

# 画像情報特論 (3)

- ビデオ圧縮: H.264/AVC

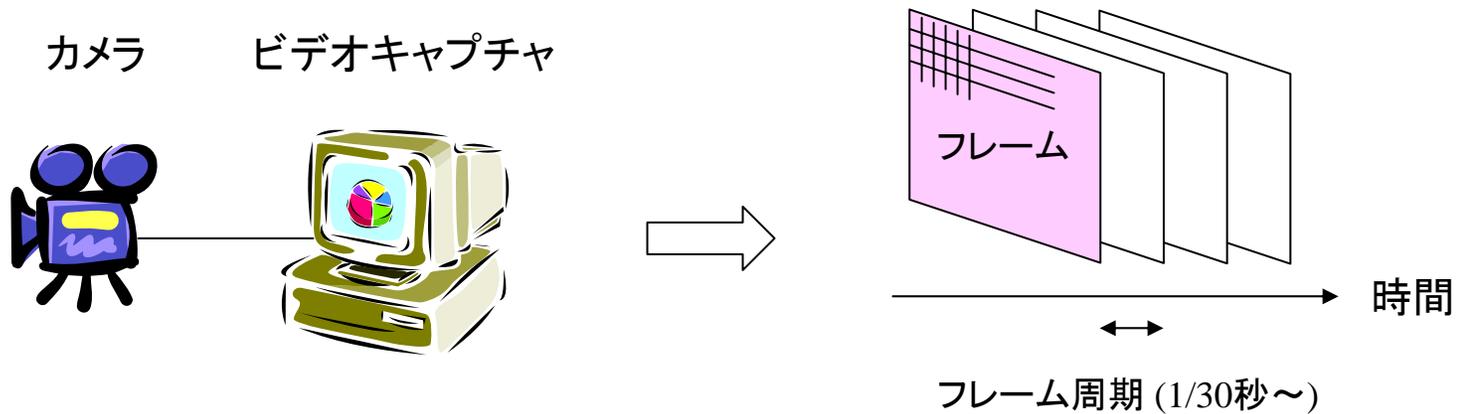
情報理工学専攻 甲藤二郎

E-Mail: [katto@waseda.jp](mailto:katto@waseda.jp)

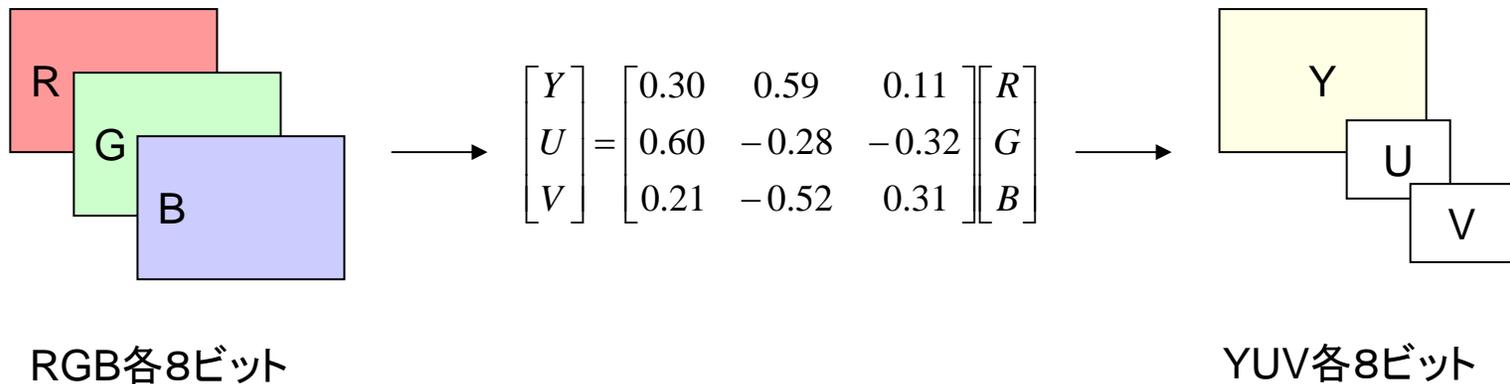
# ビデオ圧縮の基礎

# デジタルビデオ (1)

- 時間方向・空間方向のサンプリング

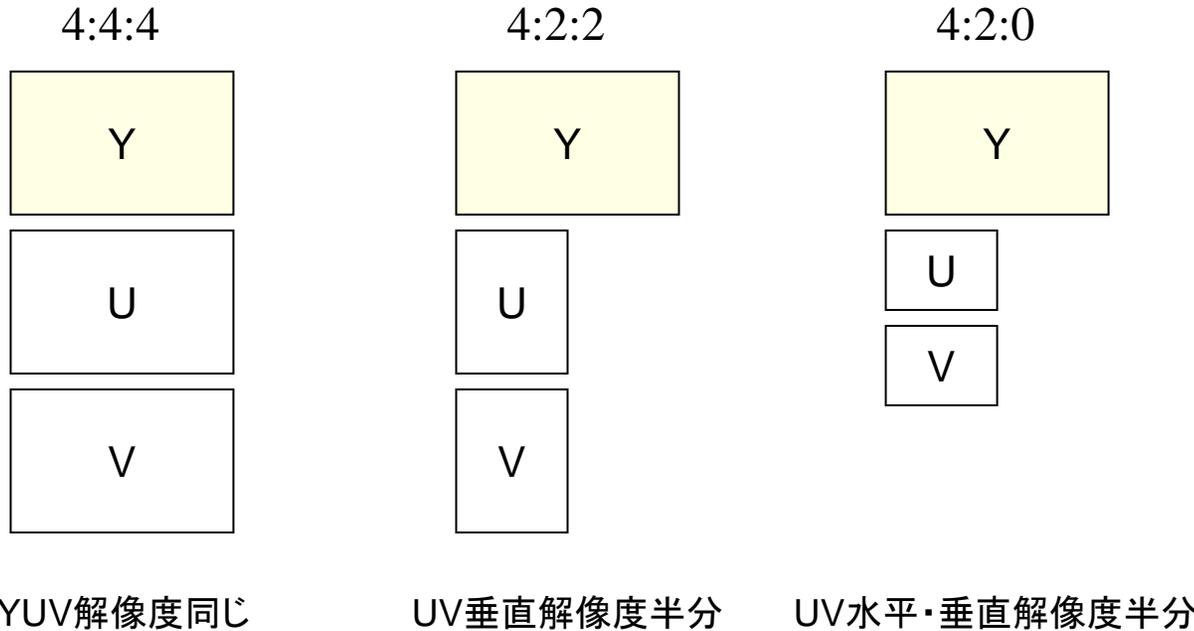


- RGB / YUV 変換



# デジタルビデオ (2)

- CCIR 601 フォーマット

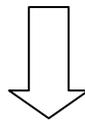


- 通常のビデオ圧縮： 4:2:0 フォーマット
- 高画質ビデオ圧縮： 4:2:2 フォーマット

# デジタルビデオ (3)

- 莫大な情報量 (RGB各8ビット無圧縮の場合)

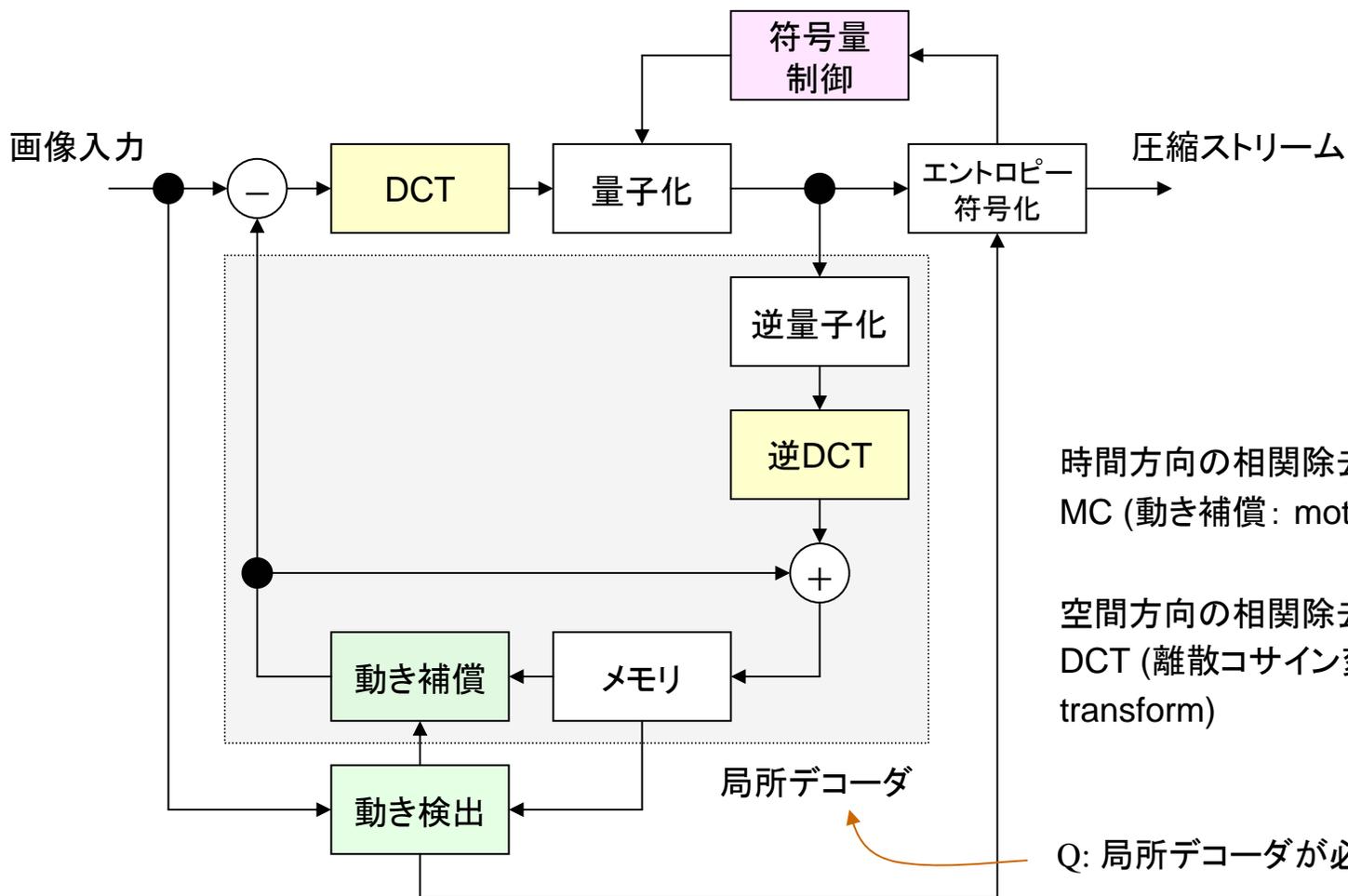
用途	解像度	データ量
TV会議	352x240	21Mbit/s
TV	720x480	83Mbit/s
HDTV	1920x1080	498Mbit/s



データ圧縮の必要性

# ビデオ圧縮の仕組み

- MC+DCT ハイブリッド予測符号化 (30年間変わらない方式)



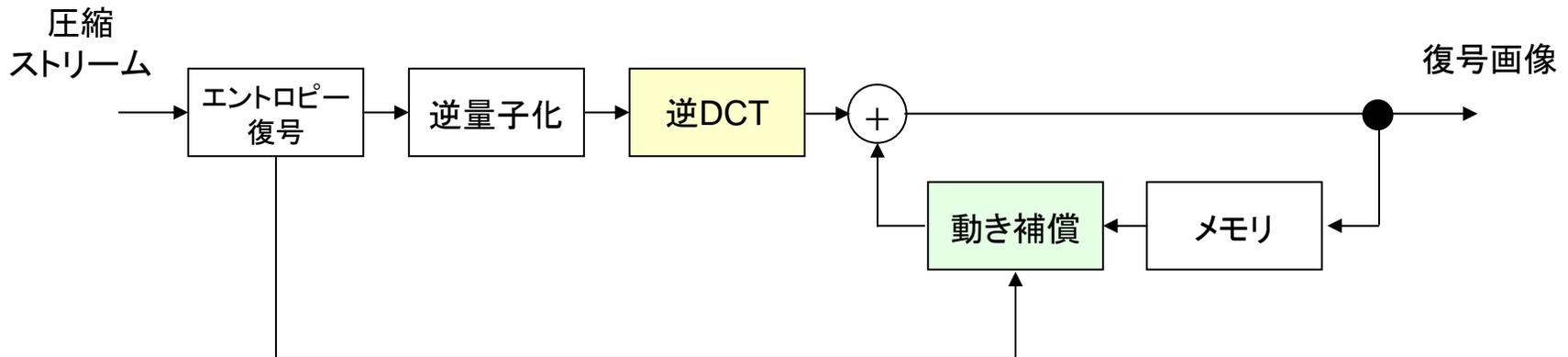
時間方向の相関除去:  
MC (動き補償: motion compensation)

空間方向の相関除去:  
DCT (離散コサイン変換: discrete cosine transform)

