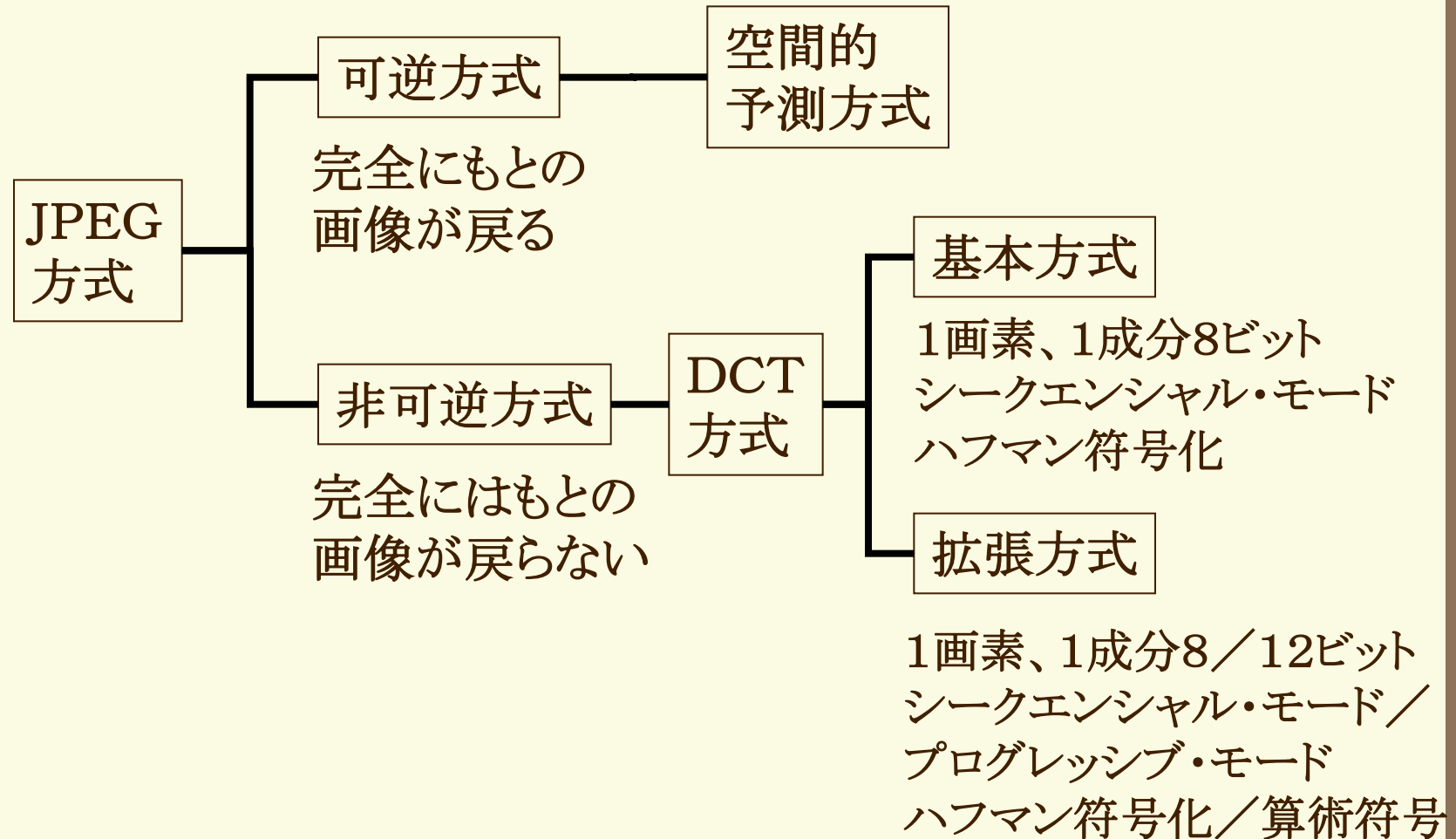


2.5 JPEG圧縮

JPEG(Joint Photographic
Experts Group)



ビットレートと画像品質

0.25-0.5bpp: 普通の品質

0.5-0.75bpp: 良好な品質

0.75-1.5bpp: 優良な品質

1.5-2.0bpp: 原画と同品質

JPEGにおける4つの主要な処理モード

シーケンシャル符号化

プログレッシブ符号化

ロスレス符号化

階層符号化

JPEGにおける4つの主要な処理モード

シーケンシャル符号化

画像はDCTを基礎とした左から右へ、上から下へのラスタースキャン方式で符号化される

プログレッシブ符号化

画像は狭いチャネルバンド幅のアプリケーションに対して、同じ空間解像度で、多重走査により符号化される。

(伝送時間は長くなるが、視聴者は粗いレベルから高精細へ、段階的に内容を見ていく)

ロスレス符号化

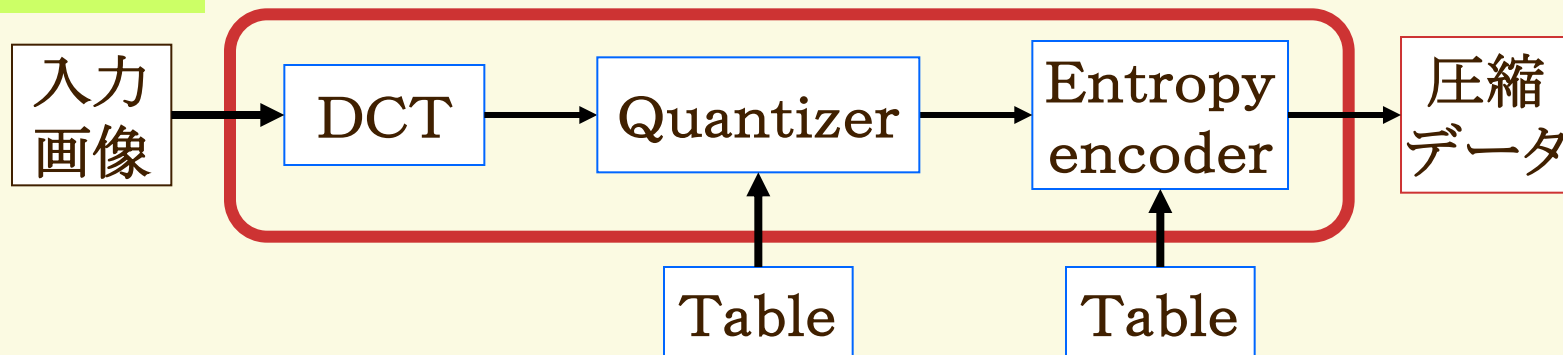
画像はあらゆるサンプル源の画質を損なわずに正確に再生できることを保証するように符号化される。無駄な情報は削除されるが、圧縮率は低い。

