

# 画像符号化の説明

山下幸彦

大学院理工学研究科  
東京工業大学

平成 23 年 9 月 5 日

# 発表内容

---

1. はじめに
2. 画像符号化の基盤技術
  - ハフマン符号化
  - 算術符号化
  - ランレンジス符号化
  - 予測符号化
  - 変換符号化
3. 標準的画像符号化法
  - JPEG
  - MPEG 2
  - MPEG 4 AVC
4. サブバンド符号化

# 画像符号化の重要性

---

## 画像の情報量

- 画像の情報量は大きい。

SDTV (Standard Television) の例 :

- 色 : 3 (R, G, B)
- 明るさ : 8 [bit]
- 横 : 720 [画素]
- 縦 : 480 [画素]
- 時間 : 30 [frame/s]

$$3 \times 8 \times 720 \times 480 \times 30 \simeq 248.8 [\text{Mbit/s}]$$

HDTV (High Definition Television)

$$3 \times 8 \times 1920 \times 1080 \times 30 \simeq 1.48 [\text{Gbit/s}]$$

- 地上波テレビ : 18 [Mbit/s] 程度  
⇒ データ量を圧縮する必要がある。

# 非可逆符号化 vs. 可逆符号化

---

## 非可逆符号化 (lossy)

- 原画像が復号画像と異なる。
- その差は人間の目には分かりにくいようにする。
- ブロック歪み, リンギング(モスキート雑音)
- データ圧縮率に優れる。
- 一般に用いられる画像符号は非可逆符号化

## 可逆符号化 (無歪み符号化) (lossless)

- 原画像が復号画像が全く同じ。
- 圧縮率は1/2程度
- 医用画像などに用いられる(データの保存が法律で必要)。

今回は, 非可逆符号化を扱う。

# エントロピー符号化

---

- **符号化**：アルファベット(整数)列を0, 1の系列に直すこと。
- **符号**：1つのアルファベットに割り当てられた0, 1の系列  
例：  
符号：0 → 00, 1 → 01, 2 → 1001  
アルファベット列：2011 → 1001000101
- **エントロピー**：平均情報量(符号の出現確率で決まる量)
- 平均符号長はエントロピー以下にはできない。
- エントロピー符号化：平均符号長をエントロピーに近づける符号化
- 原則：  
頻繁に現れるアルファベットに短い符号  
あまり現れないアルファベットに長い符号を割り当てる。
- **コンパクト符号**：平均符号長が最小になる符号

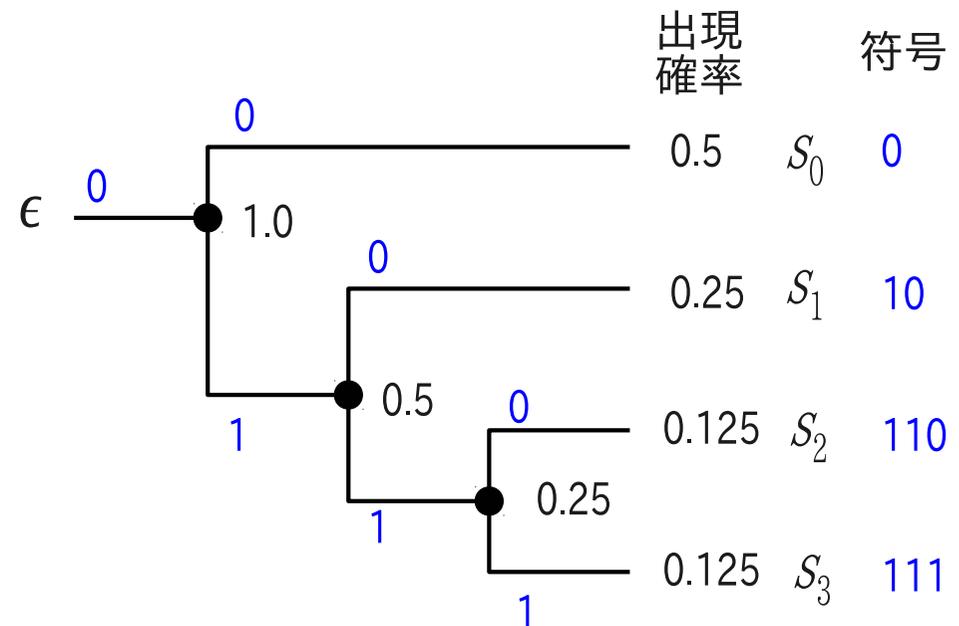
# ハフマン (Huffman) 符号化

- アルファベットの出現確率が与えられたときに、**コンパクト符号を作ることができる。**
- 符号化テーブルを使って符号を決める。
- 画像符号化では、本当に確率を使って符号を決めるのではなく、**符号化テーブルを使って符号を決める方法一般に使われている。**
- 具体的な符号化テーブルの例は、JPEGで説明する。

## ハフマン符号の符号化構成法

平均符号長：

$$0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75$$



# 算術符号化

---

- MITのP. Eliasによって提案
- IBMのG. G. Langdon, J. J. Rissanenが完成(特許)
- JPEG2000のMQ coderに使われている。
- 符号化アルゴリズム(原理):
  1. 初期の対象区間:  $[0, 1)$
  2. 現在の対象区間を, シンボルの確率に比例させた長さの区間に分割
  3. 符号化するシンボルに対応する区間を対象区間とする。
  4. 2へもどる
- 区間に含まれる1つの小数でその区間を表す。
- シンボル列の出現確率が高い方が, 長い区間になる。
  - ⇒ 一般には長い区間には桁数の短い小数が含まれる。
  - ⇒ アルファベット列の出現確率が高いほど短い符号になる。
- 復号は小数が含まれる区間を, アルファベットの順番で調べる。

## ランレングス (run length) 符号

---

- 画像符号化では，符号化したいアルファベットに0が続くことが多い。

### zero run

- 連続している0の個数 (**ゼロランレングス**) を情報とする。  
 $0 \rightarrow 1, 00 \rightarrow 2, 000 \rightarrow 3, 0000 \rightarrow 4, \dots$
- 連続して0が続く回数の出現確率は，その値の種類に依存する。  
 $\Rightarrow$  0が続く値もエントロピー符号化する。
- 残りがすべて0のときのための，専用の符号を割り当てる。
- 具体的な例は，JPEGで説明する。

# 量子化

---

- 画像の画素値やその変換係数は一般には**連続量**
- 符号化するために，整数値に変換する。
- 線形量子化

$Q$  : 量子化係数

$u$  : 入力連続値

$v$  : 量子化された値

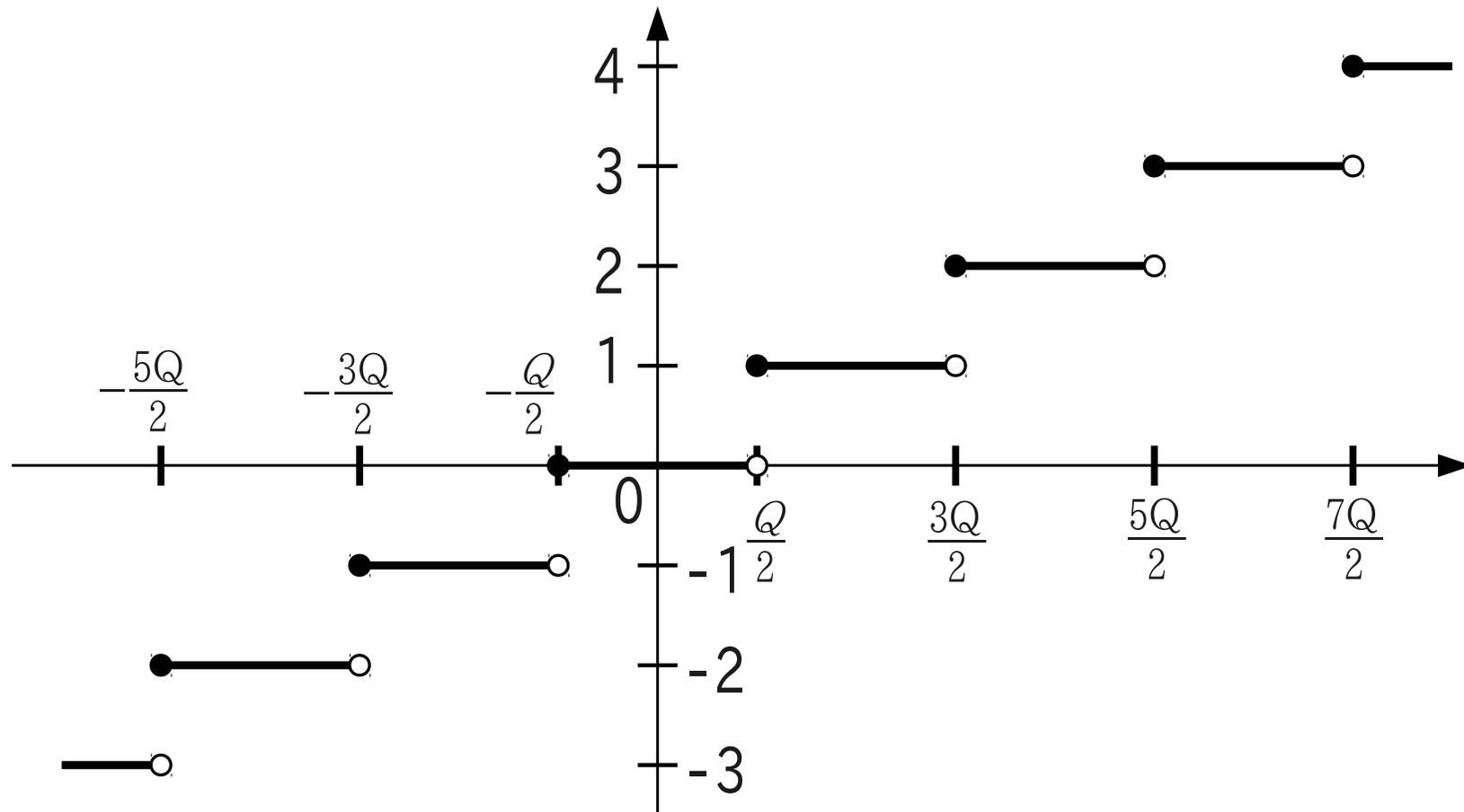
$$v = \text{floor} \left( \frac{u}{Q} + 0.5 \right)$$

- 逆量子化 :

$u_0 = Qv$  : 出力連続値

- $u$  と  $u_0$  には， $\pm \frac{1}{2}Q$  の誤差が生じる。
- デッドゾーン (0にする範囲を広くする。)  
0を符号化する効率が良いためである。

# 線形量子化



- $Q$  が大きいと，送信する整数が小さくなり誤差が大きくなる。

# 予測符号化

- 差分符号化 (最も簡単な例)

- 直前の値を予測値とする。
- $x_1, x_2, x_3, \dots$  を送りたい信号列とする。
- $x_{n+1}$  と  $x_n$  の差は小さいことが多い。
- $x_{n+1} - x_n$  を送る。

値が小さいため, 少ないbit数で送ることができる。

n	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
入力値	120	130	134	137	140	144	138	142	134
差分	120	10	4	3	3	4	-6	4	-8

## 局所復号器 (ローカルデコーダ) 1/2

---

- 予測符号化で量子化を行う場合は，注意が必要
- 不完全な情報を送っている。
- 先の例で量子化係数10で量子化する場合：

n	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
入力値	120	130	134	137	140	144	138	142	134
差分	120	10	4	3	3	4	-6	4	-8
差分の量子化	120	10	0	0	0	0	-10	0	-10
復号結果	120	130	130	130	130	130	120	120	110

