1　コンピュータにおける画像ファイル

コンピュータで扱うことのできるファイルは，基本的には 0 と 1 の情報で記述され，**バイナリデータ** (binary data) と呼ばれている．バイナリとは，２進数という意味である．したがって，文字，図形，画像等を 0 と 1 の記号に置き換えて保存しているのである．

一方，コンピュータは 1byte（8bit）を一つの単位としてデータを処理する．つまり 8 桁の 2 進数を基本として処理がなされている．例えば，この 1 バイトで整数を表現するとすれば，0〜255 の値を表現することができる．文字情報のみの**テキストデータ** (text data) も実はバイナリで表現されている．英数字は，ASCII **コード**によって，1 バイトの値と文字とが対応付けられている．

1byte を表現するのに，一般に 16 進数が用いられている．1byte（8bit）を 4bit づつに分割し，2 つの値で表す．4bit で 0〜15 の値を表現できるので，16 × 16 = 256 の表で表現できる．なお，16 進数は，10〜15 をアルファベット A〜F で表す．表 1 はその ASCII コード表である．例えば，ASCII コードにおける K は．上位バイト 4，下位バイト B に位置し，16 進数で 4B，10 進数で 75，2 進数で 1001011 としてコンピュータ上では表現される．

表 1　ASCII コード表

		上位バイト							
		0	1	2	3	4	5	6	7
下位バイト	0	NUL	DLE	SP	0	@	P	‘	p
	1	SOH	DC1	!	1	A	Q	a	q
	2	STX	DC2	”	2	B	R	b	r
	3	ETX	DC3	#	3	C	S	c	s
	4	EOT	DC4	$	4	D	T	d	t
	5	ENQ	NAC	%	5	E	U	e	u
	6	ACK	SYN	&	6	F	V	f	v
	7	BEL	ETB	’	7	G	W	g	w
	8	BS	CAN	(	8	H	X	h	x
	9	HT	EM	)	9	I	Y	i	y
	A	LF/NL	SUB	*	:	J	Z	j	z
	B	VT	ESC	+	;	K	[	k	{
	C	FF	FS	,	<	L	\	l	—
	D	CR	GS	-	=	M	]	m	}
	E	SO	RS	.	>	N	^	n	~
	F	SI	US	/	?	O	_	o	DEL

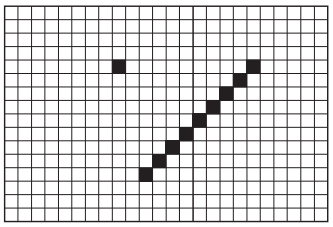
テキストファイルとは，この ASCII コードにしたがって，記述されているファイルを指している．

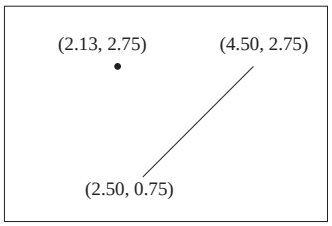
ところで，漢字は文字の種類が極めて多いので，1byte では表現しきれない．そこで，2byte のコードで漢字を表現しており，別のコードで漢字の対応表が規定されている．現在は，JIS，Shift-JIS, EUC, 区点等のコードがある．Windows 系の OS では Shift-JIS コードが，携帯電話などでは区点コード，UNIX 系の OS では EUC コードが採用されており，適宜コード一変換が必要である．多くの電子メールは，メールソフト自身が漢字コード変換の機能を有しているため，ローカルの OS に対応した漢字変換が自動的になされる．

画像ファイルは，通常バイナリ形式で格納されており，明るさ（輝度）の情報を数値で示している．

1.1 ラスタデータとベクタデータ

画像のデータモデルの形式は大きくラスタ型とベクタ型に分類することができる．下図に各データモデルそれぞれの概念図を示す．





ラスタ型のデータは，格子状に対象領域を区切り，その格子点毎にデータが納められている形式である．画像の取得に用いられる CCD 等を用いたセンサは，通常ラスタデータを出力する．位置情報は直接データ内に表現されていないが，格子点の座標から投影面での位置座標を導くことができる．格子点の座標は，左上点が原点となり，右方向が列番号 u，下方向が行番号 v として扱っているのが普通である．数学における座標は，X 座標を右方向にとると，Y 座標は上方向にとる右手系である．しかし，ラスタデータにおける UV 座標の V は下向きとなっており，左手系である．テレビは，かつてブラウン管と呼ばれる装置で左上から右下に向かって光を捜査して映像を表現していた．つまり左上点が原点だったわけである．このようなことから画像においては，左手系の座標が使われている．なお，列は column，行は row である．表計算ソフトにおけるデータも左上から右下に向かって表現されるので，この点においては矛盾はない．画像において，格子点は画素 (pixel) に相当するが，この画素を表計算ソフトで例え，セル (cell) と呼ばれることもある．

