画像解釈言語 PILS†

森　英雄††　赤羽秀友††

画像解釈言語 PILS は，胃X線写真の認識と診断のアルゴリズムをスマートに記述するため開発した問題向き言語で，LSI 11/23 上に PASCAL を用いてインプリメントしてある．かなりの汎用性を備えているので，対象が胃X線像でなくても終局的にシルエット，すなわち白黒面画像で表わされるものならば何にでも適用できる，応用範囲の広い言語である．その特長は1）画像データを階層構造をなす6種類の画像要素で抽象的に表現する，2）画像要素の属性や関係をアクセスする 53 種類の画像演算子がある，3）画像要素の探索専用の文がある，4）画像要素の連想記憶や最大最小選択の機能を有する文がある，5）プログラマが専用の画像演算子を定義できる，こと等である．

1. まえがき

筆者らは10年この方胃 X 線写真自動スクリーニングの研究を続けてきた[1]．その内容は一口でいうと，名医の脳裏にある診断のプロセスを計算機へ移植する作業といえる．医師の診断プロセスは，胃X線像から所見（診断の因子となる特徴）を見つけ出す画像処理過程と，所見をもとに判別する診断過程から成ると考えられる．胃X線診断の場合，この所見は100から200 種類くらいあるといわれている．問題はこの所見抽出のアルゴリズムをいかに簡潔にかつわかりやすく記述するかということである．筆者らが以前アセンブラで書いたところ，3つの所見のプログラミングから検証までに1年を要し，でき上がったプログラムの改良発展も不可能に近かった．セマンテックな所見抽出のアルゴリズムをスマートに記述する画像処理用の言語が欲しいという切実な要求がここに生れた．

画像解釈言語 PILS (Picture Interpretation Language for Silhouette) は，こうした要求から生れた，画像の抽象的表現や探索機能，連想記憶機能，最大・最小選択機能等を有する問題向き言語である．

2. 画像データの表現

PILS では画像をマクロ輪郭とミクロ輪郭の 2段階で表現する．マクロ輪郭は対象の面積，高さ，幅，構造等のおおまかな特徴を表現するとともに，ミクロ輪郭のガイドとしての役割を果たす．ミクロ輪郭は対象の辺縁の細かな凹凸と濃淡勾配を表現する．**図1**は原画像(a)から抽出したマクロ輪郭(b)とミクロ輪郭(c)である．

2段階の表現を選んだのは，

（1） 人間も周辺視と中心視の2段構えで見てい

（a） 原画像
（a） Original image

（b） マクロ輪郭
（b） Macro-contour

（c） ミクロ輪郭
（c） Micro-contour

図1 マクロ輪郭とミクロ輪郭
Fig. 1 Macro and micro contour.

† PILS: Picture Interpretation Language for Silhouette by HIDEO MORI and HIDETOMO AKABANE (Department of Computer Science, Yamanashi University).

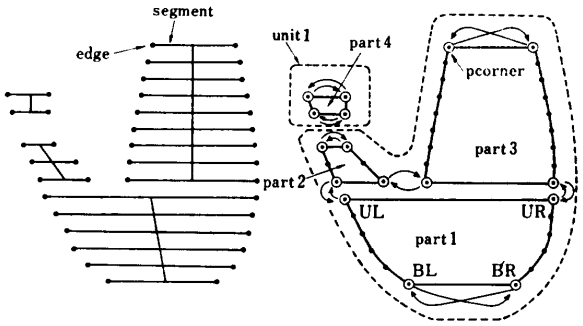
†† 山梨大学工学部計算機科学科

る[2].

(2) 全画面を細かく走査する必要がないので，メモリの節約と処理時間の短縮ができる[3]からである．

画像データの表現で重要なことは，画像をどのような単位に分割 (segmentation) するかということである．画像は構造化して，すなわち，幾つかの部分が組み合わさってできたものとして表現しなければならない．そのとき，画像をあまり小さな部分に分割し過ぎると，認識に用いる特徴は幾つかの部分にまたがってしまい抽出に不便だし，逆に分割し足りなくて大きな部分のままだと特徴はその中に埋もれてしまうので，結局もう一度分割し直さなくてはならなくなる．