Q: 局所デコーダが必要な理由を説明せよ

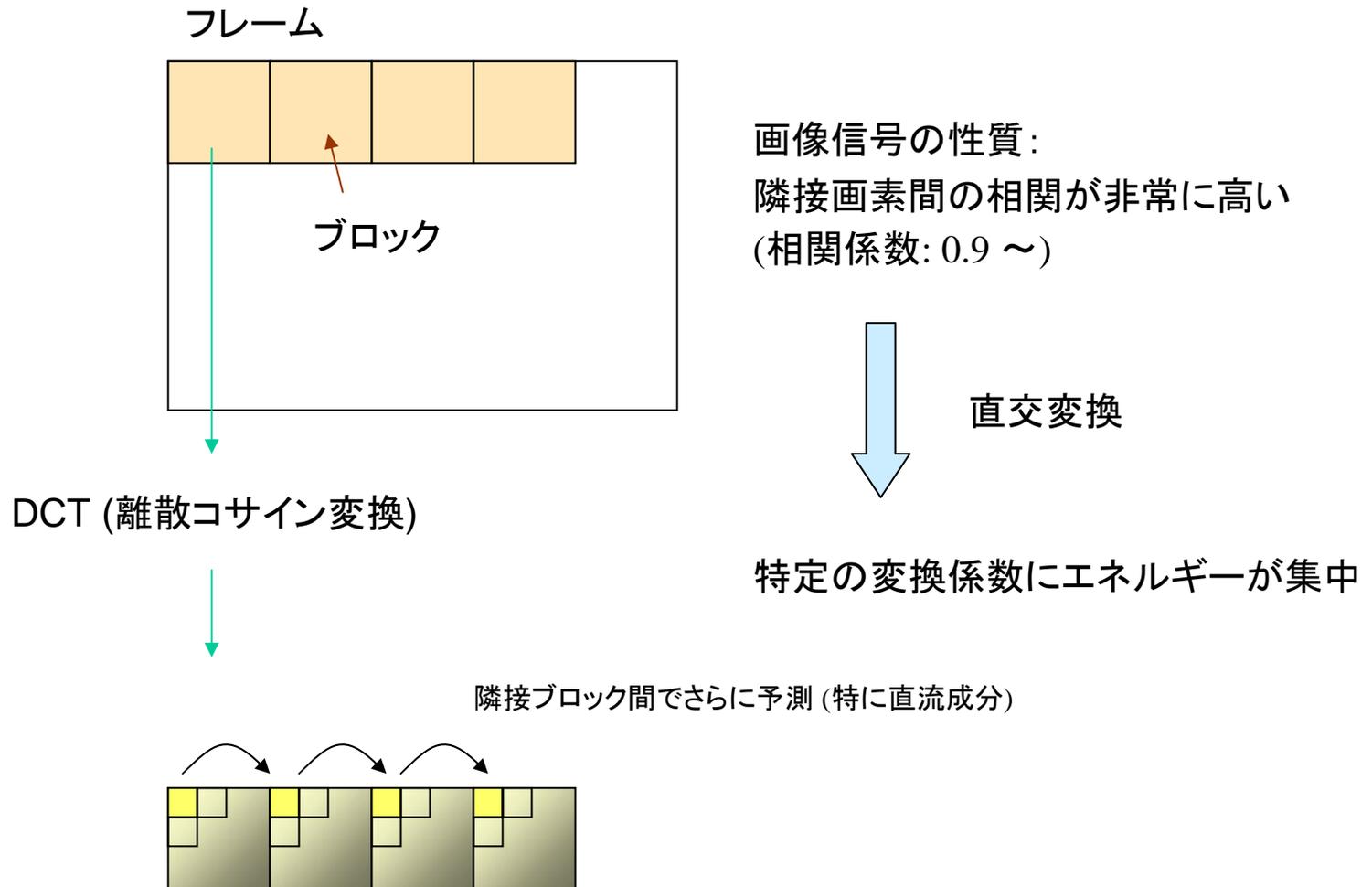
# ビデオ復号の仕組み

- エンコーダのローカルデコードに同じ



# フレーム内符号化

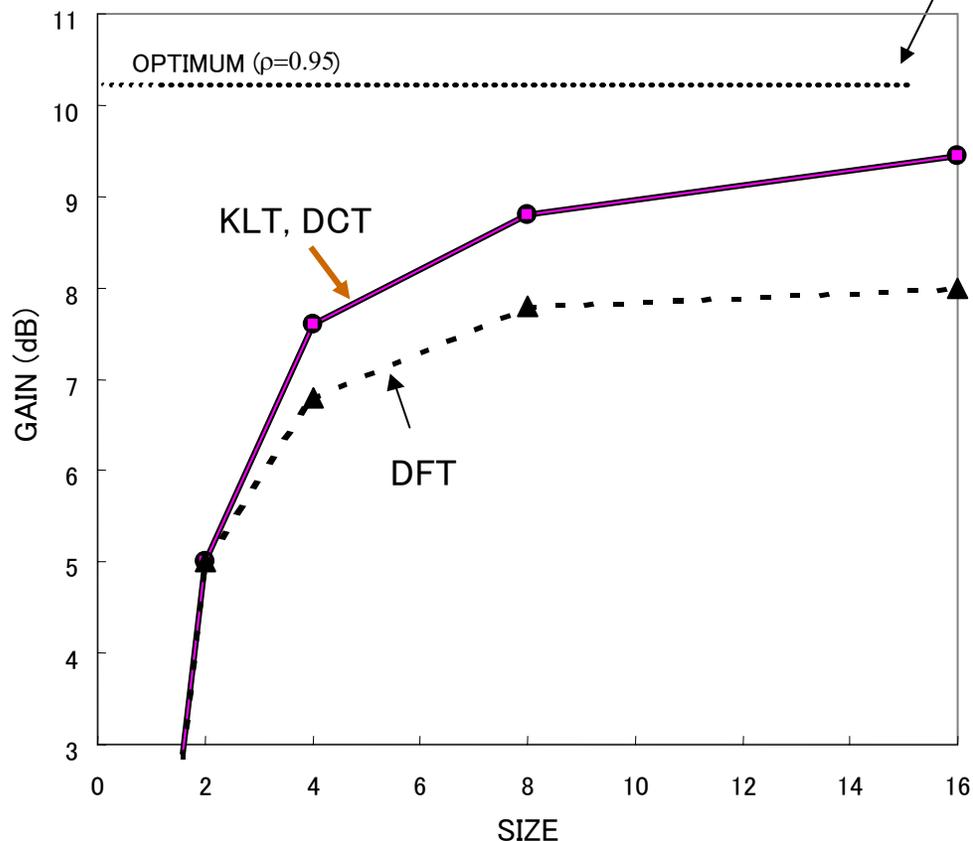
- DCT



# 直交変換 (1)

## • DCTが使われる理由

圧縮効率 KLT, DCT, DFT の符号化利得の比較 理論的最適値



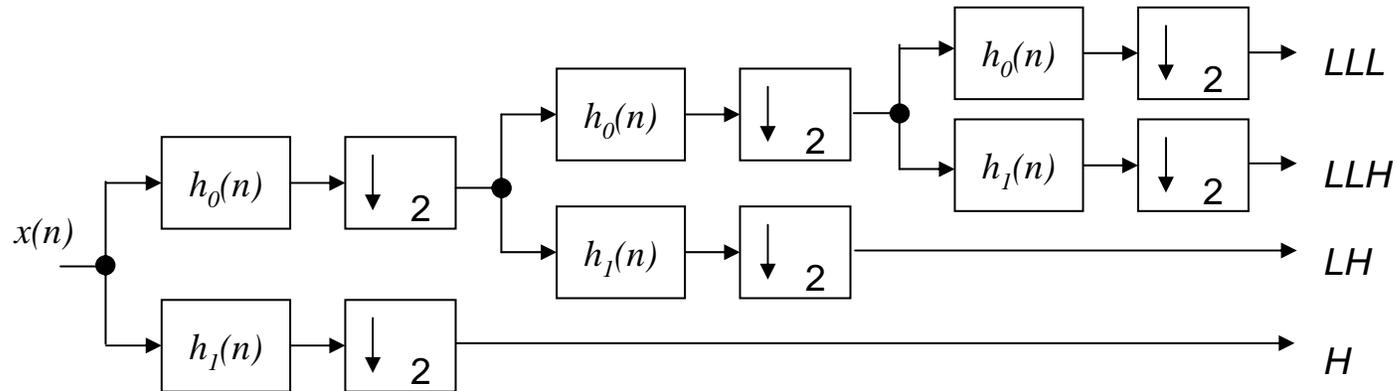
KLT: 理論的に最適な直交変換。

DCT: 相関の高い入力に対する KLT への漸近性、及び高速アルゴリズムが存在。通常は 8x8 サイズの DCT を使用。

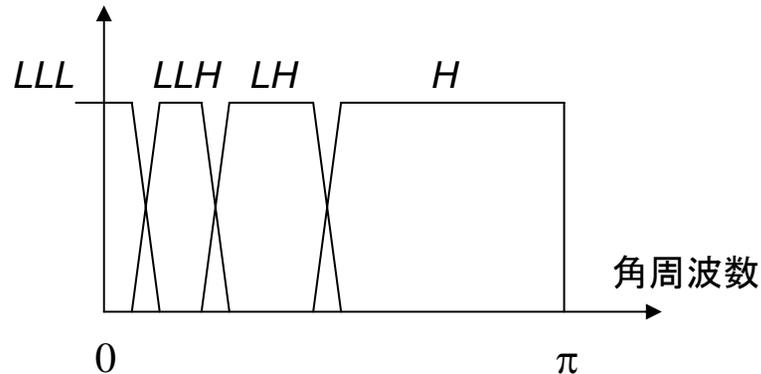
→ 直交変換の  
ブロックサイズ

# 直交変換 (2)

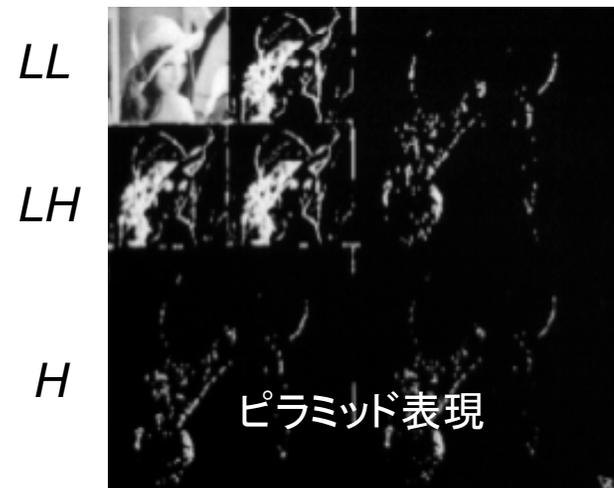
- Wavelet 変換 (対抗)



2分割フィルタバンクのツリー接続

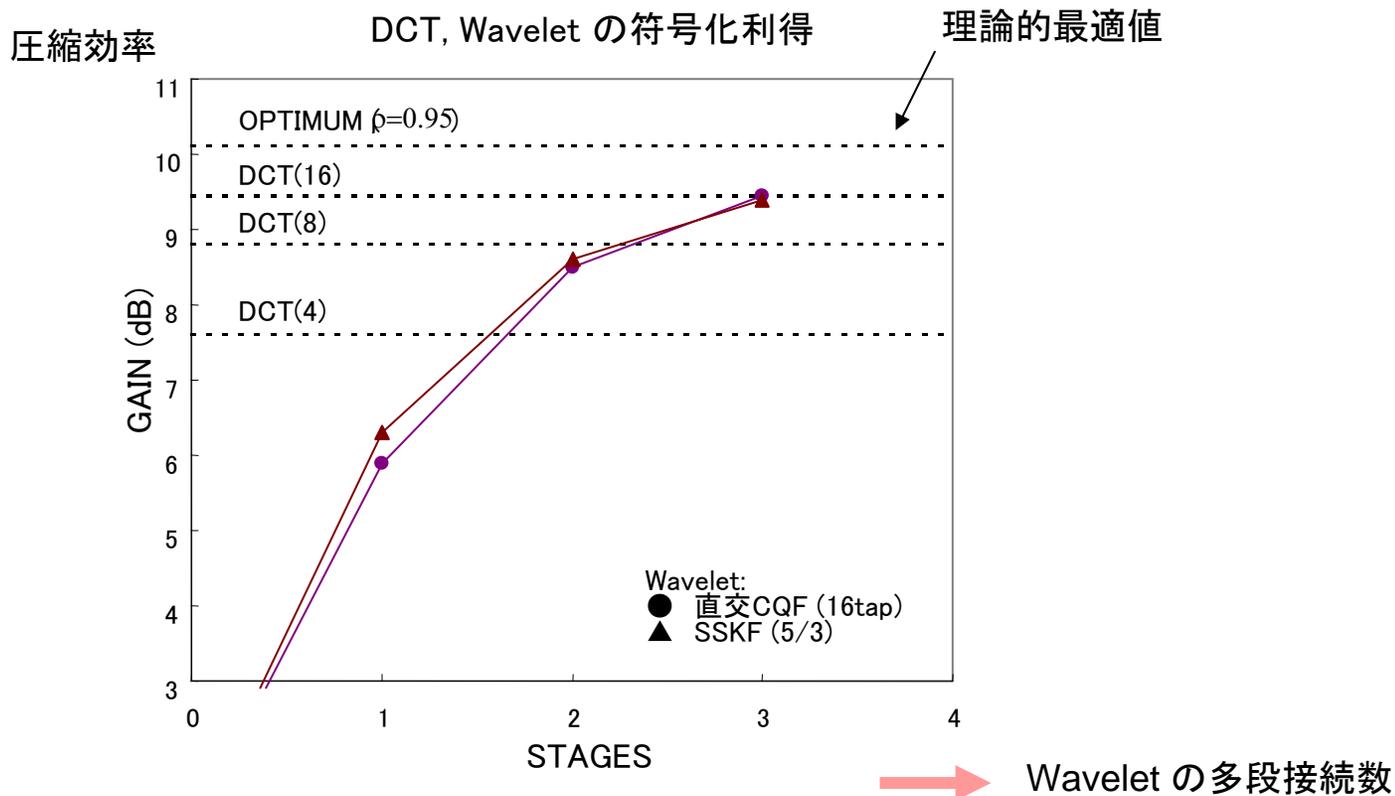


- 長所: ブロックひずみが少ない
- 短所: ブロック動き補償と相性が悪い



# 直交変換 (3)

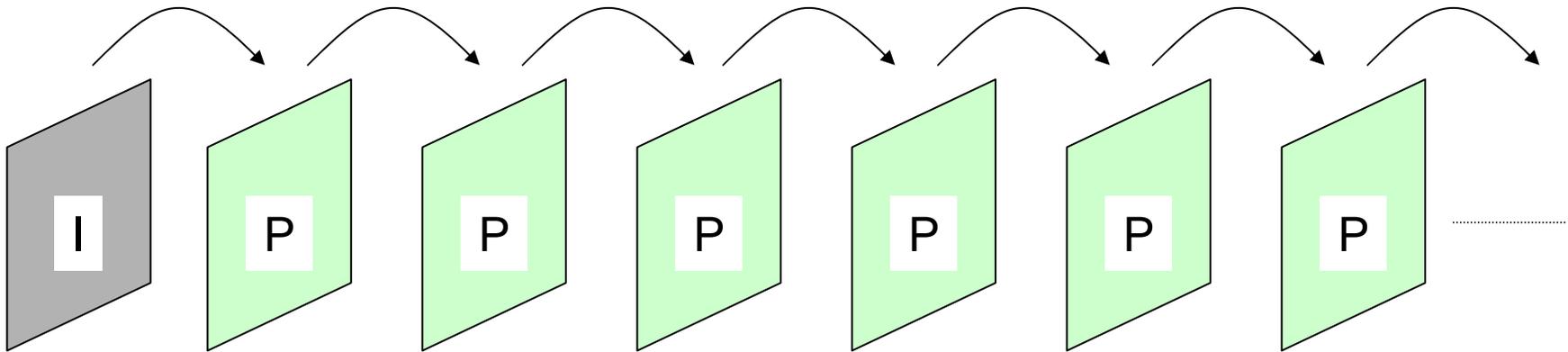
## • DCT と Wavelet の比較



- DCT: 動画 (ビデオ) 圧縮
- Wavelet: 静止画圧縮 (JPEG-2000)

# フレーム間符号化 (1)

- IP 予測



ビデオ信号の性質:

隣接フレーム間の相関が非常に高い  
(相関係数: 0.9 ~)



