

標準画像データフォーマット

イメージプロセッシング研究連絡会

1. はじめに

電子計算機によるデジタル画像処理はアナログ処理に比し融通性に富み、精度、再現性に勝り、しかもICの発展に支えられて演算コストが低下しつつあるので、衛星、気象、医用、非破壊検査、リモートセンシング等広い分野で実用になりつつある。しかし、画像データが多岐にわたるため入出力機器、処理アルゴリズムなどの定量的相互比較評価が行えず、また特定の応用の為の訓練セットも研究者毎に重複して作られ互換性のない状態である。従って、標準となる画像のデータベースを作り、相互比較評価が可能になる様にし、これに基づいて各分野で蓄積される画像データが互換性をもって共存出来るようにすることが急務であると考えられる。

この様な見地からイメージプロセッシング研究連絡会では、縁起検出、セグメンテーション、投影からの立体像の再生等基本的アルゴリズムの比較、評価が行える様なデジタル画像、並びに画像入出力機器の評価を行えるような写真等の標準入力及びデジタル標準出力のフォーマットを定め、基本的なものから作成する作業を開始している。その中で、先ず、標準画像を研究者相互間で交換できるようにするための標準画像データフォーマットを定めることにし、検討を行って来たが、案としてまとまり、この報告をした。

ここでは、標準画像データフォーマットの案とその解説と例題につき紹介するが、画像の交換例ほどもを加え、情報処理学会誌上に掲載される予定であるので、御批判を頂ければ幸である。

2. 標準画像データフォーマット

画像データは磁気テープ(MT)を媒体として交換することを原則とするため、ここではMT上にデータを書いたり、読んだりする場合に必要なフォーマットについて述べる。しかし、MT以外の媒体(磁気ディスクなど)を用いる場合もここで述べるフォーマットを使うことができる。MT上のデータアクセスで最も重要な情報は、データがどのように区別されて配列されているかである。つぎに画像データがいかばる性質を持っているかも必要な情報である。以上の情報を画像データとは別にし、ヘッダ部として画像データの前に配置させることにする。このヘッダ部は固定長(512バイト)とし、いったんヘッダ部の情報が分れば、画像データはヘッダ部で指定されたフォーマットで書かれているので、容易に読むことができる。したがってヘッダ部のフォーマットを予め決めておくことが必要になる。

ヘッダ部フォーマットを決めるさいに考慮した点を以下に示す。

- (1) 画像データの区切り方と配列の順序には融通性を持たせ、入出力の容易さだけでなく、標準フォーマットで書かれたMTを直接処理する場合の効率を考へる。すなわちデータの区切りを単に水平走査線単位とし、任意の矩形領域とすることによって、輪廓の追跡や部分的な表示の効率を上げ、また矩形領域の重りを許すことにより、局所演算の効率向上を計

ることをできる。

- (2) 計算センターでFORTRANを用いても簡単に読めるようにEBCDICコードを用いる。
- (3) ヘッダ部には重要な項目から順に並べ、不要な項目の値を0としておけば、デフォルトとして扱う。