PILS の画像データ構造をマクロ輪郭から説明する．全画面を等間隔の帯状領域 (zone) に分割する．帯状領域を粗く走査して対象の **edge** を抽出する．対象の左右の edge の対を **segment** という．中心と長さが連続的に変化する segment の集合を **part** という．part の底辺と上辺の4つの角をそれぞれ BL (bottom left), BR (bottom right), UL (upper left), UR (upper right) と名付ける．part と角の対を **pcorner** という．pcorner から **arrow** と呼ばれるポインタが出ており，隣接する part を連結する*．連結した part の集合を **unit** という．1つの原画像に含まれる unit の集合を **whole** という．edge, segment, …, whole を総称して画像要素という．**図2**は各画像要素の説明図で，**図3**はそのデータ構造図である．unit を構成する part の中で，図2の part 2, part 3 のように行止まりで上に連結するものがない part を頭 (head) といい，part 1 のように脚となる part を尾 (tail) という．part 4 は頭がありかつ尾である．



水平線分は segment を表わし，中心線で結ばれた segment の集合が part を表わす．

図 2 マクロ輪郭
Fig. 2 Macro-contour.

* segment を連結して part を生成する作業と，part を連結して unit を生成する作業は1 pass で実現しているが，ここでは論じない[4].

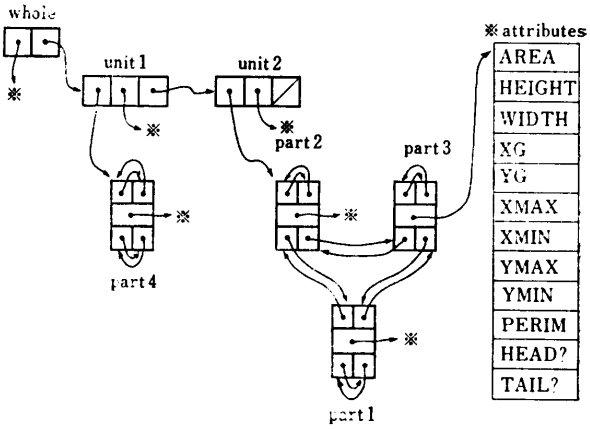
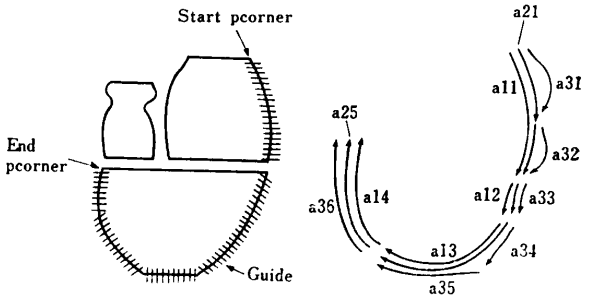


図 3 マクロ輪郭 (図2) のデータ構造
Fig. 3 Data structure of macro-contour.



(a) マクロ輪郭　(b) arc 列
(a) Macro-contour　(b) Arc sequences

図 4 ミクロ輪郭の表現
Fig. 4 Representation of micro-contour.

各画像要素は幾つかの属性で抽象化する．幾つかの属性とは，その画像要素の面積 (図3では AREA)，高さ (HEIGHT)，幅 (WIDTH)，重心座標 (XG, YG)，X座標の最大・最小値 (XMAX, XMIN)，Y座標の最大・最小値 (YMAX, YMIN)，周辺長 (PERIM, ただし whole と unit に対してのみ適用，頭か尾か (HEAD? TAIL?, ただし part に対してのみ適用) をいう．

ミクロ輪郭は，**図4**(a)のようにマクロ輪郭の edge の全体，またはその一部をガイド (plan ともいう) にして辺縁を密に走査して3本のミクロ輪郭線を抽出したものである．ミクロ輪郭線を3本で表わしたのは，一般に辺縁の濃淡勾配は場所によって異なり，ただ1本の曲線で表わすのは難しく，正確に表わすにはセマンテックな知識がいるからである．3本の輪郭線の混み具合から辺縁の濃淡勾配がわかる．各ミクロ輪郭線は図4(b)のように尖点や変曲点で分割し，向きのついた弧の列で表わす．この弧を arc という．arc はその始端の座標 (X0, Y0)，終端の座標 (X1, Y1)，始端から終端へ引いたベクトルの x 成分・y 成分 (DX,

