

# 情報の符号化

## 情報科学 (2)

2022.4.15 N. Shinoda

# 本回の概要

## 概要

今回は、現在使われているメディアのデジタル表現の基礎となり、高等学校の情報科の内容とほぼ重なる内容の復習を主とする。

## 目標

情報の符号化の視点で、基礎的なデジタル表現を理解する。



# 1. 二進数 (二元符号)

情報の符号化

情報のやりとり

文字 → 2進数 (二元符号) → 文字

画像 → 2進数 (二元符号) → 画像

音 → 2進数 (二元符号) → 音

動画 → 2進数 (二元符号) → 動画

**二元符号**とは

2種類の記号を使って何かを表す

0/1, 表/裏 など

アルファベットは26(52)元符号

(だいたい)

10進数の数え方	2進数の数え方	8ビットの2進数	ランプの点灯で示すと
0	0	00000000	●●●●●●●●
1	1	00000001	●●●●●●●●
2	10	00000010	●●●●●●●●
3	11	00000011	●●●●●●●●
4	100	00000100	●●●●●●●●
5	101	00000101	●●●●●●●●
6	110	00000110	●●●●●●●●
7	111	00000111	●●●●●●●●
8	1000	00001000	●●●●●●●●
9	1001	00001001	●●●●●●●●
10	1010	00001010	●●●●●●●●
11	1011	00001011	●●●●●●●●
12	1100	00001100	●●●●●●●●
⋮	⋮	⋮	⋮
99	01100011	00001111	●●●●●●●●
100	01100100	00010000	●●●●●●●●
101	01100101	00010001	●●●●●●●●
⋮	⋮	⋮	⋮
254	11111110	11111110	●●●●●●●●
255	11111111	11111111	●●●●●●●●
256	100000000		
257	100000001		
⋮	⋮		

↑  
8ビットの2進数では11111111 (255)までしか数えられません。 ED2



# 1. 二進数 (二元符号)

## 2進数とn進数

2進数：0と1だけで数を表す方法

10進数：0から9の10種類の記号（文字）を使って数を表す方法

8進数：0から7の8種類の記号（文字）を使って数を表す方法

16進数：0から9とAからFの16種類の記号（文字）を使って数を表す方法

n進数とは数値  $x$  が

$$x = \dots + a_3 \times n^3 + a_2 \times n^2 + a_1 \times n^1 + a_0 \times n^0 \quad (a_0, a_1, \dots < n)$$

であるとき、 $\dots a_3 a_2 a_1 a_0$  とする表記のこと

例)

$$10\text{進数 } 123_{(10)} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 100 + 20 + 3 = 123_{(10)}$$

$$2\text{進数 } 1101_{(2)} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13_{(10)}$$



# 1. 二進数 (二元符号)

## 2進数の整数

1 byte (バイト) (=8 bit (ビット)) で表現できる整数の範囲

$0000\ 0000_{(2)} \sim 1111\ 1111_{(2)}$

8桁の0と1の組合せは

$2^8 = 256_{(10)}$ 種類 (これを数値  $0 \sim 255_{(10)}$  または  $-128_{(10)} \sim +127_{(10)}$  として表現する)

2 byte (=16 bit) では

$0000\ 0000\ 0000\ 0000_{(2)} \sim 1111\ 1111\ 1111\ 1111_{(2)}$

$2^{16} = 65,536_{(10)}$  ( $0 \sim 65,535_{(10)}$  または  $-32,768_{(10)} \sim +32,767_{(10)}$  として表現する)

※大きな数を表現するには多くの桁が必要

※本資料では、 $256_{(10)}$ は10進数での表記、 $1111\ 1111_{(2)}$ は2進数での表記を示す。



# 1. 二進数 (二元符号)

## 練習問題

### 問題1-1

以下の数値を10進数で表しなさい

$$0111\ 1110_{(2)} =$$

$$7E_{(16)} =$$

$$176_{(8)} =$$



## 2. 文字の表現

文字コード

「文字コード」

1文字に数値を1対1に対応させた時の数値を文字コードという。

どの文をどの数値をと対応させるかは任意だが、共通させていないと共用できない。

「文字コード表」

利用する文字群を対応させた文字コードの表

文字のデジタル化の例

JIS X 0201コード表に基づいて、1文字ずつ文字コードに置き換える。

This is a pen.

