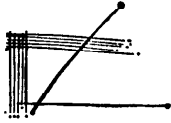


展 望



画像処理とソフトウェア技術†

棟 上 昭 男††

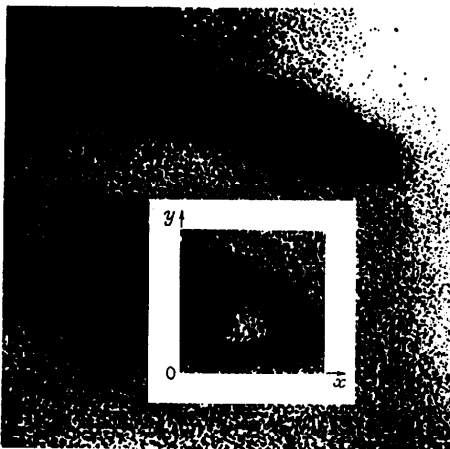
1. はじめに

画像情報の計算機による取扱いには元来困難な問題が数多く含まれていて、ソフトウェアという物の技術には結び付き難い面が多かった。しかしながら最近の計算機技術の進歩は、このような障害を少しずつ取り除きつつあり、プログラムやシステムとしての実体を伴った画像処理を考えることも可能となってきた。

図-1(b)は(a)のような濃淡画像を数値データとして計算機内に取り入れ、その変化の様子を示したものである。画像処理とは、このような配列の形で与えられる濃淡の数値データ（これを信号レベルのデータあるいは低レベル・データと呼ぶ）を別の形の数値データに変換したり、そのデータ全体とか、一部分に関する性質についての記述や測定値を求める操作、およびその逆操作のすべてを含んでいる。

表-1 はこのような画像処理の分野を入出力データ

の形式によって分類したものであるが¹⁾、計算機による画像処理には、このように信号の世界と記号の世界の一方または双方を取り扱う処理全体が含まれていて、様々なプログラム技術が必要とされる。この小論では、主に信号レベルの画像データに適切な変換を行って別の画像を求めたり、解析や認識を行ったりする処理についてのソフトウェア上の問題点について考えることにする。同様の問題については、異なった観点からこれまでもいくつかの解説が行われているので、これらもあわせて参考にされることをおすすめしたい^{2),3)}。



原画像データ
図-1 (a) (ドット密度で濃淡を表示)

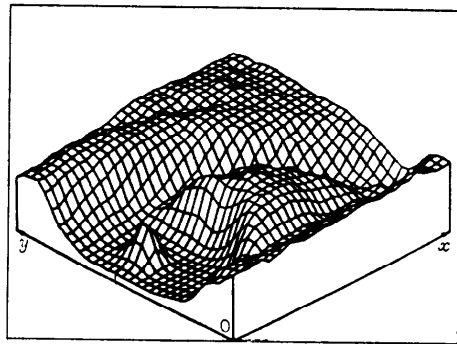


図-1 (b) 白枠内の濃淡の変化の様子

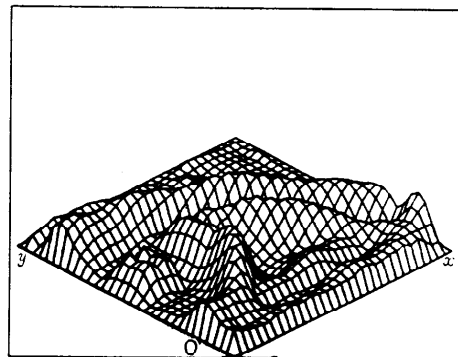


図-1 (c) 濃淡の変化の大きさ(明るさの勾配)

† Image Processing and Software Technology by Akio TOJO (Computer Science Division, Electrotechnical Laboratory).

†† 電子技術総合研究所ソフトウェア部情報システム研究室

表-1 画像処理の分野分類

種別	入出力データ	処理内容と例
変換	画像→画像	鮮明化, 強調, 雑音除去, 補正, 修正, 復元
解析 認識	画像→ $\left\{ \begin{array}{l} \text{数値} \\ \text{論理値} \\ \text{記述} \end{array} \right.$	認識・分類 (文字認識, 細胞分類など) 解析・記述 (情景解析, X線写真解析など)
表示	$\left. \begin{array}{l} \text{数値} \\ \text{モデル} \end{array} \right\} \rightarrow \text{画像}$	測定データ表示 (多次元スペクトル, 断層写真像など), 可視化 シミュレーション・計算結果 (構造物設計, 図柄デザインなど)

2. 画像処理の特徴

もう一度図-1(b)を見てみよう。このグラフの上方向の値は原画像の各標本点の明るさに対応している。通常の計算機処理では、その精度は8 bits 以内であることが多い。これは通常許容できる時間内で画像データを数値化しようとする場合に達成できる有効桁数が各種の雑音によって高々6~8 bits に押えられること、1つの画素を計算機内の処理単位 Byte に対応させる方が都合が良いこと、一つの視野内での人間の視覚の明るさに対する分解能が6 bits 程度を越えないと思われること、などをその理由としている。

一方空間的な分解能について考えてみると、図-1(b)の場合は 32×32 の画素から構成されているが、このように少ない画素数で目的とする処理を行うことができるのはまれで、少なくとも 128×128 以上、場合によっては 2000^2 から 4000^2 に達する画素を扱わねばならないこともある。空間的な分解能は本来処理目的に応じて最適の値に設定されるべきものであるが、解析とか認識といった処理においても、まだそのための理論が確立されているわけではなく、許される範囲内で大き目の分解能で標準化されたデータを出発点に処理アルゴリズムに関する実験等を行っているのが現状である。

したがって比較的控え目に 256×256 程度の画像を処理する場合でも、処理のある段階までは原データ、中間結果、最終結果等のデータのための少なくとも $100 \sim 200$ kBytes 程度の領域が必要となり、さらに色彩情報も扱う場合には、この数倍のデータ領域が必要となる。分解能を上げれば、その2乗に比例して必要なデータ領域が大きくなるのはもちろんである。

信号レベルの画像処理では、このように大量のデータをどのように管理し能率良く処理してゆかが重要なポイントとなり、ソフトウェア的にも様々な工夫が

行われている。また原データには多くの不要情報が含まれるので、できるだけ早い段階で必要な情報を抽出し、情報の圧縮をはかることも重要である。

図-1(c)は(b)に示すデータの変化の度合を表示したもので、画像中のコントラストの強さを示している。その尾根に相当する部分は画像中の輪郭とか陰影の縁に対応しており、画像の重要な特徴情報の一つである。情報を圧縮する一つの方法は、このような特徴の位置、方向、強度などによって原データを近似することであるが、実はこのような特徴の抽出は、原データの正確な解析や認識の結果はじめて可能になるという性格のものであることが多く、信号データから記号レベルへの一方向的な処理の流れに沿って情報の圧縮を行ってゆくことは必ずしも可能ではない。本来は画像の認識とか理解といった処理そのものが、高度の情報圧縮過程であると考えられるべきものであるから、画像処理の対象や目的によっては、原データやそれに近いデータを最後まで保持しなければならない場合も少なくない。

記号レベルの処理では、信号データから得られる特徴とその属性に関する測定値、相互の関係、判定を行うための何らかの形の辞書といったものを扱う。これらは多くの場合リストやグラフの形で表現されていて、配列構造の信号データの特定の部分や、それらの間の関係に対応している。したがって記号レベルの画像処理は記号処理の側面のほかに、多くの場合信号処理的側面も合せ持つことになり、広い範囲のデータ型を能率良く扱える機能が必要とされる。