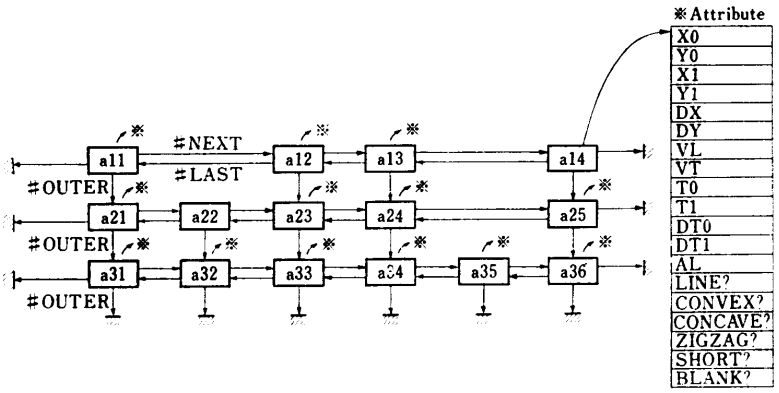


図 5 図4(b)に示すミクロ輪郭のデータ構造
Fig. 5 Data structure of the micro-contour showin in Fig. 4 (b).

DY), そのベクトルの長さ (VL), そのベクトルの方向 (VT), 始端および終端における接線ベクトルの方向 (T0, T1), 隣接する arc の接線ベクトルとの差分 (DT0, DT1), 弧の長さ (AL), 弧の形状 (LINE?, CONVEX?, CONCAVE?, ZIGZAG?, SHORT?, BLANK? のいずれか) 等の属性で抽象化する. **図5** にミクロ輪郭のデータ構造を表わす.

PILS は整数, 実数, 文字列, 論理 (TRUE か FALSE の値をとる) からなる算術データを扱える. 画像要素名や算術データの変数名は, 先頭が英字で始まる 10 字以内の文字列で, 末尾の2文字 (論理だけは1文字) でデータの型がわかるようにつけなければならない. **表1** にその規則を示す.

3. 画像演算子

次の言語設計上の問題点は, 画像要素の属性の呼出し (access) やポインタをたぐる操作を, いかにわかりやすく表現するかという問題である.

PILS では XPP という名の part の面積を XPP#AREA で表わす. #AREA を画像演算子という. 属性を求める画像演算子は, 図3や図5に示す属性の名前の前に # をつければ良い. ある画像要素に対して親子または兄弟関係にある画像要素を求める操作も画像演算子で表わす. その中の幾つかを **図6** に示す.

画像演算子を含む画像式の書き方は

〈画像要素名〉#〈画像演算子名〉{#〈画像演算子名〉}

である. この記法は日本人にとって大変わかりやすい. # を格助詞"の"で置き換えて読めば良いからである. たとえば, XPP#BL#ARROW#PART#AREA は part XPP の左下の pcorner とポインタ arrow で結ばれている part の面積を表わす.

表 1 画像要素と変数の名前のつけ方
Table 1 Naming rule of picture element and variable

画像タイプ	画像要素名
whole	ααWW
unit	ααUU
pcorner	ααCC
part	ααPP
edge	ααEE
arc	ααAA

変数タイプ	変数名
整数	ααII
文字列	ααSS
論理	αα?
実数	αααβ

αα: 先頭が英字の8字以内の英数字
ββ: WW, ···, AA II, ···, ? 以外の英数字

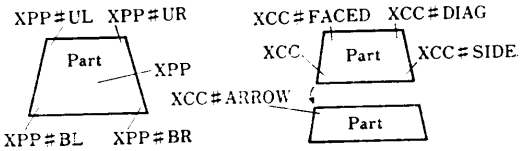
4. 画像入力文

次の問題点は文 (ステートメント) をいかに過不足なく用意するかということである.

PILS の標準画像入力装置は, FSS ビデオメモリ内蔵の TV カメラである. プログラマは生の画像データを配列の形で処理することはしない. 画像入力文を通して貰うデータは, すべて構造化した画像データ, すなわち, whole を頂点とするマクロ輪郭か, arc 列で表わされるミクロ輪郭である. 画像入力文は MACRO 文と GETARC 文の2つからなる.

MACRO 文はマクロ輪郭を抽出し構造化する. データとして最上位の画像要素である whole を引き渡すので, プログラマはそれに名前を与えねばならない. 文型は

MACRO 〈whole 型要素名〉
[AXIS 〈座標軸の回転, 平行移動のパラメータ〉]
[ZONE〈帯状領域内のサンプリングパラメータ〉]
[FILTER 〈フィルタのパラメータ〉]



XPP#BL は, part XPP の左下隅の pcorner を示す.
XCC#FACED は, pcorner XCC と上下方向に向かい合った pcorner を示す.

