

# 画像情報特論 (4)

## - デジタル圧縮とメディア表現

### (1) ビデオ圧縮

2003.05.16

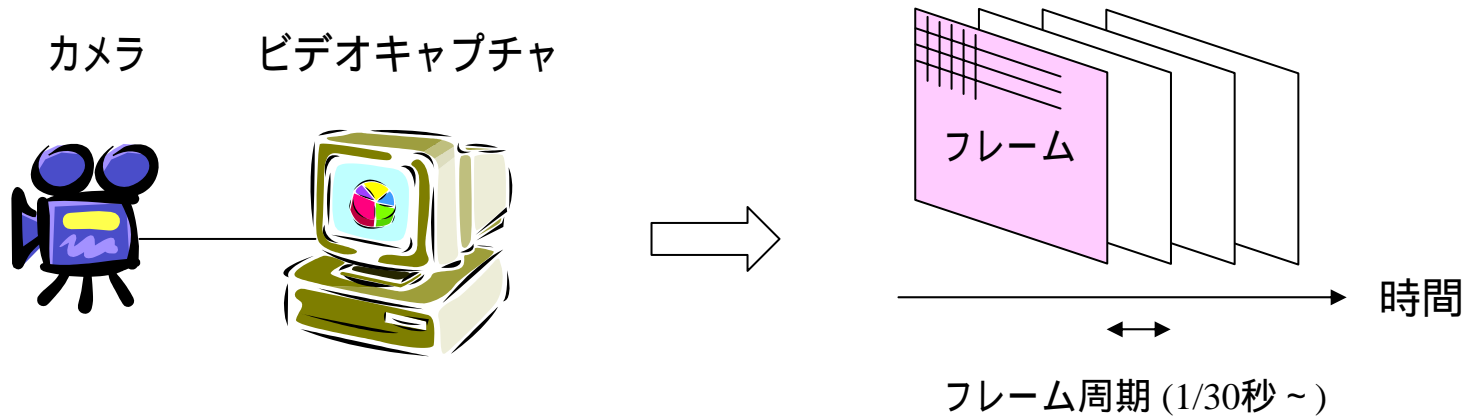
情報ネットワーク専攻 甲藤二郎

E-Mail: katto@waseda.jp

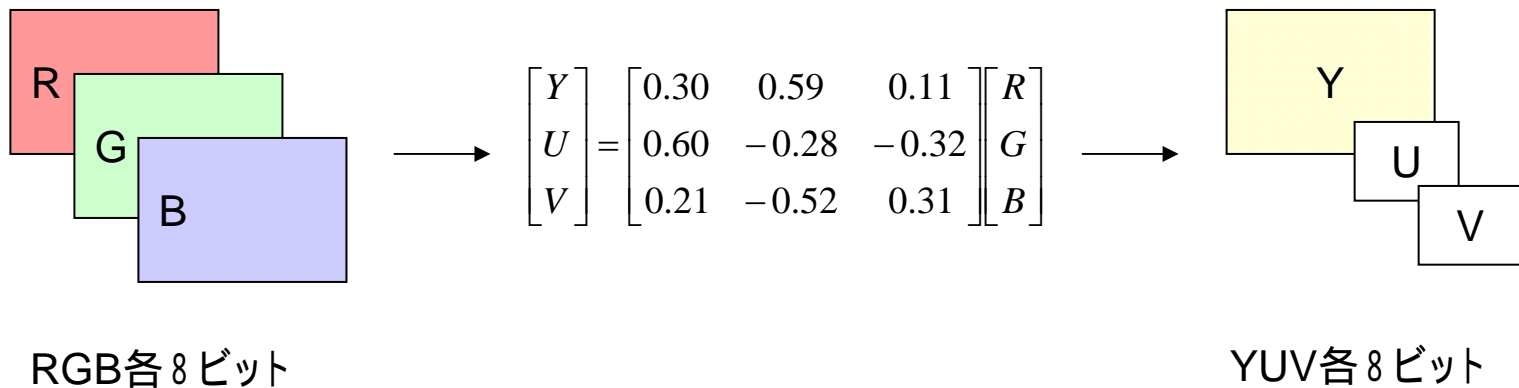
# ビデオ圧縮の原理

# デジタル動画 (1)

- 時間方向・空間方向のサンプリング