3. 信号レベル・データの表現と処理

上述のように信号レベルの画像処理では、ほとんどの場合大量のデータ領域が必要とされる。このように大きな領域を主記憶上に割当てるとは不可能なことが多いから、信号レベルの処理では何らかの方法で外部記憶を利用することを前提としなくてはならない。

もともと仮想配列機能を持ったシステムや、仮想記憶システムを用いる場合には見かけ上このような問題はないが、そのかわり配列の割当て方向や、ページの大きさなどに注意を払わないと、処理中の画像の走査方向などによっては非常に能率の悪くなる場合があるのは、ほかの処理の場合と同様である。写真データのような静止画像の場合には必要な画素をその都度読み込み、写真を読み出し専用の記憶媒体のように利用することも考えられるが、読み込みの速度や安定度などの点で

問題が多く、特別な場合以外はあまり用いられていない。むしろ後述のように画面の中で詳細に調べるべき部分を何らかの経済的な方法で早目に見付け出し、そのような小領域のみ高い分解能で入力して処理する方式の方が現実的であろう。

さてこのような画像データを計算機上に格納し、処理するにはどのような方法が考えられるであろうか。

3.1 ラスタ・バッファ切替方式

信号レベルの画像処理の多くは、各画素自身、またはその近傍画素の値のみで結果の定まる近傍演算が多い。そのような処理では各時点で、処理を行っている画素の周囲のデータが常に保持できるようにすればよい。たとえば各画素の上下左右といった最近傍画素に対して一定係数のマスク処理を行う演算（先の図-1(c)もこのような処理結果の一例である）は、図-2(a)のように 3×3 の係数行列より成る窓で画面を走査しながら、その順に出力を求めてゆく処理であると考えられるが、1走査線分のデータ（以下ラスタ・データを外部記憶との入出力転送のブロック長（または

その整数倍）に対応させることによって、図-2(b)のように最小4ラスタ分のバッファを持てば処理を行うことができる。

この場合、1ラスタ分の処理が終わるごとに、バッファおよび被演算ラスタへのポインタを切替えることによって、物理的なデータ移動を避けることが可能となる。また実際には処理と外部記憶に対するデータ転送を並行させることによって処理速度の向上をはかることができるが、この場合には点線で示すようにさらに2ラスタ分のバッファを追加すればよい。

さて上の例では与えられた画像に対して1種類の演算をほどこす場合を考えたが、実際には平滑化処理を行った結果に対して縁検出を行うといった具合に、何種類かの演算をカスケードに実行する必要があることが多い。そのような場合には、処理結果はいちいち外部記憶に戻さずに、図-2(c)のように演算段数に応じたバッファ領域を用意して、次々にこれを切替えながら使用してゆく方が能率が良い（図ではスイッチは省略してある）。 $N \times N$ の近傍演算を p 段実行する場合には、 $Np+3$ ラスタ分のバッファ領域を準備しなければならないことになるが、これは多くの場合、通常の小形計算機などでも十分に準備可能な大きさである。処理の内容によっては、単純なカスケード処理ではなく、同一データに異なる演算をほどこすことにより別々の結果を得たり（データの流れの分岐）、異なるデータや演算結果の間で演算を行う（データの流れの合流）必要がある場合もあるが、同じような考え方で取扱ってゆくことができる⁴⁾。

話は少々横道にそれるが、図-2(c)に示す方式は、各処理に独立な演算装置を用いることができる場合には並列処理システムとして実現できる。実際にバッファ・メモリは遅延線であると考えてよいから、CCDのようなシフト・レジスタ型のメモリと演算ユニットを組合せれば、外界から得られるビデオ信号の速度に同期した実時間処理も可能なわけで、実際にそのような試みもあるようである⁵⁾。

3.2 2次元仮想配列

上に述べたラスタ・バッファ方式は、信号レベルの画像全体に対して、その内容に関係なく近傍演算で実現できる均一な処理をほどこす場合に適している。しかしながら処理の目的によっては、このような条件を満足しない場合も多い。たとえば画像の中のある特徴に着目して、これを追跡しながら局所的な処理を行ってゆくような場合である。このような場合にはラス

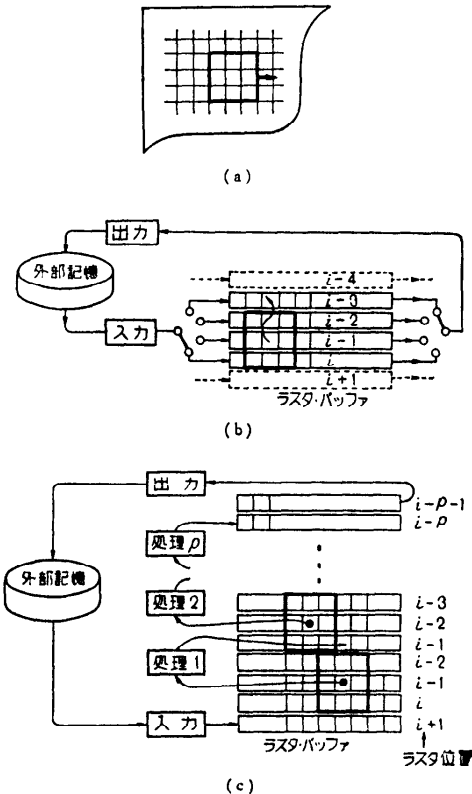


図-2 ラスタ・バッファ切替方式による近傍処理

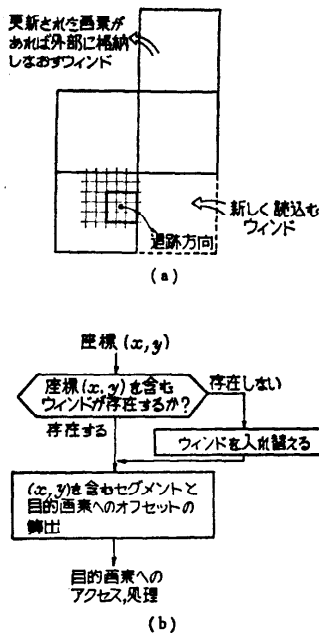


図-3 2次元仮想配列方式

タ・バッファ方式のように逐次的なアクセスを基本とするものは具合が悪く、ランダム・アクセスを許す方式が望ましい。

図-3(a)はこれを実現するために利用される2次元仮想配列方式を模式的に示したものである。この方式では画像はあらかじめある一定の大きさのウィンドに分割されていて、そのうちの数枚が主記憶上に保持される。与えられた座標位置の画素データにアクセスするには、その座標を含むウィンドが主記憶上に存在するかどうかを検査し、無ければ適当なアルゴリズムに従ってウィンドを入れ替えた後、画素データを読出したり更新したりする。入れ替えの方法としては、通常のページングと同様に最も古いものについて行う方式のほか、距離の最も遠いウィンドを入れ替える方法なども考えられる。いずれにせよこのようなソフトウェア・ページングのもとで、座標値を引数として対応画素にアクセスするための関数をいくつか用意しておけば、ユーザはウィンドの存在を気にせずに、任意の大きさの画像について処理プログラムを書けることになる。

ところで、この方法を用いれば、プログラムの書きやすさや保守性は向上するけれども、このようなページングに対するハードウェア・サポートは通常皆無で

あるから、実際には各画素へのアクセスのたびに図-3(b)に示すような境界検査が必要となり処理速度の点では問題が残る。このような問題を軽減するための便法として、与えられた位置の近傍データを一まとめにして別の作業領域に転送するための関数を用意しておく方法がとられることもあるが⁶⁾、これはせっきくのプログラミング上の見通しの良さを損なうことになるのであまり望ましいこととはいえないであろう。