階層符号化

画像は複数の空間解像度で符号化される。従ってより解像度の低い画像は、より空間解像度の高い画像の再生を行うことなしに、その表示やアクセスを許す

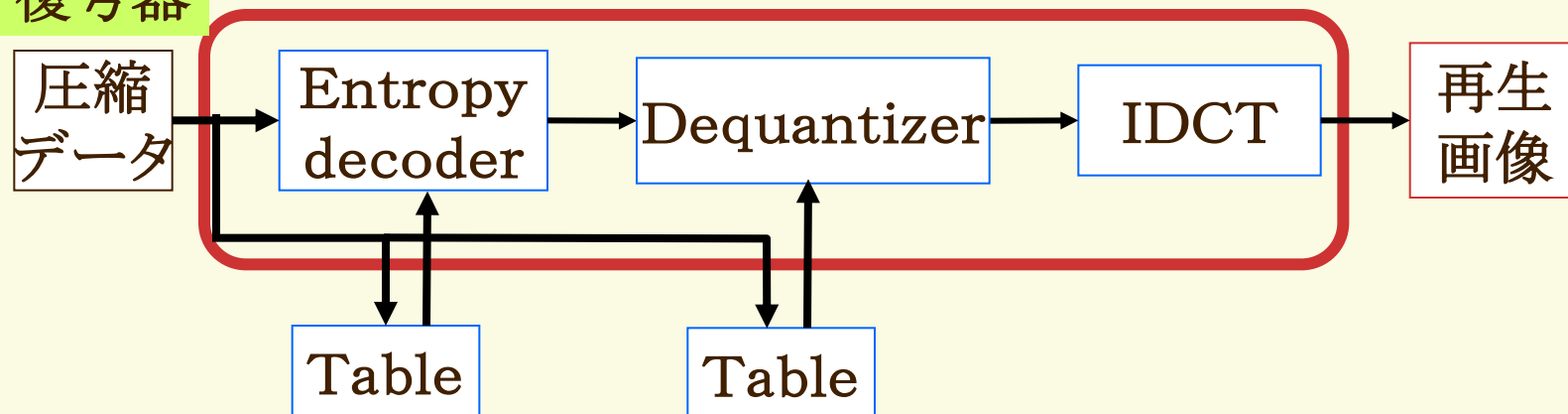
基本的な流れ

符号器



圧縮率 1/10以下 識別できない
1/20 やや画質劣化
1/30以上 画質劣化が目立つ

復号器



DCTプロセス

(A) DCT

(1) 画像を8x8画素のブロックに切り出す

(2) DCT変換を行う

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{j=0}^7 \sum_{k=0}^7 f(j, k) \cos \frac{(2j+1)u\pi}{16} \cos \frac{(2k+1)v\pi}{16}$$

$$C(w) = \begin{cases} 1/\sqrt{2} & w=0 \\ 1 & w=1 \sim 7 \end{cases}$$

(3) この変換で生成された8x8のDCT係数を成分毎に分け並べる

1ブロックあたり	DC成分	1個
	AC成分	63個 (ジグザグスキャンして並べる)

.....

逆DCT変換

$$f(j, k) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) F(u, v) \cos \frac{(2j+1)u\pi}{16} \cos \frac{(2k+1)v\pi}{16}$$