2.1 ヘッダ部フォーマット

表1にヘッダ部フォーマットを示す(図1を参照)。以下に各項目を説明する。尚、MTから画像データを読み込むのに最低必要な項目に下線を施す。

表1 ヘッダ部フォーマット

項目番号	項目	データ長 (バイト数)	先頭位置	摘要
1	データ名	12	1	12文字未満の場合は後を空白とする
2	フレームの大きさ	4	13	FX, FYの順に各2バイト
3	サブフレームの大きさ	4	17	SX, SYの順に各2バイト
4	サブフレーム数	4	21	NSX, NSYの順に各2バイト
5	濃淡情報			NBIT, EBIT, LR, SCALEの順に各1バイト
	データビット数(NBIT)	1	25	8の約数/倍数。(NBIT=0 → NBIT=8)
	有効ビット数(EBIT)	1	26	EBIT ≤ NBIT (EBIT=0 → EBIT=NBIT)
	格納形式	1	27	0:右詰め, 1:左詰め
	濃淡表現	1	28	0:輝度α数値, 1:濃度α数値等
6	論理リード長(LLENG)	4	29	LLENG=0 → LLENG=SX*SY*NBIT/8
7	物理リード長(PLENG)	2	33	PLENG ≤ LLENG. (PLENG=0 → PLENG=LLENG)
8	画素配列	1	35	0:水平方向上から, 1:垂直方向左から, 2:水平方向下から
9	サブフレーム配列	1	36	同上
10	隣接情報	4	37	ML, MR, MU, MDの順に各1バイト
11	入力装置	6*	41	FSS L L L L L L ITV L L L L L L 等
12	入力方式	6*	47	単色/色彩, 単眼/ステレオ等
13	全視野の大小	4	53	PX, PYの順に2バイト
14	有効領域	8	57	EXS, EYS, EXE, EYEの順に各2バイト
15	フレームの始末	4	65	FXS, FYSの順に各2バイト
16	日付	6*	69	$\frac{MM}{日} \frac{DD}{月} \frac{YY}{年}$ (年は西暦のF2桁)
17	場数	6*	75	
18	拡張用予備	52	81	
19	コメント	380*	133	
計		512		

注) * : 文字コード(EBCDIC)

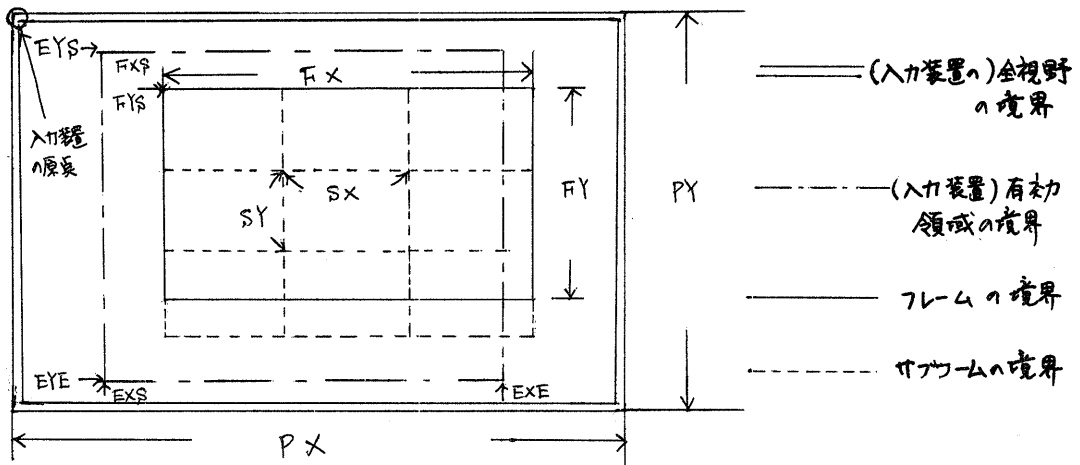


図1 画像データの切り出し

(1) データ名 (NAME)

EBCDICコード12文字以内とし、12文字未満の場合は後を空白とする。

(2) フレームの大きさ (FX , FY)

記録媒体上に存在するデータ領域をフレームとする。(FX , FY の単位は画素数)。

(3) サブフレームの大きさ (SX , SY)

大きさは画面で一度に X イン・ X エリで処理できればいい場合等の画面分割の単位。 SX , SY は必ずしも FX , FY の約数でなくともいい。この場合、右端下端のサブフレームにはフレーム外の画素を0として埋める。

(4) サブフレーム数 (NSX , NSY)

水平方向、垂直方向とそれぞれのサブフレームの個数を NSX , NSY とする。従って一画像の全サブフレーム数は $NSX * NSY$ となる。

(5) 濃淡情報

画像の濃淡に関する下記の4項目を順に各バイトずつ割り当てる。

5.1 データ・ビット数 ($NBIT$)

記録媒体上での画像の濃淡レベルのビット数 ($NBIT$) は8の約数/倍数のみを認める。約数の場合、1バイト中に $8/NBIT$ 画素分パックされているものとし、倍数の場合、 $NBIT/8$ バイトで一画素と考える。但し FX , FY , SX , SY 等はすべて画素数で与えるので、データ量は $NBIT/8$ 倍となる。デフォルトとして、 $NBIT=0$ は $NBIT=8$ として扱う。