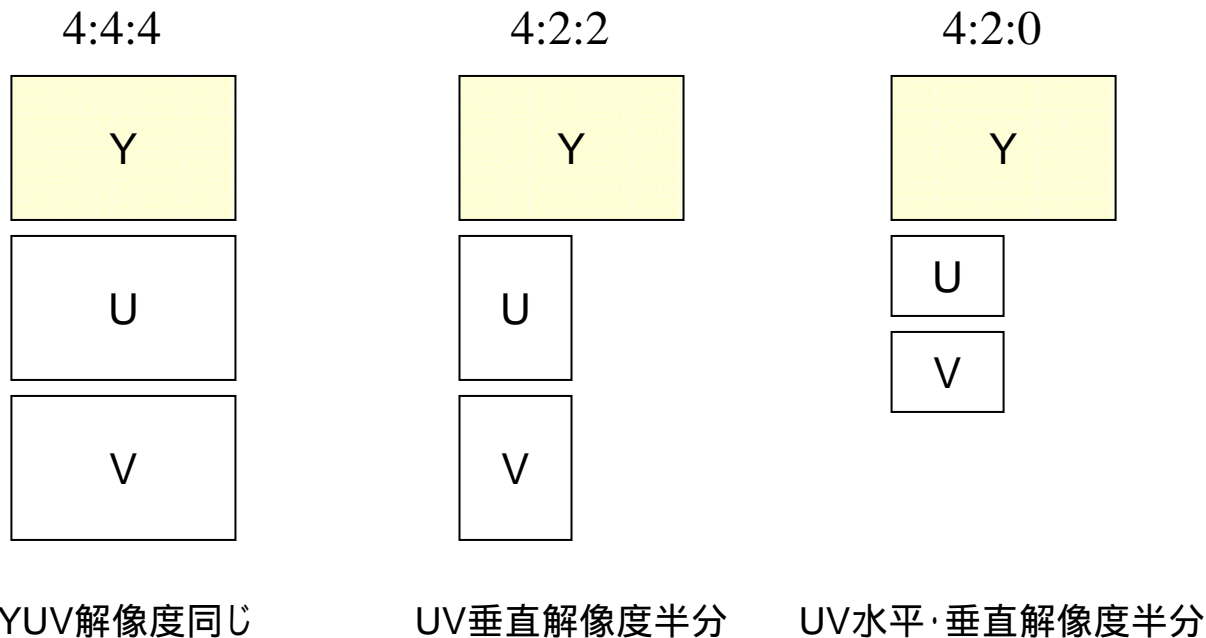


- RGB / YUV 変換



# デジタル動画 (2)

- CCIR 601 フォーマット

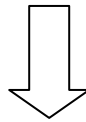


- 通常のビデオ圧縮： 4:2:0 フォーマット
- 高画質ビデオ圧縮： 4:2:2 フォーマット

# デジタル動画 (3)

- 莫大な情報量

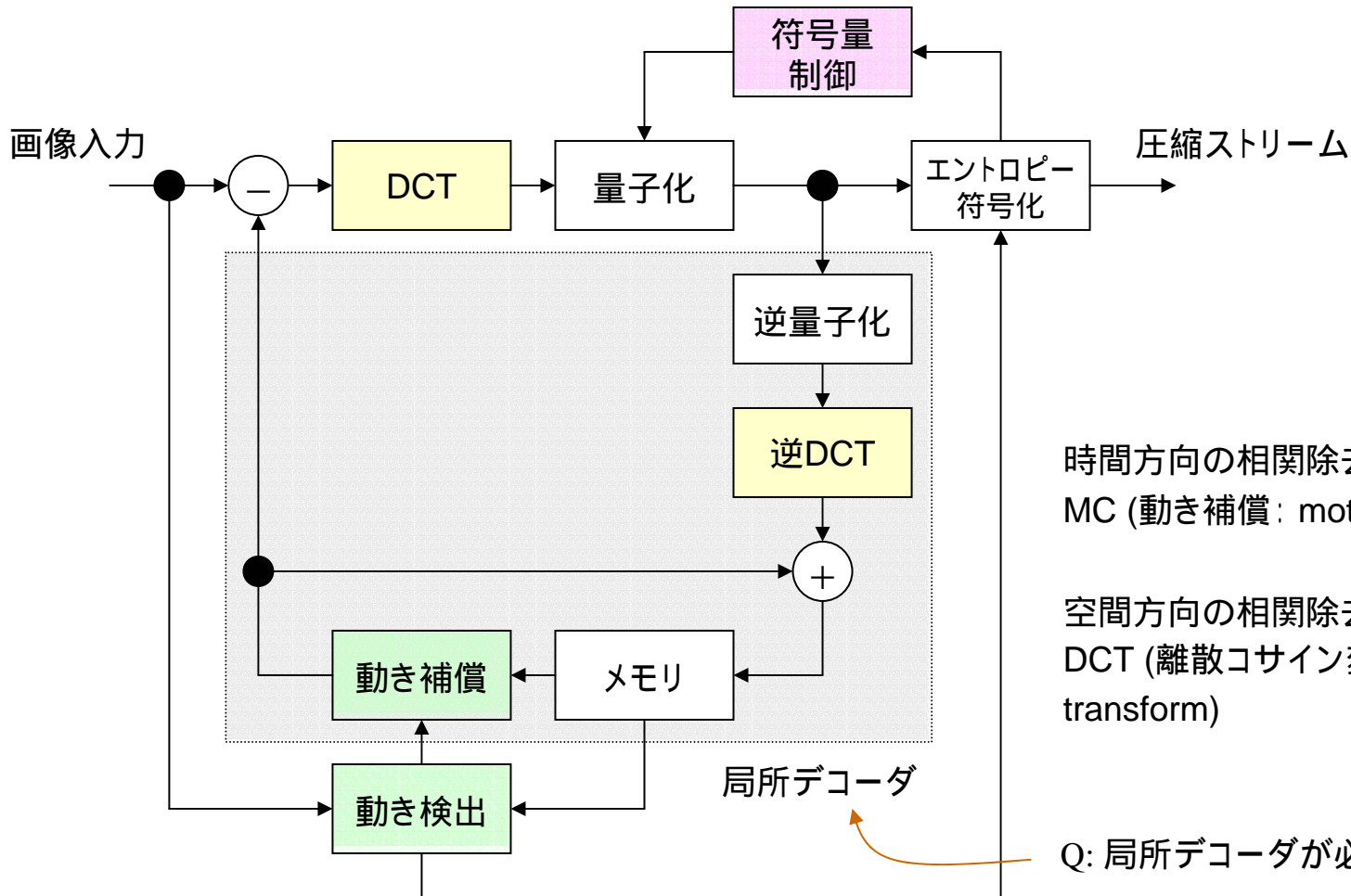