図 6 関係画像演算子
Fig. 6 Rerational picture operators.

[ORGANIZE〈構造化のパラメータ〉]

である．AXIS 以下のパラメータの全部または一部を省略しても良い．省略すると既定値が用いられる．

GETARC 文はマクロ輪郭の edge をガイドに3本のミクロ輪郭線を抽出し，それらを arc 列に変換する．3つの arc 列の最初の要素，図4では a_{11}, a_{21}, a_{31} がそれに相当する，を引き渡すので，プログラマはそれに名前をつけなければならない．文の型は

GETARC〈arc 型要素名〉{,〈arc 型要素名〉}

[BY〈ミクロ輪郭抽出と arc 列変換のパラメータ〉]

TRACE〈ガイドの出発点となる pcorner〉

TO〈ガイドの終点となる pcorner〉

である．BY で始まるパラメータは省略して良い．

5. 探索文

パターン認識で最も重要な操作の1つは探索 (searching) である．探索とは，〈探索範囲〉の画像要素の中から〈目的画像要素の条件〉を満たす画像要素を見つけ出すことである．これは画像要素を次々とたどって行く繰返しの手続きをつくり，その中に条件文を入れればできるが，決まりきっているわりには煩雑で，間違いやすい．スマートに表現する必要がある．

マクロ輪郭の探索はその探索の仕方によって2つに分けられる．1つは〈探索範囲〉の中の画像要素を余す所なく調べる仕方で，その探索の順序はどうでも良い．これを PARASCAN 文としてまとめる**．文の型は，

PARASCAN
SCOPE〈探索範囲を表わす画像要素の式〉
COMPONENT〈被探索の単位となる画像要素名〉
TARGET-1〈目的画像要素1が満たすべき条件式〉
[TARGET-2〈 〃 2 〃 〉]
[TARGET-1, 2〈目的画像要素1と2がともに満たすべき条件式〉]
1:〈目的画像要素1の発見時に実行する文〉
[2:〈 〃 2 〃 〉]
[1,2:〈目的画像要素1または2の発見時に実行する文〉]
PARAEND

である．目的画像要素は3つ以上あっても良い．whole に作用して unit を得たり，unit に作用して part を得たりする画像演算子は備えていないので，そうしたいときは，whole (unit) を SCOPE にし unit (part) を COMPONENT とする PARASCAN 文を用いて探索しなければならない．

もう1つの探索の仕方は，〈出発点となる画像要素〉と〈次にたどるべき画像要素を与える式〉を与えて，〈探索続行条件〉を満足している間，あるいは，〈探索打切り条件〉に到達するまで探索を続ける仕方である．これを SEQSCAN 文にまとめる．文の型は

SEQSCAN [−TRM]
FROM〈被探索画像要素名〉
:=〈出発点となる画像要素式〉
[WHILE〈探索続行条件式〉]
TARGET-1〈目的画像要素1の満たすべき条件式〉
:
1:〈目的画像要素1の発見時に実行する文〉
:
STEP〈被探索画像要素名〉
:=〈次の画像要素を与える式〉
[UNTIL〈探索打切り条件式〉]
SEQEND

SEQSCAN-TRM と書くと，目的画像要素が1つでも見つかれば探索を中止 (terminate) する．TARGET 文と実行文の書き方は PARASCAN 文と同じである．

ミクロ輪郭が作る arc 列の探索には SKIP 文を用いる．SKIP 文は〈探索の最初の arc〉を与え，その arc から正順方向に arc 列をたどって行き，〈目的条件〉を満たす arc が見つかったら，その arc に〈arc 型要素名〉をつけ，なければ〈arc 型要素名〉の値は EMPTY にして〈ラベル〉の指す文へ分岐する．SKIP 文の型は

SKIP〈arc 型要素名〉
:=〈探索の最初の arc を与える式〉
TARGET〈目的画像要素が満たすべき条件式〉
[EXIT〈ラベル〉]