5.2 有効ビット数 ($EBIT$)

画像の実際の濃淡レベルのビット数を $EBIT$ とする。これは8の約数/倍数でなければ切り上げて約数/倍数とし、これを $NBIT$ としている。 $EBIT \leq NBIT$ 。デフォルトとして、 $EBIT=0$ は $EBIT=NBIT$ として扱う。たとえば、($EBIT=6$, $NBIT=8$)、($EBIT=3$, $NBIT=4$) などの組合せがある。

5.3 格納形式 (LR)

$EBIT < NBIT$ のとき、有効ビットをデータビット領域内で右詰めにするか左詰めにするかの2通りがある。(図2参照) $EBIT = NBIT$ のときは、

LRの値は全く影響しない。

5.4 濃淡表現(SCALE)

各画素の数値のモフ意味をコード(数値)で与える。

SCALE = 0 輝度の数値
(e.g. NBIT = 8 のとき 黒 = 0 → 白 = 255)

SCALE = 1 濃度の数値
(e.g. NBIT = 8 のとき 白 = 0 → 黒 = 255)

その他 log scale, square, root 等適宜割り当てる。

(6) 論理レコード長 (LENGTH)

サブフレームを論理レコードの単位と考える。このデータ長は、LENGTH = SX * SY * NBIT / 8 (バイト) デフォルトで、LENGTH = 0 の場合はユーザーが、SX、SY、NBIT からこの値を計算してから、データ部を読む。

(7) 物理レコード長 (PLENGTH)

物理レコードは最大 4096 バイトとする。論理レコードがこれより大きい場合適宜に物理レコードで区切り、ZMT に入れる。LENGTH が PLENGTH で割り切れない時も、最終レコードの長さも PLENGTH とする。この場合、余白部は 0 とする。デフォルトで、PLENGTH = 0 の場合は PLENGTH = LENGTH として扱う。

(8) 画素配列 (PODR)

サブフレーム内の画素の配列順序をコード(数値)で与える。図3に配列順序を示すが、これ以外は適宜定める。

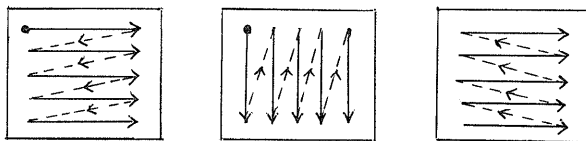
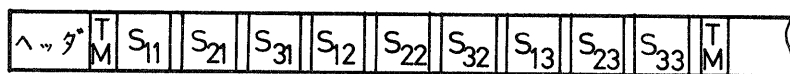


図3 画素配列のコード

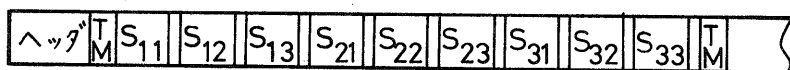
(9) サブフレーム配列 (WODR)

フレーム内のサブフレームの配列順序をコード(数値)で与える。図4に配列順序を示すがこれ以外は適宜定める。

WODR = 0 MT 上では



WODR = 1



WODR = 2

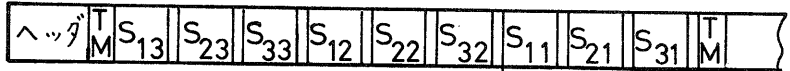
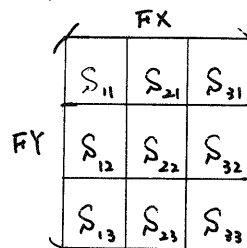


図4 サブフレーム配列のコード

例



(10) 隣接情報 (ML, MR, MU, MD)