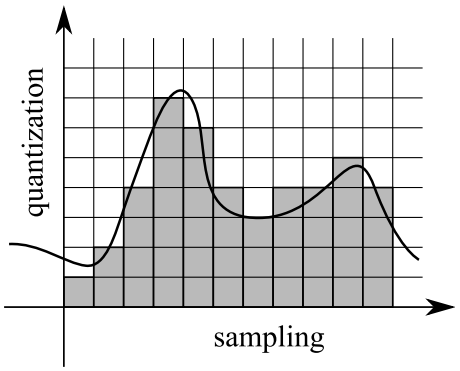
ベクタ（ベクトル）型は，対象物の座標情報と属性情報（点か線か多角形か？線の太さは？等，データに付随する情報）が直接納められている．地理情報システムではベクタ型を扱うことが多い．施設等を点で，道路等を線で，区画等を面で表現するためである．これらについては，地理情報システムの章で解説する．

位置情報の精度は，ラスタ型の場合，格子点間隔を細かくすることで高精度化が可能であるが，高精度化に伴ってデータ量が膨大なものとなる．したがって対象領域が広く非常に高い精度が要求されるデータのモデルとしては適さない．これに対してベクタ型は，位置座標を直接記述することができるため簡単に高精度のデータを構築することが可能である．データのハンドリングに関しては，ラス

タ型の方が有利である．位置座標が固定化されているため構造が単純で，ある空間におけるデータを時系列処理するとき等，変化の状況を把握するには都合のいいデータ型といえる．

1.2 量子化と標本化

アナログの情報をデジタル情報に変換することを A/D **変換** (analog-digital conversion) と呼んでいる．コンピュータで扱える画像は二次元のデジタル情報であるが，音等の一次元のデータを例に標本化と量子化について解説する．下図は，連続して変化している現象を示したものである．音でいえば，時間的な音の強さの変化を図示したものと考えても良い．横軸方向が時間で，縦軸方向が音の強さを表しているものとする．



この音の情報は，時間と伴に連続的に変化しているが，デジタルデータとして保存するときは，ある一定時間間隔ごとにデータを区切って表現することになる．このように領域を区切ってデータを取得することを**標本化**（sampling）と呼ぶ．領域の間隔が細かいほど，品質が高いものとなる．次に，区切ったデータごとの音の強さの値も同時に取得しなければならない．このときも一定間隔ごとに区切ってデータを取得する．例えば，データを 1 バイトの範囲で表現するときは，強さを 256 段階に区切ってデータの値を表現することになる．このことを**量子化**（quantization）と呼んでいる．データの品質は，標本化だけでなく量子化も重要な要素となる．

画像は，ある領域を格子で区切り，その格子点ごとに輝度の情報が収められている．このとき，格子で領域を区切って情報を取得することが標本化である．そして，その各格子における明るさの情報を数値に変換することが量子化となる．なお，標本化された一つ一つの格子は**画素**（pixel）と呼ばれている．

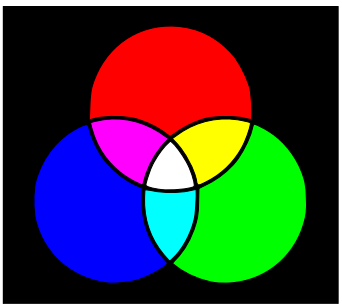
画像は，一般に CCD や CMOS 等の撮像素子を用いて標本化がなされる．デジタルカメラであれば，その撮像素子の数が多いほど分解能の高い画像となる．つまり，標本化は画像の分解能に直接影響を与える．一方，輝度情報は，量子化ビットの数によって輝度レベルの細かさが決まる．例えば，量子化ビットが 8 ビットの場合は，256 階調で輝度が表現される．高価なデジタルカメラであれば，量子化ビットは 10 ビット以上のものもあり，階調の高い画像が取得できる．つまり，量子化は画像の階調に直接影響を与える．