である．〈ラベル〉は整数である．

6. 連想記憶操作文

1つの探索文で見つかる目的画像要素は，ただ1つとは限らない．そこで見つけるそばから次々と記憶するためのメモリが必要になる．このメモリは，画像要

** PARASCAN は parallel scanning, SEQSCAN は sequential scanning の略で，心理学者の Sternberg が反応時間を指標にした一時記憶の呼出しの実験結果を説明するのに用いた用語である．実際には PARASCAN を sequential 走査に変換して実行するが，機能的には parallel に走査したのと同一なのでこのように名づけた．

素とそれにつけた名前の対，または，画像要素とその評価値の対を記憶しておき，対の一方を指定して他の一方を読出すという連想記憶方式を採ることにする．プログラムのわかりやすさとメモリの利用効率を考えたからである．さらに，ある条件を満たす画像要素を捜し出すメモリ探索機能と，評価値の大きい順，または，小さい順に画像要素を読み出す最大，または，最小選択器の機能をつけ加える．この機能は，評価値として標準パターンとその画像要素間の類似の程度を表わす距離をとり，その値の小さい順に標準パターンをとれば，カテゴリ分類ができて便利である．

メモリは 10 組あり MM-0, MM-1, …, MM-9 なる名前がついている．1つのメモリに色々な型の画像要素と評価値の対を混ぜてしまうことができる．評価値は整数か実数，または，10 字以内の文字列である．

メモリは使用に先立って CLEAR 文で帰零しなければならない．1つの CLEAR 文で複数のメモリを帰零できる．文の型は次の通りである．

CLEAR 〈メモリ名〉{,〈メモリ名〉}

メモリへ画像要素と評価値の対をしまうのには PUT 文を用いる．文の型は

PUT 〈画像要素を表わす式〉, 〈評価値を表わす式〉
INTO 〈メモリ名〉

である．同一画像要素を同一メモリに何度 PUT しても良い．メモリにしまわれるのは画像要素1個分だけで，その評価値は整数型と実数型に限り加算される．これは便利な機能で，たとえばプログラム例3のように，unit にドーナツ状の孔が空いているかどうか調べるのにも役立つ．

メモリから評価値を指定して対する画像要素を取り出すときは TAKEPIC 文を用いる．文の型は

TAKEPIC 〈取り出した画像要素をしまう画像要素名〉
〈取り出すべき画像要素の評価値を与える式〉
FROM 〈メモリ名〉
[EXIT 〈求める画像要素がないとき行くラベル〉]

である．EXIT は省略できる．

メモリから評価値の大きい順に画像要素を取り出したいときは TAKEMAX 文を，小さい順に取り出したいときは TAKEMIN 文を用いる．その文の型は

TAKEMAX (または TAKEMIN)
〈取り出した画像要素をしまう画像要素名〉
[〈取り出した評価値をしまう変数名〉]
FROM 〈メモリ名〉
[EXIT 〈求める画像要素がないとき行くラベル〉]

TAKE 文は一種の探索文で，TARGET で指定した条件を満たす画像要素と評価値の対をメモリの中から捜し出す．文の型は次の通りである．

TAKE 〈取り出した画像要素をしまう画像要素名〉
[〈取り出した評価値をしまう変数名〉]
[TARGET 〈求める画像要素と評価値の対が満たすべき条件式〉
FROM 〈メモリ名〉
[EXIT 〈求める画像要素がないとき行くラベル〉]

7. 画像データの構造の変更

画像入力した後で画像データを操作して，1つの画像要素を2つ以上の要素に分割し，そのデータ構造を変更することはできるが，原画像にない画像要素を生成したり，逆に実在する画像要素を消去したりすることはできない．このような操作は上位のセマンテックな解釈レベルの話であると考え，画像の解釈を支える画像データはあくまで事実に忠実でなければならないと考えたからである．

マクロ輪郭の分割は CUT 文と REORG 文の2段階で行う．CUT 文はマクロ輪郭を分割する線を定義するだけで，分割の実行とマクロ輪郭の再構造化は REORG 文で行う．文の型は

CUT 〈分割線の始点座標〉, 〈分割線の終点座標〉
REORG 〈whole 型要素名〉
BY 〈構造化のパラメータ〉

ミクロ輪郭線を arc に分割する仕方は何通りもあり，どれを用いるかは，どのような特徴を抽出しようとするかによって決まる．したがって，ミクロ輪郭線に何種類かの性質の異なる特徴が存在する可能性があるときは，分割の仕方を何通りか変えて特徴抽出を試みなければならない．