用途	解像度	データ量
TV会議	352x240	21Mbit/s
TV	720x480	83Mbit/s
HDTV	1920x1080	498Mbit/s



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 (20年間変わらない方式)



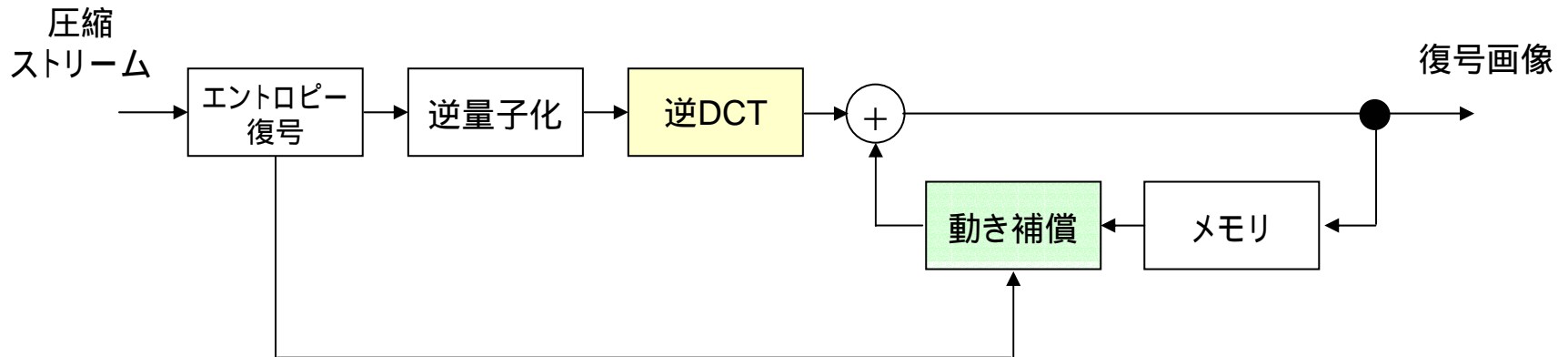
時間方向の相関除去:  
MC (動き補償: motion compensation)