したがって、この方式は必ずしも処理能率を最優先させなくてもよい、色々なアルゴリズムを開発したり比較するための実験システムに適した方式であるといえる。

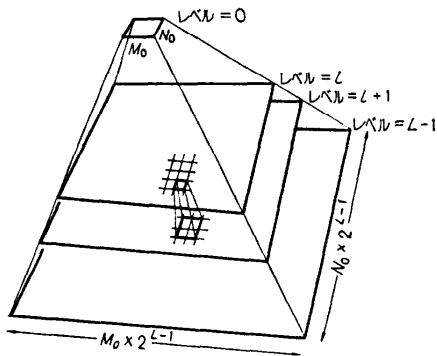
3.3 ピラミッド・データ構造と4進木表現

信号レベルのデータを処理しようとする場合、画面上のすべての部分について同じような詳しきで処理を行う必要の無い場合がしばしば起る。たとえば均一な性質を持った背景領域については、それ以外の目的とする対象が含まれていそうな領域と同じように処理を行う必要は無いことが多い。逆にこのような領域を選び出すには、最終的な処理精度に必要なとされる分解能よりも粗い分解能で処理を行う方が具合の良いこともある⁷⁾。

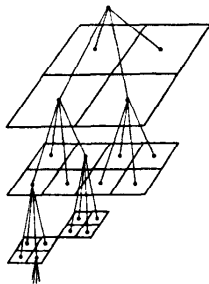
一般に粗い分解能でおよその見当をつけ(プランを作り)必要に応じて部分的に詳しく解析する、あるいはよりおおまかに大局的な状況を調べる、といった処理方式を想定し、あらかじめ何段階もの分解能の画面を用意しておこうというのがピラミッド形データ構造である⁸⁾。

図-4に示すようにこの方式では、データはL層の階層で表現されていて、各層は1段下の層のデータのいわば要約となっている。たとえば第l層の分解能は第l+1層の分解能の1/2となっていて、第l層の各画素の値は、その下位の対応する4画素の値の平均値であるとか、最大値と最小値などに対応している。

このようなデータ表現は、記憶領域をある程度犠牲にしても、無駄な処理を省いて全体としての処理能率を高めようとするもので、最近の記憶価格の低廉化と比較的よくマッチするものではあるが、とはいっても画像データが実在の記憶領域に対してかなり膨大であるという状況には変り無く、実際の画像処理でこのようなデータ表現を完全な形で実現し利用している例はあまり無い。またこのような構造を想定したプロセッシング・コーンと呼ばれる並列処理ハードウェア(マイクロコンピュータ・ネットワーク)の提案もある



(a)



(b)

図-4 ピラミッド・データ構造(a)と画像の4進木(b)

が、まだその能力、可能性等について一部で検討が行われている段階である⁹⁾。

このデータ構造で興味ある点の一つは、画像の特徴抽出、セグメンテーション、帯域圧縮等との関係である。すでに述べたようにピラミッド構造導入の動機の一つは、詳しく調べるべき部分と、そうでない部分を、できるだけ早い段階でうまく仕分けして無駄を省きたいという点にあるのであるが、画像を蓄えたり伝送したりする場合にも同じ要求は当然あるわけで、ある特定の領域が一定の明るさや色を持っているなら、その領域の形状や位置に関する情報とその属性という形で符号化した方が効率の良い場合も考えられる。

また効率の点を別にしても、たとえば伝送速度の遅い通信路を通して送られてくる画像を表示して観測する場合に、左上隅から最終的な精度でゆっくりと画像が現れてくるよりも、まず粗い精度で全体の様子が示され、時間とともに詳細が浮び上がってくる方が望ましいことが多い。ピラミッド状の構造で表現された画

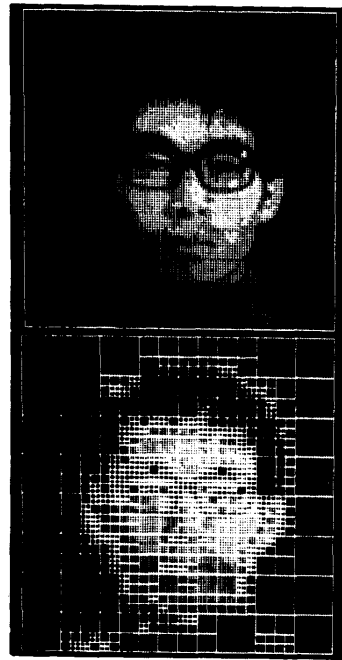


図-5 画像の4分割処理例

像を最上段から転送すれば、このような要求は原理的に満たされるわけで、実際にこのような方式に関する提案もいくつかなされている^{10), 11)}。

また均一な属性を持った領域に関しては、それ以上領域を小さく分割して格納したり転送したりする必要は無いわけで、この点を利用するとデータの圧縮が可能となる。図-4(b)はこれを模型的に示したものであるが、図に見られるようにこれは画像を4進木で表現することに相当しており、この4進木は与えられた画像の一種の領域分割木になっている。

図-5は明度情報のみを用いてこのような分割を行った例であるが、小さな領域に分割された部分は濃淡の縁のように性質の急激に変化する部分に対応しており、一種の特徴抽出にもなっている。しかしながら4進木の構造自体は画像データの内容とは独立に定められるために、各領域の距離的な近さであるとか、あるいは性質の類似性といったものが木構造上に直接的に反映されないという具合の悪い点もあり、その取扱いは工夫を要する¹²⁾。なおピラミッド・データ構造に関しては、画像処理を離れて、一般的な行列計算の場面に適用しようとする試みもある¹³⁾。

3.4 ビット・スライス表現

これは画像データの並列処理を目的とした最も古い

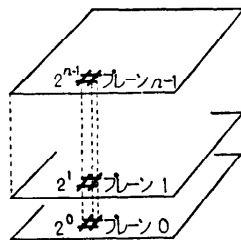


図-6 ビットスライス表現

計算機の一つ ILLIAC-III や、類似の構造を持つ CLIP 4 などに用いられている表現法で、濃淡データの各桁に対応させたビット・プレーン上での並列演算を基本とするものである¹⁴⁾。濃淡精度が2値以上の場合には、その必要桁数に応じてビット直列に処理を行う(図-6 参照)。

この表現法は、もともと ILLIAC-III のシミュレータとして開発され、その後独立に画像処理用プログラム・パッケージとして一部に利用されていた PAX にも採用されている¹⁵⁾。その最大の利点は任意濃淡精度の画像データを無駄なく記憶することが容易な点にあるが、実際の画像データはもともと全体を主記憶上に保持することが困難なことが多く、また記憶コストの低廉化に伴い処理効率の向上がより重視されるようになった結果、現在では特殊な目的以外にはあまり用いられない。

記憶装置の大容量化と低価格化の傾向は、今後もしばらくは衰えることがないだろうから、これからの画像処理では画像データをいかにして保持するかという空間的な問題はそれほど重大なものとはなくなってくるにちがいない。しかしながら演算素子の速度には、集積回路の集積度のような急速な進歩はあまり望めないし、その一方では取扱う画像の分解能、したがってデータ量は、実際の要求を満たすまで大きくなっていくであろうから、今後はできるだけ無駄な処理を省き、また並列的なアクセスや処理を許すハードウェアおよびソフトウェア方式がより重視されることになるものと思われる。