- 誤差が累積してしまう。

## 局所復号器 (ローカルデコーダ) 2/2

- 符号器側で復号結果を再構成しその差分を量子化して送る。

差分 = 入力値 - 局所復号器出力

送信データ = 差分を量子化したもの

n	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
入力値	120	130	134	137	140	144	138	142	134
局所復号結果との差分	120	10	4	7	0	4	-2	2	-6
差分の量子化結果	120	10	0	10	0	0	0	0	-10
局所復号結果	120	130	130	140	140	140	140	140	130
復号結果	120	130	130	140	140	140	140	140	130

## 変換符号化 (1/4)

---

- ベクトル値を符号化するとき、各要素を符号化するのではなく、線形変換(直交変換)してから、係数を符号化する。
- $s_i$  ( $0 \leq i \leq N$ ):  $N$ 次元ベクトル  $s$  の第  $i$  要素の値
- $\sigma_i$ :  $s_i$  の標準偏差 (分散の平方根)
- $s_i$  は連続値であるので、その値を完全に送ることはできない。
- $s_i$  を符号化するとき、精度を一定とすると、その値を送るための平均符号長の下限は以下のようなになる。

$$\log_2 \sigma_i + \text{const} \quad [\text{bit}]$$

- 従って、ベクトルの平均符号長の下限は以下のようなになる。

$$\sum_{i=1}^N \log_2 \sigma_i + \text{const} \quad [\text{bit}]$$

## 変換符号化 (2/4)

---

- 画像を直交行列変換  $U$  することを考える。

$$t = Us$$

$U$  は等長変換である。

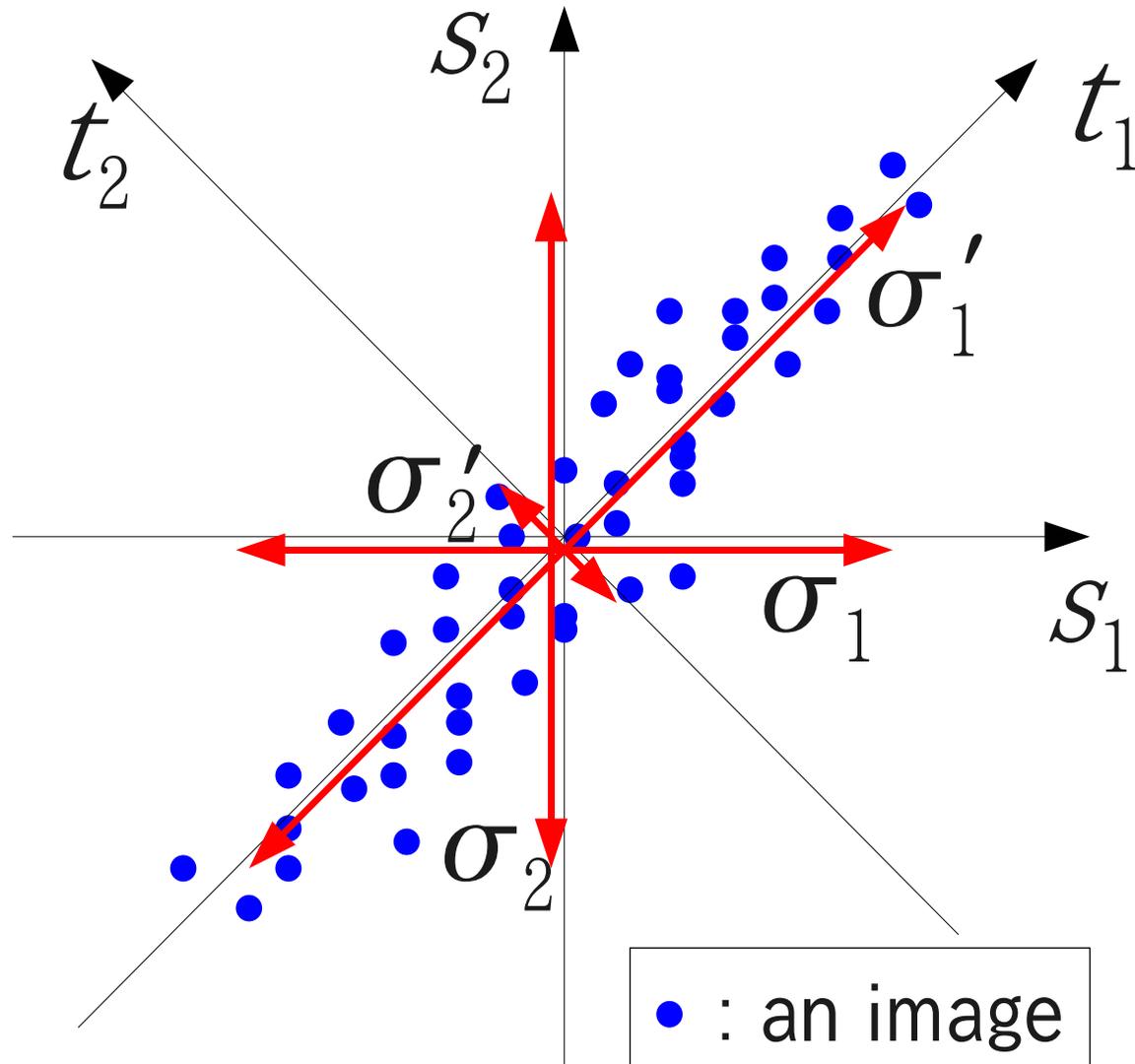
$$\|Us\| = \|s\|$$

- 座標値の精度と画像の精度の関係は  $s$  と  $Us$  で変わらない。
- $\sigma'_i$  :  $t_i$  の標準偏差 (座標値の標準偏差は変化する。)
- $s_i$  を画素  $i$  における画素値と考える。
- 画像の統計的性質は移動不変であるから, 画素値の分散  $\sigma_i$  の値は全て等しい。

$$\sigma_1 = \sigma_2 = \sigma_3 = \cdots = \sigma_N$$

- しかし,  $\sigma'_i$  は等しいとは限らない。

# 变换符号化 (3/4)



$$\sigma_1 = \sigma_2$$

$$\sigma'_1 \neq \sigma'_2$$

## 変換符号化 (4/4)

---

- 例 :

$$\sigma_1 = \sigma_2 = \cdots = \sigma_{11} = \sqrt{10}$$

$$\sigma'_1 = 10, \sigma'_2 = \sigma'_3 = \cdots = \sigma'_{11} = 1$$

分散の合計は同じ。

$$\sigma_1^2 = \sigma_2^2 + \cdots + \sigma_{11}^2 = 110$$

$$(\sigma'_1)^2 + (\sigma'_2)^2 + (\sigma'_3)^2 + \cdots + (\sigma'_{11})^2 = 110$$

- 平均符号長の下限の差は以下のようなになる。

$$\left[ 11 \log_2 \sqrt{10} \right] - [\log_2 10 + 10 \log_2 1] \simeq 14.9[\text{bit}]$$

- 信号の分散(パワー)を少ない成分に集めると, 平均符号長が短くなる。
- このための最適な変換は, **Karhunen-Loève transform (KLT)**.

# 画像符号化の標準規格

---

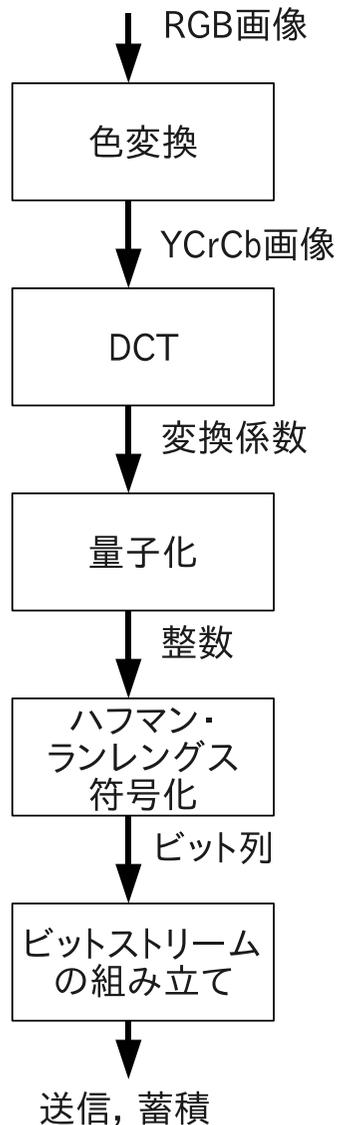
## JPEG

- Joint Picture Experts Group
  - ISO (International Organization for Standardization , 国際標準化機構)
  - ITU-T (International Telecommunication Union , 国際電気通信連合)-  
(Telecommunication Standardization Sector , 電気通信標準化部門)
- **静止画像**のための規格  
パソコン , デジタルカメラ

## MPEG

- Moving Picture Experts Group
- **動画**のための規格  
ビデオCD , DVD , デジタル放送
- ITU の , H.261 ~ H.264 とほぼ同様

# 静止画像符号化方式の標準 (JPEG)



- 色変換:  
RGB空間から,  $YCrCb$  空間へ変換
- 離散コサイン変換 (DCT)
- 量子化: DCT係数の量子化
- ハフマン・ランレングス符号化:  
量子化された係数の符号化
- ビットストリームの組み立て:  
各種符号列(画像のサイズ, テーブル, 係数)を1つのストリームにする。

## 色変換

---

- 人間は色に関する分解能が明度に関する分解能よりも低い。
  - 錐体：明るさを感じる。密度が高い (1.2~1.4億個)。
  - 杆体：色を感じる。密度が低い。  
(500~600万個，赤60%，緑30%，青10%)
- 明るさの成分と色の成分を分けて処理する。
- **RGB-YCbCr変換**