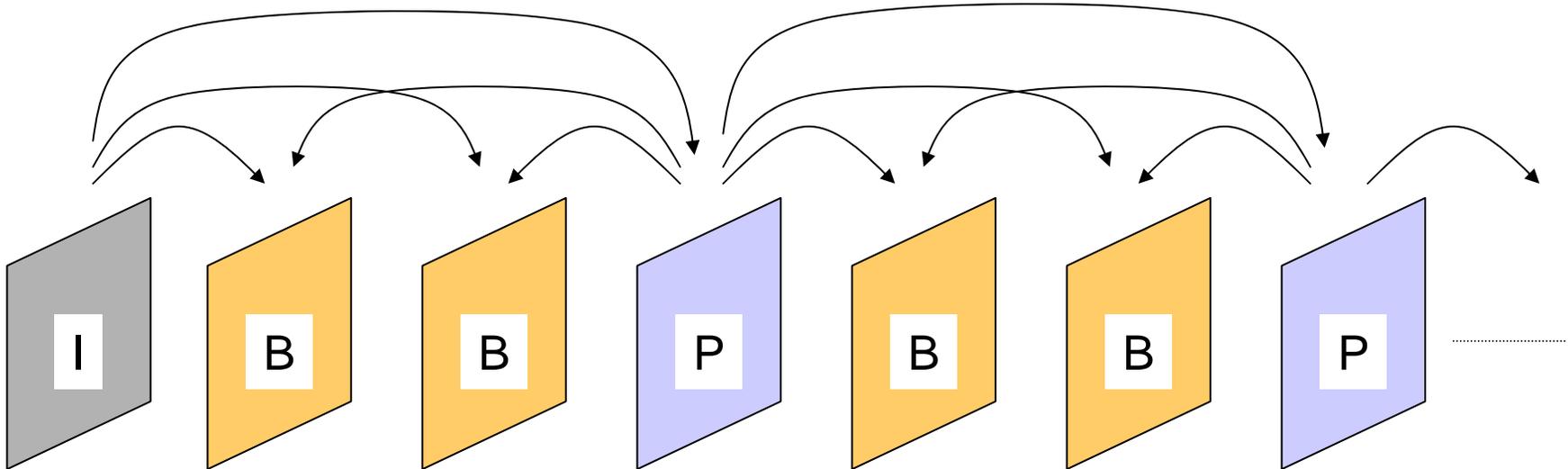
フレーム間の予測誤差がほとんどゼロ

- I: Iピクチャ (フレーム内符号化)
- P: Pピクチャ (フレーム間符号化)

さらに動き検出・動き補償予測

# フレーム間符号化 (2)

## • IPB 予測



片方向で予測を行うより、両方向で予測を行うほうが予測効率が高い (ただし、フレーム間の距離に依存)

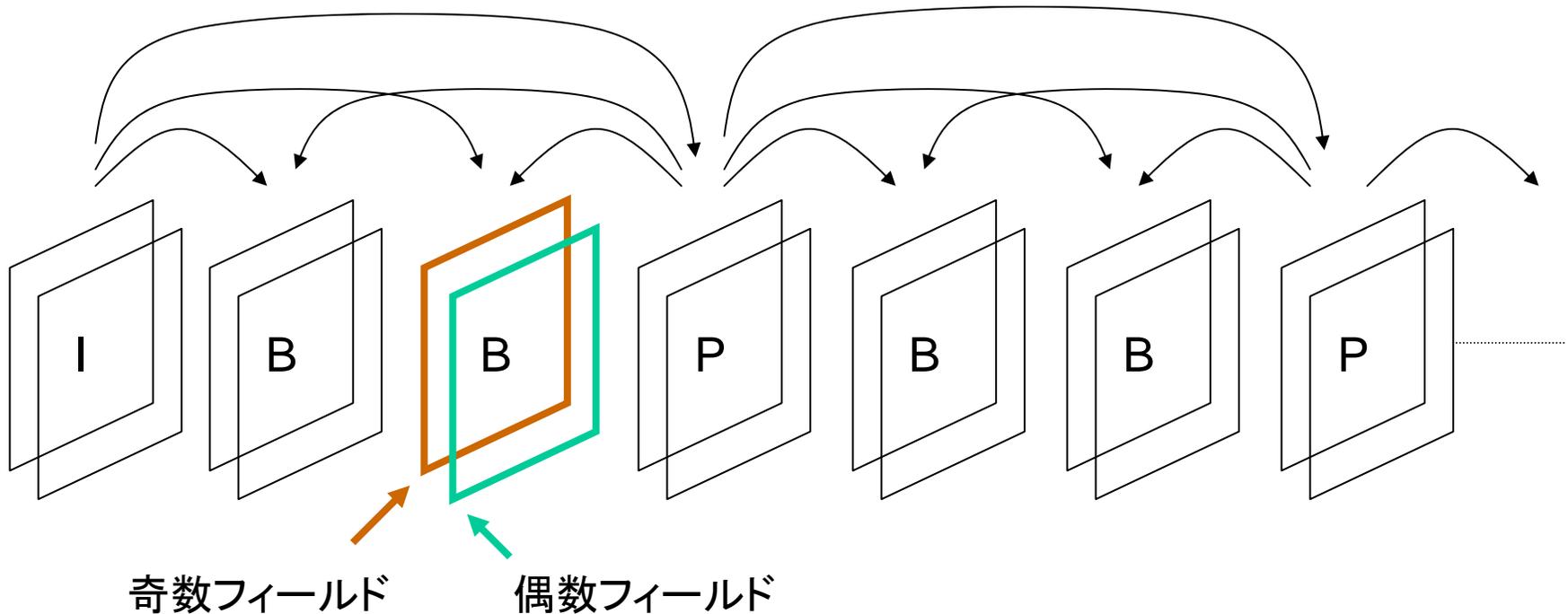


- I: Iピクチャ (フレーム内符号化)
- P: Pピクチャ (片方向予測)
- B: Bピクチャ (両方向予測)

→ 予測効率の改善

# フレーム間符号化 (3)

- フレーム・フィールド適応予測



デジタルTV放送に対応 (MPEG-2)

- 動き補償: フィールド予測、フレーム予測、デュアルプライム予測
- DCT: フレームDCT、フィールドDCT

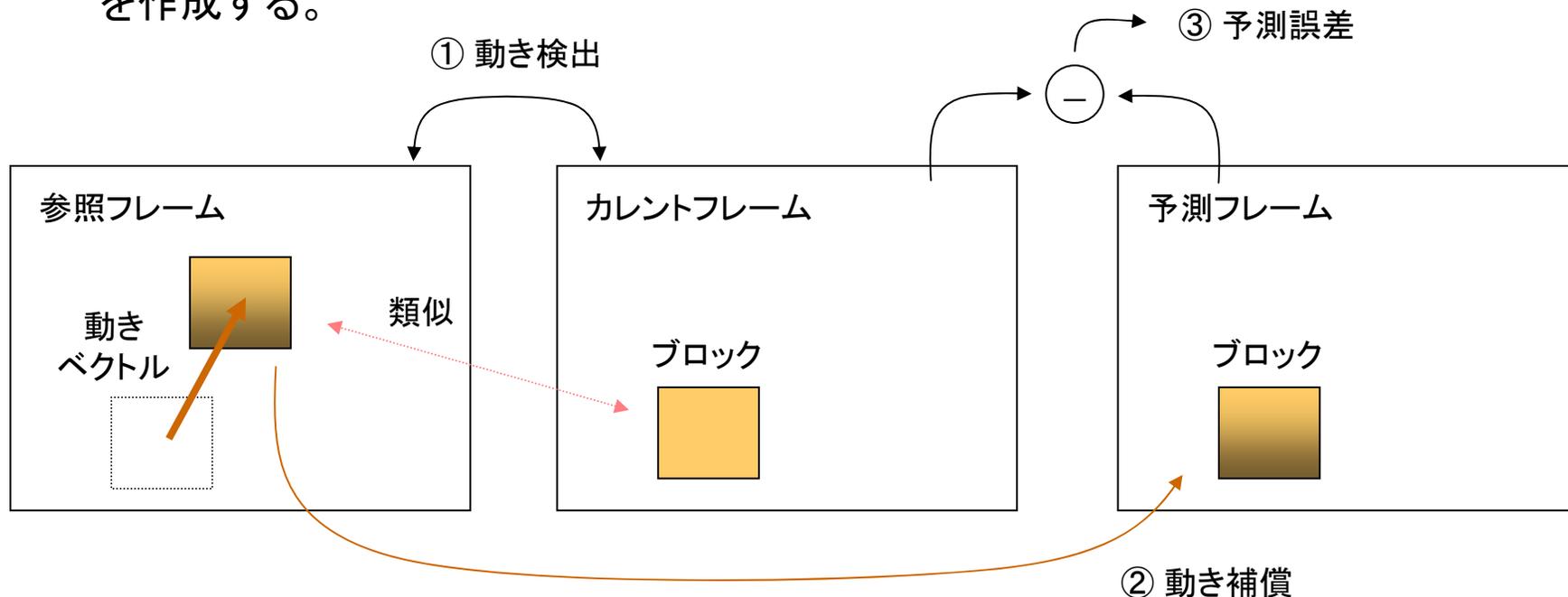
# 動き検出と動き補償 (1)

- 動き検出 (ブロックマッチング):

過去の画像 (参照フレーム) から、現在の画像 (カレントフレーム) に最も類似しているブロックを探索し、動きベクトルを求める。

- 動き補償:

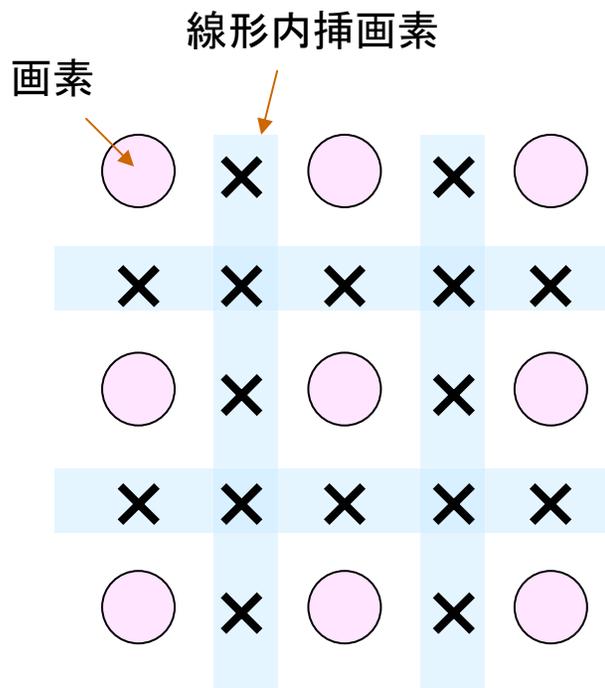
動き検出で求めた動きベクトルから、カレントフレームの予測画像 (予測フレーム) を作成する。



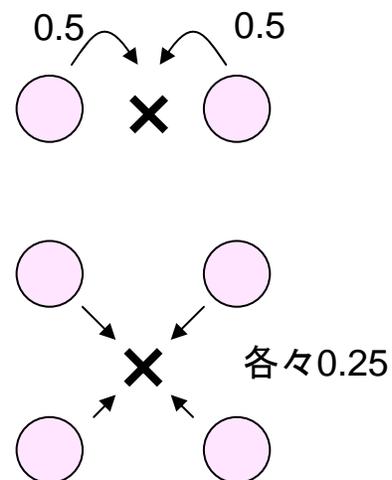
# 動き検出と動き補償 (2)

- 半画素精度動き補償:

線形内挿を行い、0.5 画素精度の動きベクトルを算出し、予測画像を作成。



内挿フィルタ:

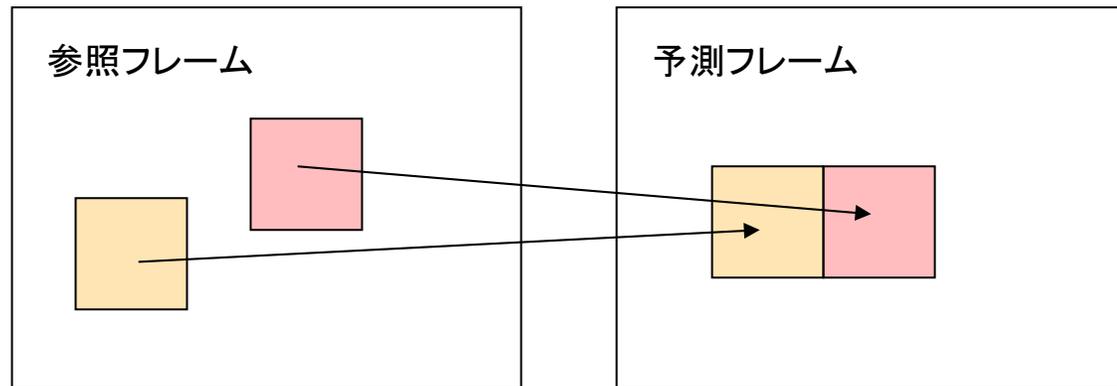


# 動き検出と動き補償 (3)

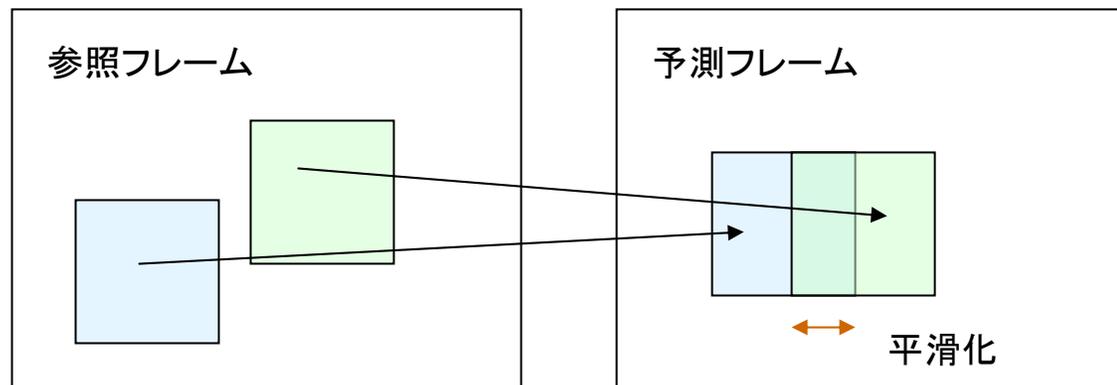
- オーバーラップ動き補償:

隣接ブロックの動きベクトルも利用し、ブロックの平滑化加算によって予測画像を作成。

通常ブロックマッチング



オーバーラップ動き補償



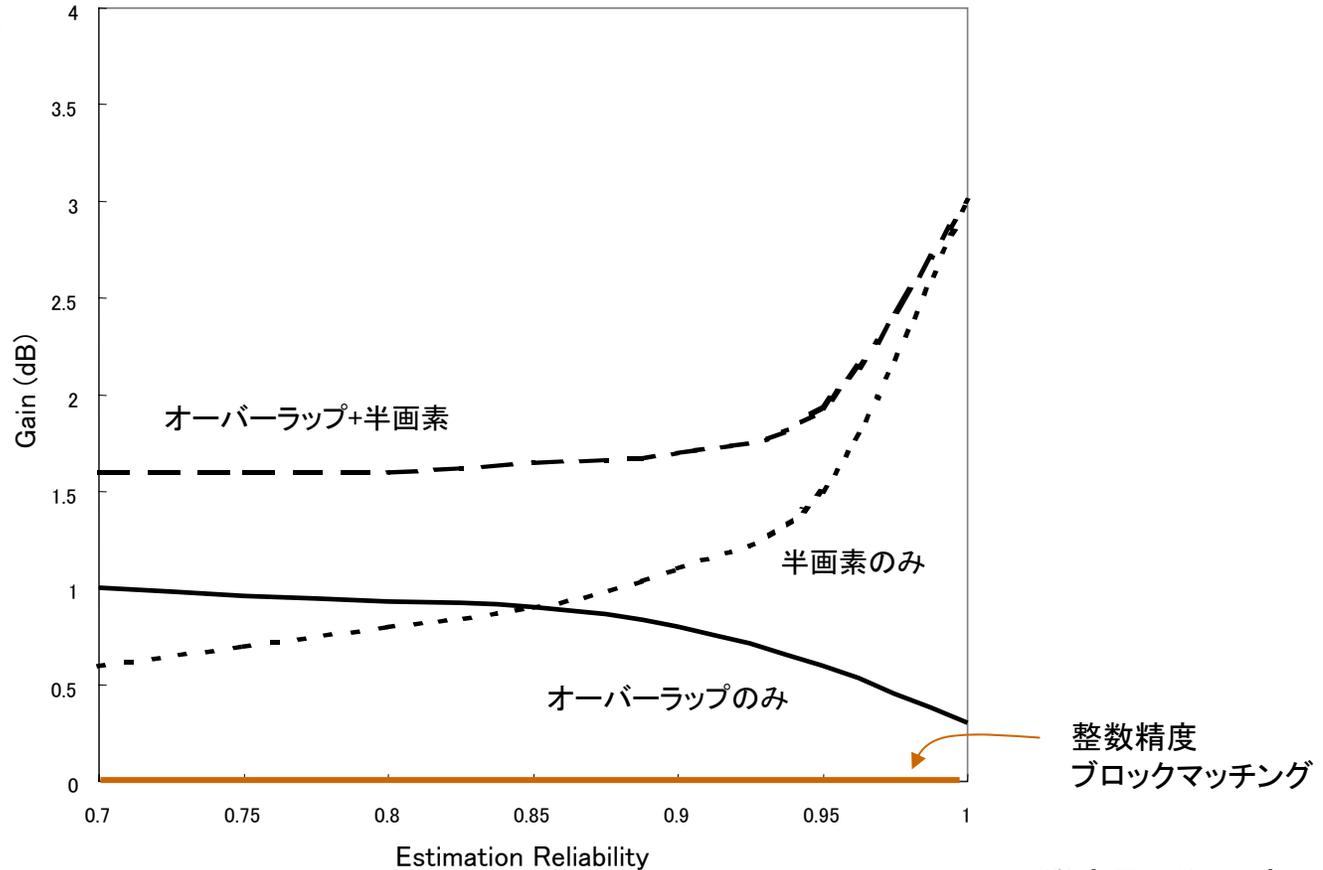
平滑化: 台形ウィンドウ、  
コサインウィンドウなど。

# 動き検出と動き補償 (4)

## • 特性比較

整数画素精度・ブロック動き補償に対する  
半画素精度・オーバーラップ動き補償の予測利得

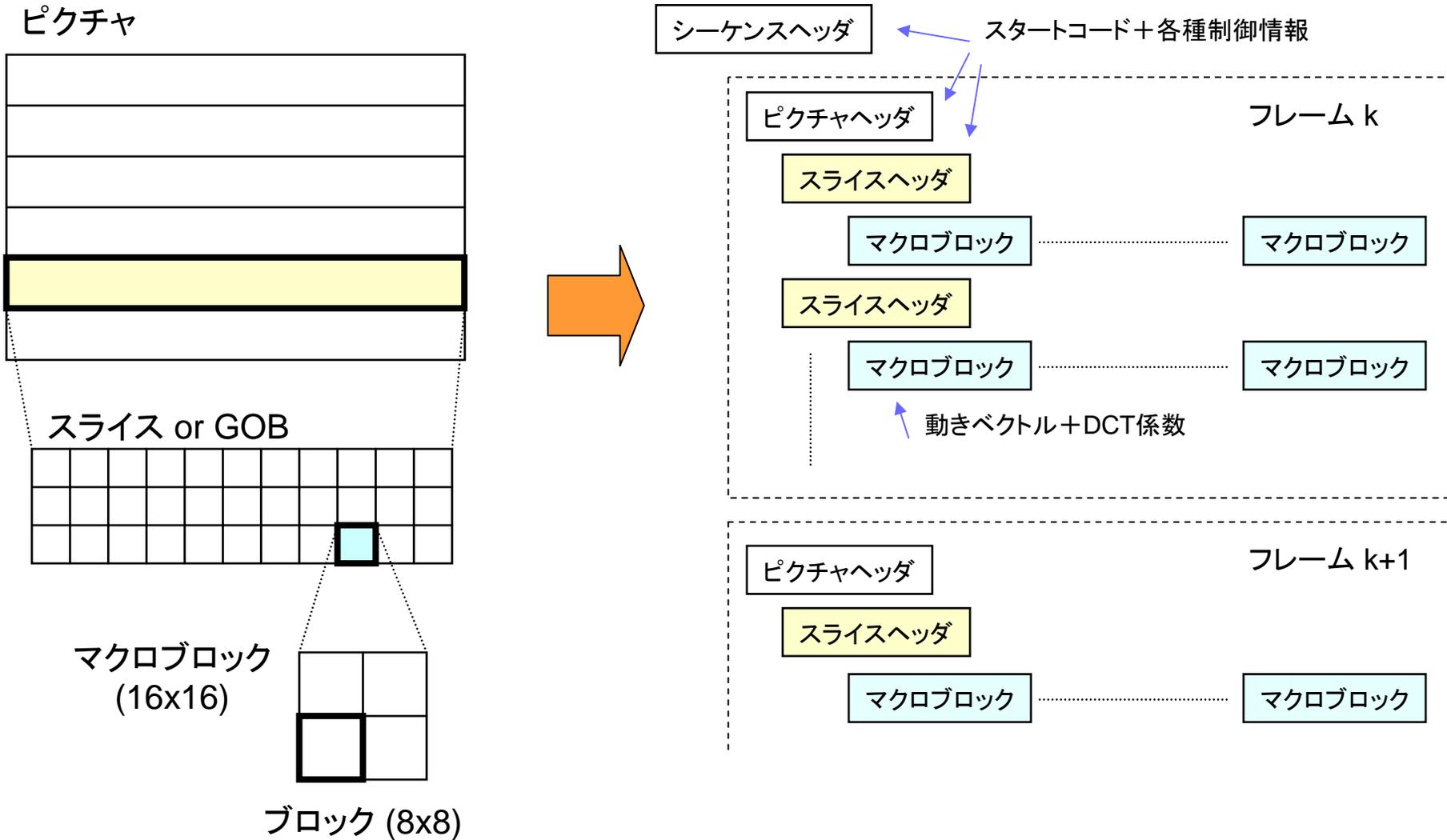
予測誤差の  
低減効果



動き予測しにくい  
画像の場合

動き予測しやすい  
画像の場合

# ビットストリーム構造



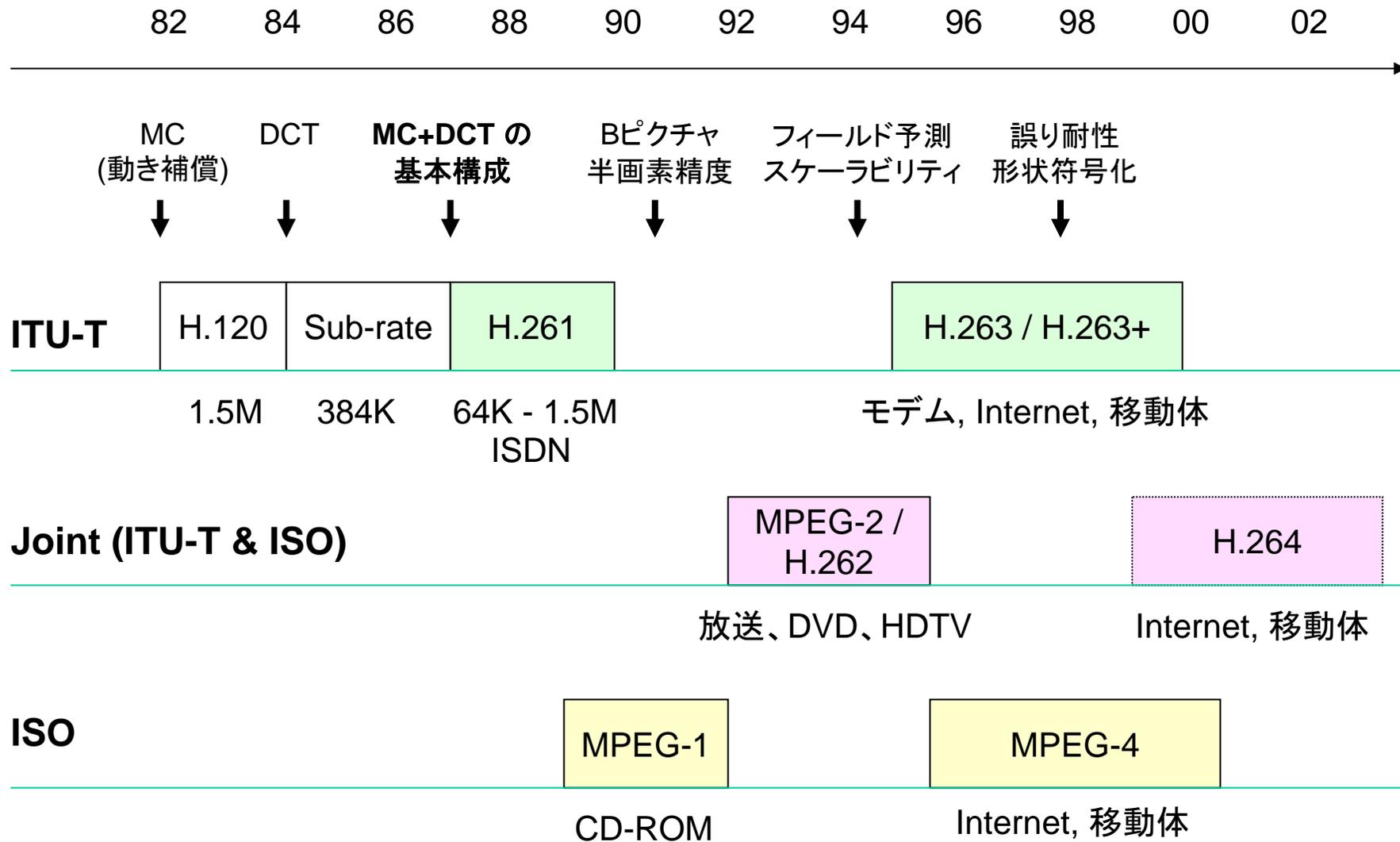
# 國際標準方式

# 国際標準方式 (1)

団体	名称	時期	符号化レート	当初の用途
ITU-T	H.261	1990年	64kb/s~2Mb/s	ISDN用テレビ電話
	H.263	1996年	数十kb/s~	アナログ回線用テレビ電話
	H.263+	1998年	数十kb/s~	インターネット、移動体
	H.264	2003年	数十kb/s~	インターネット、移動体
ISO	MPEG-1	1992年	~1.5Mb/s	CD-ROM
	MPEG-2*	1995年	数Mb/s~数十Mb/s	デジタル放送
	MPEG-4	1999年	数十kb/s~	インターネット、移動体

\* MPEG-2/H.262、H.264/AVC (MPEG-4 Part 10) はISOとITU-Tのジョイント規格

# 国際標準方式 (2)



# 国際標準方式 (3)

- 代表的な機能の比較

名称	MC+DCT	1/2画素	IPB予測	フィールド	形状符号化	再同期	スケーラビリティ
H.261	○	-	-	-	-	-	-
H.263	○	○	△	-	-	-	-
MPEG-1	○	○	○	-	-	○	-
MPEG-2	○	○	○	○	-	○	○
H.263+	○	○	△	-	△	○	○
MPEG-4	○	○	○	○	○	○	FGS
H.264	○	○	○	-	△	○	SVC

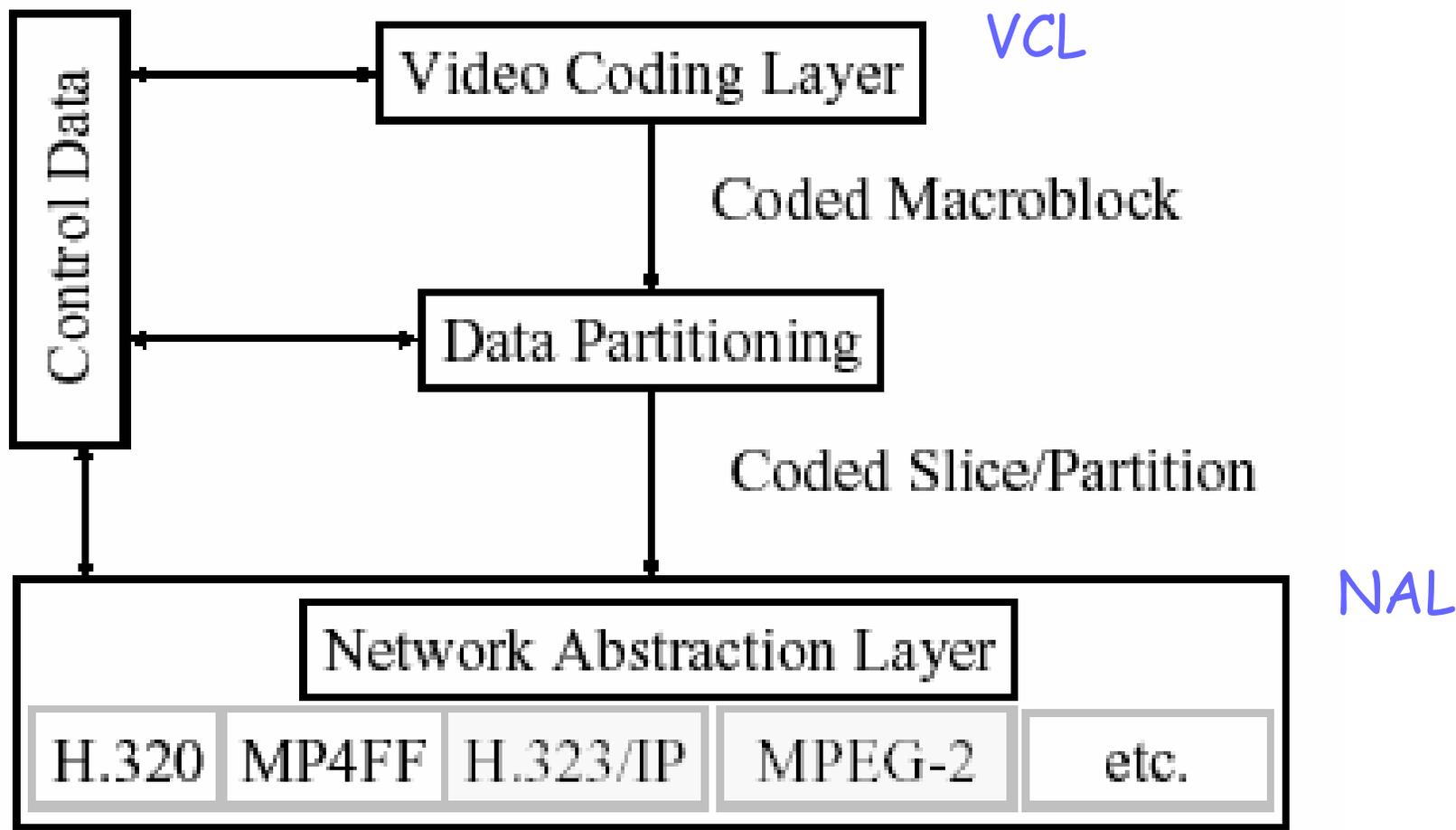
インターネット放送で有効  
+ 符号量制御 (後述)