文字コード  
(10進数表記)

84 104 105 83 32 105 83 32 97 32 112 101 110 46

文字列データ

84,104,105,83,32,105,83,32,97,32,112,101,110,46

2進数の文字列データ

1010100,1101000,1101001,1010011,0100000,1101001,1010011,0100000,1100001,0100000,1110000,1100101,1101110,0101110

パソコンに記憶



ED2





## 2. 文字の表現

### 文字コード表

**JIS 8bit コード表** (=ASCIIコード+半角カタカナ等) = JIS X 0201コード表

※ JIS: 日本工業規格 (Japanese Industrial Standard)

※ ASCII: American Standard Code for Information Interchange

**JIS 16bit コード表** (情報交換用として使われる)

漢字, 英数字, ひらがな, カタカナ, 記号等

**shift JIS コード表**: パソコンで古くから使われていた

**euc コード表**: ワークステーション等で使われていた

**UNICODE 表**: 全世界で使われている/いた文字を一つの表にする





# 2. 文字の表現

JIS8bit/ASCII コード

ASCII コードの例

文字 '0' =  $00110000_{(2)} = 48_{(10)}$

文字 '1' =  $00110001_{(2)} = 49_{(10)}$

文字 'A' =  $01000001_{(2)} = 65_{(10)}$

文字 'a' =  $01100001_{(2)} = 97_{(10)}$

文字 'P' =  $01010000_{(2)}$

文字 'e' =  $01100101_{(2)}$

文字 'n' =  $01101110_{(2)}$

		上位		下位															
		10進	16進	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10進	16進	2進	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
0	0	0000	NUL	DLE		0	@	P	`	p				-	9	!			
1	1	0001	SOH	DC1	!	1	A	Q	a	q				.	7	F	h		
2	2	0010	STX	DC2	"	2	B	R	b	r				!	4	7	k		
3	3	0011	ETX	DC3	#	3	C	S	c	s				!"	5	8	l		
4	4	0100	EOT	DC4	\$	4	D	T	d	t				."	6	9	m		
5	5	0101	ENQ	NAC	%	5	E	U	e	u				-.	7	0	n		
6	6	0110	ACK	SYN	&	6	F	V	f	v				./	8	1	o		
7	7	0111	BEL	ETB	'	7	G	W	g	w				/.	9	2	p		
8	8	1000	BS	CAN	(	8	H	X	h	x				0.	0	3	q		
9	9	1001	HT	EM	)	9	I	Y	i	y				1.	1	4	r		
10	A	1010	LF	SUB	*	:	J	Z	j	z				2.	2	5	s		
11	B	1011	VT	ESC	+	;	K	[	k	[				3.	3	6	t		
12	C	1100	FF	FS	,	<	L	\	l	l				4.	4	7	u		
13	D	1101	CR	GS	=	=	M	]	m	]				5.	5	8	v		
14	E	1110	SO	RS	.	>	N	^	n	-				6.	6	9	w		
15	F	1111	SI	US	/	?	O	_	o	DEL				7.	7	0	x		



## 2. 文字の表現

表の読み方

大文字 'A' の文字コードは：

二進数の上位4桁=0100

二進数の下位4桁=0001

$$0100\ 0001_{(2)} = 2^6 + 2^0 = 65_{(10)}$$

16進数の上位1桁=4

16進数の下位1桁=1

$$41_{(16)} = 4 \times 16^1 + 1 \times 16^0 = 65_{(10)}$$

※0,1行目は「制御文字」(後述)

		上位	10進	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		16進	2進	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
下位	10進	16進	2進	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0000	NUL	DLE	0	0	0	P	-	p								
1	1	1	0001	SOH	DC1	1	1	A	Q	a	q			.	7	f	k		
2	2	2	0010	STX	DC2	2	2	R	b	r				「	4	g	l		
3	3	3	0011	ETX	DC3	3	3	C	S	c	s			」	5	h	m		
4	4	4	0100	EOT	DC4	4	4	D	T	d	t			.	2	3	4		
5	5	5	0101	ENQ	NAC	5	5	E	U	e	u			-	3	4	5		
6	6	6	0110	ACK	SYN	6	6	F	V	f	v			?	4	5	6		
7	7	7	0111	BEL	ETB	7	7	G	W	g	w			?	5	6	7		
8	8	8	1000	BS	CAN	8	8	H	X	h	x			+	6	7	8		
9	9	9	1001	HT	EM	9	9	I	Y	i	y			?	7	8	9		
10	A	A	1010	LF	SUB	:	:	J	Z	j	z			1	3	4	5		
11	B	B	1011	VT	ESC	+ :	:	K	[	k	[			2	4	5	6		
12	C	C	1100	FF	FS	<	<	L	\	l	l			+	3	4	5		
13	D	D	1101	CR	GS	=	=	M	]	m	]			3	4	5	6		
14	E	E	1110	SO	RS	>	>	N	^	n	-			3	4	5	6		
15	F	F	1111	SI	US	/	?	O	_	o	DEL			+	4	5	6		