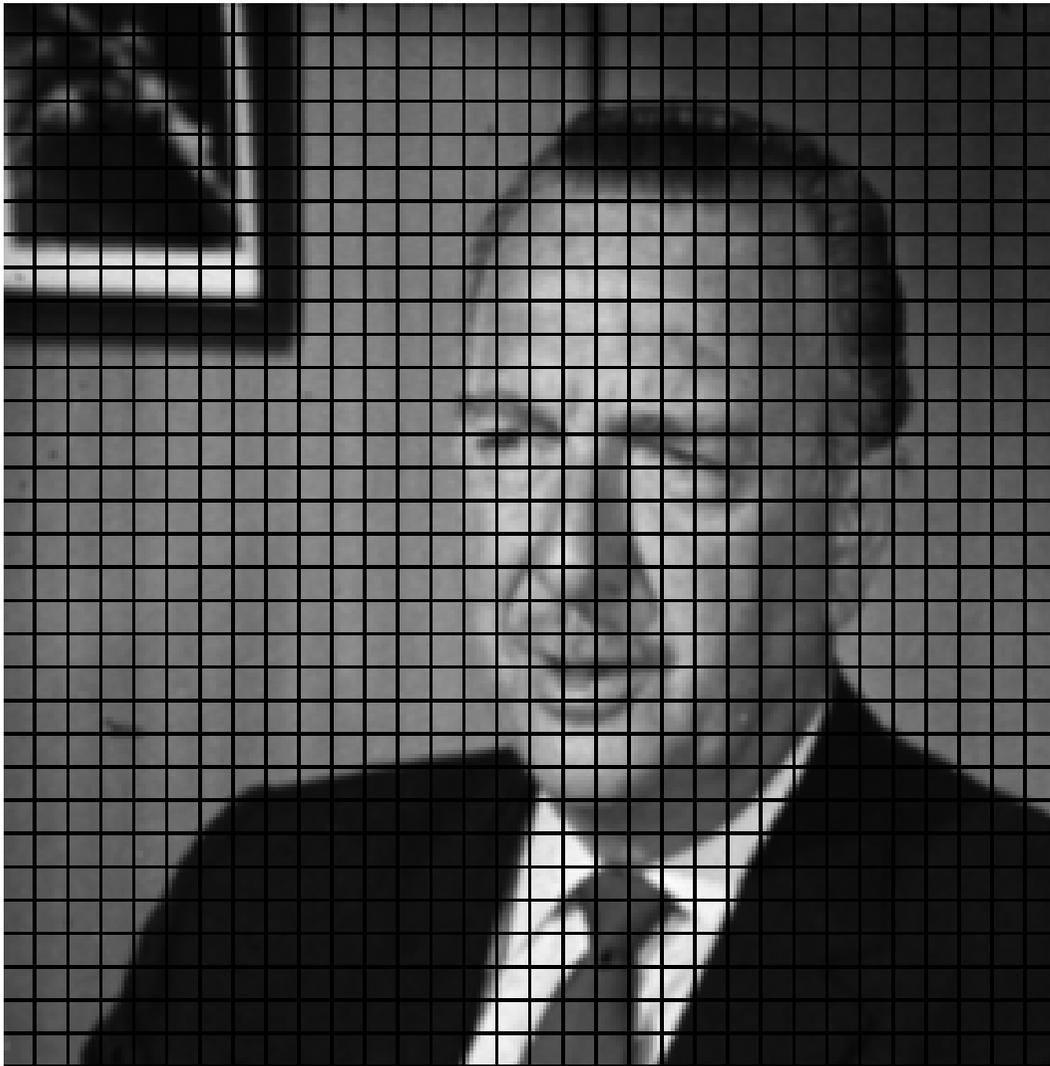
$$Y = 0.29900R + 0.58700G + 0.11400B$$

$$Cb = -0.16874R - 0.33126G + 0.50000B + 128$$

$$Cr = 0.50000R - 0.41869G - 0.08131B + 128$$

- 色に関するデータ (CbとCr) は間引く ( $1/2 \times 1/2$ )。

# ブロック分割



縦8×横8画素のブロック  
に分割する



以降，基本的にはそれぞれ  
のブロックを**独立に処  
理**する。

# DCT

---

- 信号のエネルギーを少数の成分に集める。
- DCT基底 (正規直交基底) :

$$f_{\text{DCT}}^n(m) = C_n \frac{1}{\sqrt{N}} \cos \left\{ \frac{\pi}{N} m \left( n + \frac{1}{2} \right) \right\},$$

where

$$C_n = \begin{cases} 1 & (n = 0) \\ \sqrt{2} & (n \neq 0) \end{cases}.$$

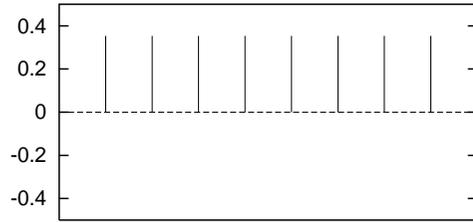
- **DCT係数** = 信号と  $f_{\text{DCT}}^n$  の内積
- DCTを表す  $A = [f^0 \ f^1 \ \dots \ f^{N-1}]^T$  は, 直交行列
- 入力画像を表すベクトルを  $x$  とすれば, 変換係数  $y$  は次のようになる。