大画面をサブフレーム単位に分割処理する場合、局所演算を行おうとサブフレームの端の部分の演算を行おうためには他のサブフレームのデータが必要となるため、処理の効率が下がる。これを防ぐため、あらかじめ何画素分か余分に隣接する部分をつけ加えておいて一つのサブフレームとする方法が考えられる。図5にML, MR, MU, MDを示す。なおSx, Syは、この隣接部分を含むものとするが、Fx, Fyは、画像本来のデータの大きさを示す。隣接情報を用いない場合はMU=MD=ML=MR=0としておくはよい。

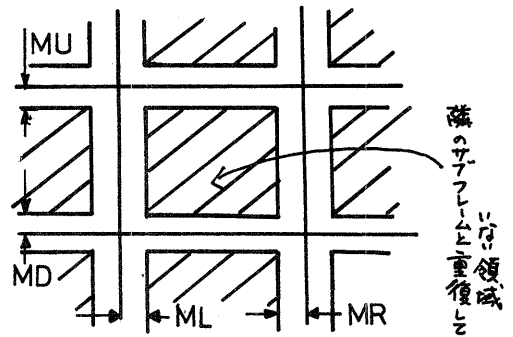


図5 隣接情報を含んだサブフレームの構成

(11, 12) 入力装置, 入力方式

特別なコードは定めないので適当な文字で識別できるようにする。

(13, 14, 15) 全視野の大きさ, 有効領域, フレームの視覚

図1に示されているが、これは入力装置の特性(位置による位置決め誤差, シェーディングなどが問題になり、これを補正する必要がある場合に用いられる。

(16, 17) 日付, 場所

いずれも文字で、場所などはどの研究室で入力したものがわかるように適当に定める。

(18) 拡張予備

将来ハッタ部フォーマットとして定めなければならない事項が全じた場合のための予備領域である。

(19) コメント

これまでの項目で書くことのできなかった情報を文章として文字で表わしてコメントとする。

2.2 MT上のフォーマット

前述のハッタ部と、画像データを図6に示すように配列する。MTは原則として9トラックで、記録密度は800bpiあるいは1600bpiとする。1本のMTに複数の画像を格納することを許し、最後の画像データの終りにはター

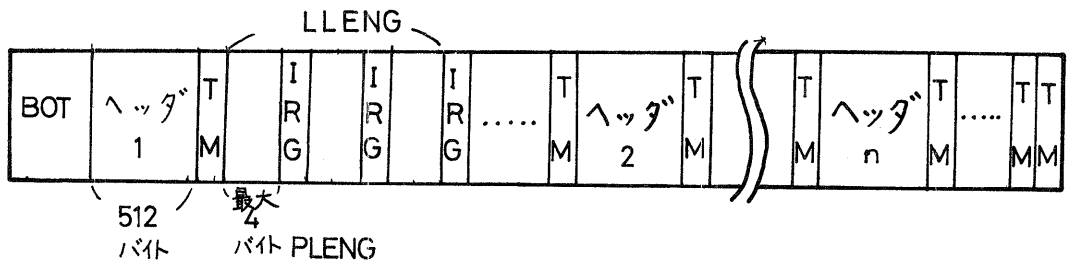


図6 MT上のフォーマット

マーク (TM) を2つ書入れる。ここでは1つの画像データが複数のMTにまたがっているマルチリールは許さばい。最初のヘッダ部の前には何か書かばい。レタが、2MTへのアクセスはラベルはし方式で行う。

2.3 例題

これまで述べた標準フォーマットで画像データを作成することは、表1を一見しただけでは複雑なように思われるが、実際に通常のデータを作ることは容易である。ここでは典型的な画像データを標準フォーマットでMTに書く例を示す。はじめの2例は比較的単純な例で、3番目の例は隣接情報を用いる複雑な場合を、最後にフォートランでMTを読むのに都合の良い例を説明する。

