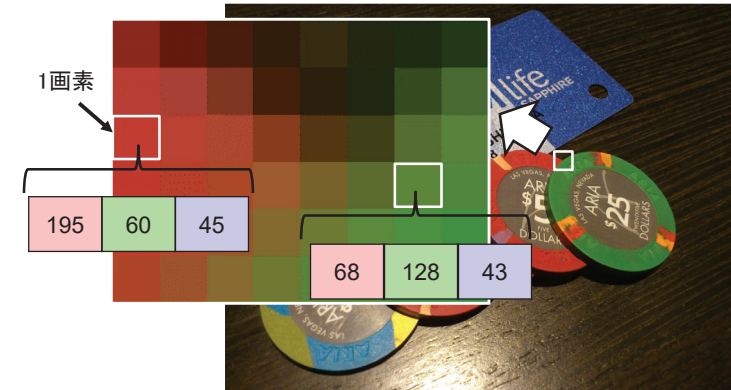


テキストファイルとバイナリファイル

1

デジタル画像データの基本

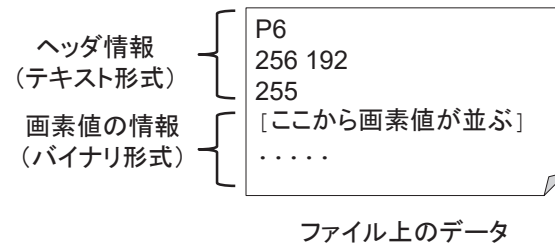
- 画素ごとに赤, 緑, 青の3色の情報を持っている
- 赤, 緑, 青(RGB)の組み合わせで色を表現



2

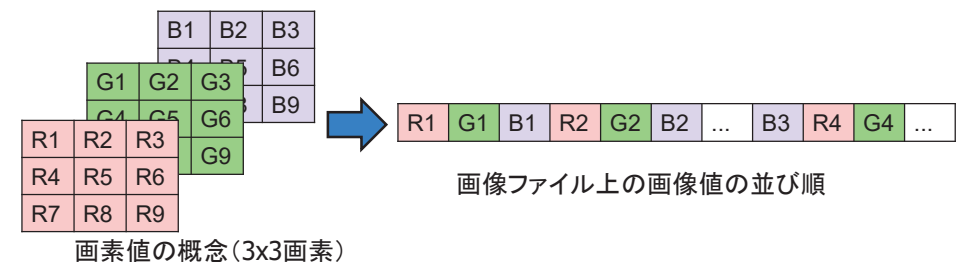
PPM画像(P6)

- 画像に関する情報をヘッダとして持つ画像
 - 縦画素数と横画素数の情報が含まれる
 - 余計な情報が少ないため, 処理しやすい



画素値の情報

- 基本的にR, G, Bの順に数値が一行に並ぶ
 - メモリは1次元なので, 1次元に展開される
 - 画素値はテキストではなく**バイナリデータ**として保存



4

テキストファイルとバイナリファイル

■ テキストファイル

- 一般的には、ASCIIコード表で文字に変換できる数値(0x00から0x7F, つまり符号なし整数だと0から127)のみが含まれるファイル

■ バイナリファイル

- それ以外のファイル
- PPMファイル(P6)はバイナリファイル

テキストファイル

■ メモリ上はバイナリデータ(数値)が保存される

■ テキストエディタで開くと、ASCIIコードに対応している文字を表示

	0	1	2	3	4	5	6	7
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	NAK	%	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	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

エディタで開いた
test.txt

hello



ASCIIコード表

アドレス	...	11	12	13	14	15	16	17	...
メモリ上の数値	...	?	68	65	6C	6C	6F	?	...

(注: 以後、アドレスとメモリ上の数値は16進数を意味し、0x表記は省略)

バイナリファイル

■ メモリ上はバイナリデータ(数値)が保存される

■ テキストエディタで開いたときに、 ASCIIコードの文字に対応する とは限らない

エディタで開いた
img01.ppmの
画素値の部分

+ ,



	0	1	2	3	4	5	6	7
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	NAK	%	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	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

ASCIIコード表

アドレス	...	11	12	13	14	15	16	17	...
メモリ上の数値	...	?	FF	F3	2B	99	60	?	...

偶然+に対応

偶然'に対応

PPMファイルはバイナリファイル

■ 1画素の各色を1バイトで表現

■ 画素値は0から255(0x00から0xFF)なので、 ASCIIコード表の文字とは対応しない

■ 画素値は"テキスト"としては読み込めない

テキストエディタで
開いたimg01.ppm

P6
256 192
255
}}/ ' aa#" _// ...

} ヘッダはテキストデータ

} 画素値の部分は意味を持った
テキストにならない

10	11	12	13	14	...
E3	7D	7D	2F	C5	...

画素値の部分のメモリの状態

1画素のRの値(1バイト)