$$y = Ax$$

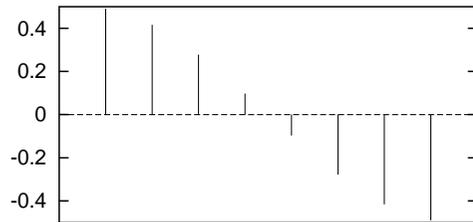
- 高速計算法が存在する。

# DCT 基底

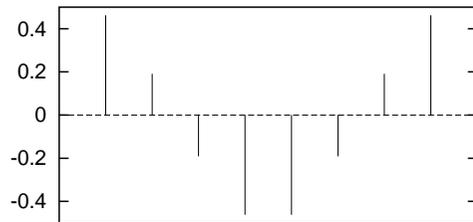
$n = 0$



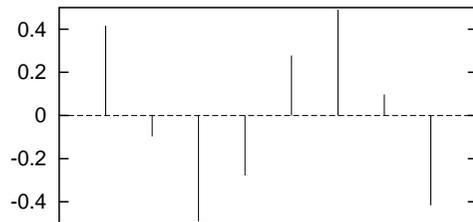
$n = 1$



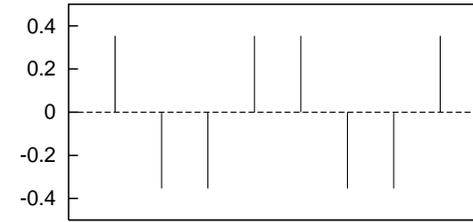
$n = 2$



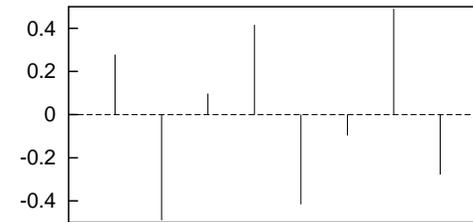
$n = 3$



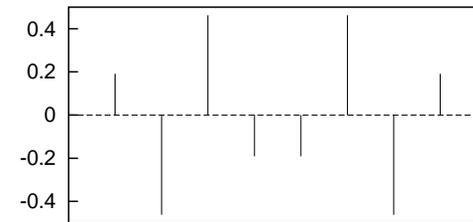
$n = 4$



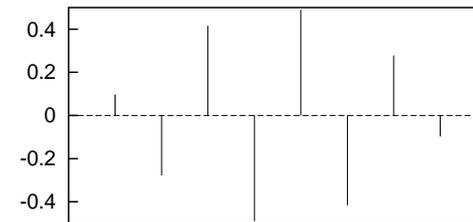
$n = 5$



$n = 6$

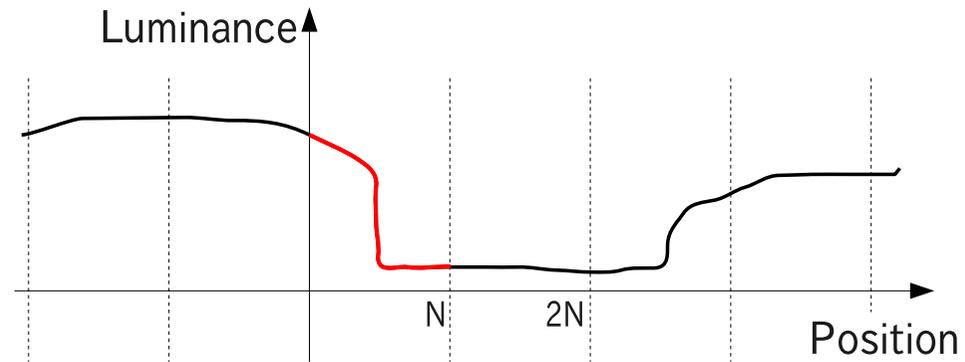


$n = 7$

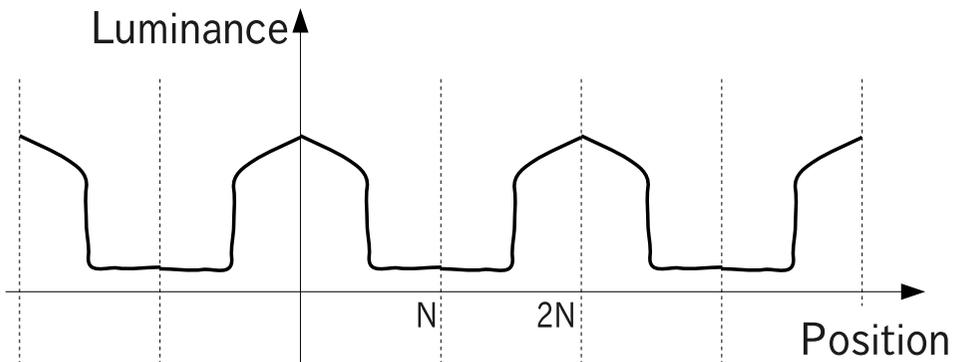


# DCT と DFT (1/2)

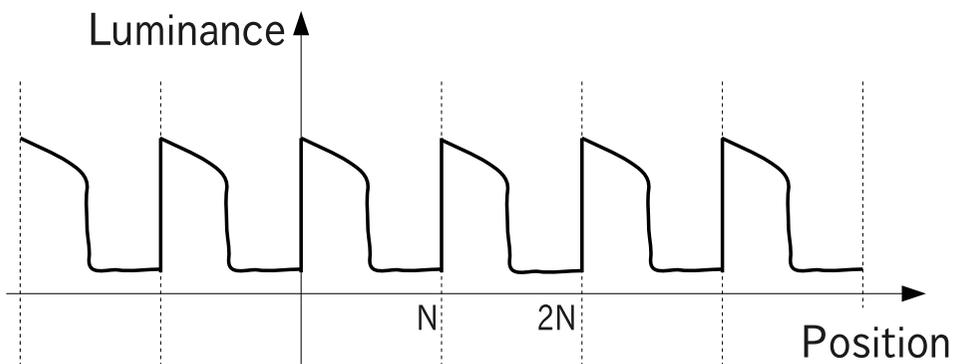
原信号



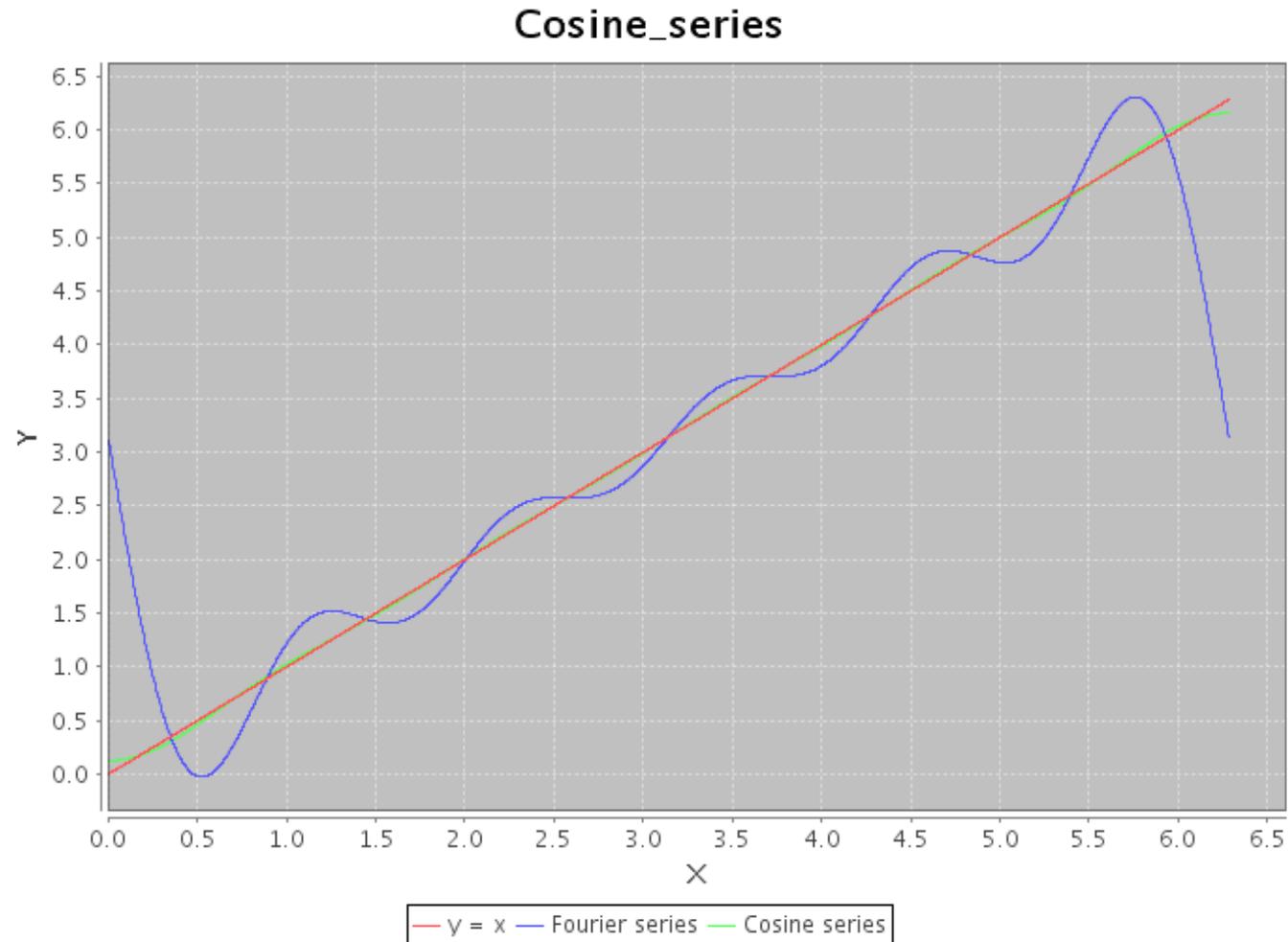
DCT



DFT



## DCTとDFT (2/2)



- $y = x$  を , DCT, DFT 共にはじめの5項で近似

# DCTとKLT

---

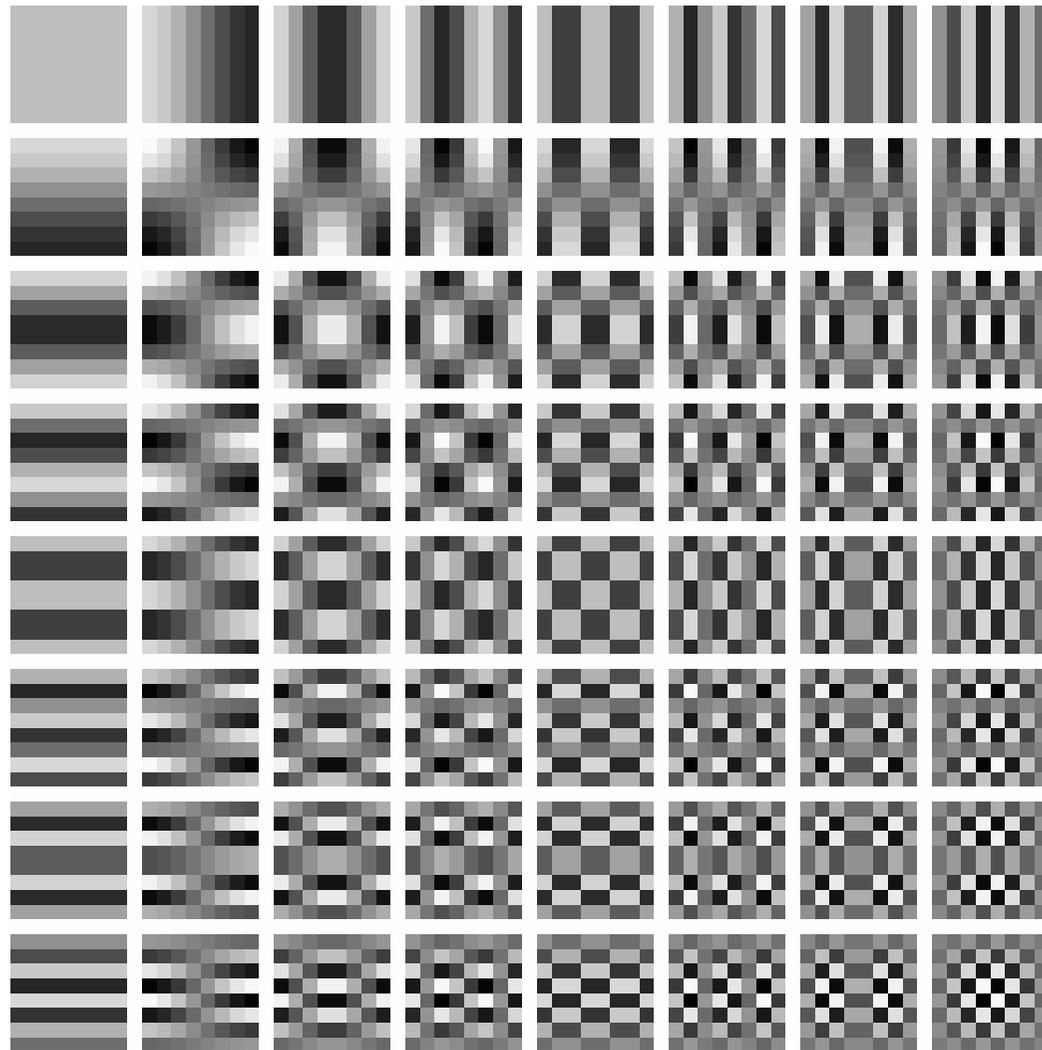
- $X_i, X_j$ : 第  $i$  成分と第  $j$  成分を表す確率変数
- $X_i$  と  $X_j$  の相関

$$E_{X_i, X_j} X_i X_j$$

- この相関が,  $\rho (< 1)$  を定数として,  $|i - j|$  の指数関数  $\rho^{|i-j|}$  で与えられるとき,  $\rho \rightarrow 1$  の極限で, KLTはDCTで与えられる
- 現実的にも画素間の相関は高い。
- **DCTを用いる問題: ブロック歪み (Block distortion)**
  - 変換ブロック境界に見える不連続性
  - 視覚的に目立つ。
  - 復号画像に対して画像処理 (エッジ抽出など) をするときに誤差が生じる。

# 2次元DCT

---

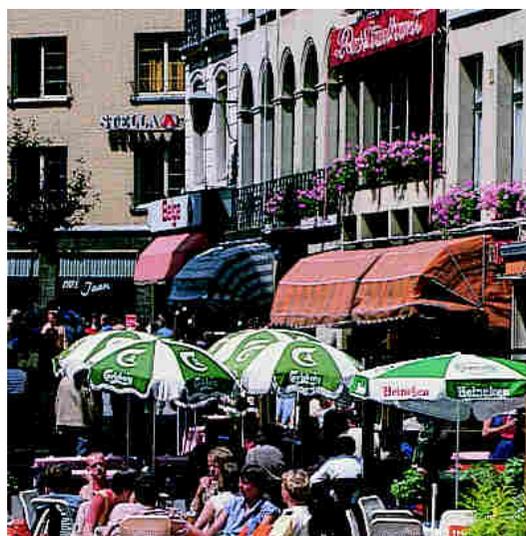


# ブロック歪み

原画像



JPEG



# 量子化

---

- **線形量子化**を用いる。
- DCT係数の周波数成分によって、異なる量子化係数を使う。  
高周波成分の誤差は低周波成分に比べて目立たない。
- Y と CbCr には異なる量子化係数を使う。
- **量子化テーブル**：各係数に対する基準となる量子化係数のテーブル
- 量子化テーブルの値に定数を乗算して、実際の量子化係数とする。
- 標準の量子化テーブルの値は決められているが、JPEG データの中に独自の量子化テーブルを埋め込むことが可能

## 量子化テーブル (輝度成分, Y)

---

V \ H	0	1	2	3	4	5	6	7
0	16	11	10	16	124	140	151	161
1	12	12	14	19	126	158	160	155
2	14	13	16	24	140	157	169	156
3	14	17	22	29	151	187	180	162
4	18	22	37	56	168	109	103	177
5	24	35	55	64	181	104	113	192
6	49	64	78	87	103	121	120	101
7	72	92	95	98	112	100	103	199

# 量子化テーブル (色差成分, CbCr)

---

V \ H	0	1	2	3	4	5	6	7
0	17	18	24	47	99	99	99	99
1	18	21	26	66	99	99	99	99
2	24	26	56	99	99	99	99	99
3	47	66	99	99	99	99	99	99
4	99	99	99	99	99	99	99	99
5	99	99	99	99	99	99	99	99
6	99	99	99	99	99	99	99	99
7	99	99	99	99	99	99	99	99

## 画像 (1ブロック)

---

V \ H	0	1	2	3	4	5	6	7
0	123	71	82	129	180	171	171	186
1	128	100	110	140	164	161	176	195
2	148	139	151	162	170	179	181	194
3	173	177	182	184	187	194	191	192
4	183	184	188	189	183	189	179	177
5	187	183	186	183	175	177	181	172
6	183	182	181	183	172	176	183	170
7	173	170	165	164	158	163	164	156

# DCT变换系数

---

V \ H	0	1	2	3	4	5	6	7
0	1343.80	-75.40	1.22	20.43	22.00	10.22	4.90	1.92
1	-83.81	-112.16	1.56	23.44	40.38	-5.45	7.17	8.55
2	-97.00	-42.23	8.17	22.50	24.42	3.17	-3.14	6.69
3	-54.82	-2.59	-5.68	13.94	10.55	2.80	-0.70	4.09
4	0.75	2.12	-8.46	12.25	2.00	-5.34	2.42	-0.59
5	21.78	-3.74	-4.44	4.79	-1.96	0.32	-0.85	0.86
6	-8.79	1.67	-0.39	3.42	0.16	-0.69	3.57	1.04
7	-0.78	0.54	-4.23	1.48	0.38	0.09	2.75	-2.61

## DCT变换系数 (量子化後)

---

DCT V \ H	0	1	2	3	4	5	6	7
0	84	-7	9	1	0	0	0	0
1	-7	-9	0	1	0	0	0	0
2	-70	-3	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

## DC係数のハフマン符号化

---

- 対象：DC係数(0, 0)を量子化した整数
- 差分符号化：直前のブロックの値との差を送る。
- 出現確率に基づく本当のハフマン符号化ではない。
- その整数を表すためのbit長がカテゴリになる。
- カテゴリに符号を割り振る。
- 1つの整数は、カテゴリの符号と値(+極性)を表すもののbit列となる。
- 標準のハフマンコードブック(符号化テーブル)が決まっている。  
次ページ：DC係数差分(輝度成分)のハフマンコードブック
- JPEGデータの中に独自のコードブックを埋め込むことが可能  
カテゴリーは変えることはできない。

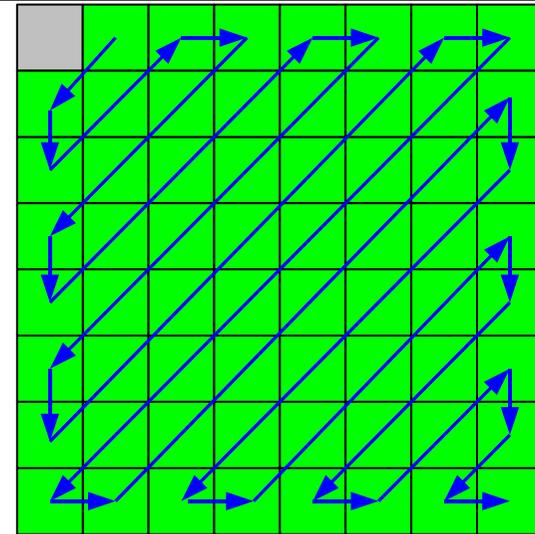
# DC係数差分 (輝度成分) のハフマンコードブック

量子化値	カテゴリー	符号	付加ビット
0	0	00	なし
-1, 1	1	010	0, 1
-3, -2, 2, 3,	2	011	00, 01, 10, 11
-7, ..., -4, 4, ..., 7	3	100	000, ..., 111
-15, ..., -8, 8, ..., 15	4	101	0000, ..., 1111
-31, ..., -16, 16, ..., 31	5	110	00000, ..., 11111
-63, ..., -32, 32, ..., 63	6	1110	000000, ..., 111111
-127, ..., -64, 64, ..., 127	7	11110	0000000, ..., 1111111
-255, ..., -128, 128, ..., 255	8	111110	00000000, ..., 11111111
-511, ..., -256, 256, ..., 511	9	1111110	000000000, ..., 111111111
-1023, ..., -512, 512, ..., 1023	10	11111110	0000000000, ..., 1111111111
-2047, ..., -1024, 1024, ..., 2047	11	111111110	00000000000, ..., 11111111111