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

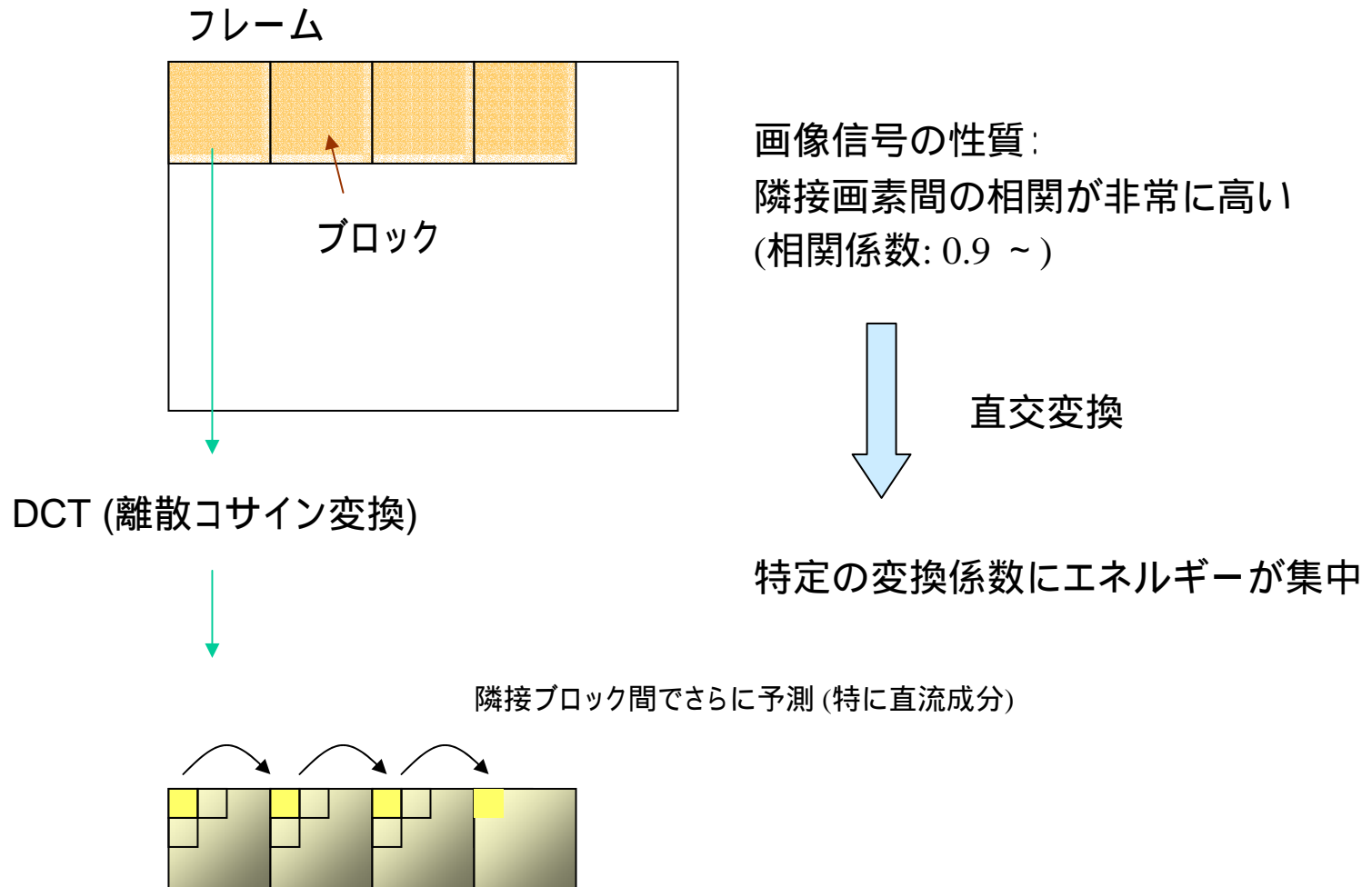
# ビデオ復号の仕組み

- エンコーダのローカルデコードと同じ



# フレーム内符号化

- DCT

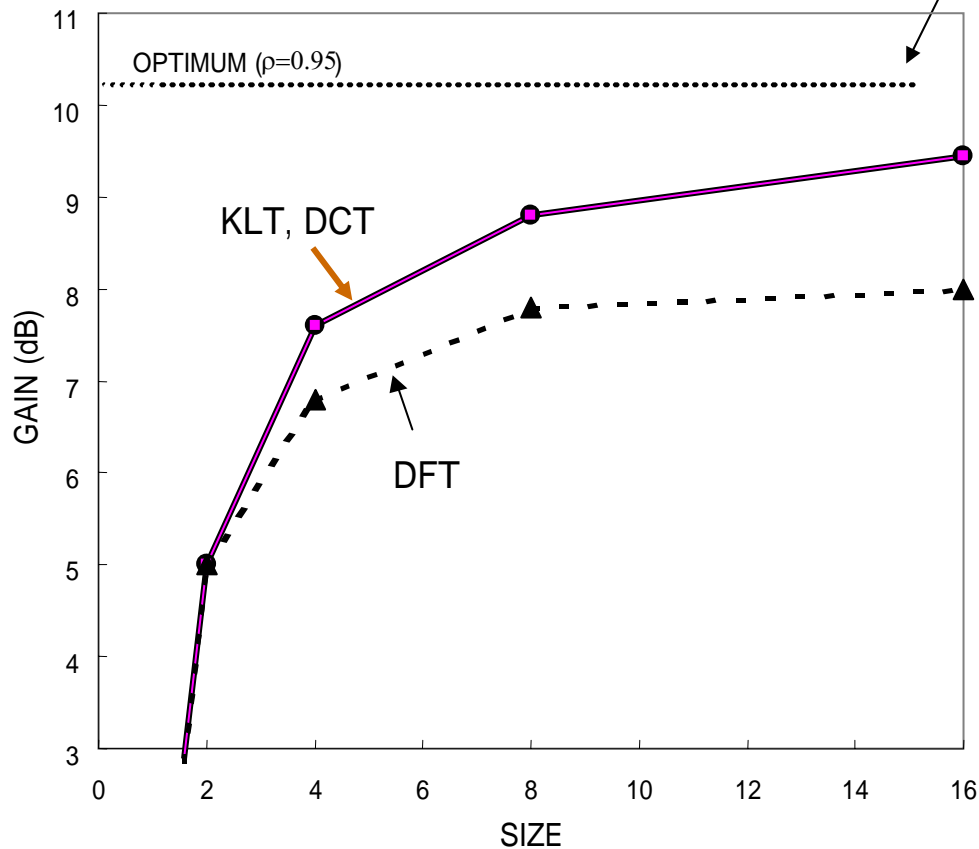