DCT係数

8×8

DC

AC

DC成分 $F(0,0)$

AC成分 $F(1,0)$

$F(0,1)$

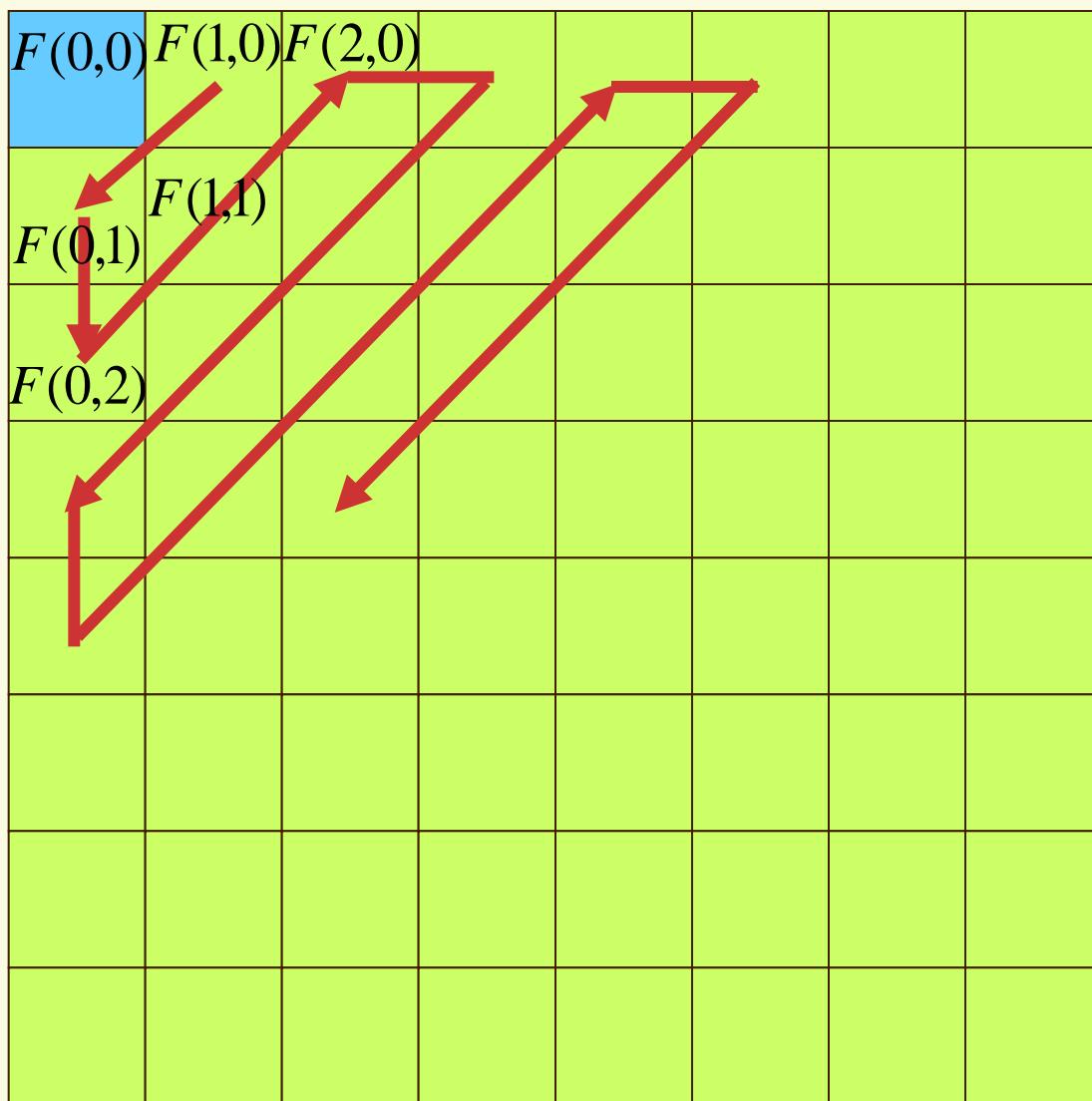
$F(0,2)$

$F(1,1)$

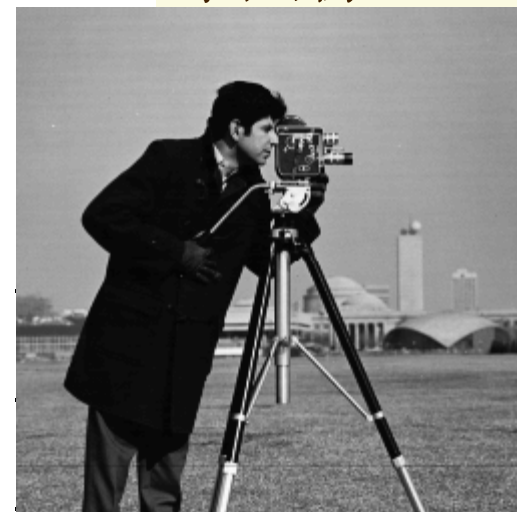
$F(2,0)$



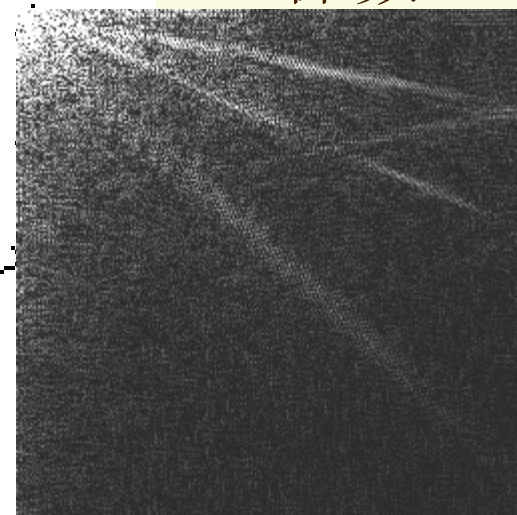
ジグザグスキャンして並べる



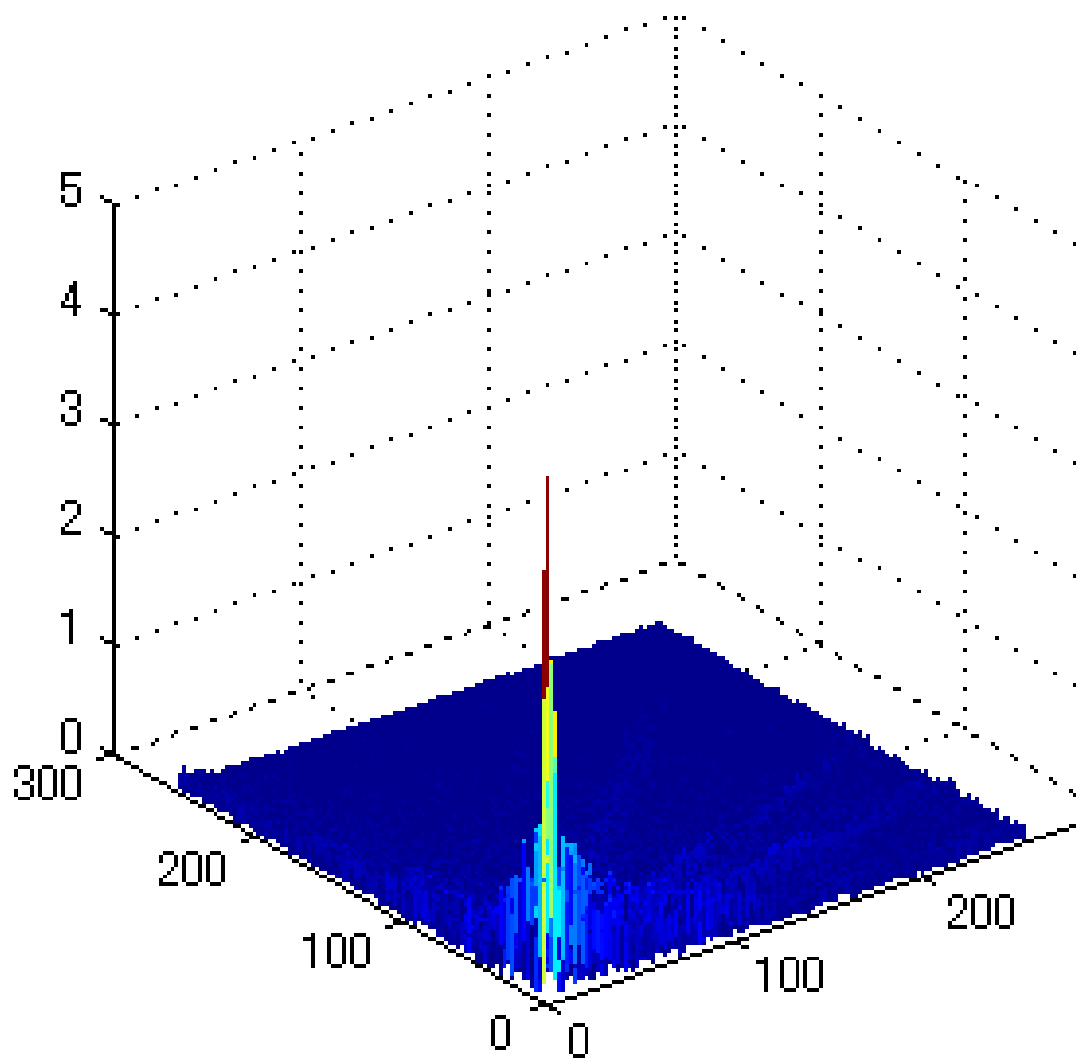
原画像