例: 0 → 00, 10 → 1011010, -27 → 11000100

# AC係数のランレングスハフマン符号化

- 対象：AC係数  $((0, 0)$  以外) を量子化した整数
- **ジグザグスキャン**：右図
- 0が多く続く  
⇒ ランレングス符号化



- ジグザグスキャン中に0でない値が現れたときに，
  - それまで続けてきた0の数
  - 自身の値のbit数(サイズ)に対して1つの符号を与える(2次元ハフマン符号)。
- その符号の後に自身の値(+極性)を表す付加ビットを出力する。
- 走査の残りが全て0の場合は，そのための符号を出力して，そのブロックに対する出力を終了する。

## 動画画像符号化 (MPEG)

---

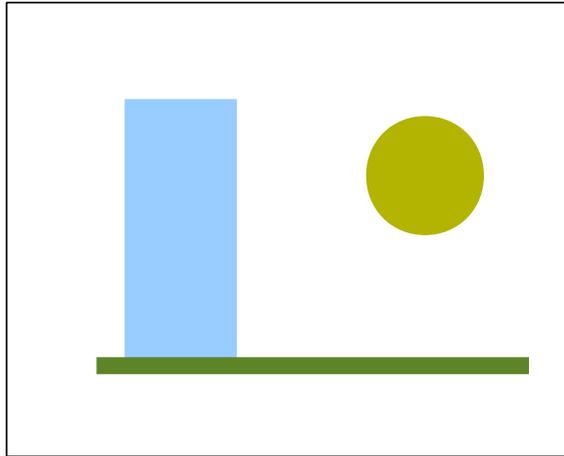
- **動画画像** : 時系列上の複数の画像
- MPEG = **動き補償予測** + JPEG
- **動き補償予測**
  - 時間的に近い画像間の類似性が高い。⇒ **予測符号化**
  - **対象画像** (target image) : これから符号化する画像
  - **参照画像** (reference image) : 対象画像を符号化する前に送信済みで、対象画像を符号化するときに利用できる画像
  - **予測画像** : 参照画像から生成する対象画像を予測した画像
  - 参照画像そのものを予測画像とすると、画像の内の動きによる誤差が無視できない。
  - 動きを補償して予測画像を生成する。
- 基本的には復号のための規格 (符号化器の設計には自由度がある)

## 動き補償予測

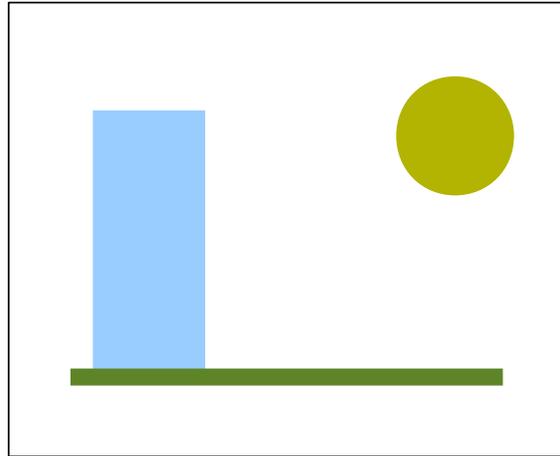
---

- 動き補償予測画像の生成法
  - 対象画像をブロックで分割する (MPEG 2 では ,  $16 \times 16$  画素)。
  - 参照画像の中から上記の各ブロックの画像と似ている領域 (ブロックと同じ大きさ) を探す。
  - **動きベクトル** : ブロックの座標と似ている部分の座標の差
- 符号器側では , 参照画像は復号済みなので , **動きベクトルさえあれば , 予測画像を生成できる。**
- 動きベクトルはハフマン符号化によって送る。
- 動きベクトル : 1 画像 ,  $1/2$  画素単位