4. 画像に関する知識の表現と利用

入力装置から取り入れられる画像データは、対象が何であるにせよ数値データが縦横に並んだものにすぎない(場合によっては少々異なるフォーマットが用いられることもあるが)。このような信号レベル・データの各画素の値や相互の関係、統計的な性質などをも

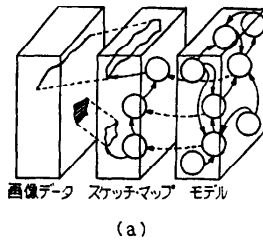
とに、特定の対象を抽出し、認識する操作では、対象に関する情報を何らかの形で蓄えておき、これを用いて目的とする対象を探したり、目的に到達するための戦略を選択したりすることが必要となる。

たとえば、航空写真の解析について考えると、森や畑、市街地といった対象が画像として捉えられた場合、それぞれに対応する領域がどのようなスペクトルを持ち、また形状を有するかとか、それぞれの間の相互の制限条件や、もっと一般的な法則(たとえば撮像系の位置や方向と像との関係であるとか、照明系と反射光成分の関係など)といった情報が必要となる。もちろん処理対象が変わって、たとえば細胞の顕微鏡写真の解析を行う場合には、異なった種類、内容の情報が必要とされる。

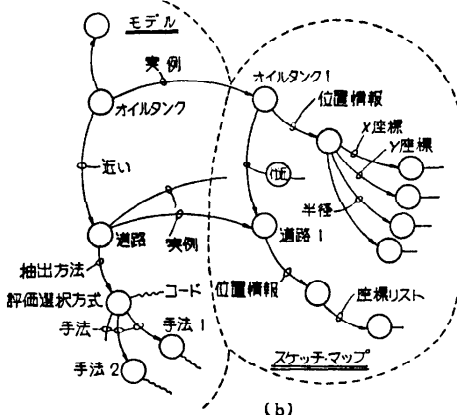
このような情報は、一般に対象に関するモデルとか知識とか呼ばれている。前者は対象の物理的な特性や幾何学的形状、配置の制限条件などに依存する、画像としての見え方についての、何らかの形の抽象化された記述を指す場合が多い。これに対して後者はこのような情報を含み、さらに個々の対象や、画像としての性質に直接的には結びつかない一般的法則や環境条件まで含めたより広い範囲の情報をさすことが多い。

画像の認識とか理解とか呼ばれる処理は、与えられた画像を調べて、その画像の性質や特徴と、このようなモデルとの対応づけを行い、対応づけのできた部分の結果に基づいて選ばれた適当な解析手法や推測のもとに、不明の部分さらに詳細に調べ、全体の構造を明らかにしてゆく過程であるといえるだろう。強い制限を付けた文字認識や、形状が比較的簡単で種類の少ない部品の認識の場合のように、問題が比較的簡単な場合には、このような階層的処理構造をとらずに、抽象度が低く信号レベルに近い情報の直接比較で十分な場合もあるが、問題を少し一般化するとこのような方式ではうまくゆかなくなる。手書きの文章を機械に支障なく読ませることはまだ不可能であるが、これを可能にするためには、文字の形に関する情報を、非常に大きな幾何学的変形に対しても不変な成分に要約して保持し利用できるようにすることが必要であろうし、さらには人間のようにある程度の誤植や伏字があっても、その部分の正しい文字を推定し、意味を理解できるようにするための一般的な知識や、言語理解能力も不可欠であろう。

このようなモデルや知識の表現法としては、ネットワーク的なものや表的なものが考えられている。また



(a)



(b)

図-7 ネットワーク型画像モデル
全体構成(a)と具体例(b)

その要素の一部に画像例を信号レベルデータの形で保持したり(これを記号による記述に対して iconic な記述と呼ぶことがある¹⁶⁾、各要素に付随する属性を調べるためのルーチンを保持したりすることにより、対応づけの処理のさいに必要なに応じてそれらを参照したり、起動したりできるようにする方式も考えられている。

図-7 はネットワーク的なモデルを用いる場合の例で、実際に与えられた画像とモデルの対応づけを行った結果は、スケッチ・マップと称するグラフ構造データに蓄えられる¹⁷⁾。スケッチ・マップは与えられた画像の解釈であり、また記述でもあって、同図(b)の航空写真の場合についてみると、その中のオイルタンクであるとか、道路といった要素は、モデル中の要素の一つの実現値としてそれらと結びつけられており、また解析の結果得られたそれらの属性値や関係といったものもスケッチ・マップ上に蓄えられる。スケッチ・マップ上の要素と実画像データとの間の対応付けは、座標や広さの情報により行われる。

このシステムでは、モデル中に特定の目標を見出す

ためのルーチンが一般に複数個、適用条件や期待される結果とともに保持されており、また状況に応じてこれらのいずれかを選択するための評価ルーチンも保持されている。応用プログラムはこのようなモデルを適当な手順で用いてスケッチ・マップを求め、これをもとに与えられた画像に関する質問に答えたり、特定の性質を持った領域の広さなど必要とされる測定値を求めたりすることができる。応用プログラムがどのような手順でこのようなモデルを利用するかは、当然対象により異なり、胸部X線写真のように対象の位置や形状がおおよそ定まっている場合には、ある程度固定のプランに基づいてトップダウン的に処理を進めることができるが、一般的には入力画像の信号レベルデータの性質に基づいて何らかのプランを動的に作成し、これに基づいてモデルを利用する処理を開始することが必要である。

このような対象画像に関するモデルあるいは知識の表現法にはどのような性質が要求されるであろうか。少なくとも処理手法の研究段階では、情報の蓄積や検証・修正および利用が容易で、しかも異なった分野にも適用可能な柔軟性を持ったシステム、したがって色々な側面からみてモジュラリティの高いシステムであることが必要であろう。たとえば、単純な構造から段階的に知識を付加してゆくことが可能であるとか、画像の構成要素の性質や構造、相互関係といった対象に関する静的な記述、あるいはデータ構造と、これらを見出すための手続き、あるいは制御構造とが独立した形になっているというようなことである。このような意味では、先に述べた例はさほど見通しの良い構造を持っているとはいえない。

ARPA の音声理解プロジェクトで採用されたことでよく知られている黒板モデル (blackboard model) は、このようなモジュラリティを保つことが比較的容易であるという点では好ましいものであると思われる¹⁸⁾。これは知識の表現法というよりは、これを利用して外界から与えられる信号データを解釈してゆくための一つの一般的な枠組であるが、これを画像の場合に適用することが、いくつか試みられている。

図-8 はその基本構造を模式的に示したもので、黒板と称する大域データ構造と、その内容に従って動作する互いに独立な解析・解釈ルーチン群 (Knowledge Source 略して KS と呼ばれる)、および黒板に蓄かれた情報を監視し、その内容と問題に応じて適切な KS を起動してゆくためのモニタが主要な構成要素と

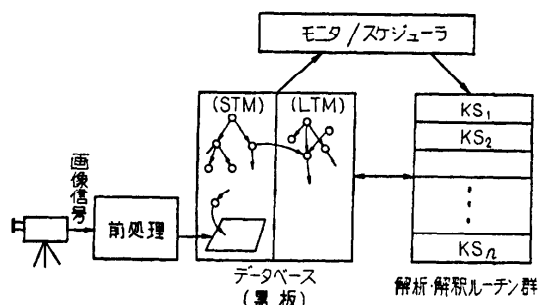


図-8 黒板モデルの一般的構造

なっている。図中の LTM は Long-Term Memory の略で、対象の構造についての宣言的な記述を保持する読出し専用のデータである。一方 Short-Term Memory STM は、入力画像に応じて作られてゆく一時的なもので、図-7 のスケッチ・マップに対応している¹⁹⁾。