1.3 色の表現

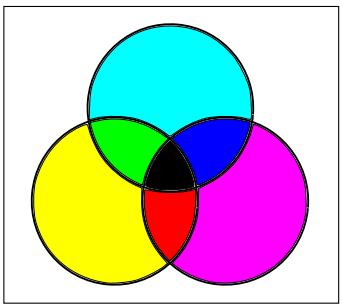
コンピュータのモニタにおいて，色を表現するには，光の 3 原色である赤 (R)，緑 (G)，青 (B) の組み合わせを利用する．人間の目で認識できる光，いわゆる可視光の電磁波の波長は，0.4〜0.7μ m であった．長い波長から順に赤，緑，青の三つに区切って，それぞれの色の強さが色情報となる．

それぞれの色について量子化された輝度情報が，色として表現されるのである．例えば，量子化ビットが 8 ビットの情報において，R=0, G=0, B=0 のときは黒，R=128, G=128, B=128 のときは灰，R=255, G=255, B=255 のときは白となる．このような，色の表現法を**加法混色** (additive color mixing) という．

通常のモニタは RGB 各 8bit の階調を映し出すことができ，その色の数は，256^3 =16,777,216 色となる．人間の目で識別するのには，十分な色数と言え，このように RGB 各 8bit の階調で表現される画像は，フルカラー画像とも呼ばれる．なお，コンピュータのビデオメモリ（VRAM）には，RGB8bit づつに対応していないものもある．例えば，3 万 2000 色しか表示できないコンピュータもあり，この場合 RGB 各 4bit の階調（32 階調）しかビデオ用のメモリを持っていない．



additive color mixing



subtractive color mixing

アナログテレビは，RGB 信号が直接電波に乗って受信されていると思いがちであるが，電波で送られてくる信号は YCC と呼ばれる表現方法である．白黒テレビとの互換性が必要とされていたため，明るさの情報と色の情報を分けて表現している．YCC のうち，Y が明るさ（輝度）を表し，色は C_1, C_2 で表される．RGB を YCC で表すと以下のようになる．

$$\begin{cases} R &= Y + C_1 \\ G &= Y - \dfrac{0.2999C_1 + 0.114C_2}{0.587} \\ B &= Y + C_2 \end{cases} \tag{1}$$

また逆変換は，以下のようになる．

$$\begin{cases} Y &= 0.299R + 0.587G + 0.114B \\ C_1 &= R - Y = 0.701R - 0.587G - 0.114B \\ C_2 &= B - Y = -0.299R - 0.587G + 0.886B \end{cases} \tag{2}$$

なお，デジタルテレビは，RGB の動画を圧縮した信号を受信している．

ところで，プリンターやカラーコピーで色を表現するには，色彩の三原色であるシアン (C)，マゼンダ (M)，イエロー (Y)，ブラック (B) の CMYB の組み合わせを利用する．白い紙に各トナーを塗ることによって色が表現されるため，明るさを減らして表現していることから**減法混色** (subtractive color mixing) と呼ばれている．

色の表現法には，その他に HSI による方法もある．H は色相（Hue），S は彩度（Saturation），I は明度（Intensity）と呼ばれ，YCC と HSI との関係は，以下のようになる．

$$\begin{cases} H & = \tan^{-1} \frac{C_1}{C_2} \\ S & = \sqrt{C_1^2 + C_2^2} \\ I & = Y \end{cases} \tag{3}$$

衛星画像には，可視光の RGB のデータを取得していないセンサもあり，赤外域の目に見えない光を取得している場合が多い．可視光の RGB のデータを取得している場合には，それぞれのデータを RGB に割り当ててカラー表示すれば現実に近いカラーで表現される．このようなカラー表現は**トゥルーカラー画像** (true color image) と呼ばれている．一方可視光のうち，青 (B) は，大気の影響を非常に受けやすいので，B のデータを取得していないセンサも多い．そのような場合は，近赤外のデータを R，可視の赤のデータを G，可視の緑のデータを B に割り当ててカラー化する場合が多い．このようなカラー表現は，**フォールスカラー画像** (false color image) と呼ばれている．森林等に覆われている山が赤く表現されている衛星画像を見かけることがあると思うが，このような画像は，近赤外のデータを R に割り当てたフォールスカラー画像である．この他にも様々な組み合わせで色を表現することができるが，現実とは違うカラー表現は，フォールスカラー画像と呼ばれる表現である．

1.4 画像フォーマット

画像ファイルに最低限必要な情報は，全画素における輝度の情報である．カラー画像であれば，一つの画素について RGB，３つの輝度情報が必要である．したがって，単純に考えて，カラー画像のファイル容量は，モノクロ画像の 3 倍必要となる．通常の画像ファイルは，この輝度情報が画像の左上から順に記録されている．