# ITU-T H.264/AVC (MPEG-4 Part 10)

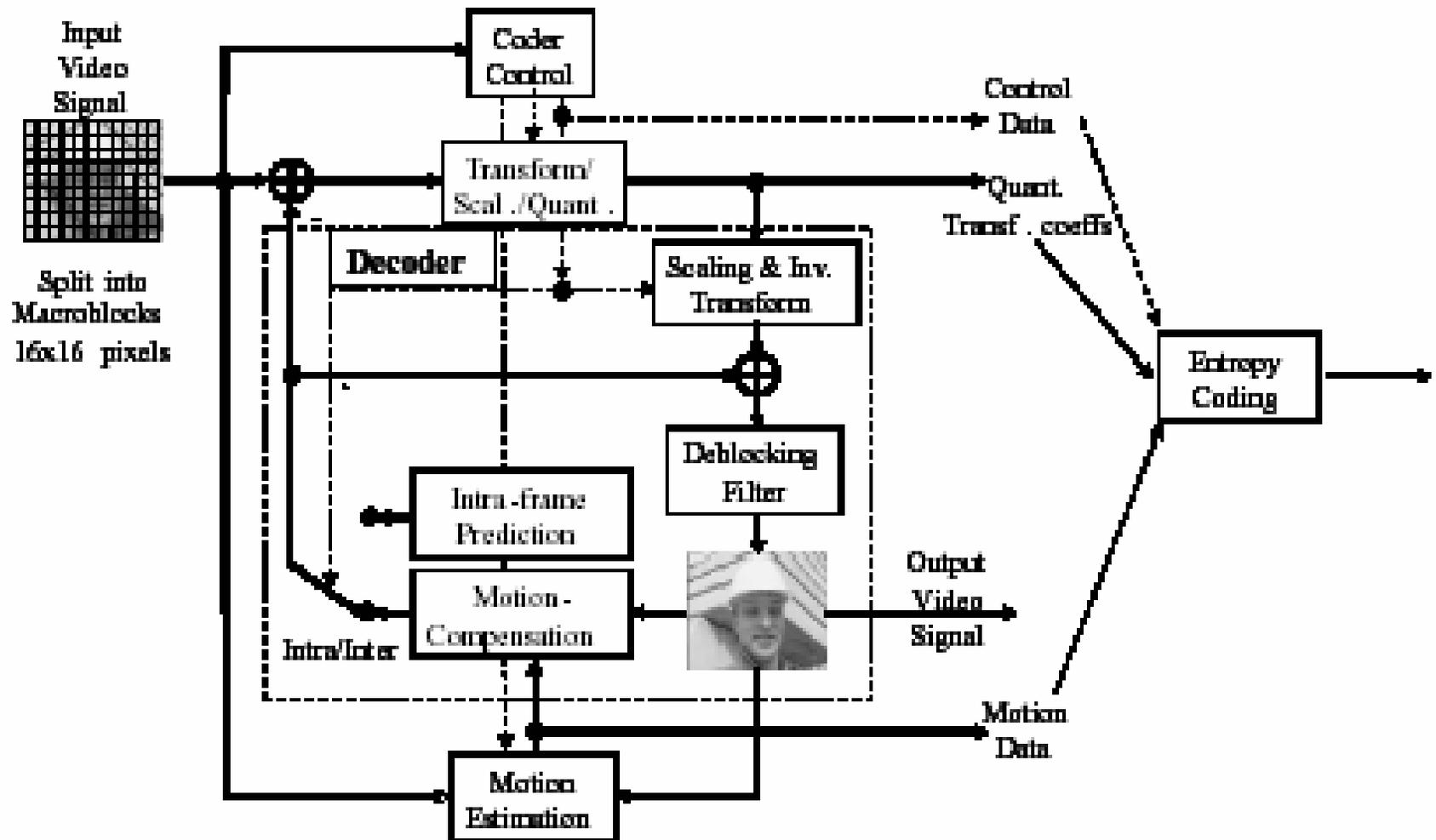
# H.264/AVC の応用

- Broadcast over cable, satellite, cable modem, DSL, terrestrial
- Interactive or serial storage on optical and magnetic devices (e.g. DVD)
- Conversational services over ISDN, Ethernet, LAN, DSL, wireless and mobile networks
- Video-on-demand (VoD) or multimedia streaming services over ISDN, cable modem, DSL, LAN, wireless and mobile networks
- Multimedia messaging services (MMS) over ISDN, DSL, Ethernet, LAN, wireless and mobile networks

# H.264/AVC の基本構成 (1)



# H.264/AVC の基本構成 (2)



# Improved Prediction

- ❑ Variable block-size motion compensation (4x4 to 16x16)
- ❑ Quarter-sample-accuracy motion compensation (1/4pel)
- ❑ Picture boundary extrapolation for motion compensation
- ❑ Multiple reference picture motion compensation
- ❑ Flexibility in picture display ordering
- ❑ Flexibility in picture referencing
- ❑ Prediction with weights and offsets (for fading)
- ❑ Motion inference in skipped area
- ❑ Directional spatial prediction for intra coding
- ❑ In-the-loop de-blocking filtering

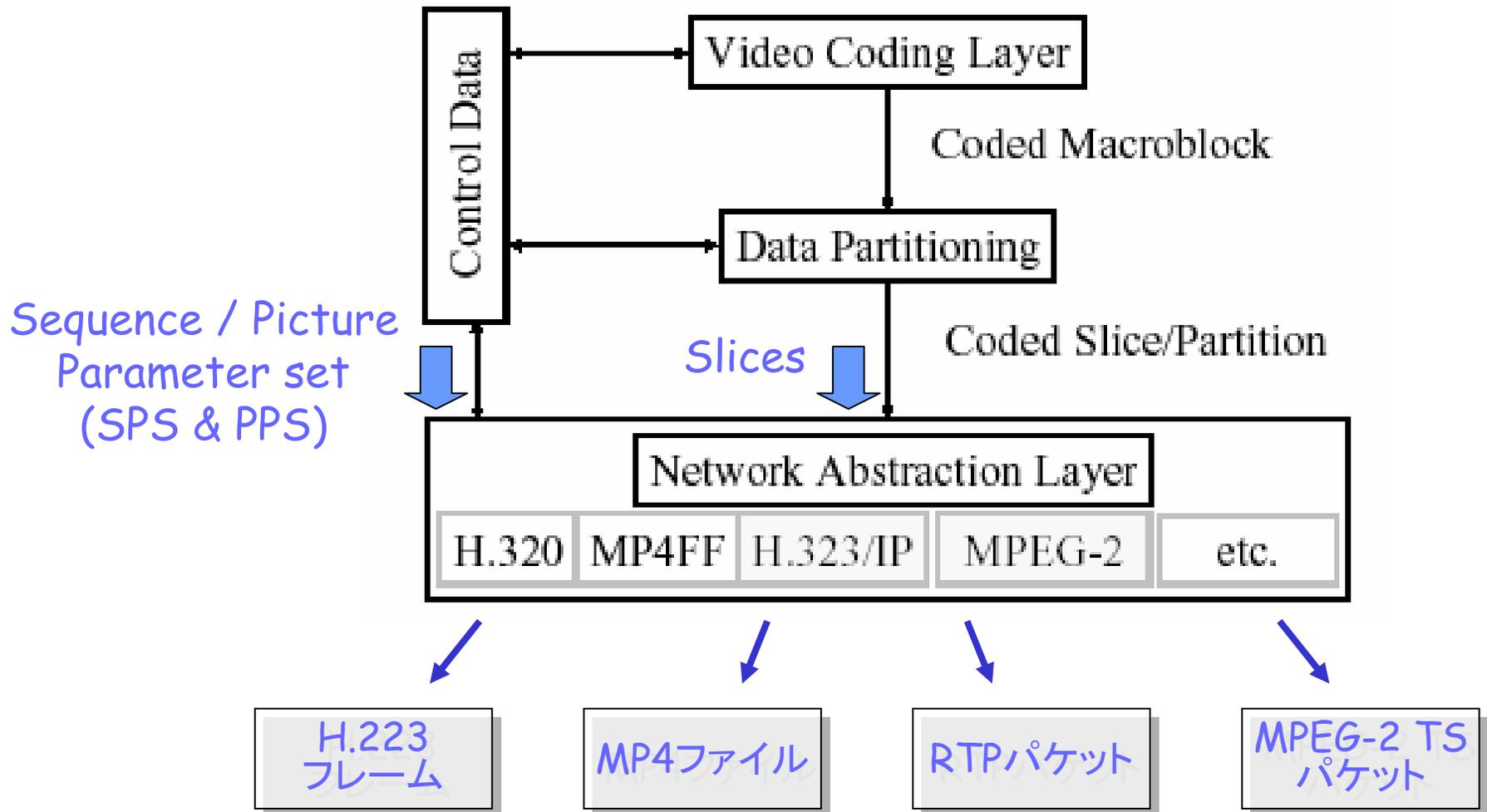
# Improved Coding Efficiency

- Small block-size transform (4x4)
- Hierarchical block transform (4x4, 8x8, 16x16)
- Short word-length transform (16-bit arithmetic)
- Exact-match inverse transform (no calculation error)
- Arithmetic entropy coding (CABAC)
- Context-adaptive entropy coding (CAVLC & CABAC)

# Robustness & Flexibility

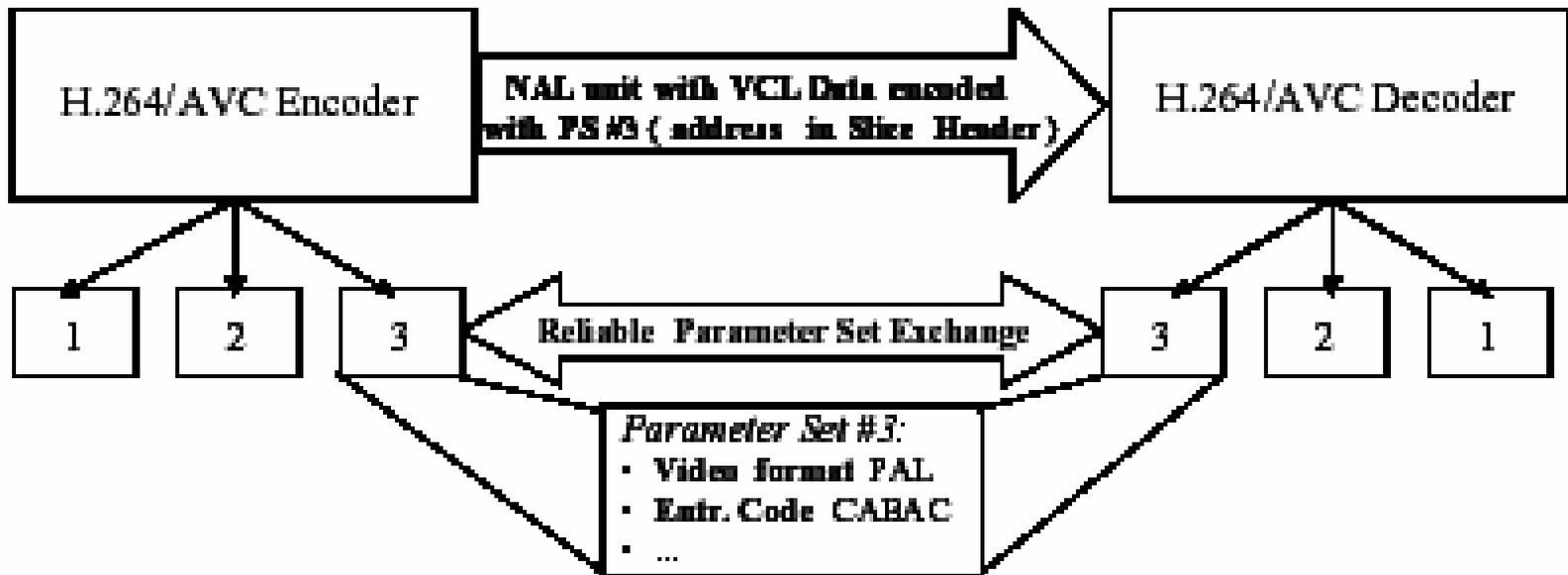
- Flexible “parameter set” structure
- NAL unit syntax structure (network abstraction)
- Flexible slice size (like MPEG-1)
- Flexible macro-block ordering by slice groups
- Arbitrary slice ordering
- Redundant picture representation (after data loss)
- Data partitioning (for error resiliency)
- SP/SI picture types (for decoder synchronization and switching)

# 1. NAL: Network Abstraction Layer



# 1. NAL: Network Abstraction

In-band signaling (e.g. by RTP)



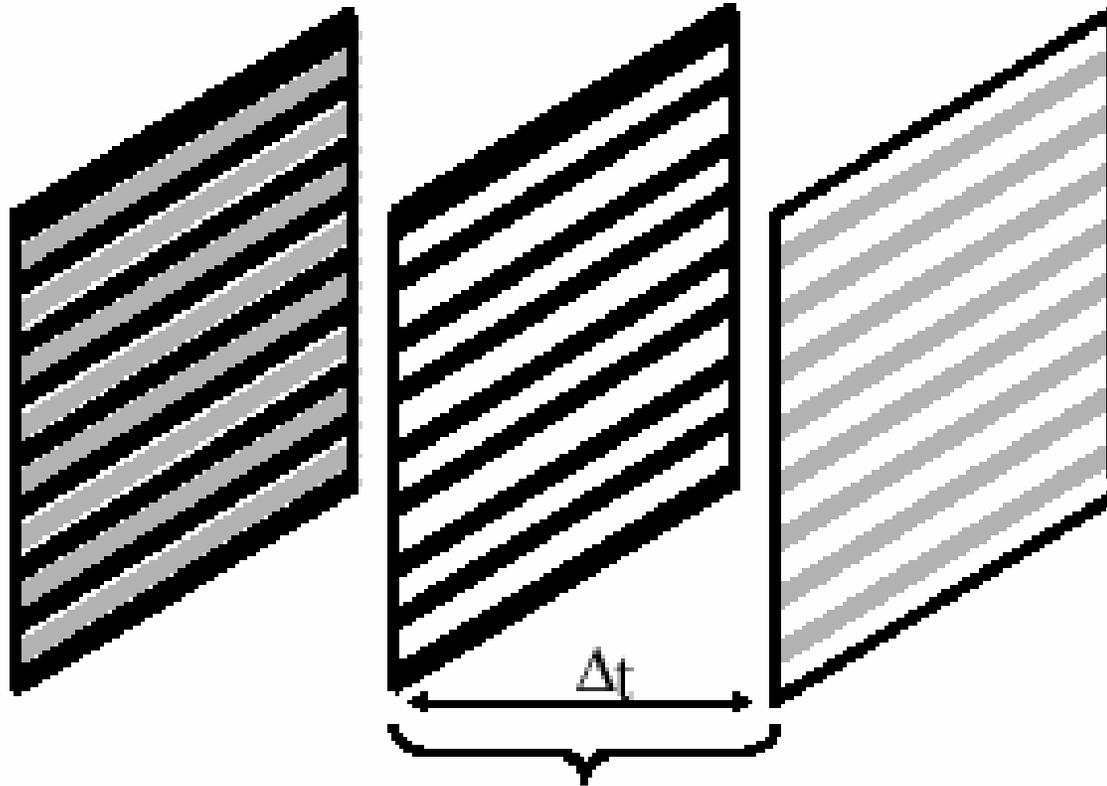
Out-band signaling (e.g. by TCP)