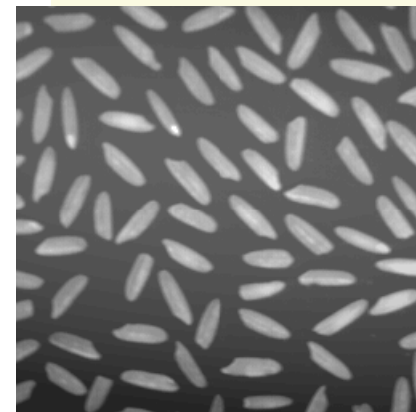
DCT係数



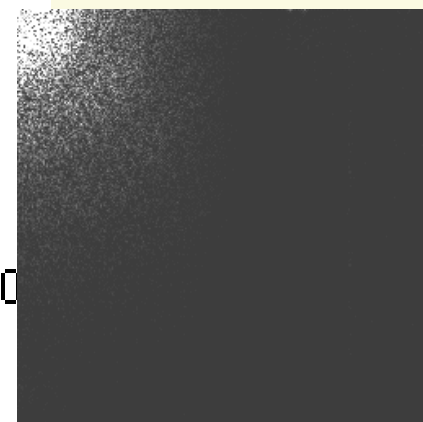
256×256



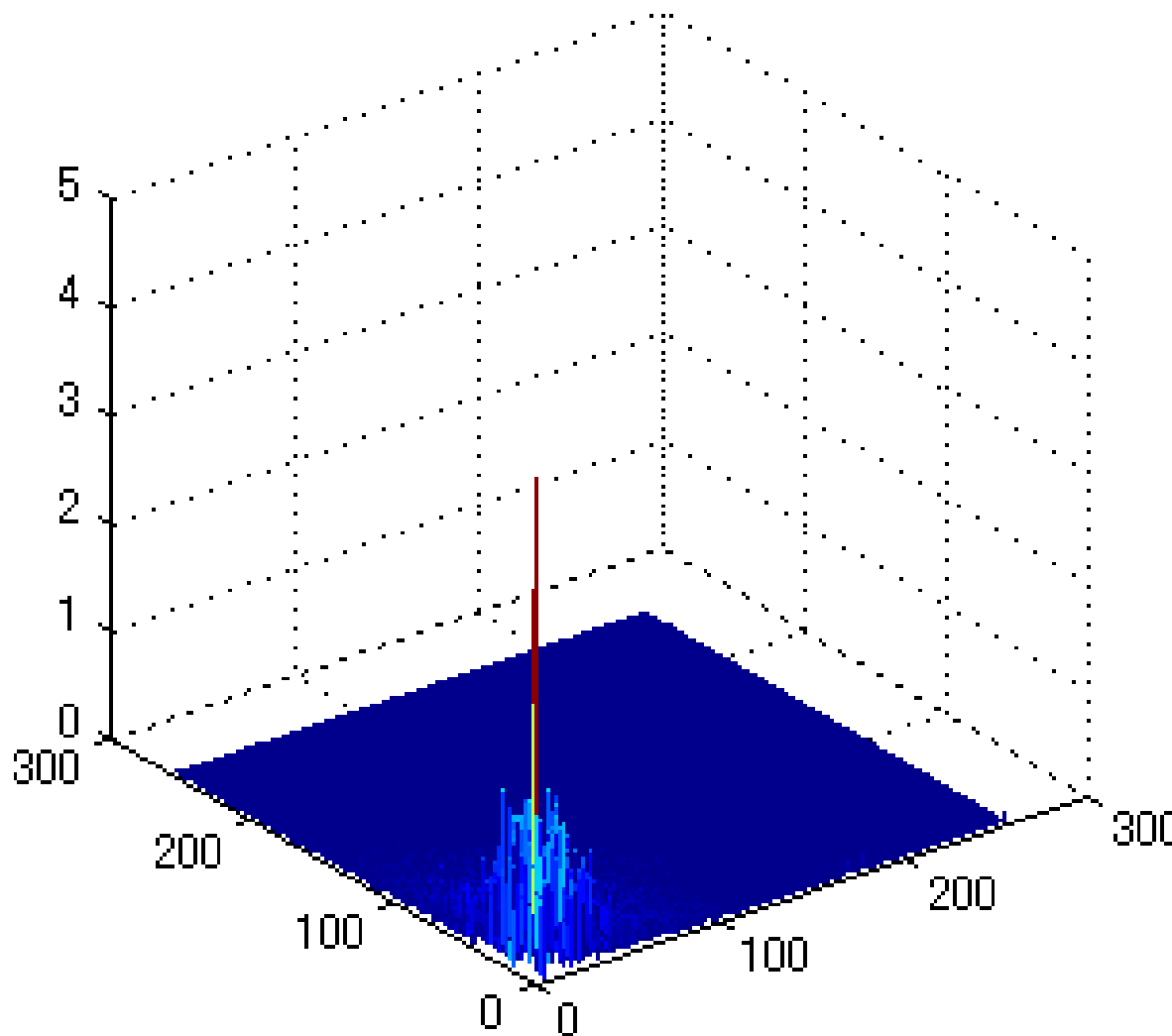
原画像



DCT係数



256×256



量子化

量子化テーブルによって係数を量子化する

考え方

人間の視覚は低周波に敏感で
高周波に鈍感



高周波成分ほど量子化幅が大きくなるように、係数の位置毎に異なる値の量子化テーブルを用いる

DC AC

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	66