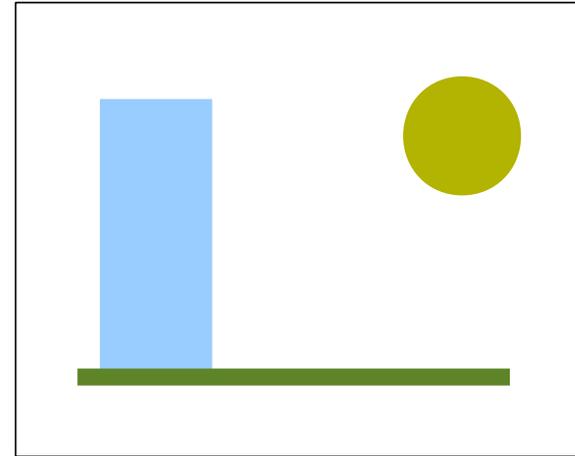
# 動き補償予測



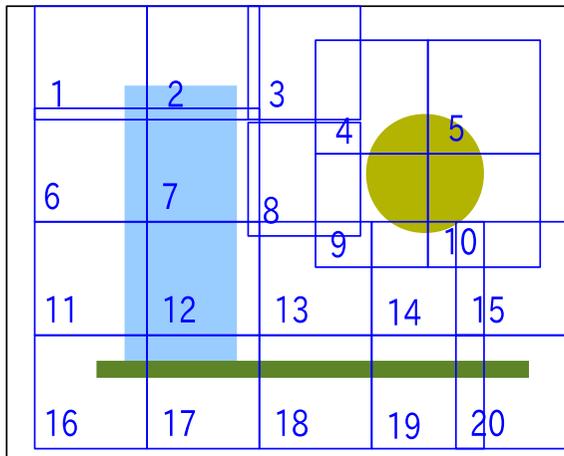
参照画像



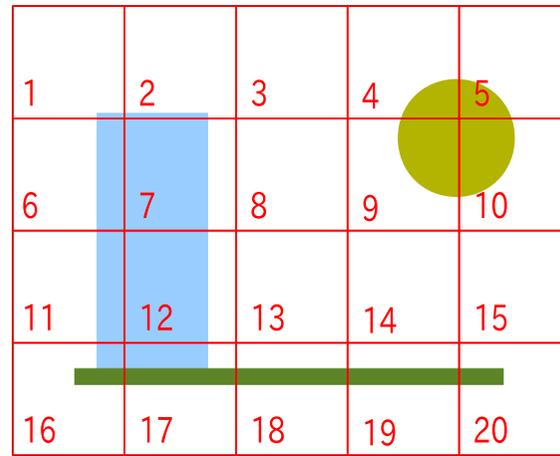
対象画像



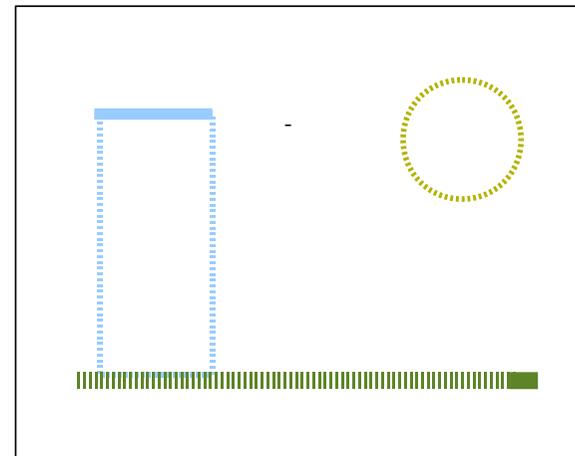
予測画像



参照画像

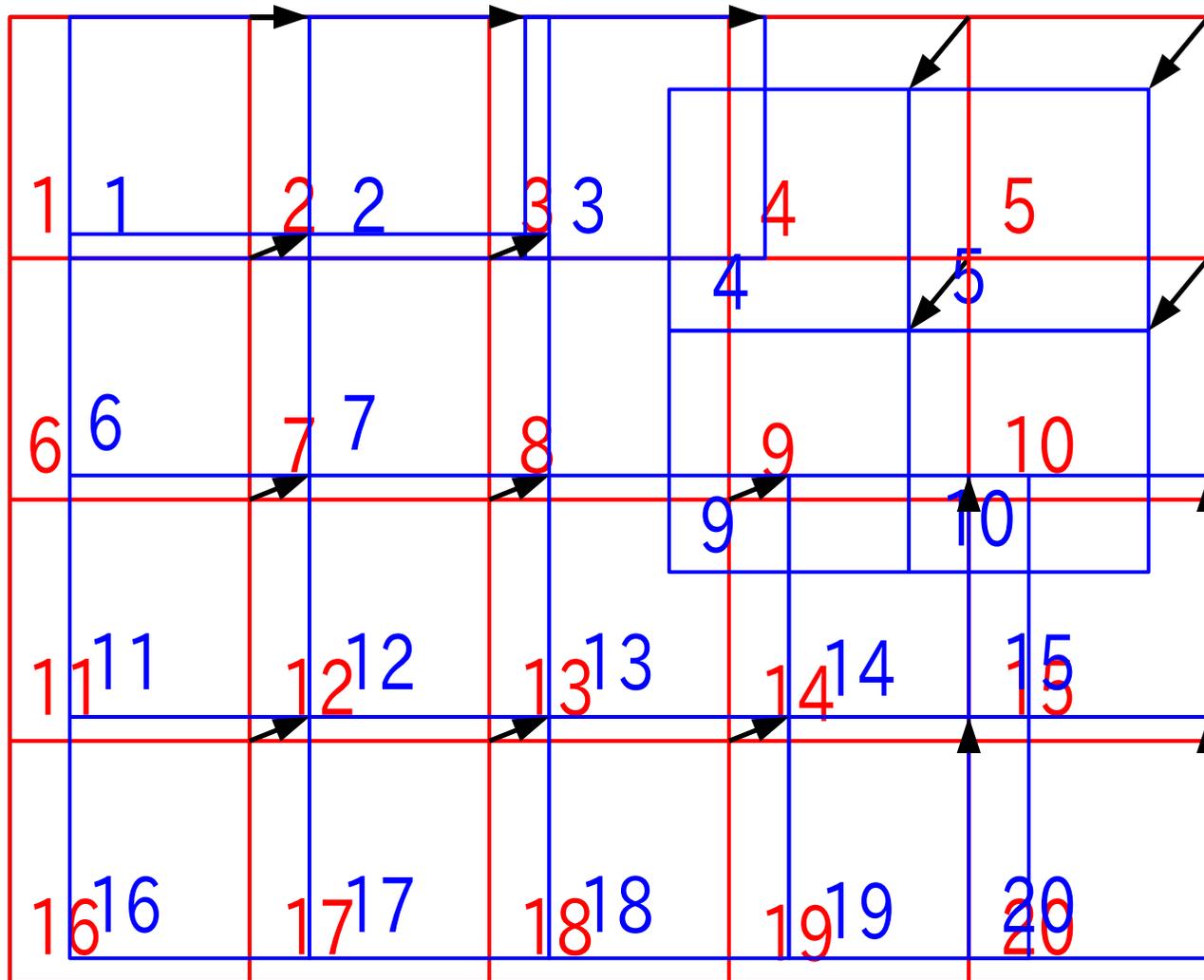


対象画像



差分画像

# 動きベクトル



# 動き補償予測 (実画像)



参照画像



対象画像



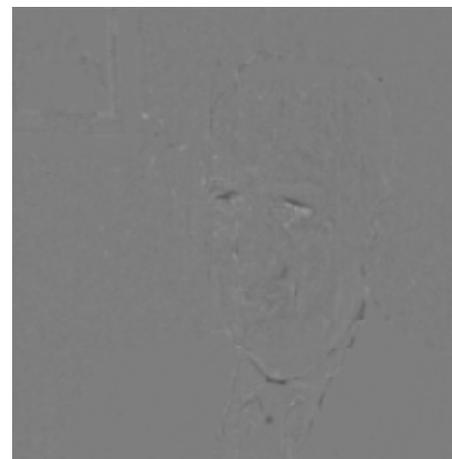
対象画像 - 参照画像



予測画像

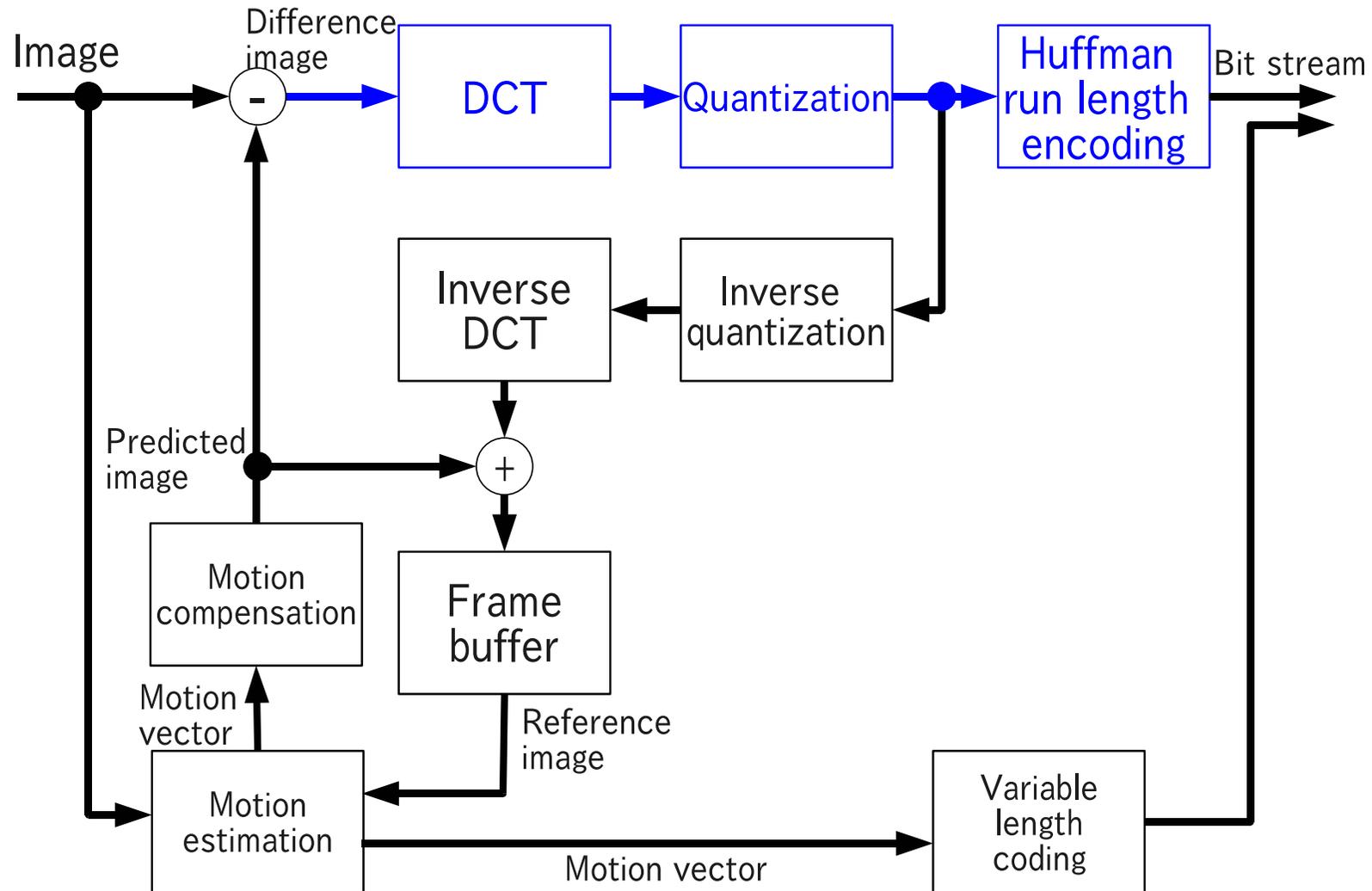


対象画像 (with MV)

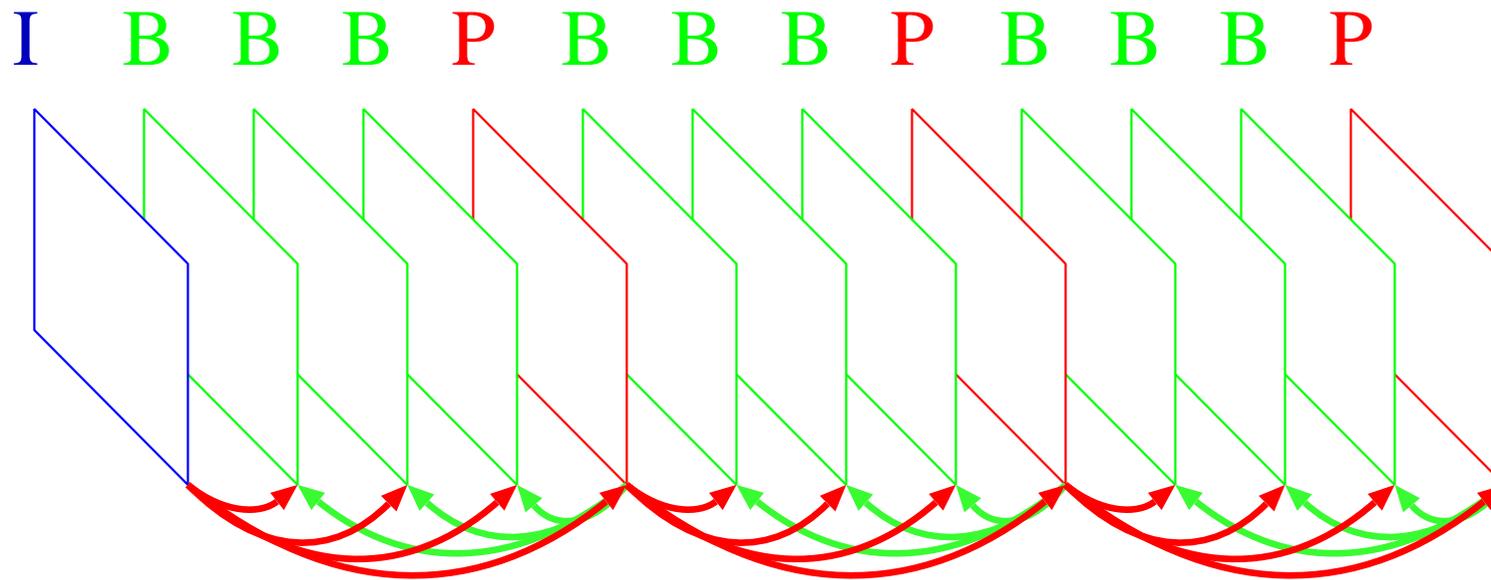


対象画像 - 予測画像

# MPEGエンコーダのブロック図

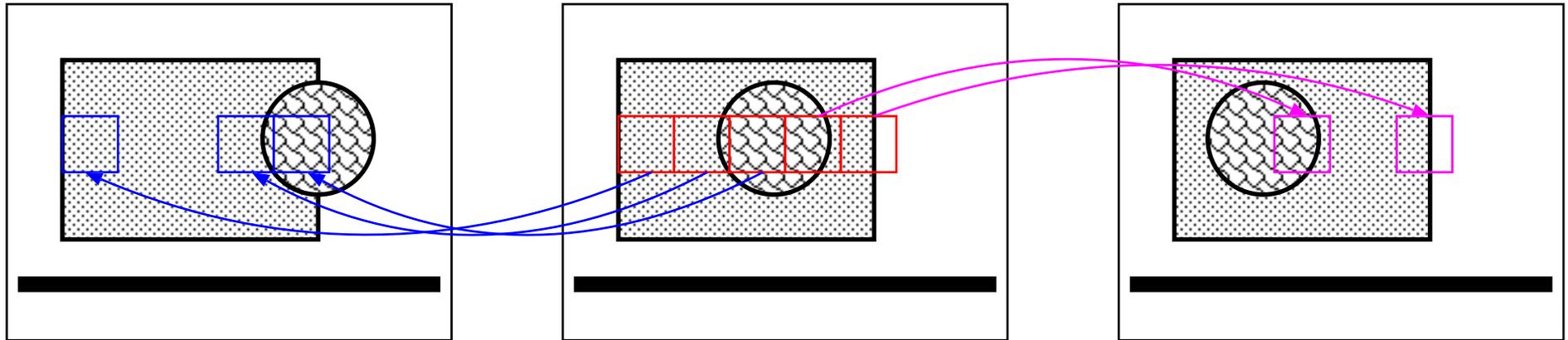


## I, P, Bピクチャ (フレーム)



- Iピクチャ：動き予測補償を行わない。
- Pピクチャ：前のIまたはPピクチャで動き予測補償
- Bピクチャ：前後のIまたはPピクチャで動き予測補償  
前のフレームでは隠れていた部分も補償できる。

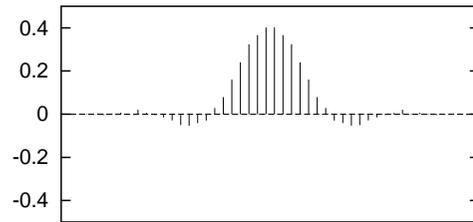
## Bピクチャ (フレーム)



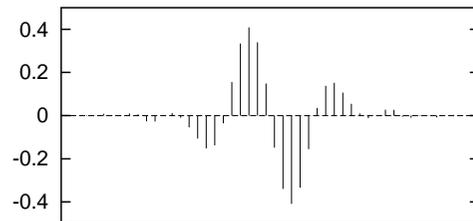
- 中央のフレームの予測画像を生成する。
- 時間的に後のフレームを使うことによって、前のフレームでは隠れていた部分の情報を使うことができる。