REARC 文は，ミクロ輪郭を表わす arc 列をパラメータを変えて再度分割し直し，新しい arc 列に作り直す．文の型は

REARC 〈ミクロ輪郭の最初の arc を指す要素名〉
BY 〈arc 列変換のパラメータ〉

8. PILS プログラム例

今までの説明の範囲内で PILS の簡単なプログラムを作って見よう．

PILS のプログラムは MEASURE で始まり MEND で終わる．文と文の区切りは改行またはスペースであ

る．探索文のように複雑な文は何行にもわたって書いて良い．indention は自由である．太字は予約語を示す．

例1　マクロ輪郭を読み取って胃を表わす unit STOMACHUU を見つけ，そのマクロ輪郭を XY プロッタに描くプログラムを書く．ただし，胃を表わす unit は面積が必ず 100 cm^2 より大であるものとする．

```
MEASURE                                    (1)
MACRO XWW                                  (2)
100: CLEAR MM-1                            (3)
PARASCAN                                   (4)
  SCOPE XWW                                (5)
  COMPONENT XUU                            (6)
  TARGET-1 XUU#AREA>100.0                  (7)
  1: PUT XUU, XUU#AREA INTO MM-1           (8)
  PARAEND                                  (9)
TAKEMAX STOMACHUU FROM MM-1                (10)
PLOT STOMACHUU                             (11)
```

例2　胃を表わす unit STOMACHUU が与えられたとき，胃角を表わす pcorner WCC を見つけ，その位置に文字 W を描き，WCC が見つからなかったらラベル 900 の文へ分岐するプログラムを作る．ただし，WCC は重心が STOMACHUU の重心より右上にありかつ最も近い part の左下隅の pcorner と定義する．

```
CLEAR MM-1                                 (12)
PARASCAN                                   (13)
  SCOPE STOMACHUU                          (14)
  COMPONENT XPP                            (15)
  TARGET-1 XPP#XG>STOMACHUU#XG             (16)
  TARGET-1 XPP#YG>STOMACHUU#YG             (17)
  1: D=SQRT(SQR(STOMACHUU#XG
       -XPP#XG)+SQR(STOMACHUU#YG
       -XPP#YG))                           (18)
    PUT XPP#BL, D INTO MM-1                (19)
  PARAEND                                  (20)
TAKEMAX WCC FROM MM-1 EXIT 900             (21)
PENUP (WCC#X, WCC#Y)                       (22)
PLOT 'W'                                   (23)
```

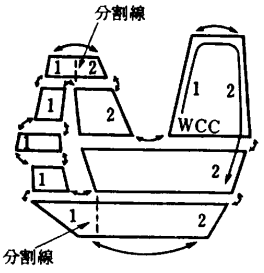


図 7　孔を有するマクロ輪郭
Fig. 7　A macro-contour having a hole in its body.

(18) 行目の SQRT と SQR はそれぞれ，平方根と平方を求める標準関数である．(22)行目の PENUP 文は，XY プロッタをペンアップしたままで WCC の位置まで移動させる文である．

例3　図1(b)のマクロ輪郭のように，十二指腸像がその上部と下部で胃像と重複して孔のあいたマクロ輪郭をつくる場合がある．これらをマクロ輪郭レベルで切り離すプログラムを作る．その方法は**図7**のように WCC を出発点にして pcorner を1つ置きにたどってマクロ輪郭を一周する．part の左側の pcorner を通ったときは，その part に1点を与え，右側を通ったときは2点を与える．孔の周りの part の中で孔の左側に位置する part は1点で，右側に位置するのは2点，その他の part は3点となる．2点の part 群の中の最上部に位置する part の左上隅と，最下位に位置する part の左下隅でマクロ輪郭を分割する．

```
CLEAR MM-1, MM-2                           (24)
SEQSCAN                                    (25)
  FROM XCC=WCC                             (26)
  TARGET-1 (XCC#BL?) OR (XCC#UL?)          (27)
  TARGET-2 (XCC#BR?) OR (XCC#UR?)          (28)
  1: PUT XCC#PART, 1 INTO MM-1             (29)
  2: PUT XCC#PART, 2 INTO MM-1             (30)
  STEP XCC=XCC#FACED#ARROW                 (31)
  UNTIL XCC=WCC                            (32)
  SEQEND                                   (33)
200: TAKEPIC XPP, 2 FROM MM-1 EXIT
     300                                   (34)
PUT XPP, XPP#YG INTO MM-2                  (35)
GO 200                                     (36)
300: TAKEMAX TOPP, Y FROM MM-2
     EXIT 400                              (37)
```

```
X0EE:=TOPP#UL#EDGE                                    (38)
CUT X0EE, (X0EE#X,
  STOMACHUU#YMAX)                                     (39)
PUT TOPP, Y INTO MM-2                                 (40)
TAKEMIN BOTTOMPP FROM MM-2                            (41)
X0EE:=BOTTOMPP#BL#EDGE                                (42)
CUT X0EE, (X0EE#X, STOMACHUU#YMIN)                    (43)
REORG XWW                                             (44)
GO 100                                                (45)
400: FEED                                             (46)
```