# 直交変換 (1)

## • 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



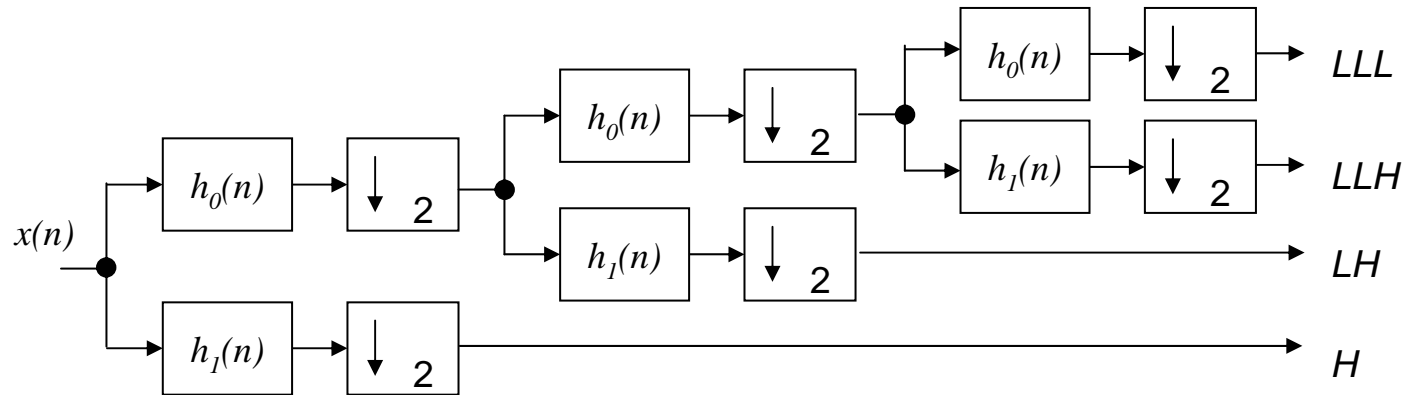
KLT: 理論的に最適な直交変換。

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。

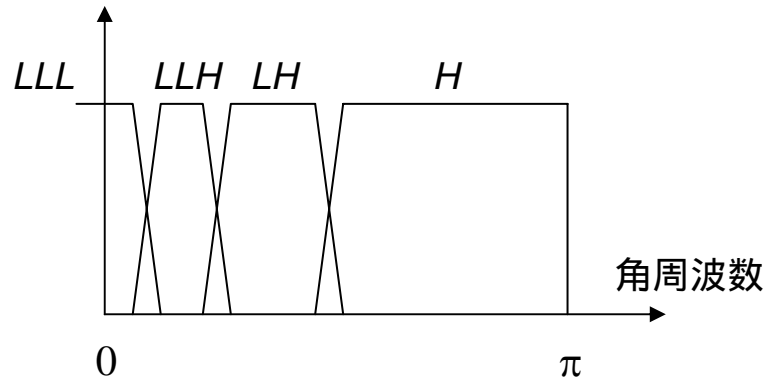
→ 直交変換の  
ブロックサイズ