# サブバンド変換

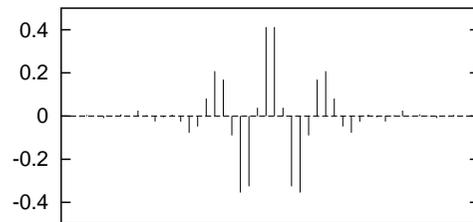
$n = 0$



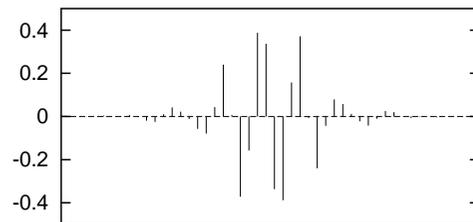
$n = 1$



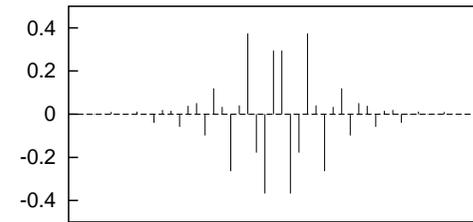
$n = 2$



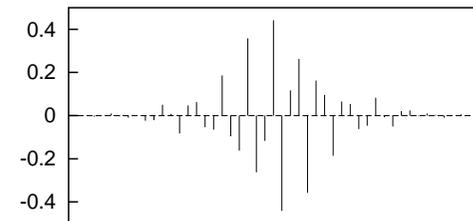
$n = 3$



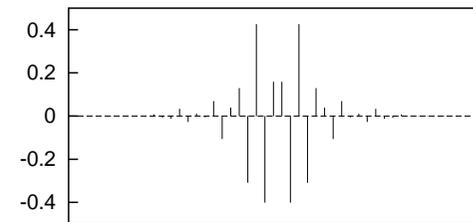
$n = 4$



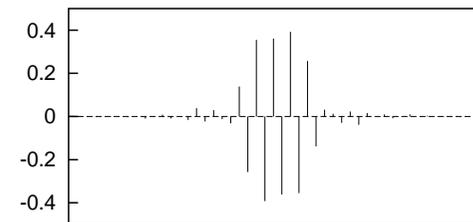
$n = 5$



$n = 6$



$n = 7$



# サブバンド動画像符号化

---

目的：ブロック歪みの少ない動画像符号化

ブロック歪みの原因と対策：

1. ブロック変換

⇒ サブバンド変換(重複変換)の適用

2. ブロック単位の動き保証予測

⇒ 適応デブロッキングフィルタの適用

○ 参照画像を切り貼りして予測画像を作成するときにブロック歪みが生じる。

ブロック歪みが生じる前の参照画像を使うことができる。

⇒ ブロック境界のエッジが本当のエッジかブロック歪みかわかる。

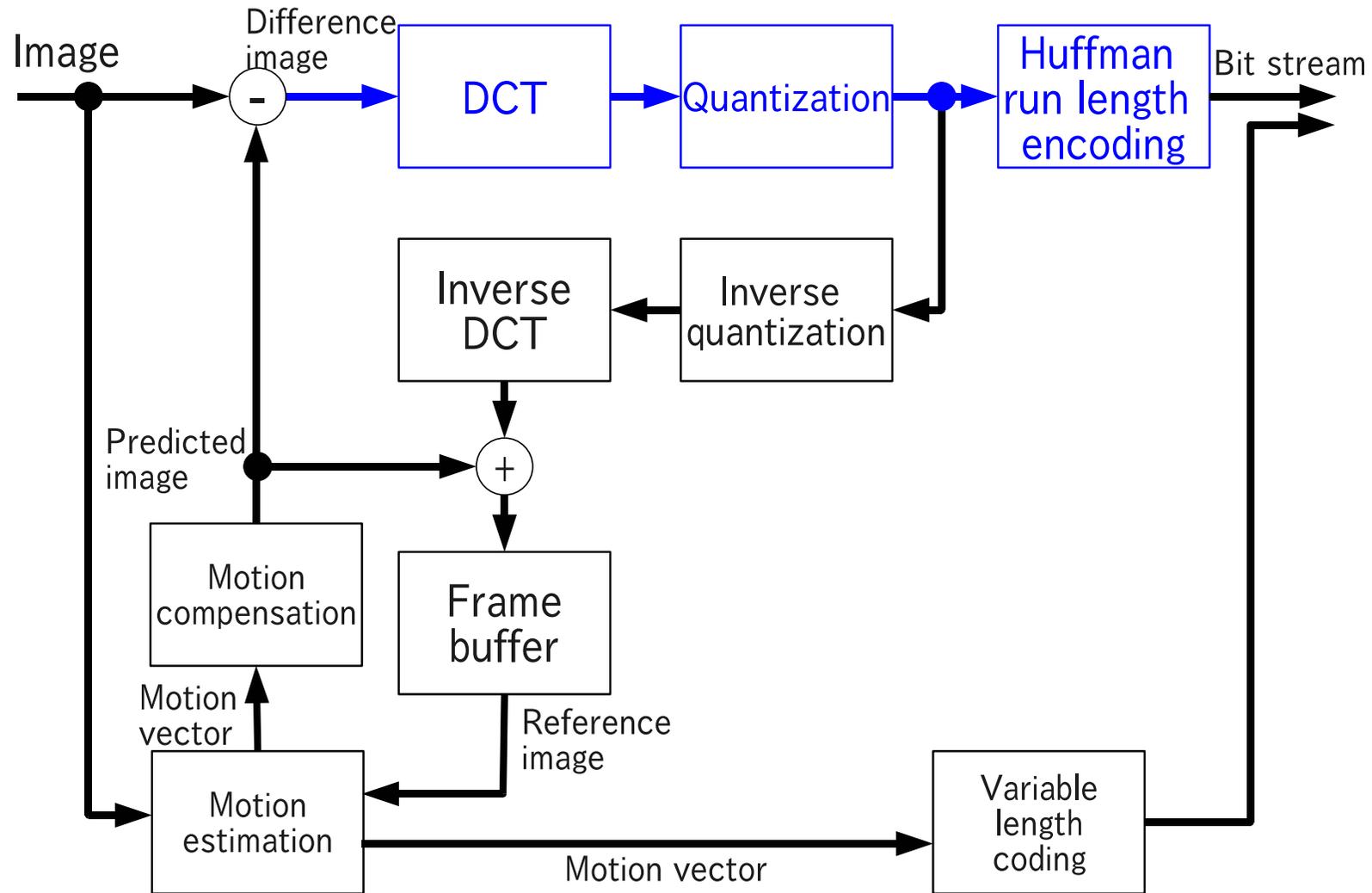
⇒ 適応デブロッキングフィルタが効率的に働く。

× 通常のブロック変換を使う方法では

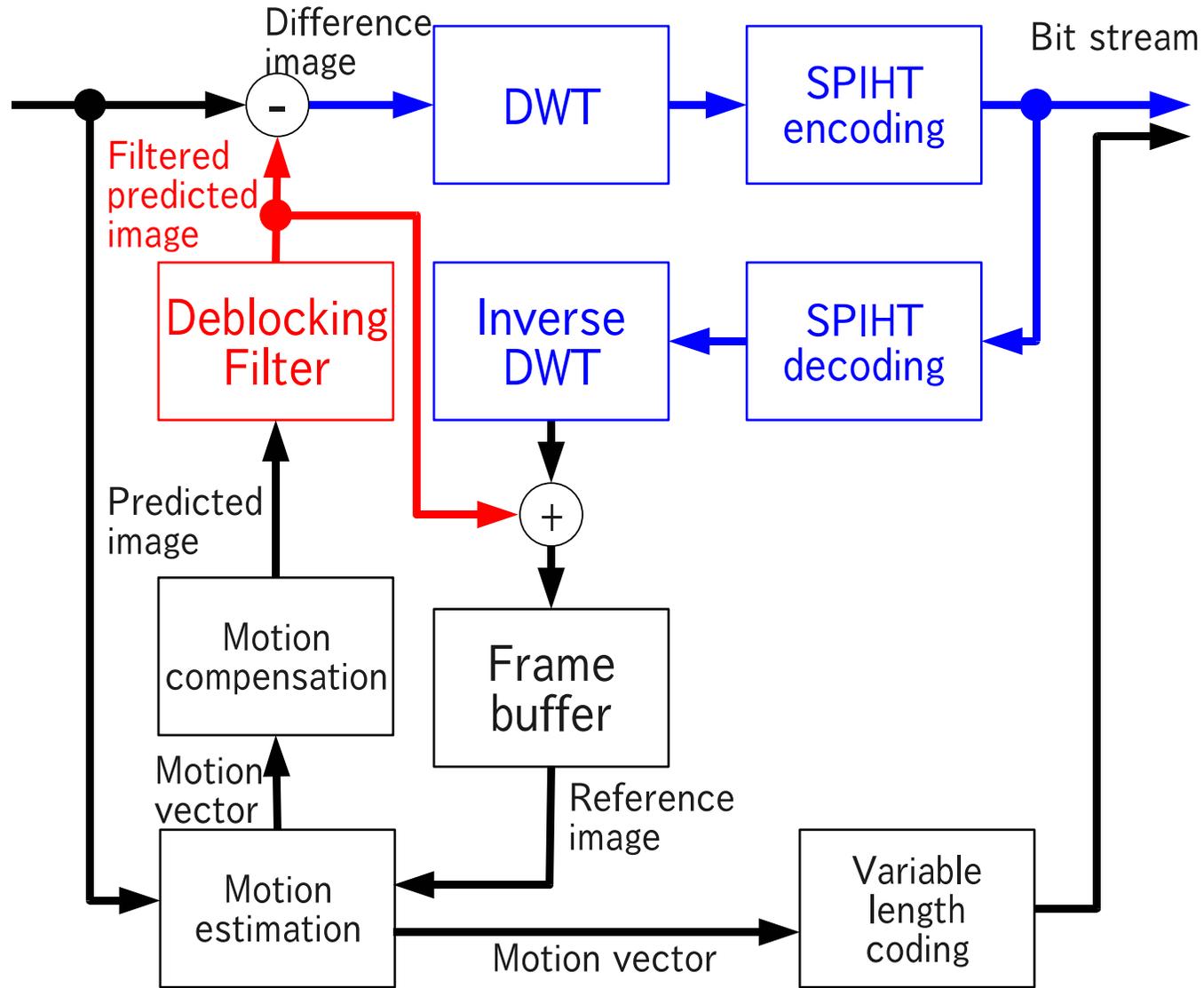
参照画像に既にブロック変換から生じるブロック歪みが生じている。

⇒ 画像のエッジを検出することが難しい。

# MPEGエンコーダのブロック図

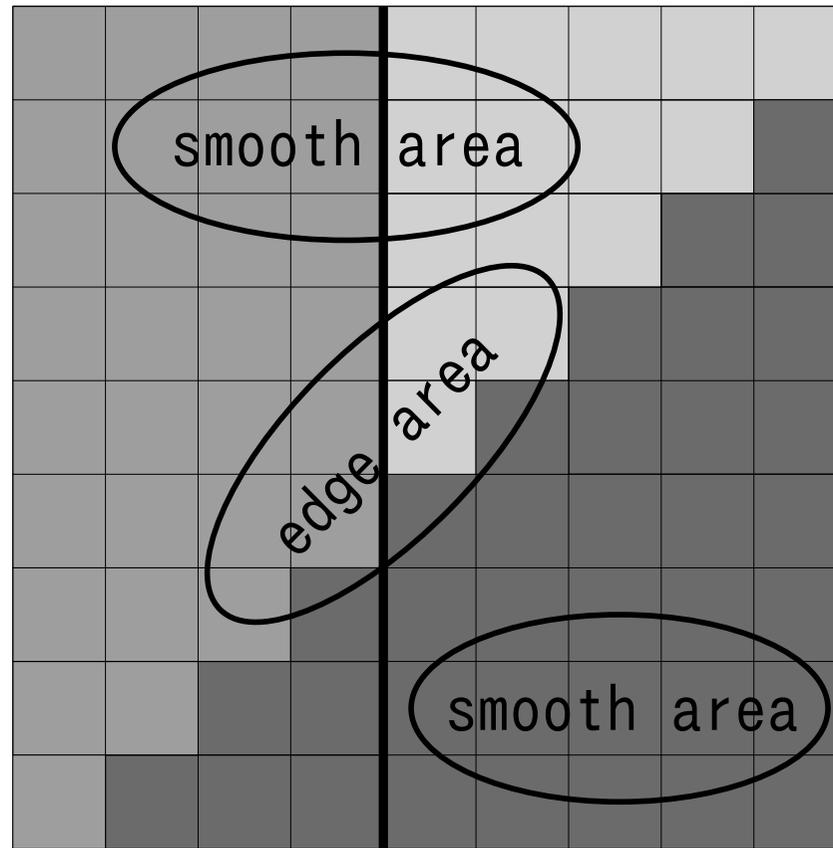


# 提案手法のエンコーダのブロック図



# エッジ適応デブロッキングフィルタ

---



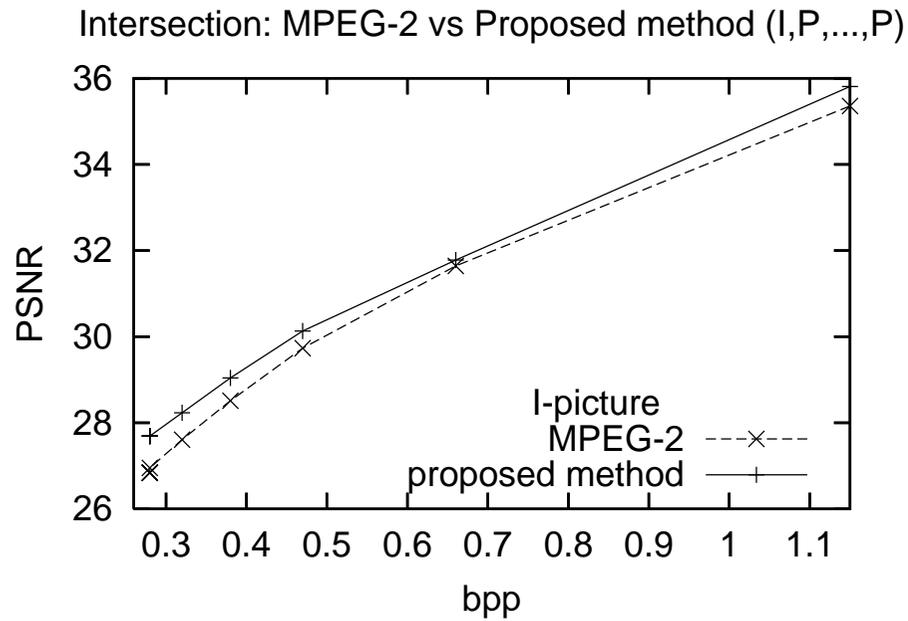
↑  
boundary of the macro block

# 実験

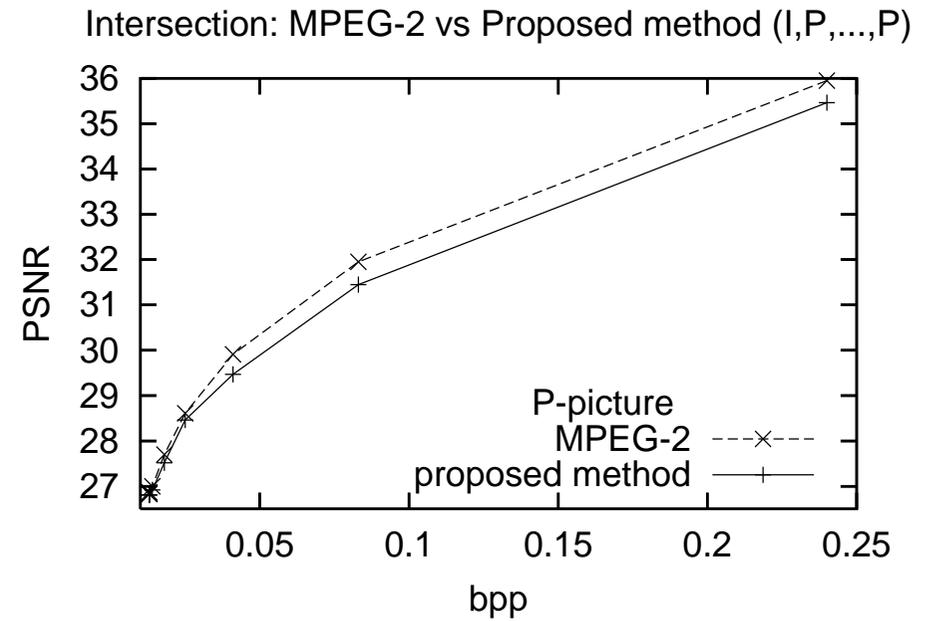


Original image (Intersection)

# 実験結果 (PSNR vs. bit rate)



I-Frame



P-Frame

# 実験結果

---



MPEG2 (ffmpeg)



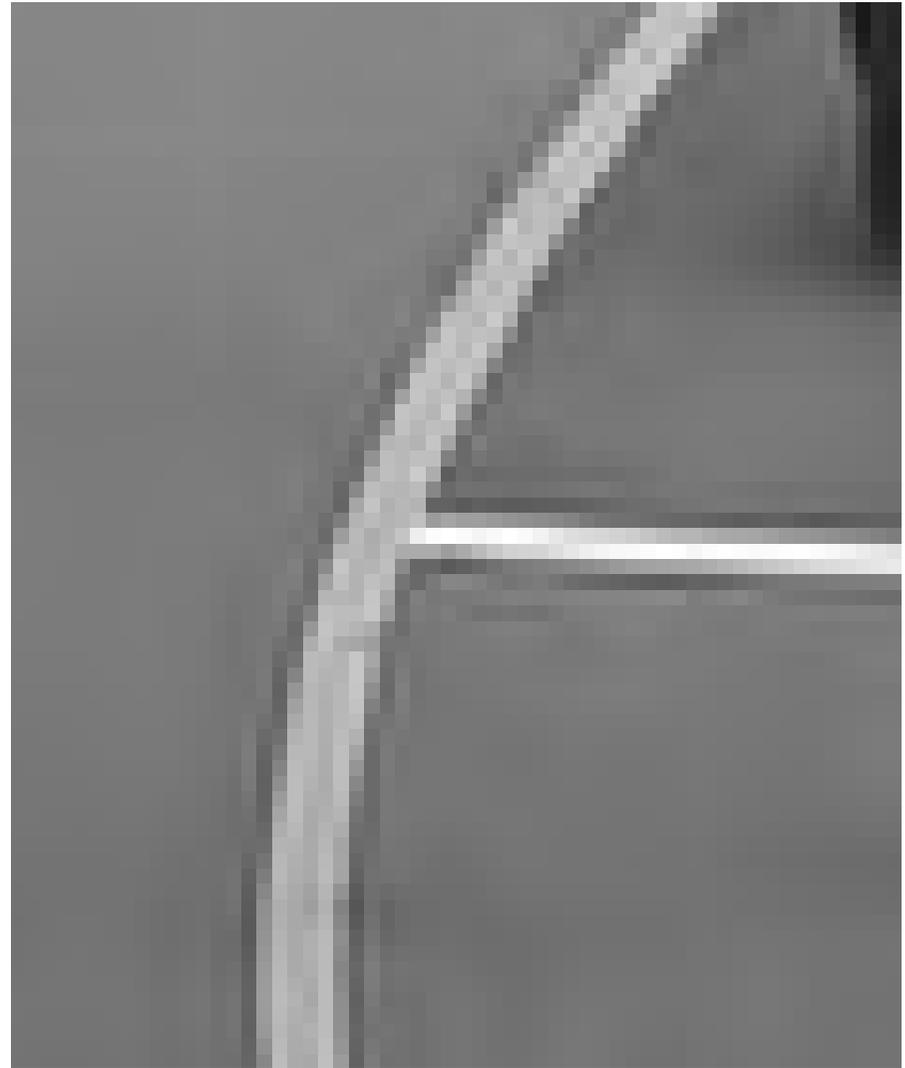
Proposed

## 実験結果 (拡大画像)

---



MPEG2 (ffmpeg)



Proposed

## まとめ

---

- 画像符号化の基盤技術
  - エントロピー符号化
  - 予測符号化
  - 変換符号化 (DCT)
- 標準的画像符号化法
  - JPEG
  - MPEG
- サブバンド符号化 (重複変換)
- 画像符号化は情報技術の集大成
- マルチメディア社会のなかで高い重要性

## 参考文献

- [1] I. Daubechies, “Orthonormal bases of compactly supported wavelets,” *Communications on Pure and Applied Mathematics*, vol.41, no.7 pp.909–996, Oct. 1988.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Transactions on Image Processing*, vol.1, no.2, pp.205–220, Apr 1992.
- [3] H.S. Malvar and D.H. Staelin, “The LOT: transform coding without blocking effects,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.37, no.4, pp.553-0559, Apr 1989.
- [4] R.L. de Queiroz, T.Q. Nguyen, and K.R. Rao, “The GenLOT: generalized linear-phase lapped orthogonal transform, ” *IEEE Transactions on Signal Processing*, vol.44, no.3 pp.497–507, Mar 1996.
- [5] S.C. Chan, “The generalized lapped transform (GLT) for subband coding applications,” *Proceedings of 1995 IEEE International Conference on Proceedings of the Acoustics, Speech, and Signal Processing*, vol.2, pp.1508–1511, 9–12 May 1995.