STM に相当する大域データ構造は、黒板モデルを用いるシステムには不可欠の要素で、各 KS はその内容の変化に応じて起動され、その結果得られる入力画像の特定領域に関する認識結果であるとか、推測結果も STM に戻される。これに対して LTM に相当する部分は、実際には図のように隔に独立することなく、KS に埋め込まれている場合もある。この場合には一般性は犠牲となるが、オーバ・ヘッドを低くできるので問題領域を限れば、より実際的な実現法となりうる。

黒板モデルは上述のように基本的にはデータ駆動形のシステムで、プロダクション・システム的な枠組の上で実現することができる。各 KS 間の情報の交換は、黒板上の情報を介してのみ行われるのが原則であり、原理的には並列的な動作も可能であるが、その場合にはモニタ中にはスケジューラのかわりに何らかの矛盾解消機構が必要とされる²⁰⁾。

画像の認識処理のための知識の表現法、利用法については、色々な方法が試みられているが、これを逐一人手で作り上げ、蓄積し、検証してゆくこと自体も非常に困難な作業である。例題を与えることによって、対象のスペクトル的な属性のみならず、その構造や周囲の要素との相互関係などを、少なくとも半自動的に取入れ、対象のモデルを作り上げてゆくことのできるような、高度の道具立ての開発がぜひとも必要であろう。

5. 画像データの管理と画像データベース

計算機で取扱う画像の数量や種類が増加し、また色々な処理手法を試みる機会が増えてくると、このような画像データや、その関連情報を、一つの統一したシステムのもとで管理し、利用できるようにすることが必要となってくる。

その発端は、(1)種々の独立なプログラムが同一の画像データを自由に参照し、処理できるようにしたい、あるいは(2)同一のプログラムで異なったフォーマットや大きさの画像を、いちいちプログラムを変更したり、パラメタを設定しなおしたりせず、自動的に処理できるようにしたいという、主として信号レベルの画像データの管理と利用に関するものであった²¹⁾。その後、実際の応用面に画像処理機能を利用しようとする動きにつれ、単に蓄えられた画像データやその一部を検索用の見出しに基づいて読出し、プログラムの中で処理できるというだけでなく、(3)画像上の指定された点や領域に関連する情報を検索したり、(4)特定の性質を持っているとか、似た性質を持った画像を取り出してくるというような、ある程度画像の内容にまで立入った検索の機能も要求されるようになってきた。

考えてみれば、我々は日常生活でも色々な形の画像データベースと呼べるものを(大部分は機械化されていないけれども)利用している。地図帖などはその典型的な例であろう。ドライブをするときなど、特定の道路に沿った地図を次々に探し出し、着目している地点が頁の境目に来たときには、その周辺の地図を張り合せて着目点を中心になるようにするとか、周囲の地形を知るために地形図を重ね合せたり、住所などを調べるために行政区分図を出して対応付けを行うというような操作はしばしば必要となるものであるが、普通これらの操作は我々の頭の中で行われている。

似たような問題、あるいはもっと複雑な問題はほかの色々な画像の場合についても考えられる。たとえば地図のほかにも、各種の設計図やレイアウト図、航空写真や衛星写真、後で参照する必要のある医療写真等等である。図面類は人間が眺めて容易に情報を読み取ることができるということだけのために2次元的な形態をとっている情報で、もともとの観測データが2次元的な航空写真等とは少々異質のようにも思えるが、現在のところすでに存在する地図や図面を機械に自由に理解させて、記号の形で記憶させておくことは可能

になっていないし、もし可能になったとしても、それが本当に望ましい形態かどうかは検討の余地のあることであるから、同列に扱っておいて良いであろう。地図等も含めて、各種図面類のデータベース化は、画像データベースの中でも今後実際のインパクトの最も大きいものの一つであると考えられる。

画像データベースが実際に有用なものとして機能するためには、上に述べたように、単に画像がある定まったフォーマットで蓄えられているというだけでなく、画像自身に関する適切な変換機能と、関連情報の効率の良い検索機能が要求される。このような機能を持ったシステムを実現する技術はまだ未熟で、色々な提案が行われている段階であるが、最近になってその環境は徐々に整いつつあるように思われる。

画像データベース・システムにおける画像データは、一般に非常に大きい単純な配列データで、構造を持った記号データとはなじみが悪い。そこで信号レベルの画像データは、記号データとは物理的にも別の系統に切り離し、そのディレクトリ情報も一つの関係データとして保持するというような構成のとられることが多い。画像データは、部分的な書替え等のあまり行われぬ、読出しのみに利用されるアーカイバルなものが多いと考えられるから、このような場合に光ディスクであるとか、場合によってはビデオディスクやマイクロフィッシュといった、従来計算機システムには比較的縁の薄かったアナログ的な記録媒体なども大幅に利用されるようになるかも知れない。また画像伝送に関連して開発の行われている帯域圧縮技術なども、対象によっては効率の良い記憶方式を実現するうえで重要であると考えられる。そしてこのように蓄えられた画像データに関する色々な形の間合せに迅速に対処できるようにするために、画像のズーム、重ね合せ、張り合せなど種々の変換操作を高速化するための専用プロセッサ類が開発され、より広く用いられるようになってくるものと思われる。

一方記号レベルの情報に関しては、原理的には通常のデータベースと同じ技術を用いることができるわけで、関係モデルによるもの、階層構造のもの、そしてネットワーク構造のものなど種々のものが提案され、試みられている^{22)~25)}。ここでの問題は、信号レベル画像データとの関連において、いかに的確に必要とする情報にアクセスし、外界からの間合せに答えられるようにするかということであり、問題によっては間合せの処理系にかなりの画像処理機能が要求されること

になる。

たとえば QPE(Query by Pictorial Example) と呼ばれるシステムでは、間合せ言語として良く知られる QBE を拡張して、間合せコマンドのパラメータとして、集計とか平均というような基本関数のほかに、ある程度の画像処理関数を許すことを試みている²³⁾。これによって、たとえば間合せ表上の指定の一部を、表示された画像上の点や領域をカーソルなどで指定することに置き替えたり、画像内の特定の目標物を抽出するための処理手続きを新たな関数として定義し、これを基本関数のかわりに使用して、間合せを行うことを可能にしている。

このような間合せの形態自体は、人間と画像データベースの間のインタフェースの実現法の一つにすぎないわけであるから、人間工学的な側面からの検討も必要であると思われるが、機能面での考え方には注目すべき点もあると思われる。

間合せを行うものが人間であるにせよ、プログラムであるにせよ、欲しい情報が何 (what) であるかを指定すれば、その詳細な求め方 (how) を知らなくても、必要な情報にアクセスできるようにすることは、情報システム一般に課せられた課題の一つであるが、画像処理の場合についても、このような機能は必要とされる。画像データベースは、実体よりも言葉の方が先行してしまった感もあるけれども、図面にしても写真にしても、画像の形態をとる情報は今後増える一方であると考えられるから、これらを適切に管理し、自由に活用できるように、非常に高度な機能を持った画像データベース・システムを開発することは、これからの画像処理システム開発の究極の目標の一つと考えても良いのではないかと思われる。