(25)～(33)行目のプログラムは part と評価値の対をメモリ MM-1 にしまう．図7に示すように左右両側をトレースした part は，評価値が 1+2=3 になる．(27)行の #BL? は pcorner に作用して，それが BL に位置するとき TRUE を，位置しないとき FALSE を返す画像演算子である．(34)～(43)のプログラムで図7に示す分割線を決める．(44)行目の REORG 文を実行すると，マクロ輪郭の構造がすっかり変わってしまい，そのときまで使っていた画像要素名，この場合は STOMACHUU と WCC, が指す画像要素はもはや存在しなくなる．そこで，もう一度最初に戻って STOMACHUU の探索からやり直す．STOMACHUU に孔がなかったら 400 へ分岐する．FEED 文は XY プロッタの紙を1画面分だけ送る．

例4 WCC からその真上の pcorner に至るマクロ輪郭をガイドにして辺縁を走査し，3本のミクロ輪郭線を抽出し，内側のミクロ輪郭線の先頭の arc に WAA という名をつける．内側ミクロ輪郭線に辺縁不整の所見を有する arc があるとき，そこに文字Zを描き，直線化の所見があったらそこに文字Lを描くプログラムを作る．ただし，辺縁不整はジグザグ状 arc の存在で表わし，直線化は両端の接線ベクトルが急激に変化する直線状 arc の存在で表わすとする．

```
GETARC WAA TRACE WCC TO
  WCC#FACED                                           (47)
PLOT WAA                                              (48)
SKIP XAA:=WAA TARGET XAA#ZIGZAG?
  EXIT 500                                            (49)
XSS:='Z'                                              (50)
500: SKIP XAA:=WAA TO (XAA#LINE?)
       AND (XAA#DT0>8) AND
       (XAA#DT1>8) EXIT 999                           (51)
XSS:='L'                                              (52)
PENUP ((XAA#X0+XAA#X1)/2.0,
       (XAA#Y0+XAA#Y1)/2.0))                          (53)
PLOT XSS                                              (54)
999: MEND                                             (55)
```

9. プログラマによる画像演算子の定義

#AREA や #HEIGHT 等の画像演算子は，PILS コンパイラに最初から組み込まれた基本的なもので，対象が何であっても適用できる汎用の画像演算子である．この他に，ある対象にのみ適用できるような専用の画像演算子を定義できる．これによってアルゴリズムの記述はより簡潔になる．たとえば，例1のアルゴリズムを使って，whole に作用して胃を表わす unit を得る画像演算子 #STOMACH とか，例2のアルゴリズムを使って，unit に作用して胃角を表わす pcorner を見つける画像演算子 #WINKEL とかが考えられる．

使用者が定義する画像演算子は PILS プログラムの文頭で宣言する．その宣言の仕方は

OPDEF ⟨結果を表わす引数名⟩=⟨オペランドを表わす引数名⟩ #⟨新画像演算子名⟩

[GLOBAL ⟨名前⟩{,⟨名前⟩}]
⟨PILS で記述した画像演算の本体部⟩
OPEND

結果を表わす引数名とオペランドを表わす引数名は，画像要素名や変数名と同様に，名前の末尾の2文字で引数の型を示す．

GLOBAL 文で宣言する名前は画像要素名でも変数名でも良い．ここで宣言した名前は，OPDEF の宣言の外，すなわち実行文中でも呼び出して使うことができる．OPDEF の本体部で使用するそれ以外の画像要素名や変数名はすべて local で，OPDEF の宣言の外では使えない．