## 2. Frame & Field

Progressive  
Frame

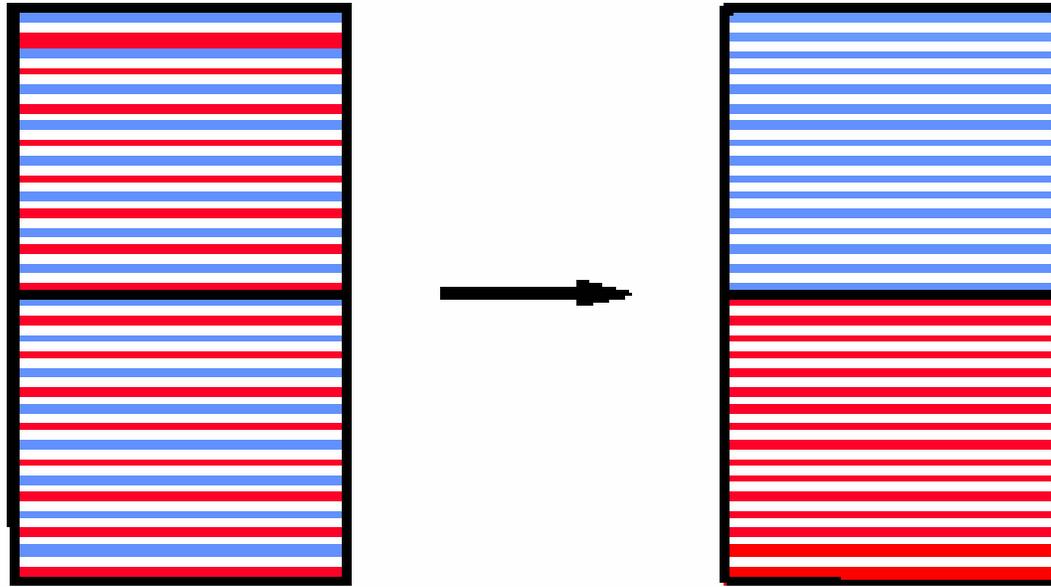
Top  
Field

Bottom  
Field



Interlaced Frame (Top Field First)

## 2. Frame & Field



A Pair of Macroblocks  
in Frame Mode

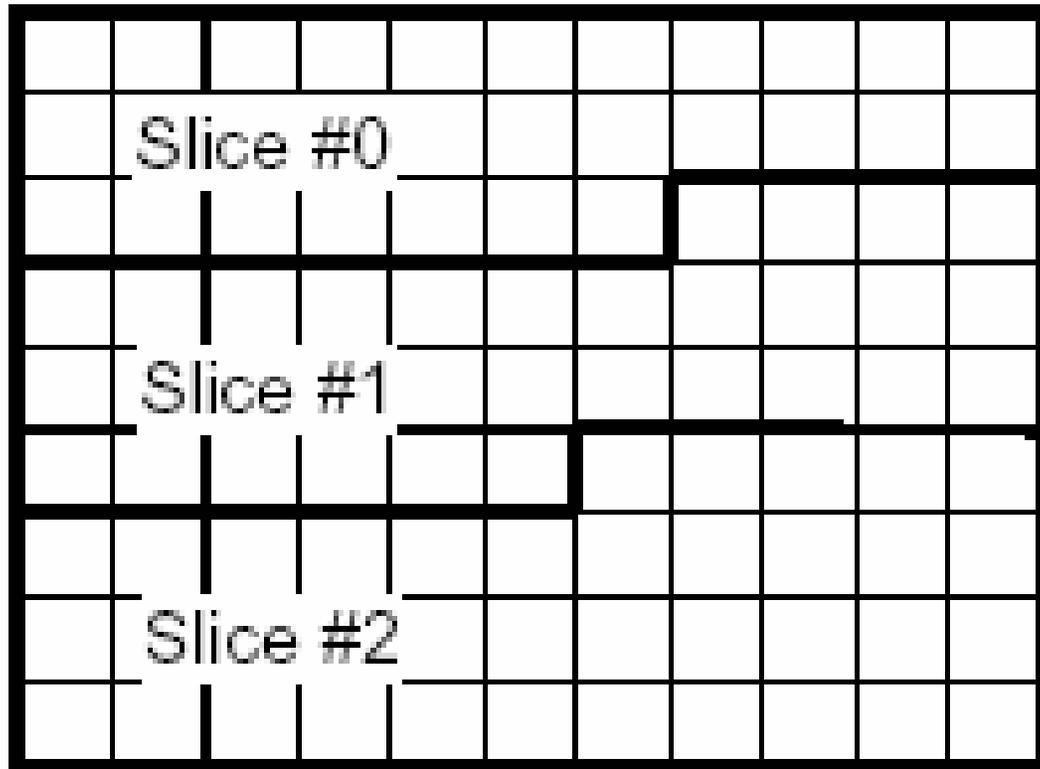
Top/Bottom Macroblocks  
in Field Mode

PAFF: picture-adaptive frame/field coding (odd/even fields)

MBAFF: macroblock-adaptive frame/field coding (top/bottom MBs)

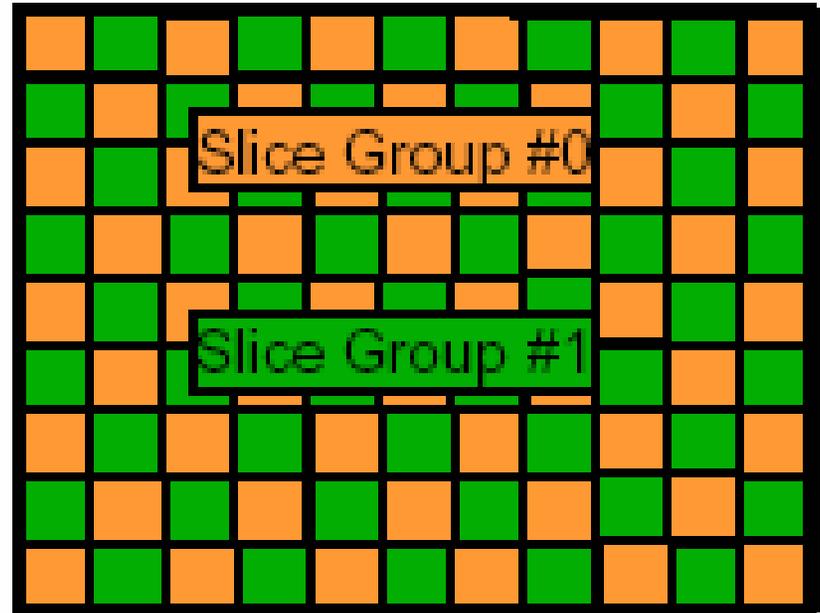
# 3. Slice

H.264/AVC の符号化の基本単位:



Not using FMO (flexible macroblock ordering)

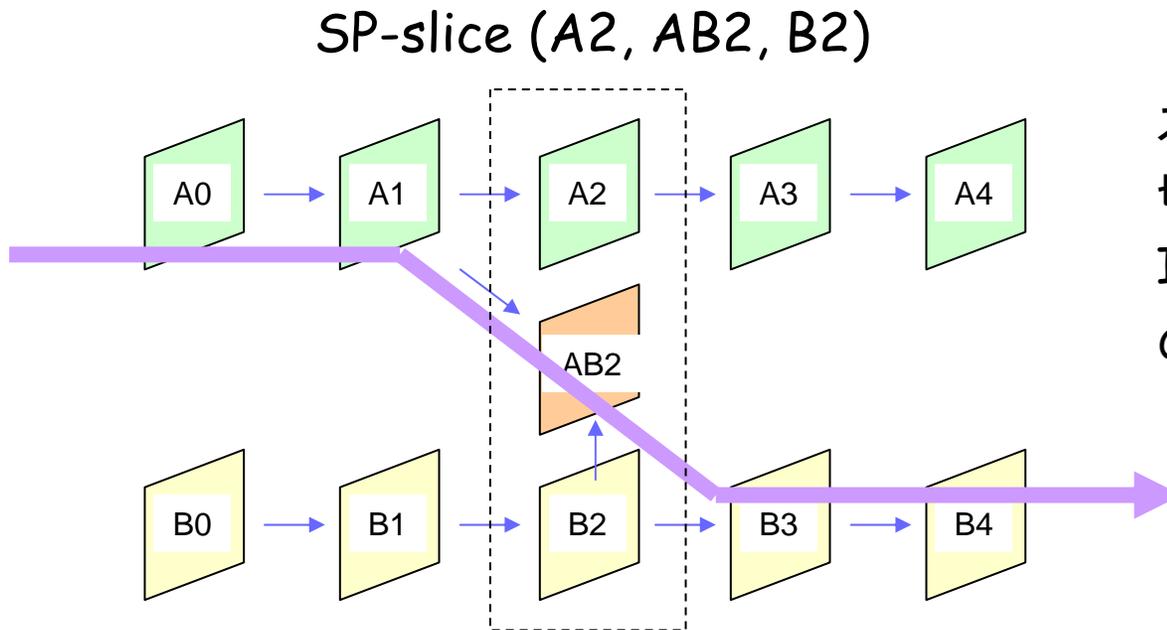
# 3. Slice



Using FMO (flexible macroblock ordering)  
= Slice groups ~ ROI (Region of Interest)

# 3. Slice

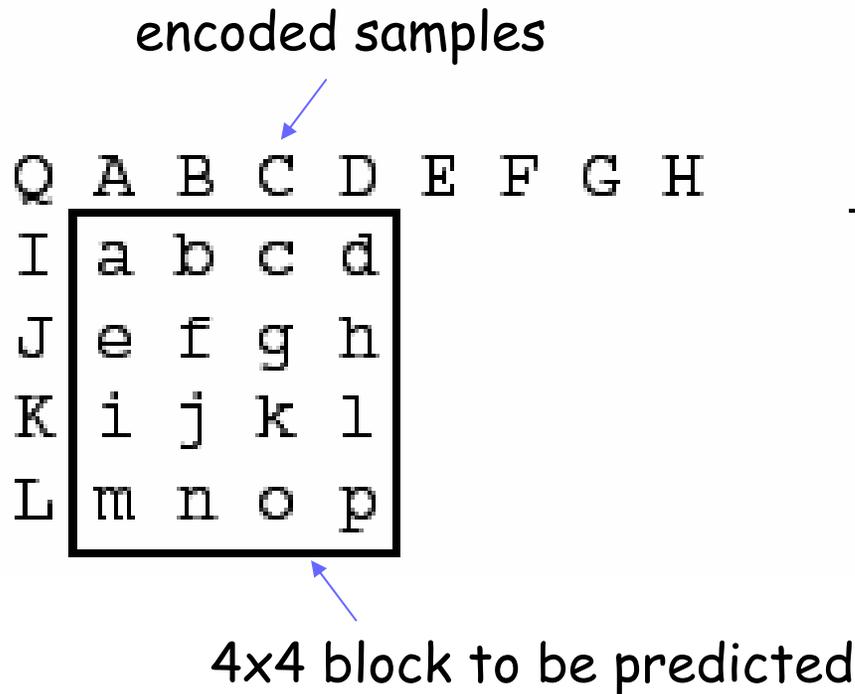
- I-slice: I macroblock only
- P-slice: I,P macroblocks
- B-slice: I,P,B macroblocks
- SP/SI-slice: (for bit-stream switching)



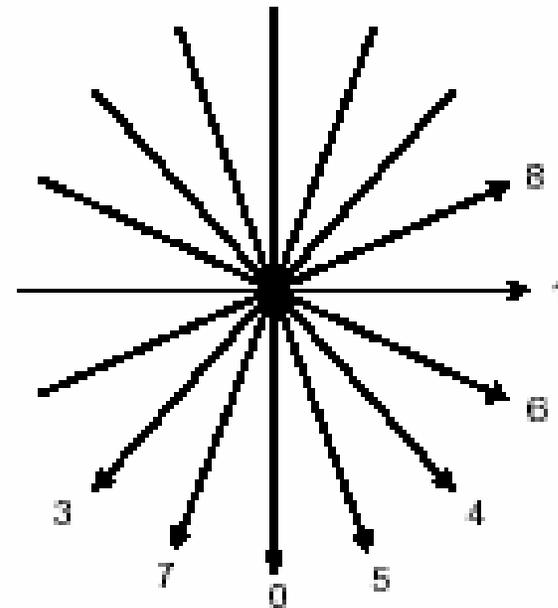
ストリーム A から B への  
切り替え、早送り等:  
I-slice よりも SP-slice  
の方が効率がよい

# 4. Intra-Frame Prediction

## □ Intra\_4x4

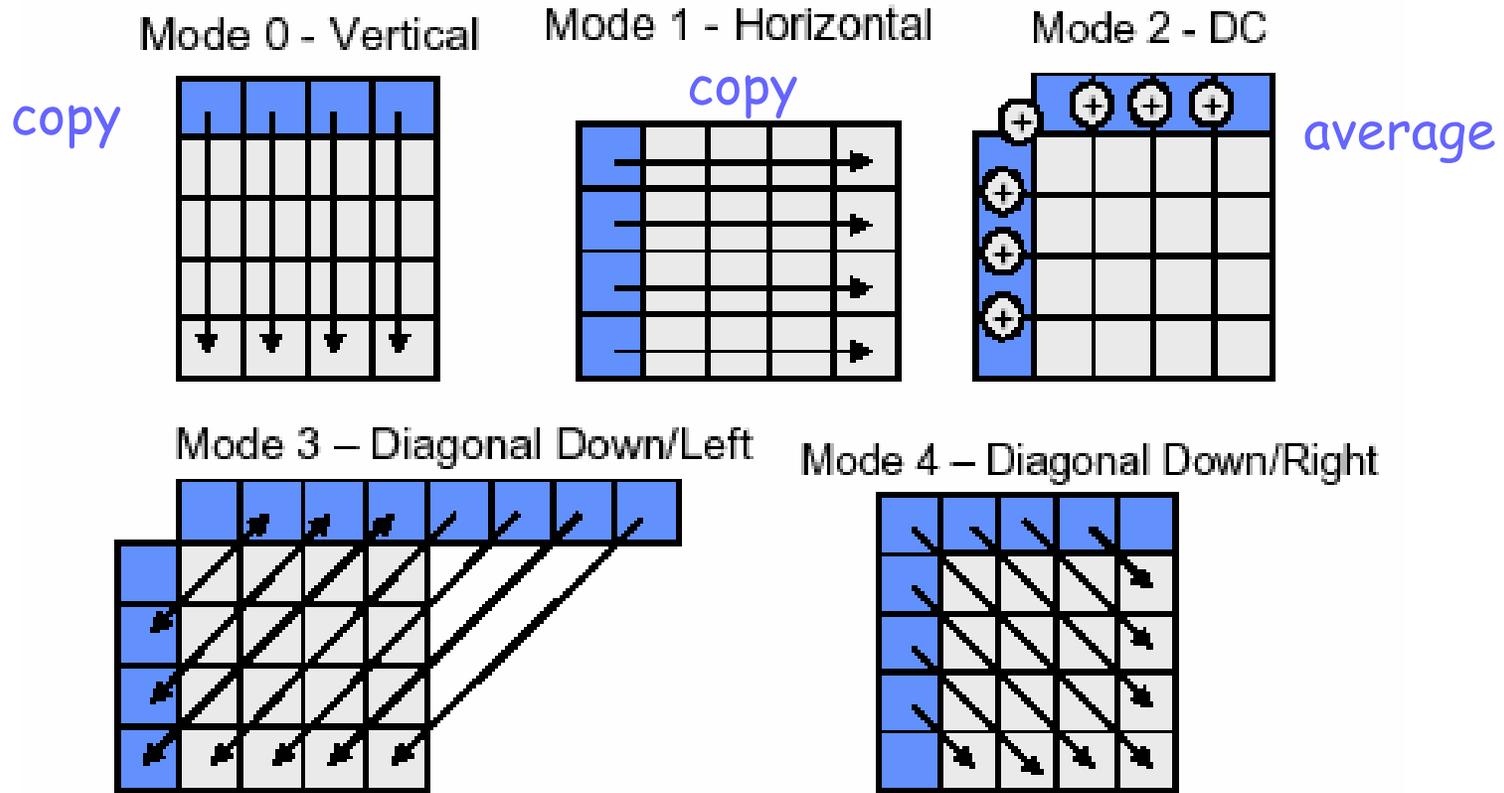


Prediction directions



# 4. Intra-Frame Prediction

## □ Intra\_4x4: nine modes



Modes 3 ~ 8: diagonal-down-left, diagonal-down-right, vertical-right, horizontal-down, vertical-left, and horizontal-up