14	13	16	24	40	57	69	57
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

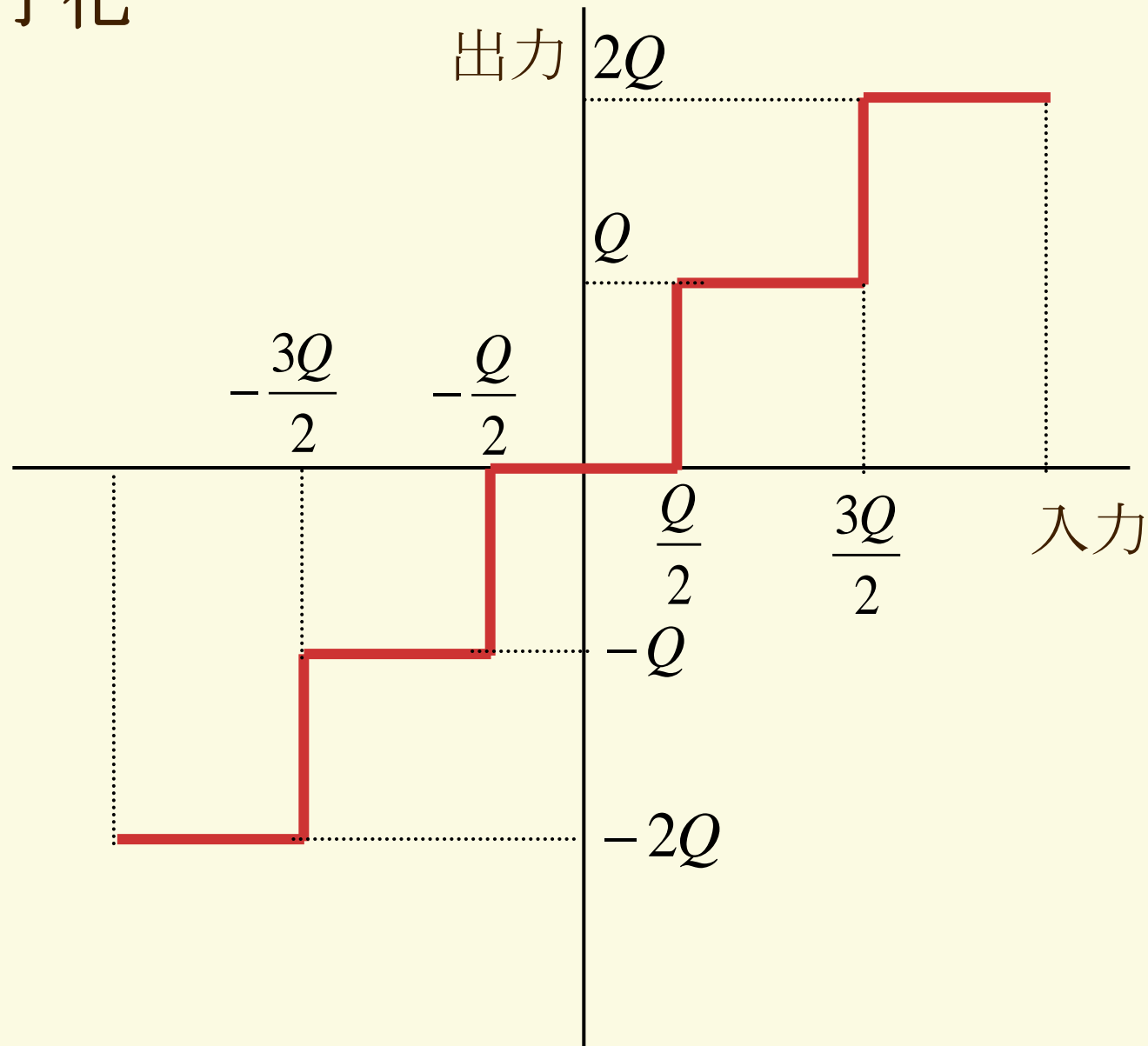
輝度信号用 量子化テーブル

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	66
14	13	16	24	40	57	69	57
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