6. プログラミング・システムとプログラム言語

これまで述べてきたような性質や構造を持った画像データを処理するプログラムを開発するための、プログラミング環境には何が要求されるだろうか。いうまでもなく画像処理の第1の特徴は、外界との間で何らかの形の画像信号の入出力が行われることであるが、そのほかに、信号レベルの処理では大量の画素データを扱わねばならないこと、4章に見られるように画像の認識とか理解を目的とする処理では、信号データから複雑な構造を持った記号データにいたるまで、広い範囲のデータ構造を取扱わねばならないこと、そして

人間との間で頻繁な交信を必要とする処理が多いこと、などの特徴を有している。

6.1 会話形処理

ここではまず主として信号レベルの画像処理アルゴリズムの開発に用いられるシステムに要求される機能について考えてみよう。このようなシステムでは、たとえば(i)同一のデータを色々な方式で処理してみても結果を比較する、(ii)特定のアルゴリズムを色々なデータに適用してみても、それぞれに対して思うような結果が得られるかどうか調べる、(iii)複数の処理手続きを組合せた操作を加えて結果を調べる、といった実験をできるだけ簡単な操作で繰り返し行えることが望ましい。

このためには、一連の実験において各種のパラメータや作業データ・セットの変更は必要なもののみで済ますことが可能であるとともに、必要な変更も対話的にできるだけ簡単に行えることが要求される。さらに複数の独立な処理プログラムによる組合せ処理を定義し、簡単に実行できることも必要である。

このような機能のある程度まで備えた会話形画像処理システムは、これまでもいくつか発表され実際に使用されている^{26),27)}。その基本的なソフトウェア構成要素には次のようなものがある(図-9参照)。

- 1) 基本オペレーティング・システム：基本的なファイル・システムと、言語プロセッサやエディタ等のユーティリティを備えた会話処理向き OS。
- 2) 専用モニタ：上記 OS のもとで動作し、コマンドの解釈・実行、作業データの管理などを行うためのもので、一般にメモリ常駐の核部分、常駐ライブラリなどのほかに、次のものから構成される。

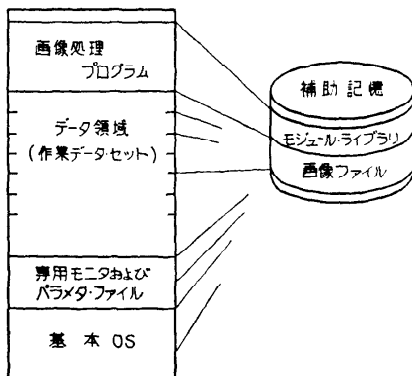


図-9 画像処理実験システム・ソフトウェアの模式的な構成図

(a) コマンド・インタプリタ：カーソル装置などを含めたオペレータ入力機器、および通常ファイルからのコマンド列の解釈・実行を行うプログラム。

(b) パラメータ・ファイル：処理データの属性や各種のパラメータ(大きさやウィンドの位置など)、処理モジュール間の交信メッセージなどを保持・管理するための、一般にメモリ常駐のファイル。

(c) 履歴管理システム：処理経歴を後で参照したり、その一部を編集して新たなコマンド列として利用したりするためのモジュール。

(d) ヘルプ・システム：誤りを犯したときの回復システム、およびオンライン・ドキュメント類。

3) 処理モジュール類：実際の問題に応じてモニタを介して起動される独立な入出力および処理モジュール類と、これらを作成するために使用される画像処理用サブルーチン・ライブラリ。

4) 作業データ・セット：使用者によって定義される処理対象データ、および中間結果などで、多くの場合何らかの仮想配列の形でメモリ上に存在する。

実際のシステムは、このような要素にどんな機能を持たせ、またインタフェースを設定するかによって、能率や使い勝手の大変異なるものとなる。たとえば作業データの管理については、データ全体を主記憶上に保持できないことが多いため、3章で述べたようなデータ管理(バッファリング)を行うことになるが、処理能率を重視する場合にはラスタ・バッファ方式を前提とすることになり、画面単位の処理とは異なるインタフェースを持った処理モジュールを用意することが必要となる²⁸⁾。

信号レベルの画像処理では、上のような配慮を行っても処理に時間がかかり、通常の計算機で会話の処理に支障の無い応答特性を得ることは困難である。そこで画像処理専用プロセッサと画像用のデータ・メモリを別に設け、信号レベルの処理はこれの中でだけ行うという方式も試みられるようになってきた²⁹⁾。この場合には先の処理モジュールは、専用プロセッサへのコマンド系列の発生、マイクロプログラムの入替え、そのほかの制御情報の入出力等が主要な内容となり、また画像データに関する記憶管理も大幅に簡略化できるので、ソフトウェアの内部構造はかなり異なったものとなる。

そのほかの使い勝手に関しては、パラメータの省略に対し適切な措置がとられるか、コマンドの省略形や好みの形式の設定が許されるか、パラメータの問合せ時

に、その時のデフォルト値を表示して修正入力させる方式がとれるか、といった会話形式の比較的外面に関するものから始まって、マクロコマンドあるいはコマンド・ファイルに許される表現や機能、処理の履歴ファイルの管理法、誤りからの回復法や保護機能、画像処理システムから出ることなくエディタなどのユーティリティを使えるかなど、種々の細かい点にどの程度配慮が行き届いているか、また基本 OS がこれらをサポートしているかが問題の鍵となる。

とくにコマンド・ファイルについては、許されるループやネスト構造、あるいはパラメタの受渡し機能や会話機能によっては、専用のマクロコマンド・システムを作る必要は無くなる。柔軟性の高いコマンド・システムは、むしろ高レベルの会話形プログラム言語であると考えた方がよいわけで、使いやすい会話形システムやパーソナル・システムの要ともいえるものである。最近では画像処理の世界でも UNIX をベースにして研究の行われている例が多いが、これもこの辺の機能の充実による所が大きいのではないと思われる³⁰⁾。

6.2 プログラム言語

次に画像処理プログラムを作成するためのプログラム言語について考えてみよう。この分野でも好むと好まざるとにかかわらず、FORTRAN はもっともよく使われる言語で、とくに信号レベルの処理ではほぼ標準語として定着している。最近の FORTRAN-77 はこの傾向を一層堅固なものとしたといえるだろう。

しかしながら先に述べたように、画像の認識とか理解を目的とする処理では、与えられた信号データを何らかの記号データに変換する過程で、モデルとか辞書を参照し、不明確な部分については再度信号データを見直してみるといった具合に、大量の数値データと複雑な構造を持った記号データを同じ世界で取扱わなければならないことが多い。このような問題でも様々な工夫を行って FORTRAN が使われているけれども、そのためには問題の本質的な部分以外にかなりの労力をさかねばならないことになる。

このような問題では、外界に近い低レベルの処理も、内部モデルに関連する高レベルの処理も同じように不自由なく表現でき、さらにこれらに必要な多様な制御構造やデータベース機能を効率良く実現できる言語が要求される。そしてまた言語が実際に使用可能であるためには、少なくとも FORTRAN 並みの効率を有する処理系や FORTRAN との結合機能、さらにそ

の処理系を長期にわたって保守できるグループや実績の存在することが必要であろう。現在のところこのような要求を満たす言語は筆者の知る限り SAIL³¹⁾ と MACLISP³²⁾ のみのようである。残念なことこの2つの言語は今のところ特定のハードウェア上でしか利用できないが、前者に関してはミニコンなどの上で使えるサブセットの開発が行われているといわれており、後者に関してもメガミニ上で開発や、専用マシンの開発が伝えられているので、近いうちにこれらの利用可能性はずっと高まるかも知れない。