# 4. Intra-Frame Prediction

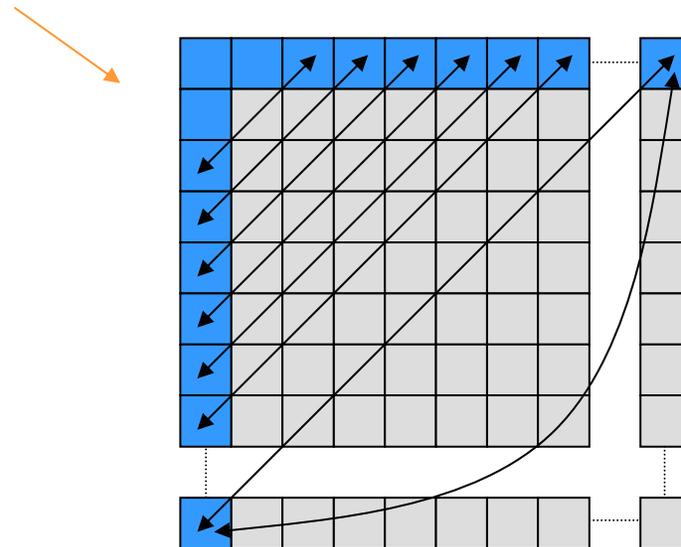
## □ Intra\_16x16: four modes

Mode 0: vertical prediction

Mode 1: horizontal prediction

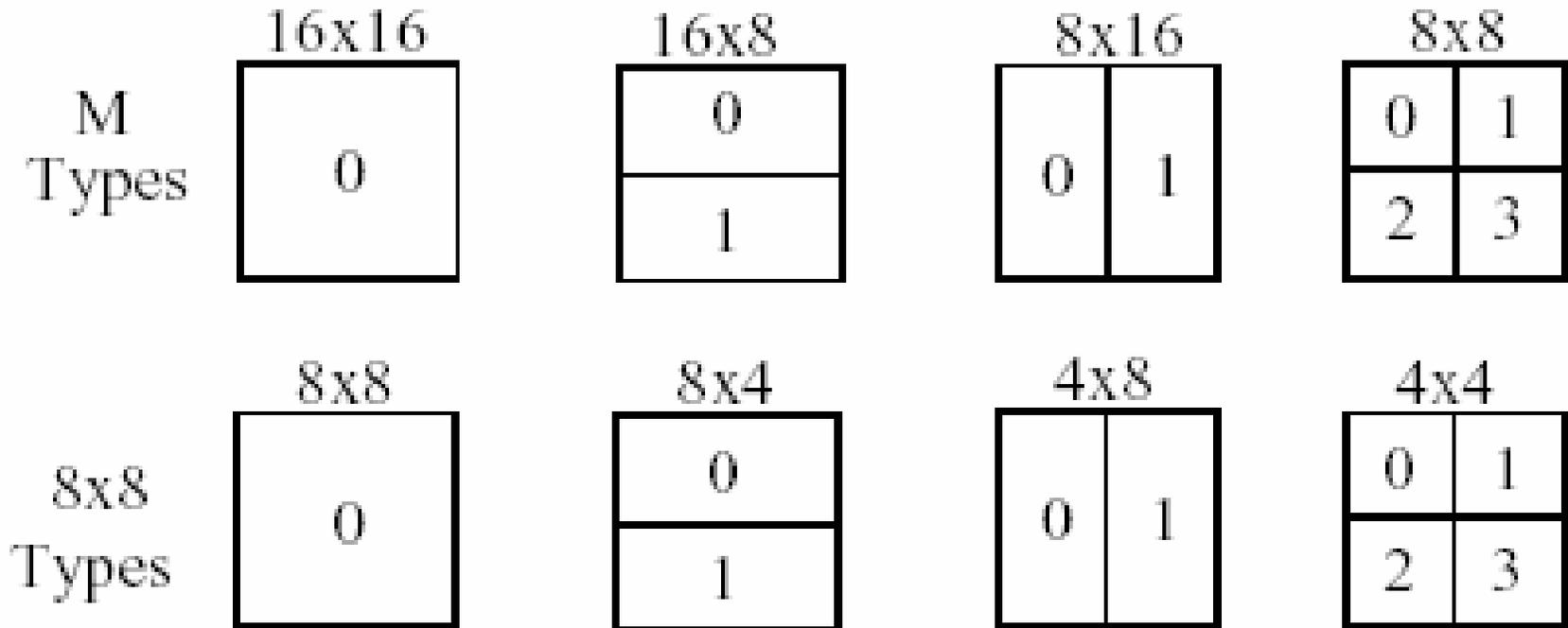
Mode 2: DC prediction

Mode 3: plane prediction



# 5. Inter-Frame Prediction

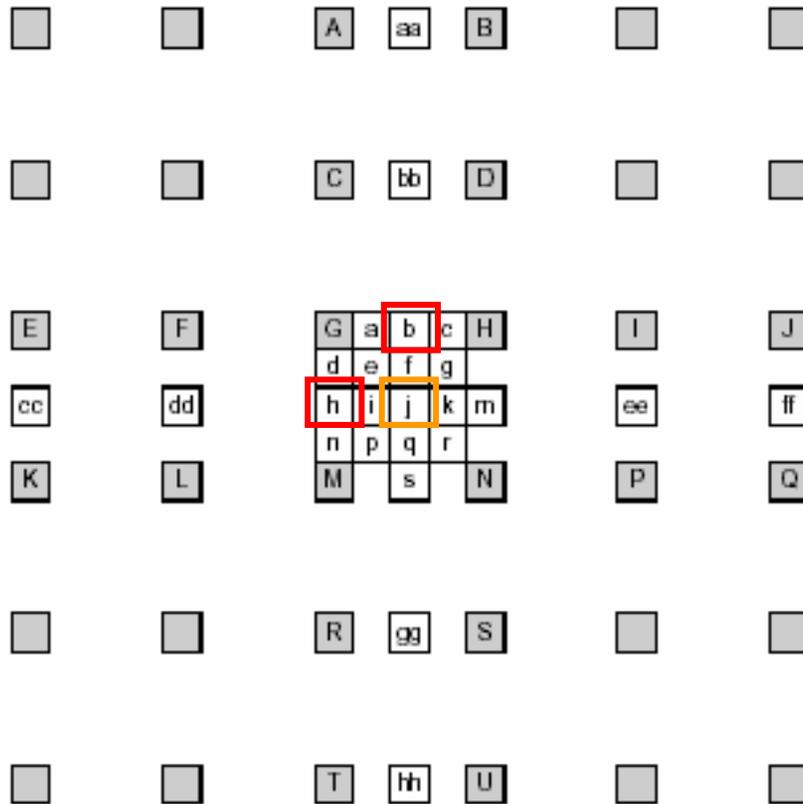
- Macroblock partitioning (for variable block-size MC)



Maximally 16 motion vectors (4x4 case)

# 5. Inter-Frame Prediction

## □ Quarter-pel accuracy motion compensation



1/2-pel accuracy: 6-tap FIR

$$b_1 = (E - 5F + 20G + 20H - 5I + J)$$

$$b = (b_1 + 16) \gg 5$$

$$h_1 = (A - 5C + 20G + 20M - 5R + T)$$

$$h = (h_1 + 16) \gg 5$$

$$j_1 = (cc - 5dd + 20h_1 + 20m_1 - 5ee + ff)$$

$$j = (j_1 + 512) \gg 10$$

1/4-pel accuracy: 2-tap FIR

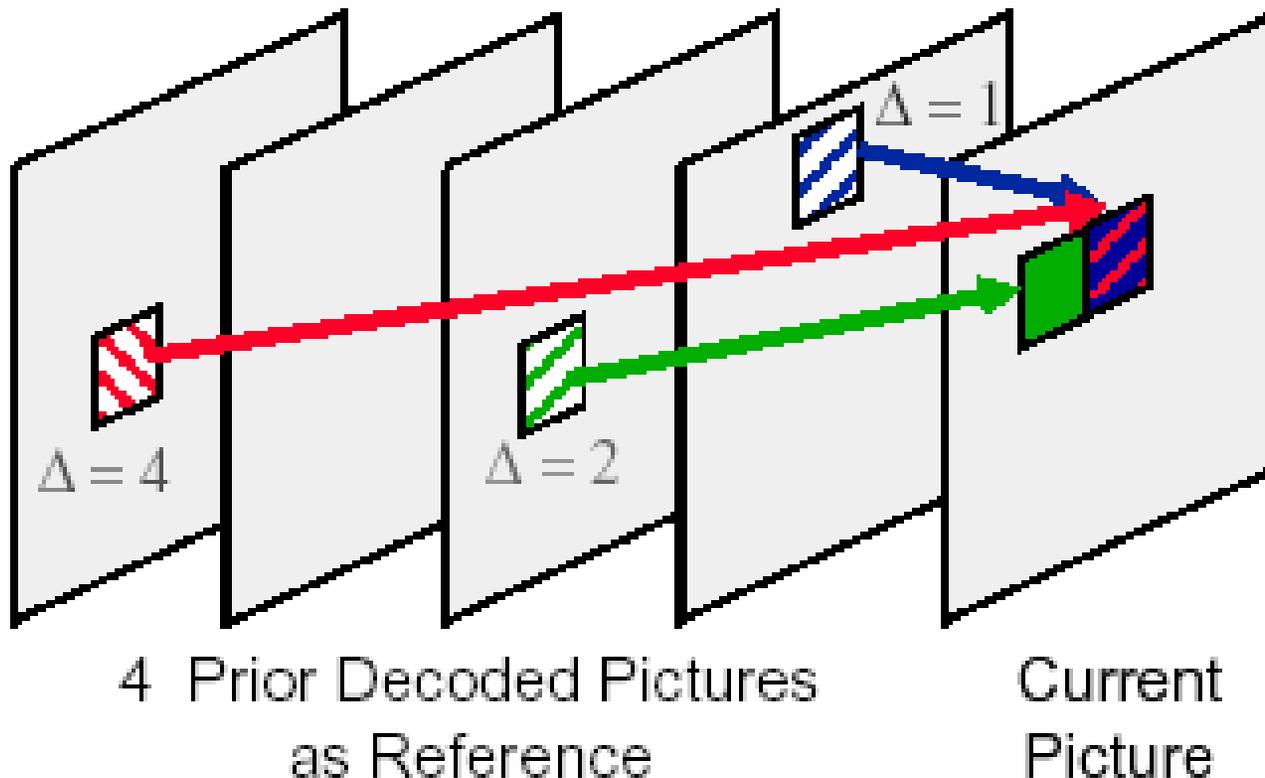
$$a = (G + b + 1) \gg 1$$

$$e = (b + h + 1) \gg 1$$

pixel

# 5. Inter-Frame Prediction

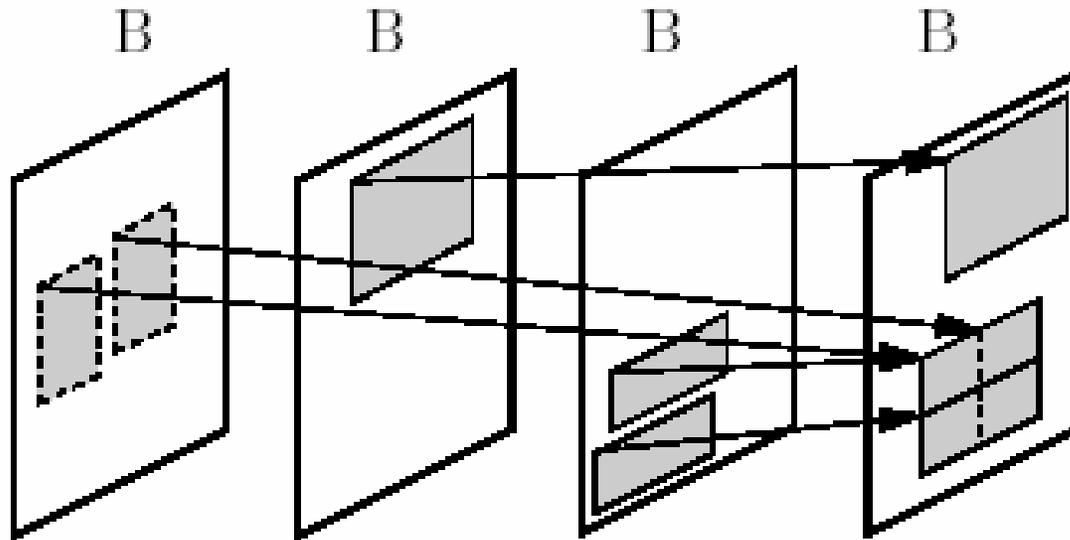
- Multi-frame (multi-reference) motion compensation



Applied to P-slices and B-slices

# 5. Inter-Frame Prediction

## □ B-slices: generalized B-pictures



- List 0 prediction
  - List 1 prediction
  - "Bi-predictive" prediction: using two reference pictures
  - Direct prediction: MV prediction and no overhead transmission
- + no coefficients

- P-Skip mode
- B-Skip mode

# 6. Transform

## □ 4x4 integer transform

$$H_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

16-bit integer arithmetic

$$G_{4 \times 4} = 5.38 \text{ dB for } \rho = 0.90$$

---

$$H_{DCT(4 \times 4)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a & b & -b & -a \\ 1 & -1 & -1 & 1 \\ b & -a & a & -b \end{bmatrix}$$

$$a = \sqrt{2} \cos \frac{\pi}{8} \approx 1.306, b = \sqrt{2} \cos \frac{3\pi}{8} \approx 0.541 \quad G_{DCT(4 \times 4)} = 5.39 \text{ dB for } \rho = 0.90$$

# 7. Entropy Coding

- **exp-Golomb code: for syntax elements except quantized transform coefficients**

code	syntax elements
1	0
0 1 0	1
0 1 1	2
00 1 00	3
00 1 01	4
00 1 10	5
00 1 11	6
000 1 000	7
000 1 001	8
⋮	⋮

Exp-Golomb code:

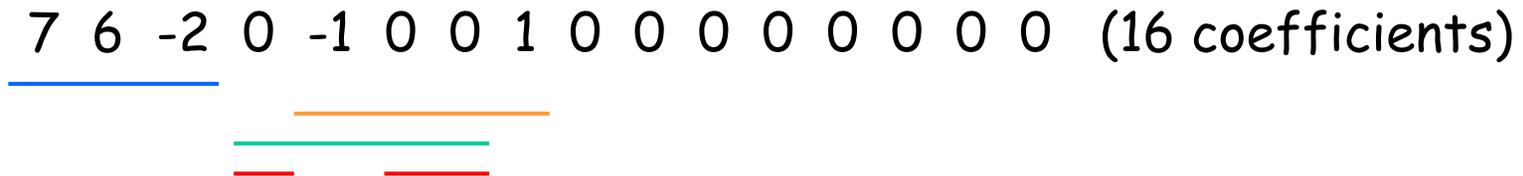
$$\underbrace{00 \dots 0}_{\text{prefix}} \quad \underbrace{0}_{\text{separator}} \quad \underbrace{xx \dots x}_{\text{suffix}}$$

0,1,2,...