色差信号用 量子化テーブル

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	66	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

量子化



DCT係数のエントロピー符号化

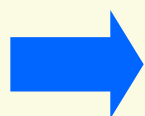
各々の係数をそのエントロピーに近いデータ量で符号化する

DC成分

符号化済み隣接ブロックのDCT係数を符号化対象のDCT係数の予測値とし、その予測誤差を符号化する。

比較的出现頻度の高い
絶対値の小さな予測誤
差値には短い符号長の
符号を割り当てる

比較的出现頻度の低い
予測誤差値には長い符
号長の符号を割り当てる



グループ番号と、同一グループ内のいずれの値かを特定する付加ビットで表現する(付加ビット長はグループ番号が小さいときに短くなっている)

グループ番号とDC成分の差

$$DC_i - DC_{i-1}$$

Group(SSSS)

Difference value

0	0
1	-1, +1
2	-3, -2, +2, +3
3	-7, ..., -4, +4, ..., +7
4	-15, ..., -8, +8, ..., +15
5	-31, ..., -16, +16, ..., +31
6	-63, ..., -32, +32, ..., +63
7	-127, ..., -64, +64, ..., +127
8	-255, ..., -128, +128, ..., +255
9	-511, ..., -256, +256, ..., +511
10	-1023, ..., -512, +512, ..., +1023
11	-2047, ..., -1024, +1024, ..., +2047

ハフマン符号表

輝度DC

色差DC

SSSS 符号長

符号語

符号長

符号語

0	2	00	2	00
1	3	010	2	01
2	3	011	2	10
3	3	100	3	110
4	3	101	4	1110
5	3	110	5	11110
6	4	1110	6	111110
7	5	11110	7	1111110
8	6	111110	8	11111110
9	7	1111110	9	111111110
10	8	11111110	10	1111111110
11	9	111111110	11	11111111110

DC成分の符号化

SSSS=0の時、追加ビットは要求されない

SSSS≠0の時、追加ビットを付加する

$$DIFF = DC_i - DC_{i-1}$$

DIFFが正の時、

DIFFの下位ビットがSSSSビット分付加される

$$DIFF = 5 \quad SSSS = 3 \rightarrow 100 \quad DIFF = 101$$

コード 100101

DIFFが負の時、

DIFF-1の下位ビットがSSSSビット分付加される

$$DIFF = -5 \quad SSSS = 3 \rightarrow 100 \quad DIFF = 010$$

コード 100010 (1の補数表現)

グループ番号とAC成分

Group(SSSS)

AC係数

1	$-1, +1$
2	$-3, -2, +2, +3$
3	$-7, \dots, -4, +4, \dots, +7$
4	$-15, \dots, -8, +8, \dots, +15$
5	$-31, \dots, -16, +16, \dots, +31$
6	$-63, \dots, -32, +32, \dots, +63$
7	$-127, \dots, -64, +64, \dots, +127$
8	$-255, \dots, -128, +128, \dots, +255$
9	$-511, \dots, -256, +256, \dots, +511$
10	$-1023, \dots, -512, +512, \dots, +1023$

AC成分のVLCテーブル R/S/code

一部 (variable length coding)

ラン/サイズ/符号語

ラン/サイズ/符号語

0	0	1010(EOB)	3	1	1110 10
0	1	00	3	2	1111 1011 1
0	2	01	3	3	1111 1111 0101
0	3	100	3	4	1111 1111 1000 1111
0	4	1011	3	5	1111 1111 1001 0000
1	1	1100	4	1	1110 11
1	2	1101 1	4	2	1111 1110 00
1	3	1111 001	4	3	1111 1111 1001 0110
1	4	1111 1011 0	4	4	
1	5	1111 1110 110	4	5	
2	1	1110 0	5	1	1111 010
2	2	1111 1001	5	2	1111 1110 111
2	3	1111 1101 11	5	3	1111 1111 1001 1110
2	4	1111 1111 0100	5	4	
2	5	1111 1111 1000 1001	5	5	

AC成分の符号化

非ゼロAC成分は合成されたR/Sで記述される

Rは前の非ゼロ値からの4bitのゼロラン(ゼロ係数の数)

Sは10分類された4ビットの値

AC成分が正の時、

AC成分の下位ビットがSSSSビット分付加される

R/S=0/2 でAC成分が2の時、

SSSS=2→01 係数のコード10 コード 0110

AC成分が負の時、符号ビットは0であり、

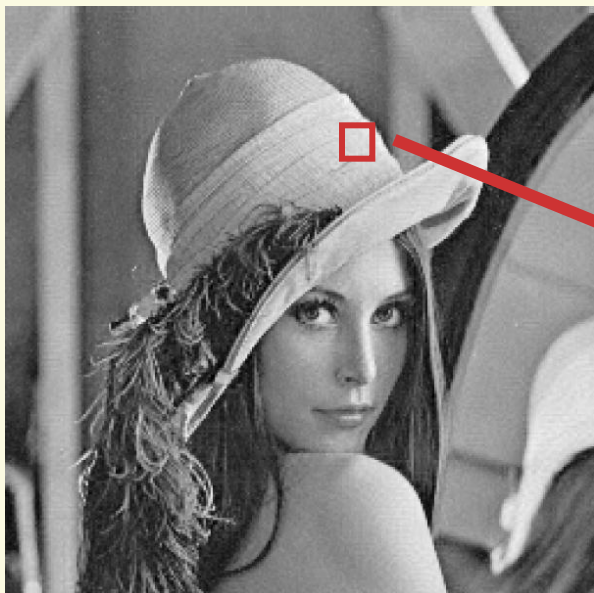
AC成分-1の下位ビットがSSSSビット分付加される

R/S=0/2 でAC成分が-3の時、

SSSS=2→01 係数のコード00(1の補数表現)