上記の2つはシステム記述性や会話処理機能、デバッグ・ツールの完備という点でもかなり強力なプログラミング・システムを形成している。高レベルのプログラム言語や豊富な実行時ルーチン群とともに、高水準で人間工学的にもすぐれたエディタや各種のユーティリティを備え、会話処理に適した一体構造の(できるだけフラットな構造の)プログラミング・システムを用意することは、画像処理に限らず高いプログラム生産性を維持するうえで不可欠の要素と考えられる。このようなシステムには、論理的な機能の強力さとともに、その時の環境やコンテクスに応じて必要最小限の情報を入力で目的とする処理を実行でき、また足りない情報や誤りに対しては、適切な指示を自動的に出せるといった「賢さ」が求められる。

このような要求を満たすために、これからはプログラム言語のみの閉じた世界でなく、応用も含めてプログラミング全体を見通した広い視点からの検討が必要となってくるだろう。言語システムと OS の接近は³³⁾、効率や信頼性の面ばかりでなく、使いやすい会話形システムの実現という点からも重要であり、将来の会話形画像処理システムや、もっと汎用の高度なパーソナル・システムがこれに沿って進むだろうことは疑いない。

7. 移植性と標準化

一つの計算機上で使用していたプログラムやデータを別の計算機上に移す問題は、情報処理のどの分野でも頭痛の種である。情報処理技術が急速に進展している一方で、変換や移植といった本来問題の本質にあまり関係の無い部分の作業に多大の労力を投入しなければならないという現実には喜劇的できさえる。このような問題は、原理的にはプログラムとかデータなどの情報が蓄積される前に、その記述法とか表現法を統一することができれば避けることのできるものであり、こ

の意味で、画像処理に関連する研究が盛んになると前後して、当学会の研究会が画像データのフォーマットの標準化や、移植の容易なプログラム・パッケージの開発を推進してきたことは意義深いことといえる³⁴⁾。

標準化の問題は、通常色々な利害が絡み合ってまとも難しく、しかも将来を十分に見通して制定しないと技術の円滑な進歩や普及に障害となることも有り得るといふ両刃の剣的な側面もあって困難なものであるが、画像処理の場合には利害の対立を生ずるまで技術が円熟していないことが幸いしているといえる。

しかしながらデータ・フォーマットについてみると、国外でも同じような趣旨で別の活動が行われていたり³⁵⁾、米国ではスタンフォード大学のフォーマット*をもとに実際のデータ交換が行われたりしているの、国際的な規格の統一ということとはなかなか困難のようである。当分の間は2, 3の代表的なフォーマットが並立して存在するということになるのではないだろうか。

プログラムの移植についてはさらに問題が多い。上述の画像処理パッケージは、どちらかといえば既存のアルゴリズムを集積し、できるだけ多くの計算機上で利用できるようにしようとするもので、その結果画面単位の処理を原則とし、入出力やファイル・システムに依存する処理は避けるような構造となっている。もちろんこれも一つの立場であって、プログラム・コードそのものは使用しないというような極端な場合でも、アルゴリズム集としては十分に意味があり、有用なものであるが、実行効率の面から実際の応用では書替えの必要な場合も生ずるだろう。このようなパッケージの問題の一つは、強力な管理者が十分に注意していないと、拡張や修正のさいに、移植性を妨げる要因が紛れ込みやすいことであると思われる。

このような問題や、OSとの整合性を考慮に入れ、逆の立場から出発して移植性を保ったパッケージを作るべきだという提案もあり、開発もある程度進められているようである³⁶⁾。このシステムでは、KernelとSurroundから成る2層構造のシステム・インタフェースを設定し、この上に現在比較的標準が良く保たれている構造化FORTRAN (RATFOR) を用いて画像処理システムを作り上げてゆこうとするものである。

* 前記の標準フォーマットは、パターン情報処理プロジェクトに関連し電線研でまとめたものを土台にしているが、実はこれもスタンフォード大学AI研究所で使われていたものを叩台にして拡張したものであった。

このインタフェースにはランダムアクセス・ファイルを始めとするファイル機能、FORTRANの配列に対する直接入出力、入出力との並行動作を可能にするための即時リターンなどの機能が用意されていて、かなりきめ細かいプログラム作りが可能ないように配慮されている。またルーチン間のインタフェースのとり方も色々工夫されていて、たとえばラスタ・バッファ方式による効率の良い近傍処理アルゴリズムなども、バッファ切替の機構を知らなくてもプログラムすることができる。

この方式では、各OSごとに標準システム・インタフェースを準備しなければならないわけであるから、そのためにある程度の初期投資が必要となるが、それ以後はかなり広い範囲のプログラムを自由に移すことが可能となる。実はこのようにシステム固有のOSを標準的なOSで包んでプログラムの移植性を確保しようという考え方は、画像処理に限らず以前から一般的にあったもので、最近もかなり多種類の計算機システムに対して仮想OS (実際には仮想UNIX) を開発するという大がかりなプロジェクトが発表されている³⁷⁾。その基本機能が実用に耐えるか否かについては十分な検討が必要と思われるが、原理的には先のシステム・インタフェースを仮想OS上に実現することも可能なわけで、画像処理システムの移植もこれを前提とすることが考えられるだろう。

「計算機の歴史は、この世界では優れたアイデアが急速に広まることを期待することがいかに馬鹿げたことかを明白に示している」というのは、上に移植可能ソフトウェアの設計に関連して著者の漏らした言葉であるが、最近の状況は必ずしもこのような悲観的な材料ばかりではなくなってきた。UNIX, CP/M, PASCALなどの普及はその好例だろう。その背景には素子技術やマイクロプロセッサ技術を含めた計算機システム技術の大幅な進歩と普及、それに伴うより優れた使いやすいシステムへの強い要求といったものが存在する。画像処理の世界でも、良いソフトウェアがさらに容易に移植し利用できるようになることを期待したい。

8. むすび

最近の計算機ハードウェア技術の進歩には目覚ましいものがあるが、これはプログラムの評価基準や作成態度にも少なからぬ影響を与えている。画像処理は、その影響の最も大きい分野の一つではないだろうか。最

近では実装主記憶が数 MB、一つのユーザ・ジョブに許される仮想記憶領域が数十 MB というミニコンも珍しくなくなってきた。しばらく前の浮動小数点演算器オプションのような気安さでベクトル演算器やレイ・プロセッサを組み込めるようになるのも時間の問題だろう。

このような状況のもとでは、従来比較的小さな主記憶を想定して発達したプログラム技術のあるもの、たとえばオーバーレイ手法や、ある種のバッファリング方式、あるいはこれまでの意味でのガーベジ・コレクション等々は、ほとんどその意味を失うことになるかも知れない。末端のユーザから見るとシステムはより透明なものとなり、おおらかなプログラム作りが許されるようになってくるだろう。しかしながらそのための前提としては、ハードウェア資源を可能な限り有効に利用し、高い実行効率を実現するための種々のレベルの最適化技術の開発と、そのような機能を組込んだシステム・プログラムの実現が是非とも必要となる。