# 7. Entropy Coding

□ **CAVLC: context-adaptive variable length coding (for quantized transform coefficients)**

7 6 -2 0 -1 0 0 1 0 0 0 0 0 0 0 0 (16 coefficients)



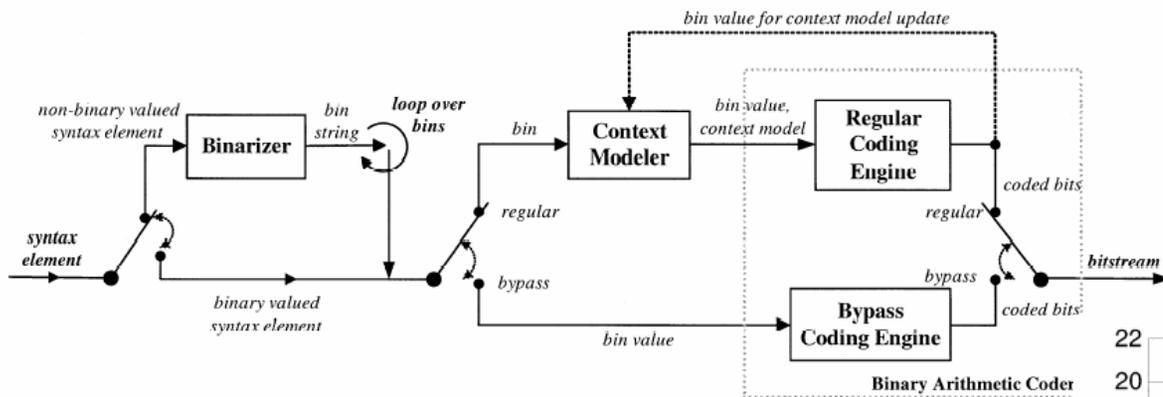
- N = 5: number of non-zero quantized coefficients
- T1s = 2: number of '±1' at the end of scan (trailing 1s)
- sign flag = '-' '+' : sign of T1s
- coefficient values in reverse order → '-2' '6' '7'
- total zeros = 3: number of zeros before the last non-zero
- run before = 2, 1: how total zeros are distributed in reverse order



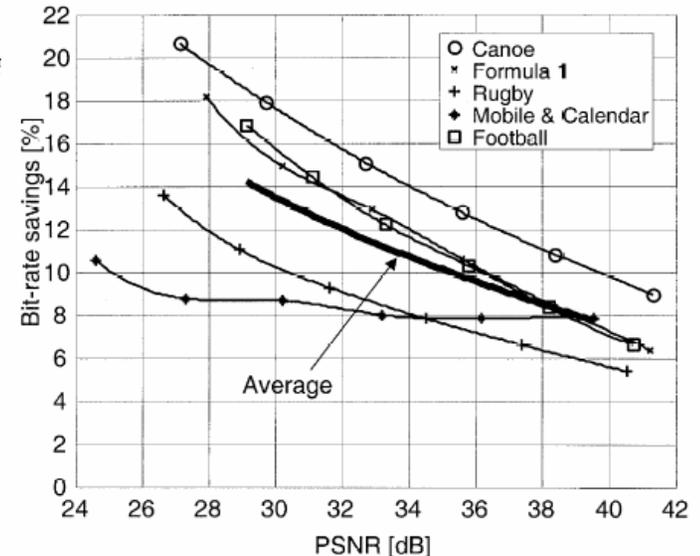
Context-adaptive: 過去の符号化結果に応じて適応的に VLC table を切替える

# 7. Entropy Coding

□ CABAC: context-adaptive binary arithmetic coding (for quantized transform coefficients)



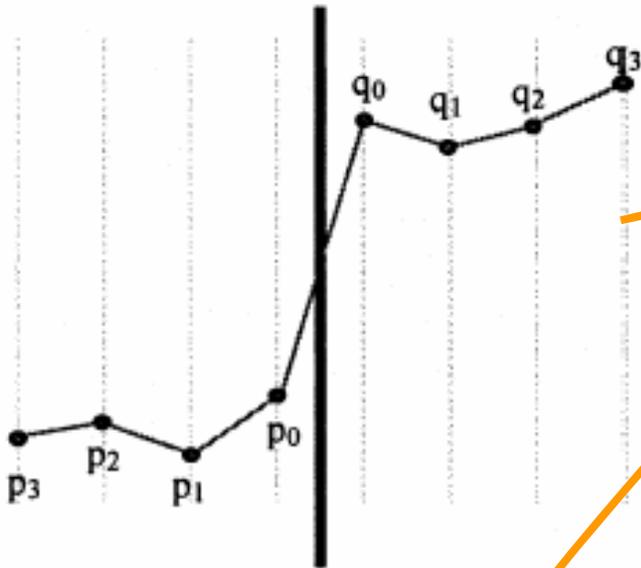
vs CAVLCの改善効果



- Binarization: 多値・二値変換
- Context modeling: 確率モデル更新・適応
- Binary arithmetic coding

# 8. In-Loop Deblocking Filter

block boundary



「ブロックひずみ」と「エッジ」をどのように  
区別するか？

$$\begin{cases} |p_0 - q_0| < \alpha(QP) \\ |p_1 - p_0| < \beta(QP) \\ |q_1 - q_0| < \beta(QP) \end{cases}$$

ならばフィルタリングを実行する。

QP: Quantization Parameter (0~51)

e.g.  $\alpha(QP) = 0.8 \cdot (2^{QP/6} - 1)$   
 $\beta(QP) = 0.5 \cdot QP - 7$

$$p'_0 = (p_2 + 2p_1 + 2p_0 + 2q_0 + q_1) / 8$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0) / 4$$

$$p'_2 = (2p_3 + 3p_2 + p_1 + p_0 + q_0) / 8$$



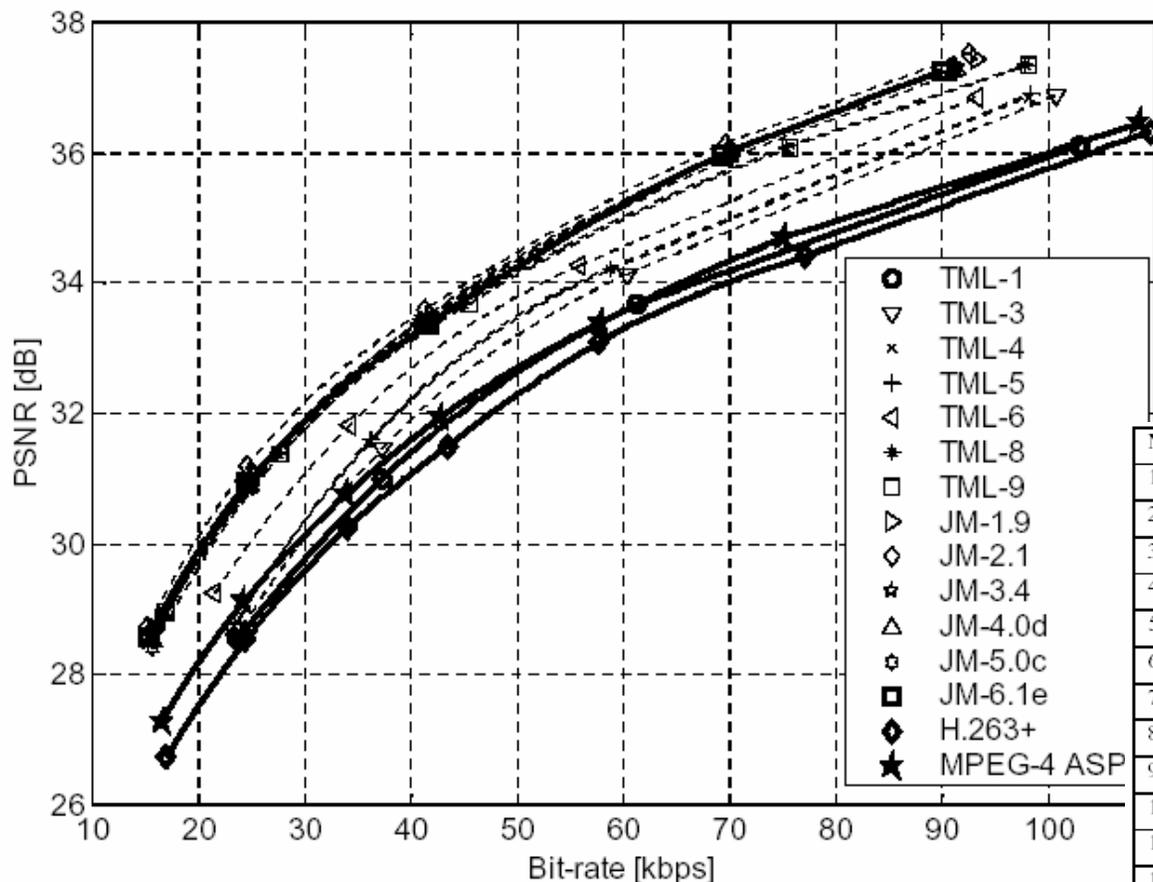
conventional



H.264/AVC

# H.264/AVC Evolution

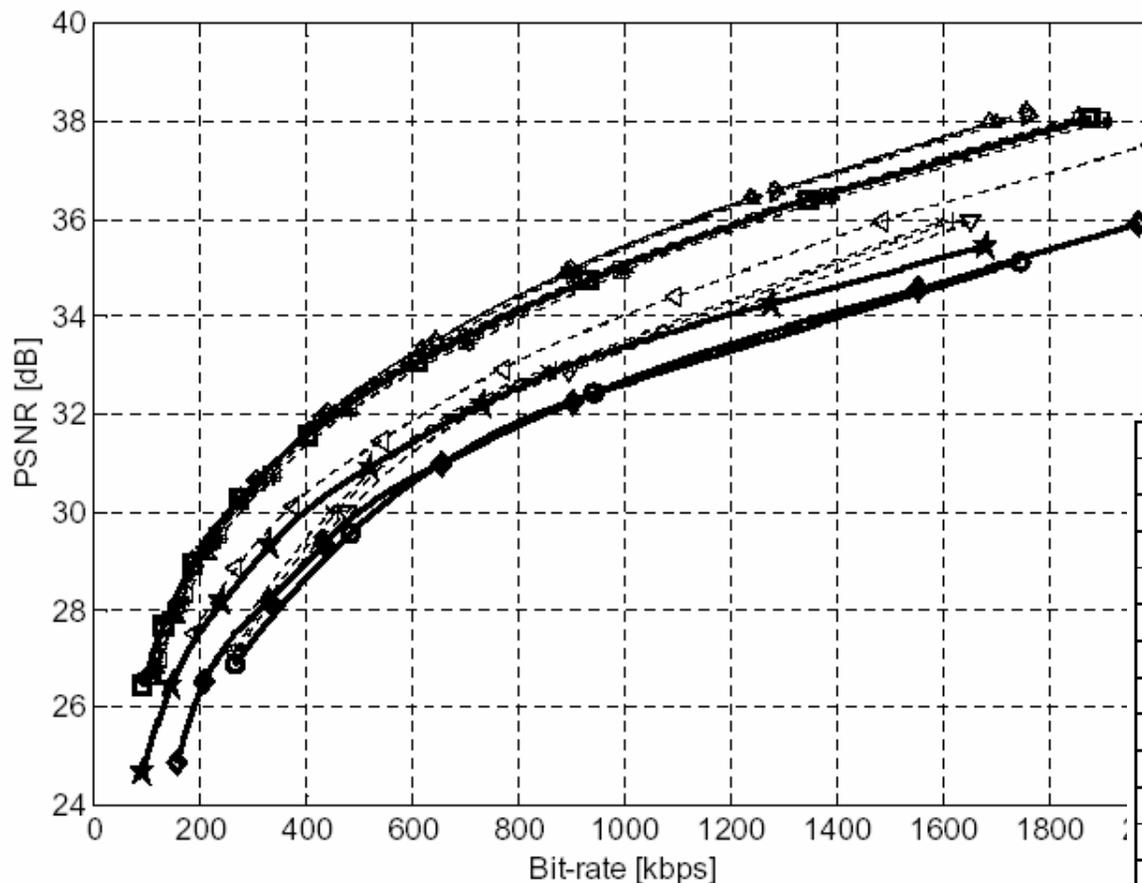
Foreman



No	TML/JM	Date	Location
1	TML-1	Aug. 1999	Berlin, Germany
2	TML-2*	Oct. 1999	Red Bank, NJ, USA
3	TML-3	Feb. 2000	Geneva, Switzerland
4	TML-4	May, 2000	Osaka, Japan
5	TML-5	Aug. 2000	Portland, OR, USA
6	TML-6	Jan. 2001	Eibsee, Germany
7	TML-7*	Apr. 2001	Austin, TX, USA
8	TML-8	May 2001	Porto Seguro, Brazil
9	TML-9	Sep. 2001	Santa Barbara, CA, USA
10	JM-1	Dec. 2001	Pattaya, Thailand
11	JM-2	Feb. 2002	Geneva, Switzerland
12	JM-3	May 2002	Fairfax, VA, USA
13	JM-4	July 2002	Klagenfurt, Austria
14	JM-5	Oct. 2002	Geneva, Switzerland
15	JM-6	Dec. 2002	Awaji, Japan
16	Final	Mar. 2003	Pattaya, Thailand

# H.264/AVC Evolution

Tempete



No	TML/JM	Date	Location
1	TML-1	Aug. 1999	Berlin, Germany
2	TML-2*	Oct. 1999	Red Bank, NJ, USA
3	TML-3	Feb. 2000	Geneva, Switzerland
4	TML-4	May, 2000	Osaka, Japan
5	TML-5	Aug. 2000	Portland, OR, USA
6	TML-6	Jan. 2001	Eibsee, Germany
7	TML-7*	Apr. 2001	Austin, TX, USA
8	TML-8	May 2001	Porto Seguro, Brazil
9	TML-9	Sep. 2001	Santa Barbara, CA, USA
10	JM-1	Dec. 2001	Pattaya, Thailand
11	JM-2	Feb. 2002	Geneva, Switzerland
12	JM-3	May 2002	Fairfax, VA, USA
13	JM-4	July 2002	Klagenfurt, Austria
14	JM-5	Oct. 2002	Geneva, Switzerland
15	JM-6	Dec. 2002	Awaji, Japan
16	Final	Mar. 2003	Pattaya, Thailand

# 圧縮効率の改善効果 (非公式)

手法	効果
CABAC	10~15%
可変ブロック動き補償	~5%
Integer 変換	~5%
複数参照ピクチャ	~5%
R-D最適化 (次回)	10~15%
総計	30%以上