コード 0100

基本符号化例



$x_{ij} =$

79	75	79	82	82	86	94	94
76	78	76	82	83	86	85	94
72	75	67	78	80	78	74	82
74	76	75	75	86	80	81	79
73	70	75	67	78	78	79	85
69	63	68	69	75	78	82	80
76	76	71	71	67	79	80	83
72	77	78	69	75	75	78	78

入力画像データは 2^{p-1} で減算した符号付き2の補数表現にレベルシフトされる

この場合、 $p=8$ で128シフト

2次元
DCT後

$$S_{uv} = \begin{pmatrix} 619 & -29 & 8 & 2 & 1 & -3 & 0 & 1 \\ 22 & -6 & -4 & 0 & 7 & 0 & -2 & -3 \\ 11 & 0 & 5 & -4 & -3 & 4 & 0 & -3 \\ 2 & -10 & 5 & 0 & 0 & 7 & 3 & 2 \\ 6 & 2 & -1 & -1 & -3 & 0 & 0 & 8 \\ 1 & 2 & 1 & 2 & 0 & 2 & -2 & -2 \\ -8 & -2 & -4 & 1 & 2 & 1 & -1 & 1 \\ -3 & 1 & 5 & -2 & 1 & -1 & 1 & -3 \end{pmatrix}$$

DC成分となる619はこのブロックの平均的階調の8倍
レベルシフトがされていないDC成分は619に1024
を加えた値となっている(1643)

量子化後

$$S_{quv} = \begin{pmatrix} 39 & -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$619/16 = 38.6 \Rightarrow 39$$

$$-29/11 = -2.6 \Rightarrow -3$$

$$22/12 = 1.8 \Rightarrow 2$$

$$-6/12 = -0.5 \Rightarrow -1$$

$$11/14 = 0.78 \Rightarrow 1$$

$$2/14 = 0.14 \Rightarrow 0$$

DC成分

前のブロックの量子化されたDC成分の係数が34であったとすると、エントロピー符号化される予測誤差は5となる

テーブルよりこのハフマン符号は100となる

$SSSS=3 \rightarrow$ ハフマンコード 100

これに予測誤差の正確な値の表現形式101が
付加される

$\rightarrow 100101$

AC成分

ジグザグスキャンによって係数を並べると

{-3 2 1 -1 1 0 0 0 0 0 -1 EOB}

-3 → 0/2 → 01

付加ビット 11 1と0を反転 00 (1の補数)

→0100

2 → 0/2 → 01

付加ビット 10

→0110

1 → 0/1 → 00

付加ビット 1

→001

{-3 2 1 -1 1 0 0 0 0 0 -1 EOB}

-1 \rightarrow 0/1 \rightarrow 00

付加ビット 1 1と0を反転 0 \rightarrow 000

1 \rightarrow 0/1 \rightarrow 00

付加ビット 1 \rightarrow 001

-1 \rightarrow 5/1 \rightarrow 1111010

付加ビット 1 1と0を反転 0 \rightarrow 11110100

EOB \rightarrow 1010

{0100/0110/001/000/001/11110100/1010}

全体では

{100101/0100/0110/001/000/001/11110100/1010}

このブロックの転送は35ビットとなる

原画像 $8\text{bit} \times 8 \times 8 = 512\text{bit}$

$512\text{bit} \rightarrow 35\text{bit}$ (6.8% 約1/15)

転送ビットレート $8\text{bit/pixel} \rightarrow 0.55\text{bpp}$

シークエンシャル方式とプログレッシブ方式

シークエンシャル方式

左上から右下へ順に画像を描画する

プログレッシブ方式

まず粗く画像を表示して、それから細部を表示する
(低周波帯域をまず符号化し、それから順に高周波帯域を符号化する)

ベースライン方式(基本方式)では、この方法は使用できない

デモンストレーション

インターリーブ方式とノンインターリーブ方式

輝度信号 Y

A_1	A_2	A_3	A_4
A_5	A_6		
		A_{N-1}	A_N

色差信号 Cr

B_1	B_2
B_3	B_4
	$B_{N/2}$

色差信号 Cb

C_1	C_2
C_3	C_4
	$C_{N/2}$

インターリーブ方式

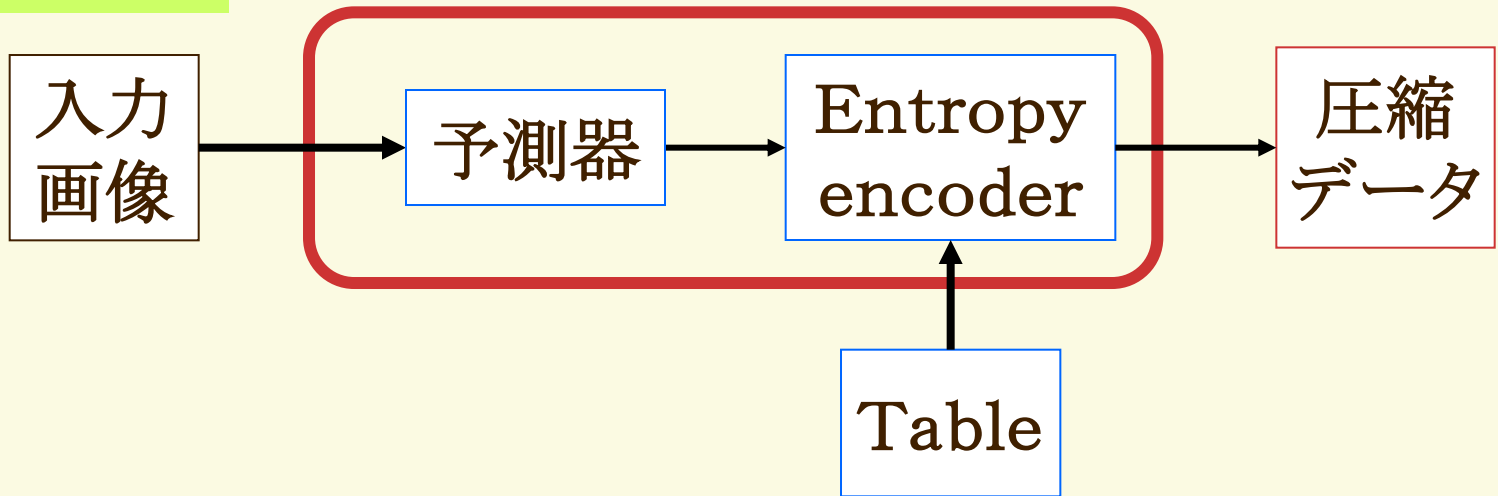
A_1	A_2	B_1	C_1	A_3	A_4	B_2	C_2	A_{N-1}	A_N	$B_{N/2}$	$C_{N/2}$
-------	-------	-------	-------	-------	-------	-------	-------	-----	-----	-----------	-------	-----------	-----------