# 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



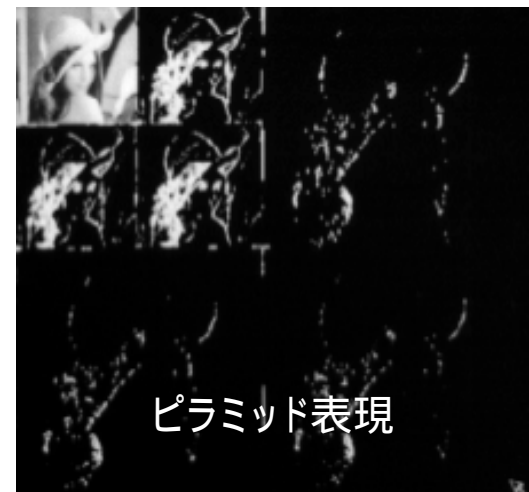
長所: ブロックひずみが少ない

短所: ブロック動き補償と相性が悪い

$LL$

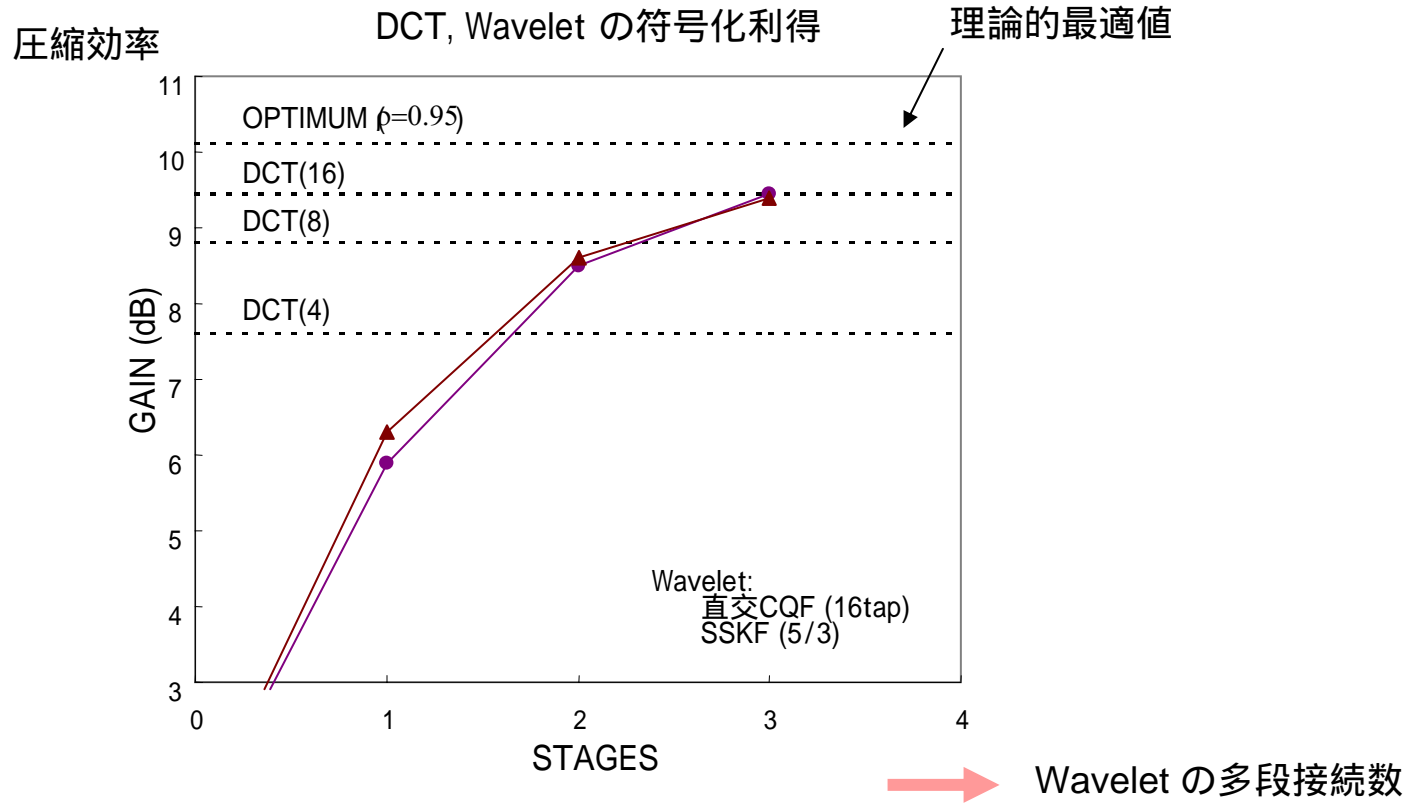
$LH$

$H$



# 直交変換 (3)

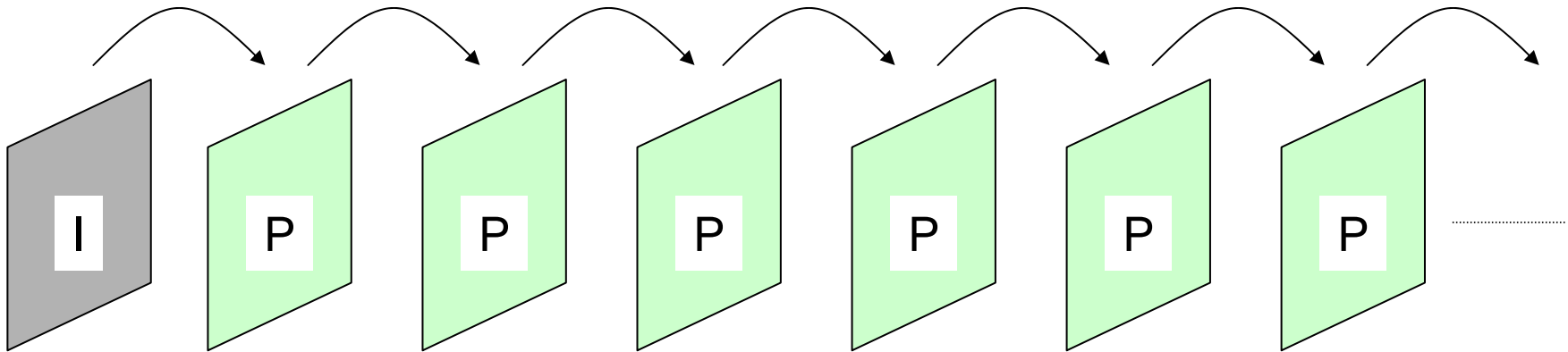
## • DCT と Wavelet の比較



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

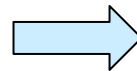
# フレーム間符号化 (1)

- 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 ~)