- [6] H.S. Malvar, “Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts,” *IEEE Transactions on Signal Processing*, vol.46, no.4, pp.1043-1053, Apr 1998.
- [7] T.D. Tran, R.L. de Queiroz, T.Q. Nguyen, “Linear-phase perfect reconstruction filter bank: lattice structure, design, and application in image coding,” *IEEE Transactions on Signal Processing* vol.48, no.1, pp.133–147, Jan 2000.
- [8] T.D. Tran, M. Ikehara, and T.Q. Nguyen, “Linear phase paraunitary filter bank with filters of different lengths and its application in image compression,” *IEEE Transactions on Signal Processing*, vol.47, no.10, pp.2730–2744, Oct. 1999.
- [9] T. Nagai, M. Ikehara, M. Kaneko, and A. Kurematsu, “Generalized unequal length lapped orthogonal transform for subband image coding,” *IEEE Transactions on Signal Processing*, vol.48, no.12, pp.3365–3378, Dec. 2000.

- [10] T.D. Tran, R.L. de Queiroz, and T.Q. Nguyen, “The variable-length generalized lapped biorthogonal transform,” Proceedings of 1998 IEEE International Conference on Image Processing, vol.3, pp.697–701, 4–7 Oct. 1998.
- [11] T. Tanaka and Y. Yamashita, “An adaptive lapped biorthogonal transform and its application in orientation adaptive image coding,” Signal Processing, vol.82, no.11, pp.1633–1647, Nov. 2002.
- [12] T. Tanaka, Y. Hirasawa, and Y. Yamashita, “Variable-length lapped transform with combination of multiple synthesis filter banks for image coding,” IEEE Transactions Image Processing, vol.15, no.1, pp.81–88, Jan. 2006.
- [13] J.M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” IEEE Transactions on Signal Processing, vol.41, no.12 pp.3445–3462, Dec. 1993.
- [14] A. Said and W.A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” IEEE Transactions on Circuits and Systems for Video Tech., vol.6, pp.243–250, June 1996.
- [15] Nasharuddin Zainal, Toshihisa Tanaka and Yukihiro Yamashita, “Moving picture coding by lapped transform and edge adaptive deblocking filter with zero pruning SPIHT,” IEICE Trans. on Information and Systems, vol. E93-D, no. 6, pp.1608–1617, June, 2010.
- [16] 小野定康, 鈴木純司, “わかりやすいJPEG2000の技術,” オーム社, 2003.
- [17] 大久保榮, 角野眞也, 菊池義浩, 鈴木輝彦, “H.264/AVC 教科書,” インプレス, 2004.
- [18] Telecommunication standardization sector of ITU, “Advanced video coding for generic audiovisual services,” ITU, 2007.