## 2. 文字の表現

### 練習問題

#### 問題2-1

Fukushima の文字コードを10進数, 16進数, 2進数で表しなさい。

#### 問題2-2

ASCII文字の大文字と小文字を交換したい。文字コードとして扱う方法を説明しなさい。

		上位		下位																				
		10進	16進	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15					
10進	16進	2進	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	A	B	C	D	E	F
0	0	0000	NUL	DLE		0	@	P	`	p				-	9	!								
1	1	0001	SOH	DC1	!	1	A	Q	a	q			,	7	F	h								
2	2	0010	STX	DC2	"	2	B	R	b	r			!	i	g	k								
3	3	0011	ETX	DC3	#	3	C	S	c	s			,"	u	f	l								
4	4	0100	EOT	DC4	\$	4	D	T	d	t			,	z	t	y								
5	5	0101	ENQ	NAC	%	5	E	U	e	u			,	2	7	2								
6	6	0110	ACK	SYN	&	6	F	V	f	v			7	h	c	3								
7	7	0111	BEL	ETB	'	7	G	W	g	w			7	l	3	7								
8	8	1000	BS	CAN	(	8	H	X	h	x			4	7	2	7								
9	9	1001	HT	EM	)	9	I	Y	i	y			9	7	7	h								
10	A	1010	LF	SUB	*	:	J	Z	j	z			1	3	h	h								
11	B	1011	VT	ESC	+	;	K	[	k	[			h	7	h	0								
12	C	1100	FF	FS	,	<	L	\	l	l			4	7	7	7								
13	D	1101	CR	GS	-	=	M	]	m	]			3	3	h	7								
14	E	1110	SO	RS	.	>	N	^	n	-			3	h	h	'								
15	F	1111	SI	US	/	?	O	_	o	DEL			4	7	7	'								