(1) ファクシミリ用標準データ

ファクシミリにおいて、各種帯域圧縮方式の圧縮率相互比較を含め、デジタルファクシミリシステムの特性評価のために標準データを作っておくと都合がよい。そこでCCITTのS-G-XIVが推奨しているファクシミリ帯域圧縮用テストチャートをファクシミリ送信機にかけ、その光電変換出力を2値に量子化、標本化して、MTに格納する場合を考える。テストチャートは9種類、いずれもA4版(210mm x 297mm)である。テストチャートのデジタル化条件は2種類推奨されているが、ここでは標本化密度7.7サンプリング/mm、走査線密度7.7本/mmを採用することにする。なお標準フォーマットのデータはテストチャート9枚分を同じ形式で作成する。このデータは図6のようにMT上に書込まれるが、各画像に関するヘッダ部の内容(データを除く)と画像データの記録方式は全く同じである。画像データは走査線1本毎に区切って一つのサブフレームとし、同時に一つの物理レコードとする。表2に最初のテストチャート(FAXTEST-A)のヘッダ部を示す。

表2 例題(1)のヘッダ部

項目番号	項目	値
1	データ名	FAXTEST-A
2	フレームの大きさ	1617, 2282
3	サブフレームの大きさ	1617, 1
4	サブフレームの数	1, 1617
5	濃淡情報	
	データビット数(NBIT)	1
	有効ビット数(EBIT)	1
	格納形式	0
	濃淡表現	1
	以下すべて0	

フレームの大きさは、 $FX = 7.7 \times 210 = 1617$, $FY = 7.7 \times 297 = 2282$, 9を切捨て、2282となる。1走査線がサブフレームになるので、 $SX = 1617$, $SY = 1$ となり、レタ

が、サブフレームは2282個である。濃淡は1ビットで表わされ、黒い部分を1とする。また濃淡データは連続して書込むことにすれば、 $NBIT = EBIT = 1$ となる。レタが、格納形式は不要になる。項目6~10はデフォールトを採用することができる。また項目11以下はこの場合には特に重要でないのど省略することにし、すべて0を入れておいてもかまわばい。

(2) 電総研バイオニクス研究室の例

電子技術総合研究所バイオニクス研究室では、画像データを同一フォーマットでディスクに蓄えて共同利用している。ディスク内のデータは、物体認識処理に直接用いられ、サブフレーム毎にコアメモリに伝送される。レタが、サブフレームは処理効率を上げるように作られている。この画像データを同じサブフレ

MTに書込む例のヘッダ部フォーマットを表3に示す。

フレームの大きさは全視野の大きさと同じで、つねに $FX = FY = 256$ とする。これは入力装置の分解能によって決定されている。サブフレームの大きさは、コアメモリの容量と、データ処理の効率との関係で決定されたが、通常はコアメモリ上にサブフレーム4個分のバッファ領域を設けておけば、前処理、追跡などの大部分の画像処理は二次記憶装置へのアクセス回数を大幅に減少させることができる。濃淡は6ビットが有効であるが、1つの濃淡データを8ビット、右づめで書込むことにしている。したがって格納形式以下はすべて0にしておいてもかまわない。

(3) 電総研図形処理研究室の例

電総研図形処理研究室ではMT上だけでなく、ディスク・ファイルや、主計算機と副計算機間のデータ転送における画像データにもこの標準フォーマットを採用することにした。画像データの一例を示すと、航空機からのリモート・センシング画像を、 $FX = FY = 1024$, $SX = SY = 128$, $NBIT = EBIT = 8$, $LENX = LENS = 8$, $LENGTH = 16384$ として扱っている。

この原画像を、局所並列演算に都合の良いように隣接情報を用いた形(表4)に変換している。一般に、 $(2n+1) \times (2n+1)$ の局所オペレータを作用させる際に画像の端の画素分は演算が出来ない。この解決法として図7のように、原画像のサブフレームから画素分余分に画像を切り出して演算後サブフレーム分(図7の矢線内)だけを再格納する方法が考えられるが、論理レコードがサブフレームであるのと同じ切り出しはやはり面倒である。部分画像の切り出し時間を短くするために、図8(斜線部が原画像では隣り合っている)のように画像を変換しておく方が、何度も局所演算を行なう場合は全体として効率が良い。

図8のような形への変換には、次

表3. 例題(2)のヘッダ部

項目番号	項目	値
1	データ名	DESKI
2	フレームの大きさ	256, 256
3	サブフレームの大きさ	32, 32
4	サブフレーム数	8, 8
5	濃淡情報	
	データビット数(NBIT)	8
	有効ビット数(EBIT)	6
	以下すべて	0