また画像処理プログラムの蓄積は今後も益々進むだろうから、使用目的に合ったプログラムを検索し、使用方法を理解し、誤りに対する対処の仕方を見出すためにはかなりの労力を必要とするようになると思われる。この問題を軽減するための画像処理プログラム検索システムの試みもあるが³⁸⁾、このようなシステムが有効に機能するためには、基礎となるデータが十分に良質なものでないと具合が悪い。

良いドキュメントが用意されているか否かは、プログラムの品質を定める重要な要因であるとも考えられるが、プログラムの作成時点から半ば強制的に、あるいは半自動的に、使用法や引数の意味についての情報を引出し、データベースへの登録や文書化の助けにできるような系統的手法の開発が望まれる。画像処理の分野でも、ソフトウェア工学や新しいプログラム技術への期待は小さくないのである。

参考文献

- 棟上昭男：画像解析技術、計測と制御、Vol. 16, No. 8, pp. 599-609 (1977).
- 長尾 真：画像処理のためのソフトウェア、電子通信学会誌、Vol. 58, No. 1, pp. 13-20 (1975).
- 伊藤貴康：画像処理ソフトウェア、電子通信学会誌、Vol. 59, No. 11, pp. 1208-1215 (1976).
- 藤村是明：ラスト走査形画像処理のプログラミング技法、情報処理学会第 21 回全国大会、7H-4 (1980).
- 詳細はあまり知られていないが、画像の実時間処理に関する日米セミナー (1978年11月、論文集は尾上守夫他編 Real-Time Medical Image Processing および Real-Time Parallel Computing として Plenum 社より最近出版された) で発表された Cytocomputer や、ハイブリッド処理による 1 チップ・プロセッサの開発を旨とする米国画像理解プロジェクトの中のスマート・センサなど。後者に関してはたとえば、Nudd, G.R.: Image Understanding Architectures, Proc. AFIPS, Vol. 49, pp. 377-390 (1980 NCC).
- 白井良明：画像データ管理システム、情報処理学会イメージプロセッシング研究会 22-1 (1979).
- Kelly, M.D.: Edge Detection in Pictures by Computer Using Planning, Machine Intelligence 6, pp. 397-409, Edinburgh University Pr., (1971).
- Tanimoto, S. and Pavlidis, T.: A Hierarchical Data Structure for Picture Processing, Computer Gr. Image Process., Vol. 4, No. 2, pp. 104-119 (1975).
- Hanson, A.R. and Riseman, E.M.: Processing Cones: A Computational Structure for Image Analysis, in Tanimoto, S. and Klinger, A.(eds.), Structured Computer Vision, Academic Pr., pp. 101-131 (1980).
- 安田靖彦, 高木幹雄, 粟野友文: 静止画像の階層的符号化法, 第9回画像工学コンファレンス, pp. 153-156 (1978).
- Tanimoto, S.: Image Transmission with Gross Information First, Computer Gr. Image Process., Vol. 9, No. 1, pp. 72-76 (1979).
- Hunter, G.M. and Steiglitz, K.: Operations on Images Using Quad Trees, IEEE Trans., PAMI-1, No. 2, pp. 145-153 (1979).
- Solntseff, N. and Wood, D.: PYRAMIDS A Data Type For Matrix Representation in PASCAL, BIT, Vol. 17, No. 3, pp. 344-350 (1977).
- Duff, M.J.B.: A User's Look at Parallel Processing, Proc. 4th IJCP, pp. 1072-1075 (1978).
- Johnston, E.G.: The PAX II Picture Processing System, in Lipkin, B.S. and Rosenfeld, A. (eds.), Picture Processing and Psychopictorics, Academic Pr. pp. 427-512 (1970).
- Tenenbaum, J.M. and Garvey, T.D., et al.: An Interactive Facility for Scene Analysis Research, Technical Note 87, SRI (1974).
- Ballard, D.H. and Brown, C.M., et al.: An Approach to Knowledge-Directed Image Analysis, Proc. 5th IJCAI, pp. 664-670 (1977).
- Ermann, L.D. and Hayes-Roth, F., et al.: The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty, Computing Surveys, Vol. 12, No. 2, pp. 213-253 (1980).

- 19) Hanson, A.R. and Riseman, E.M.: VISIONS: A Computer System for Interpreting Scenes, in Hanson, A.R. and Riseman, E.M. (eds.), Computer Vision Systems, Academic Pr. pp. 303-333 (1978).
- 20) 松山隆司, 長尾 真: 航空写真の構造解析, 情報処理, Vol. 21, No. 5, pp. 468-480 (1980).
- 21) 田村秀行: パターン情報処理研究のための画像データ・ベースとその管理, 信学技報, PRL 77-73 (1978).
- 22) Chang, S.K. and Lin, B.S., et al.: A Generalized Zooming Technique for Pictorial Database System, in Chang, S.K. and Fu, K.S. (eds.), Pictorial Information Systems, Springer-Verlag pp. 257-287 (1980).
- 23) Chang, N.S. and Fu, K.S.: A Relational Database System for Images, *ibid.*, pp. 288-321.
- 24) McKeown, D.M. and Reddy, D.R.: A Hierarchical Symbolic Representation for an Image Database, Proc. Workshop on Picture Data Description and Management, pp. 40-44 (1977).
- 25) Shapiro, L.G. and Haralick, R.M.: A General Spatial Data Structure, Proc. Conf. on Pattern Recognition and Image Processing, pp. 238-249 (1978).
- 26) 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).
- 27) 木村雄太郎, 福島重広, 相馬敬司: 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, IE 78-104 (1979).
- 28) 棟上昭男, 山口徹郎, 佐藤孝紀, 藤村是明, 内田俊一: 図形認識の一般化と実現手法, パターン情報処理システム調査・研究報告 PIPS-R-No. 25 (大型プロジェクト成果発表会論文集), 電子技術総合研究所, pp. 57-76 (1980).
- 29) 渡辺貞一, 木戸出正継, 篠田英範, 麻田治男他: 専用プロセッサを備えた濃淡図形認識システム, *ibid.*, pp. 77-92.
- 30) UNIX Time-Sharing System 特集号, BSTJ, Vol. 57, No. 6, Part 2 (1978).
- 31) Swinehart, D. and Sproull, R., et al.: SAIL/FAIL, DECUS No. 10-86, Digital Equipment Computer Users Society (1977), または Reiser, J. (ed.): Sail, AIM-289, Stanford Artificial Intelligence Lab. (1976).
- 32) Moon, D.A.: MACLISP Reference Manual, Lab. Computer Science, MIT (1974). 改訂版も出版予定らしいが現在は DEC-10/20 上にもみ存在する.
- 33) Jones, A.K.: The Narrowing Gap between Language Systems and Operating Systems, Proc. IFIP Congress, Vol. 7, pp. 869-873 (1977).
- 34) 尾上守夫他 (イメージプロセッシング研究連絡会): 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).
- 35) Bohner, M.: A Tape Format for Transferral of Image Data and Source Programs, Computer Gr. Image Process., Vol. 11, No. 2, pp. 185-191 (1979).
- 36) Hamlet, R.G. and Rosenfeld, A.: Transportable Image-Processing Software, Proc. AFIPS, Vol. 48, pp. 267-272 (1979 NCC).
- 37) Hall, D.E. and Scherrer, D.K., et al.: A Virtual Operating System, Commun. ACM, Vol. 23, No. 9, pp. 495-502 (1980).
- 38) 塩見佳久, 鳥脇純一郎, 福村晃夫: 画像処理プログラム・ライブラリのためのプログラム情報検索システム, 信学技報, IE 78-105 (1979).
(昭和55年12月16日受付)