10. ASSIGN 文

ASSIGN 文で定数名とその値を宣言することができる．定数の型は整数か実数で，名前の末尾の2文字で型を宣言する．PILS プログラムを RUN するとすぐ，ASSIGN 文で宣言した定数の名前と値のリストが CRT 端末に表示される．その時 CRT 端末から定数の宣言をやり直すことができる．その文の型は

ASSIGN ⟨定数名⟩=⟨定数⟩{,⟨定数名⟩=⟨定数⟩}

である．ASSIGN 文は PILS プログラムの文頭に書かなくてはならない．

パターン認識処理において会話型式で手続きの一部が変更できると便利な場合が多い．この場合変更の対象となるのはパラメータの値であり，アルゴリズムそのものである場合は少ない．PILS では ASSIGN 文を通して会話形式への道を開いている．

例5　8章例1のアルゴリズムを作って，wholeに作用して胃を表わす unit を求める画像演算子 #STOMACH を定義し(1)～(11)のプログラムを書き直す．

```
MEASURE                                  (1)
ASSING MINAR=100.0                       (2)
OPDEF XUU=XWW#STOMACH                    (3)
  GLOBAL MINAR                           (4)
  CLEAR MM-1                             (5)
  PARASCAN                               (6)
    SCOPE XWW                            (7)
    COMPONENT YUU                        (8)
    TARGET-1 YUU#AREA>MINAR              (9)
    1: PUT YUU, YUU#AREA INTO MM-1
                                         (10)
    PARAEND                              (11)
  TAKEMAX XUU FROM MM-1                  (12)
  OPEND                                  (13)
MACRO ZWW                                (14)
STOMACHUU=ZWW#STOMACH                    (15)
PLOT STOMACHUU                           (16)
MEND                                     (17)
```

11. PISL のインプリメント

PILS システムは LSI-11/23 上に OMSI PASCAL V-1 を用いてインプリメントした．コンパイラの大きさは 83 KB で，オーバレイをかけて 60 KB である．PILS コンパイラ部は PASCAL で書いてあるので移植が容易である．また，コンパイラの出力コードも PASCAL である．PILS プログラムの中に in line で PASCAL コードを書くことができる．したがって，PILS で記述できないような処理（たとえば濃淡画像処理）を PASCAL で書いて PILS プログラムと結びつけることができる．

画像入出力処理部はシステムサブルーチンとなっており，コンパイラとは独立しているので，画像入出力装置を変えるときは，システムサブルーチンを変えるだけで良い．

12. むすび

PILS は最初胃X線写真の認識の診断を目的として設計した問題向き言語であるが，かなりの汎用性を備えているので，対象が終局的にシルエット，すなわち白黒の面画像として表わされる画像ならば何にでも適用できる応用範囲の広い言語である．現在，胃X線診断の他に，手書き漢字認識，粒子の形状による分類と計数，植物の花や葉の形態分類，認知科学における3次元物体の認識モデル等に応用中あるいは応用を計画中である．

紙面の都合で PILS の文法を詳しく述べることができないが，詳しくは文献[5]を参照されたい．

コンパイラの製作にあたっては，移植性を考えてできるだけ標準 PASCAL で記述するように努めたが，separate compilation と EXIT 文等の若干の非標準機能の使用を免れることができなかった．また，ミニコンの 64 KB のメモリ容量に収まるように，コンパイラを製作しては文法を手直しするという手順を繰り返した．

言語設計は第一の著者が，コンパイラの設計製作は第二の著者が行った．システムサブルーチンの開発は大島昌彦君（当時本学学生）が，画像入出力装置の開発は二木弘技官が担当した．感謝の意を表する．

参考文献

1) 森，二木，中込，大関：計算機による胃X線立位充満像スクリーニングの試み，医用電子と生体工学，15巻7号，pp. 457-464 (1977)
2) 森：人はいかにものを認めるか，サイコロジー，No. 9, pp. 26-34 (1980).
3) Mori: Multi-contour detection of gastric X-ray image by organizing before identification, Proc. of 4th IJCPR, pp. 897-901 (1978).
4) 森，二木：胃X線輪郭抽出装置，公開特許番号 53-139862.
5) 森，赤羽，大島，神田：画像解釈言語 PILS の文法，山梨大学工学部研究報告 32 号，pp. 85-101 (1981).

(昭和56年8月21日受付)
(昭和56年11月18日採録)