表4 例題(3)のヘッダ部

項目番号	項目	値
1	データ名	H1-IR
2	フレームの大きさ	1008, 1008
3	サブフレームの大きさ	128, 128
4	サブフレーム数	8, 8
5	濃淡情報	
	データビット数(NBIT)	8
	有効ビット数(EBIT)	8
	格納形式	0
	濃淡表現	1
6	論理レコード長(LENG)	16,384
7	物理レコード長(PLENG)	4096
8	画素配列	0
9	サブフレーム配列	1
10	隣接情報	1, 1, 1, 1
	以下すべて	0

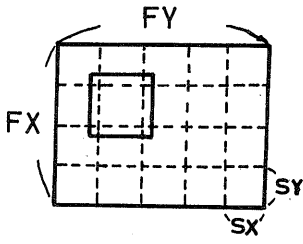


図7 サブフレームより大きき切り出し

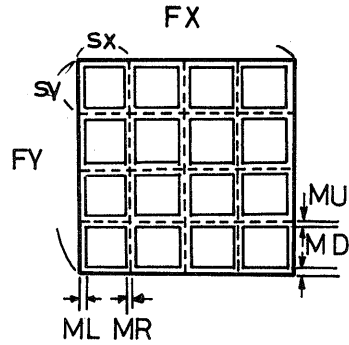


図8 隣接情報を含む画像の構成

の3つの方法が考えられる。前述の画像に 3×3 のオペレータを作用させるには、すべて $ML = MR = MU = MD = 1$ とせよ、

i) 新しいサブフレームを隣接情報分だけ大きくして $SX = SY = 130$ 、 $LENGTH = 16900$ とする。 FX, FY, NSX, NSY は変更なし。

ii) アクセス単位としてのサブフレームの大きさを SX, SY は変更せず、図7の斜線部を 126×126 とする。このため $FX = FY = 1024$ を 126 で割った余り16画素を含むためのサブフレームを右端と下端に追加して $NSX = NSY = 9$ とする。

iii) サブフレームの大きさの変更も追加もせず、原画像の上下左右の各端の8画素分を切り捨てる。即ち、 $FX = FY = 1008$ 。この場合、 $SX, SY, NSX, NSY, LENGTH$ は原画像と同じなので、ファイルの大きさを変更しなくて済む。表4はこのiii)を採用している。ただし、ディスク内ではMTのように $LENGTH \leq 4096$ の制限をつけたOSの標準に任せている。

ただし、いずれも原画像の端の部分だけは演算できないので、図5のように変換するとき原画像より外は0(又は適当な値)であるとしておく必要がある。

(4) 電統研作成のMTを名古屋大学計算センターで読んだ例。(フォートランで読み出す場合)

この標準フォーマットでは、ヘッダ部は512バイトと固定されているが、データ部の物理レコードはユーザーがヘッダ内で指定し、4096バイト以下であればよい。大型計算機のフォートランでこのMTを読み出す場合、計算機の種類によりその能力が異なるが、固定アロケーションが扱えないことが多い。この場合は、データ部の物理レコードをヘッダと同じく512バイトとするにより、アセンブラのプログラムを用意することなくフォートランのR

表5 例題(4)のヘッダ部

項目番号	項目	値
1	データ名	USC-M
2	フレームの大きさ	256, 256
3	サブフレームの大きさ	256, 2
4	サブフレーム数	1, 128
5	濃淡情報	
	データビット数(NBIT)	8
	有効ビット数(EBIT)	8
	格納形式	0
	濃淡表現	1
6	論理コード長(LENGTH)	0
7	物理コード長(LENGTH)	512
	以下すべて	0

EAD文だけで読み取ることもできる。

これを実際に試すため、南カリフォルニア大学より提供された標準画像データを電総研で表5の形でMTに書き(計算機はTOSBAC 5600/170)、これを名古屋大学計算機センターのFACOM 230-60のフォートラ上で読むことを行なった。