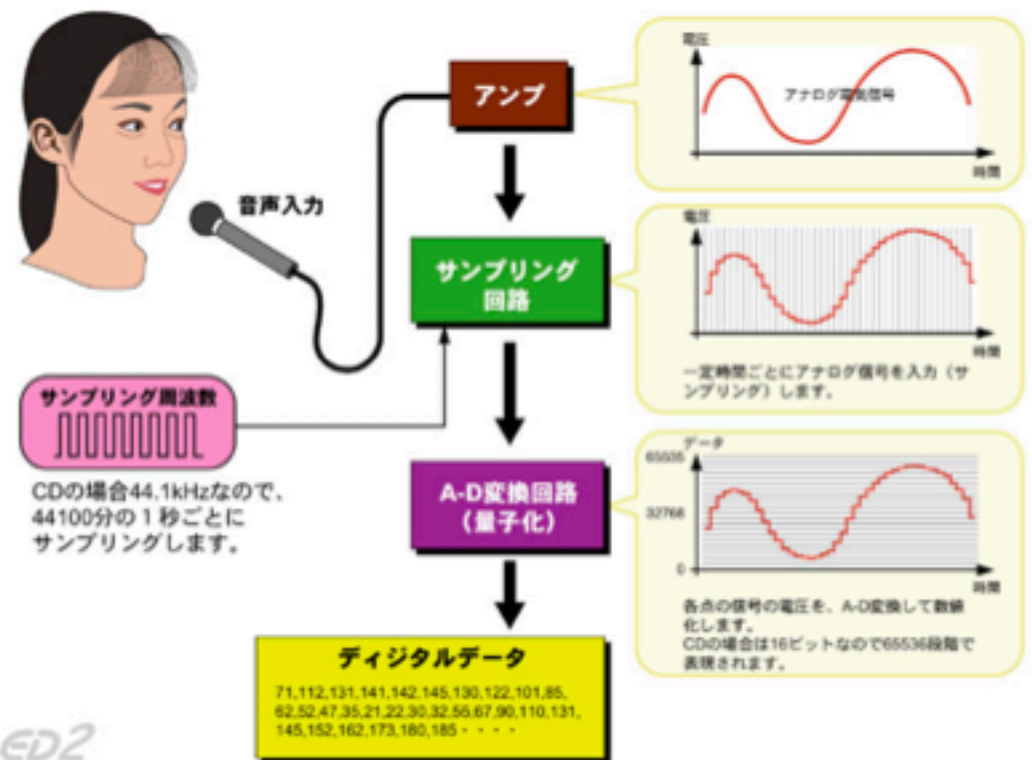
# 3. 音の表現

音の大きさを表現した数値の列  
サンプリング音 (ADPCM 等)  
音の大きさを数値で表現

用語

**サンプリング**: 時間軸の離散化  
**量子化**: 電圧軸の離散化

※ 「空気振動」を数値として表現している。



ED2



# 3. 音の表現

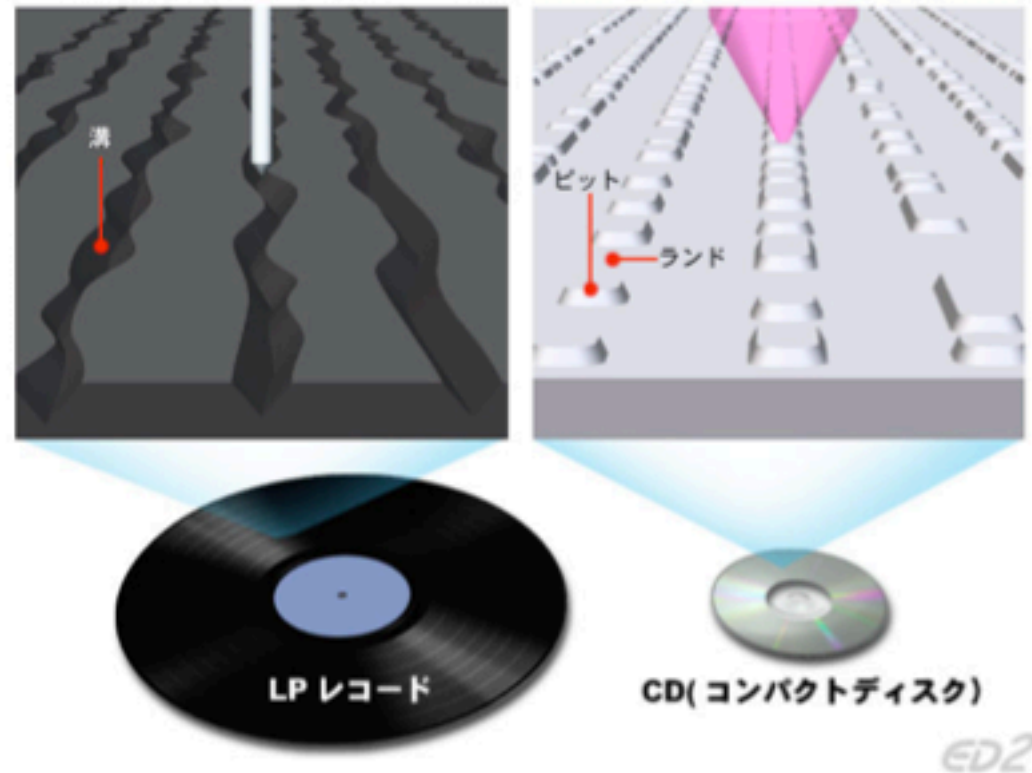
## 音楽CDの例

サンプリング：44,100回/秒  
周波数で表現すると 44.1KHz  
量子化：16bit（2進数16桁）

### サンプリングの定理

44.1KHzでサンプリングした音を再生すると、22.05KHzの高さの音まで正確に再生できる。

人間が音として聞こえる空気振動の範囲は15Hz~20KHz程度（個人差はある）なので、音楽CDで正確に再生できる音の範囲は、人間の可聴範囲の音域を含んでいる。



# 3. 音の表現

## 楽譜の表現

音そのものではなく「音楽」を符号化する表現する方法（MIDIなど）

楽器の種類，音符の長さ，音の大きさ等を数値で表現する。

※サンプリングした音よりもデータ量としては少なくなる。



# 1

## 練習問題

### 問題3-1

物理現象としての音とは何か。音の「大きさ」と「高さ」について説明しなさい。

### 問題3-2

人間の感覚としての音について、物理現象との関連を含め説明しなさい。