基本的には輝度情報の他に，画像を構成するピクセルの数やチャンネル数が画像ファイルに納められているが，画像の質，利用目的，ファイルの圧縮性に応じて様々なフォーマットが存在する．利用頻度の多いファイルフォーマットについて，以下に説明する．

1.4.1 BMP (Bitmap)

Windows 系の OS において主に利用されているフォーマット，256 色とフルカラーでの表現が可能である．ただ，画像圧縮の機能がないため，ファイルの容量が非常に大きい．拡張子は BMP となる．

1.4.2 TIFF (Tag Image File Format)

Aldus 社が考案し，様々なアプリケーションで利用されているフォーマット，256 色とフルカラーでの表現に加えて，ベクトルデータの保存も可能である．LZW と呼ばれる圧縮方式によってデータ

が圧縮される．この LZW 方式での圧縮は，可逆圧縮と呼ばれ，圧縮後に画像を復元する際，データは元の状態に戻る．拡張子は TIF となる．

1.4.3 JPEG (Joint Photographic Expert Group)

ISO でも定められたカラー画像フォーマット．フルカラーで表現された画像に適している．離散コサイン変換 (DCT) を利用した画像圧縮の機能を有し，極めて高い圧縮性を実現している．そのため，ネットワーク上ではこのフォーマットの画像がよく利用されている．ただし，この圧縮方法は，非可逆圧縮のため，圧縮後に画像を復元する際，画像が劣化する．高画質を維持するためには圧縮性を低く，低画質で良ければ圧縮性を高く設定できる．なお，離散コサイン変換 (DCT) は 8×8 画素の空間に対して実行されるため，非常に細くコントラストの高い線が 8×8 画像中にある場合は，それが滲んでしまう（ブロックノイズ）．したがって，図形が表現されている画像には向かない．

1.4.4 GIF (Graphic Interchange Format)

アメリカのネットワーク会社 Compu Serve が考案したフォーマット．全ての画像は，256 色で表現されるため，フルカラーの画像は劣化する．ただし，JPEG で発生するブロックノイズがないことから，ドロー系のソフトウェアで作成された図形画像の保存には，積極的に利用されている．

1.4.5 HDF

イリノイ大学において提案された科学データ向けのフォーマット．Hierarchical Data Format の略で，階層構造となったデータも表現できる．多くの衛星データは，この HDF 形式で配布されている．HDF データを読み込んだり，書き込んだりする場合のプログラムライブラリがイリノイ大学 NCSA(National Center for Supercomputing Applications) のサイトに公開されている．自作プログラムで扱う場合には，このライブラリを利用するのが簡単である．

1.4.6 RAW

廉価なデジタルカメラにおいては，JPEG 形式や TIFF 形式が採用されているものがほとんどであるが，高価なものになると RAW 形式での保存も可能となっている．高価なデジタルカメラにおいては，量子化の際に 10bit や 12bit の輝度情報で保存される．通常の画像フォーマットは 8bit なので，JPEG や TIFF ではカメラの性能よりも劣る画像で保存されてしまう．このためカメラで取得された生の情報を保存する機能として RAW 形式が存在する．ただ，RAW 形式は各社様々なフォーマットで記録されるため，カメラメーカーの提供している特殊なソフトウェアを利用するか，フォーマット仕様書を見て独自で読み込むプログラムを作成しなければならない．ただ，RAW 形式は単純なフォーマットで記録されているものがほとんどなので，複雑なプログラムとはならない．RAW 形式の画像を読み取るには，画像サイズ (Column 数と Row 数)，輝度情報のデータサイズ（バイト数)，RGB データの並び順，画像ファイルにおける画像データの開始位置が解れば良い．RGB の並び順については，次に示すように 3 種類存在する．

1.4.7 BIP 形式

BIP は，Band Interleaved by Pixel の略で，ピクセルごとに RGB 情報が並んでいる形式である．つまり，画像の左上の一番目から，R, G, B, R, G, B, R, G, B, ... と並んでいる形式である．BMP 形式でも採用されているものである．ただ，BMP 形式は画像の右下の画素を一番目として，逆順にデータが格納されている．

1.4.8 BIL 形式

BIL は，Band Interleaved by Line の略で，ラインごとに RGB 情報が並んでいる形式である．

1.4.9 BSQ 形式

BSQ は，Band Sequential の略で，画像ごとに RGB 情報が並んでいる形式である．

下図に，4 × 4 画素の場合の BIP，BIL，BSQ の並び順の概念を示す．