ノンインターリーブ方式

A_1	A_2	A_3	A_{N-1}	A_N	B_1	...	$B_{N/2}$	C_1	...	$C_{N/2}$
-------	-------	-------	-----	-----	-----	-----------	-------	-------	-----	-----------	-------	-----	-----------

可逆符号化方式

符号器



圧縮率 $1/2 \sim 1/3$
画質劣化はない

予測に用いられる画素

c	b
a	x

 P_x \mathbf{x} の予測値 R_a R_b R_c a, b, c の再構成値

選択値／予測式

0 *no prediction*

$$1 \quad P_x = R_a$$

$$2 \quad P_x = R_b$$

$$3 \quad P_x = R_c$$

$$4 \quad P_x = R_a + R_b - R_c$$

$$5 \quad P_x = R_a + (R_b - R_c) / 2$$

$$6 \quad P_x = R_b + (R_a - R_c) / 2$$

$$7 \quad P_x = (R_a + R_b) / 2$$

ロスレスハフマン符号化のための差分の分類

Group(SSSS)

Difference value

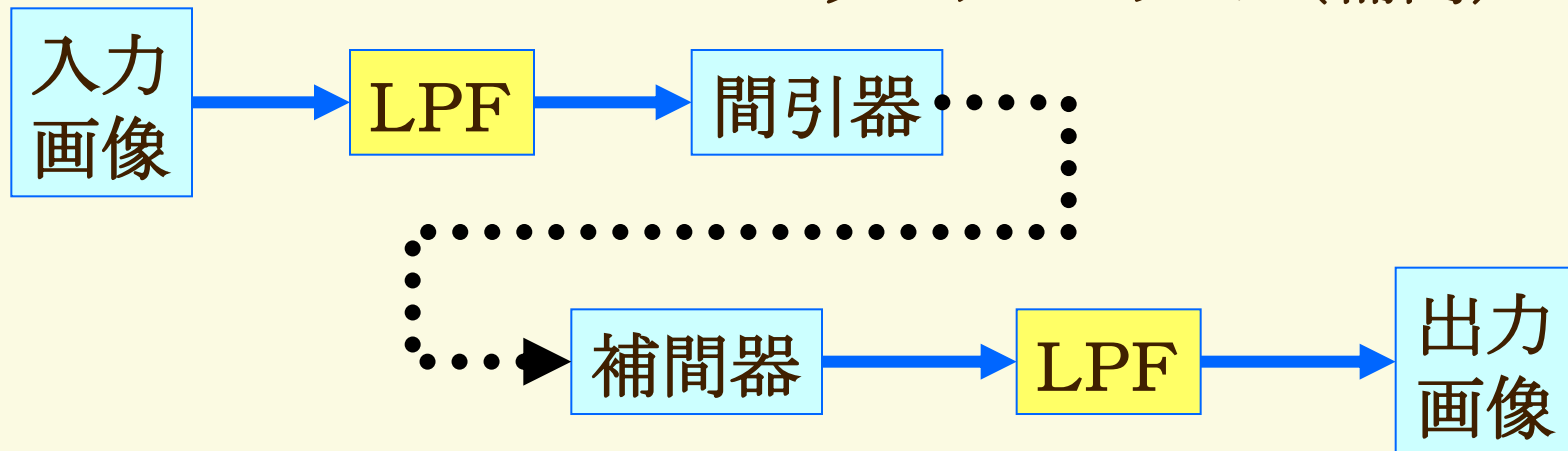
0	0
1	$-1, +1$
2	$-3, -2, +2, +3$
3	$-7, \dots, -4, +4, \dots, +7$
4	$-15, \dots, -8, +8, \dots, +15$
5	$-31, \dots, -16, +16, \dots, +31$
6	$-63, \dots, -32, +32, \dots, +63$
7	$-127, \dots, -64, +64, \dots, +127$
8	$-255, \dots, -128, +128, \dots, +255$
9	$-511, \dots, -256, +256, \dots, +511$
10	$-1023, \dots, -512, +512, \dots, +1023$
11	$-2047, \dots, -1024, +1024, \dots, +2047$
11	$-4095, \dots, -2048, +2048, \dots, +4095$
11	$-8191, \dots, -4096, +4096, \dots, +8191$
12	$-16383, \dots, -8192, +8192, \dots, +16383$

16
まで

階層的処理

ダウンサンプリング (間引き)

アップサンプリング (補間)

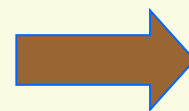


LPF

1	4	1
4	16	4
1	4	1

*

x_0	x_1	x_2
x_3	x_4	x_5
x_6	x_7	x_8



	y_4	

$$y_4 = \frac{1}{36} [x_0 + x_2 + x_6 + x_8 + 4(x_1 + x_3 + x_5 + x_7) + 16x_4]$$