### 問題3-3

「犬笛」とは何か、その特徴、発する周波数との関係も含めて説明しなさい。





## 4. 画像の表現

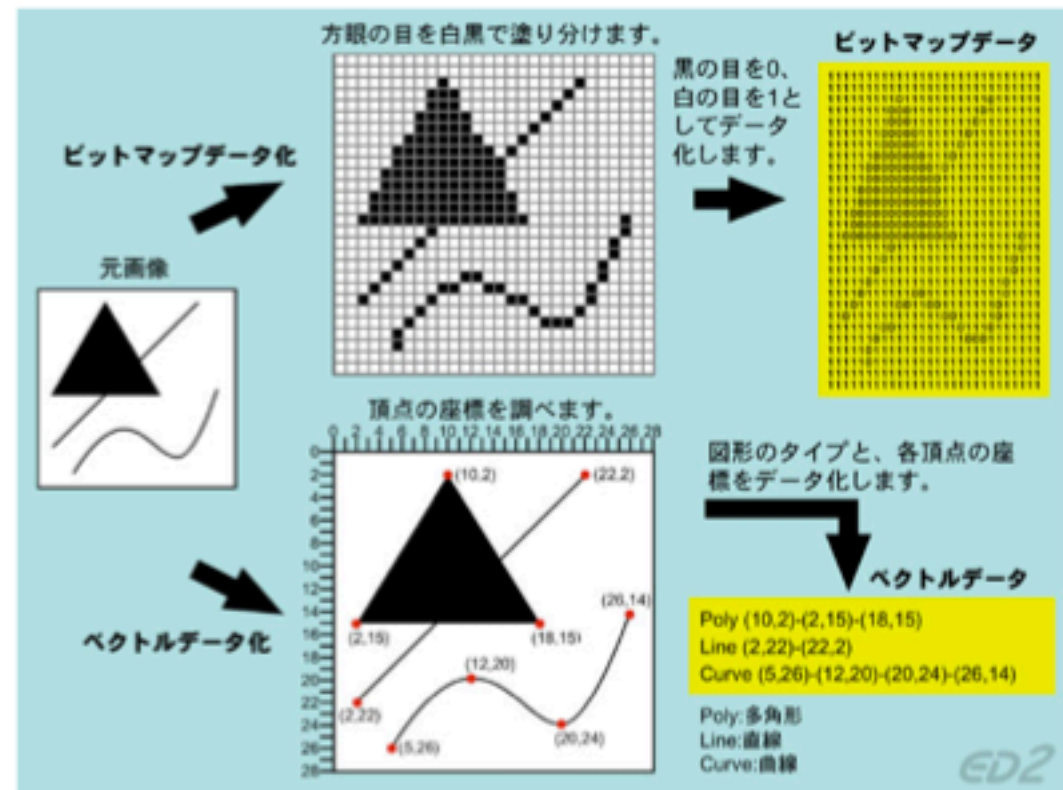
ビットマップ画像とベクトル画像

### ビットマップ画像

画像を画素（色の点/ドット/dot）の集合として表す。

### ベクトル画像

座標、線、面の種類、色、模様等の集合として表す。



# 4. 画像の表現

## 色の表現

1 **ドット(dot:画素)**の情報

R, G, B の明るさを各256段階(=8bit)で表現

1画面 =  $1,024 \times 768 = 786,432$  dot とすると、

=  $786,432 \times 3 = 2,359,296$  byte 必要

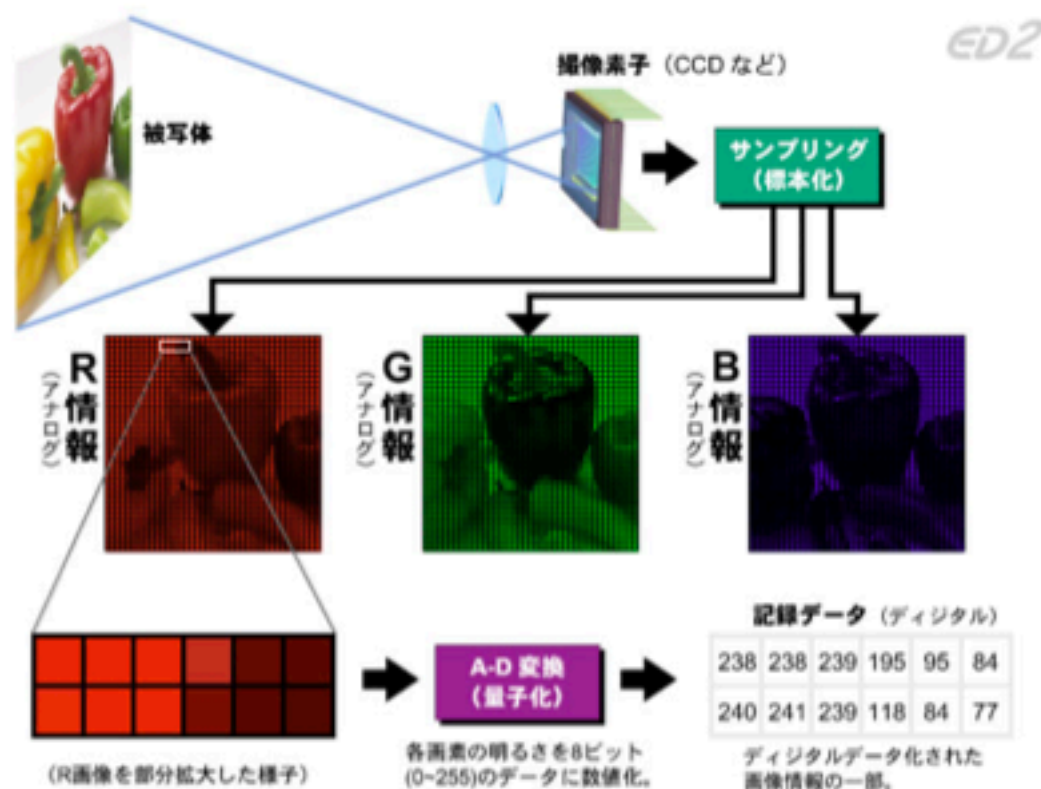
例) あるコンピュータの画面は

$1,920 \times 1,080 = 2,073,600$  dot

=  $2,073,600 \times 3 = 6,220,800$  byte

= 約6M byte (6MB)

実際には情報を「**圧縮**」することが多い (可逆/非可逆) (後述)



# 4. 画像の表現

## 練習問題

### 問題4-1

「ビットマップ画像」と「ベクトル画像」の違い、用途、長所短所について説明しなさい。

