

```
COMMAND = MODE
 * PARAMETER DEFINITION MODE *
   MESSAGE(PARAMETER,FIX)
     FIX=0 : UNDETERMINED
         1 : FIXED
         2 : LOOP VALUE
#COMMAND = D-THRS.1
 * DYNAMIC THRESHOLD *
PROCESS(1) OR ABORT(2) ? (     0 , 0 )  1,1
SOURCE PICTURE = (     0 , 0 )  1,1
DESTI. PICTURE = (     0 , 0 )  2,1
IMAGE SEGMENT SIZE = (     0 , 0 )  9,1
THRESHOLD AVERAGE = (0.0000000     , 0 )  60.0,1
THRESHOLD COEFF. OF VARIAT. = (0.0000000     , 0 )  .2,1
AVERAGE REPORT(1) OR NOT(2) ? (     0 , 0 )  2,1
ST. DEVIAT. REPORT(1) OR NOT(2) ? (     0 , 0 )  2,1

                    :
                    :
                    :
#COMMAND = MODE.1
 * RUN MODE *
COMMAND =
```

(b) Operation for parameter fixing.

```
COMMAND = D-THRS.1
 * DYNAMIC THRESHOLD *
 # THRESHOLD
COMMAND =
```

(c) Operation by parameter fixing.

```
COMMAND = MSSG
 FIXED PARAMETERS AND CORRESPONDING MESSAGES
  WILL BE PRESENTED.
COMMAND = D-THRS.1
 * DYNAMIC THRESHOLD *
PROCESS(1) OR ABORT(2) ? (     1 , 1 )
SOURCE PICTURE = (     1 , 1 )
DESTI. PICTURE = (     2 , 1 )
IMAGE SEGMENT SIZE = (     9 , 1 )
THRESHOLD AVERAGE = (60.00000      , 1 )
THRESHOLD COEFF. OF VARIAT. = (0.2000000      , 1 )
AVERAGE REPORT(1) OR NOT(2) ? (     2 , 1 )
ST. DEVIAT. REPORT(1) OR NOT(2) ? (     2 , 1 )
                    :
                    :
                    :
 # THRESHOLD
COMMAND =
```

(d) Operation by parameter fixing and message printing.

図A.1 変動閾値処理の操作例（下線部は入力を表す）

Fig. A. 1 Operational examples of a dynamic thresholding. The underlines indicate input by the operator. This is extremely simplified by parameter fixing.

```
COMMAND = EDITOR
 * EDITOR *
# : 10 TIME
# : 20 INITMT
# : 30 READMT.1
# : 40 RELSMT
# : 50 D-THRS.1
# : 60 SMOCON.1
# : 70 FRAME.1
# : 80 LABL.1
# : 90 SGMT.1
# : 100 SMOCON.2
# : 110 I-BIN.1
# : 120 FRAME.2
# : 130 LABL.2
# : 140 SGMT.2
# : 150 I-BIN.2
# : 160 FRAME.3
# : 170 CONTOR.1
# : R
# : T
 ...printing of the macro program...
# : P
 * REGIST MACRO-COMMAND *
COMMAND NAME = SPARS
 COMMENT PLEASE.
A SYSTEM FOR PATTERN ANALYSIS....
COMMAND =
```

(a) Operation for editing a macro command.

```
COMMAND = SPARS
 REAL TIME CLOCK
  WILL BE PRESENTED.
16:56: 5
 * MTO INITIALIZED *
16:56: 6
 * LOAD FROM STANDARD FORMAT MT *
IMAGE NUMBER = 0
IMAGE TOO LARGE ---
SAMPLE(1) OR QUIT(2) ? 1
LEFT-TOP CORNER : I,J = 1,1
PITCH = 4
AGAIN(1) OR NOT(2) ? 2
16:56:43
 * MTO RELEASED *
16:56:44
 * DYNAMIC THRESHOLD *
 # THRESHOLD
16:57:56
 * SMOOTH THE (0-1) IMAGE
      :
   (no input operation here)
      :
17: 0:30
 * EXTRACT CONTOUR
OUTPUT FILE NAME = CONT.2
START POINT OF SEARCH =   38   4
17: 0:49
COMMAND =
```

(b) Operation with a macro command and by parameter fixing.

図A.2 マクロコマンドによる操作例（下線部は入力を表す）

Fig. A. 2 Operational examples of a macro command. The underlines indicate input by the operator.

知識にもとづく処理は利用者がプログラムを作成して組み込むようにしている．また，システムの規模が大きくならないように，不要なプログラムを削除することもある．このように，応用システム開発のための核を提供することが本システムの基本的な考え方である．

A.2 使 用 例

なるべくマクロコマンド定義とパラメタ値固定の機能を利用し，また，知識にもとづく処理についてはそのためのプログラムを作成することにより，胃X線画像解析手順[12]を組み込んだ．**図 A.1** は胃領域抽出のための変動閾値処理における種々の対話例を示す．この例では，統計量や中間結果の出力を制御パラメタによって指示するようにしており（同図(a)），このような出力が不要と決まっていれば，パラメタ値を固定し（同図(b)），対話量を最小にできる．たとえば，同図(c)ではコマンド名のみを入力している．同図(d)は実行中に固定値を表示させるようにした例である．**図 A.2** は画像入力から胃輪郭線抽出までをマクロコマンドとして定義した例を示す．パラメタ値固定も行っているので，実行時には磁気テープ上の画像データの番号と，輪郭線データの格納ファイル名だけを入力すればよい．

(昭和 59 年 10 月 5 日受付)
(昭和 60 年 2 月 21 日採録)