### 問題4-2

「ビットマップ・ディスプレイ」とは何か説明しなさい。



## 5. メディアと情報の量


各メディアから発せられる・受け取る情報量

文字 1文字2byte とすると  
10文字/秒 = 20 byte/秒


音 1データ 2 byte とすると  
44,100データ/秒  
= 88,200 byte/秒

画像 1ドット 3 byte とすると  
1画像 = 1,240×960 ドット  
= 1,190,400 ドット  
= 3,571,200 byte  
※30 画像/秒 = 107,136,000 byte/秒

**静止画 (圧縮なし)**

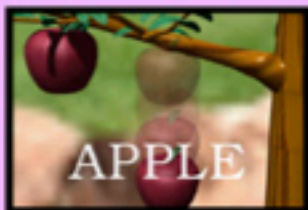


画像の大きさが720×480ドットとすると、全体のドット数は345,600ドット。1ドットごとに3バイトのデータ量なので、画像のデータ量は1,036,800バイト=約1Mバイト。




CD-ROM1枚に約650枚記録可能。

**動画 (圧縮なし)**




データ量が1Mバイトのフレームを、毎秒30フレームのスピードで表示すると、1秒あたりのデータ量は30Mバイトにもなります。



CD-ROM1枚に約21秒記録可能。

毎秒30フレーム



# 画像の出典

情報機器と情報社会の仕組み素材集（火曜の会）

<https://jnk4.sakura.ne.jp/www/jyouhou-kiki/>