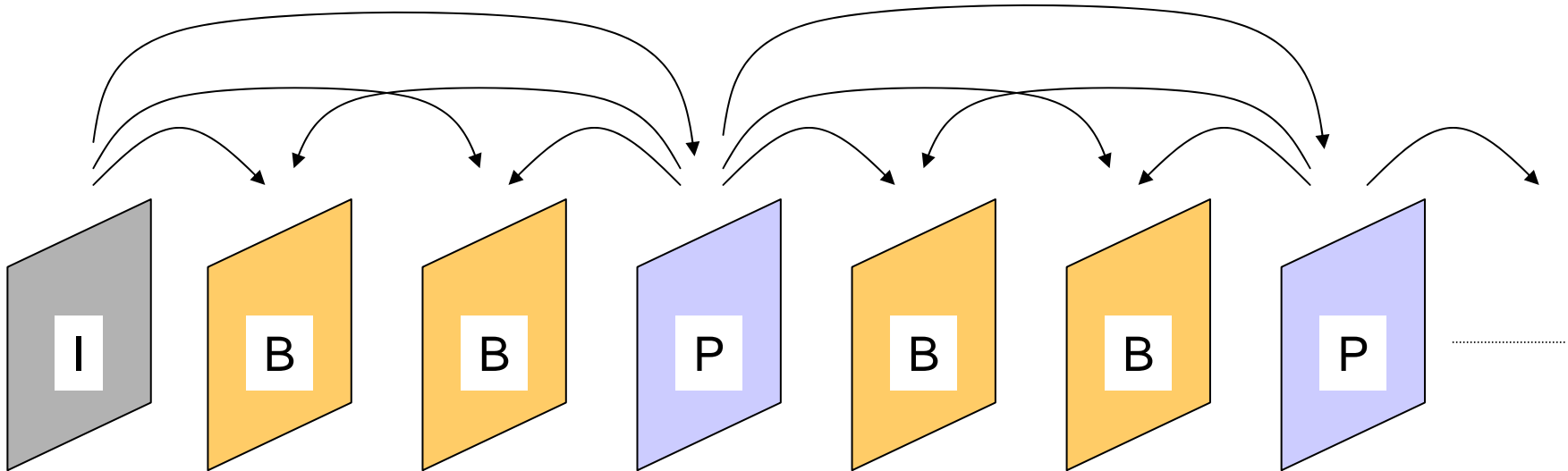
フレーム間の予測誤差がほとんどゼロ

- 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)

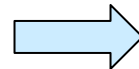
さらに動き検出・動き補償予測

# フレーム間符号化 (2)

## • IPB 予測



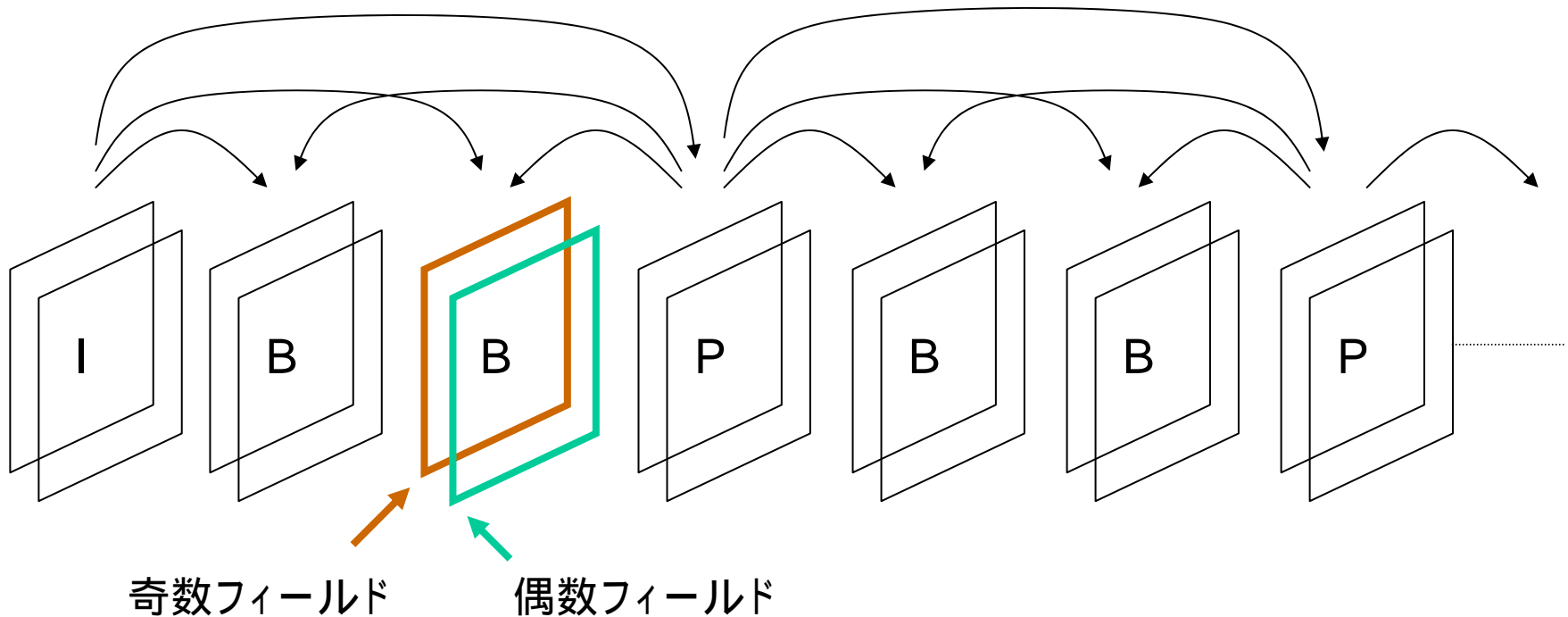
片方向で予測を行うより、両方向で予測を行うほうが予測効率が高い (ただし、フレーム間の距離に依存)



- 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)

# フレーム間符号化 (3)

## • フィールド予測



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

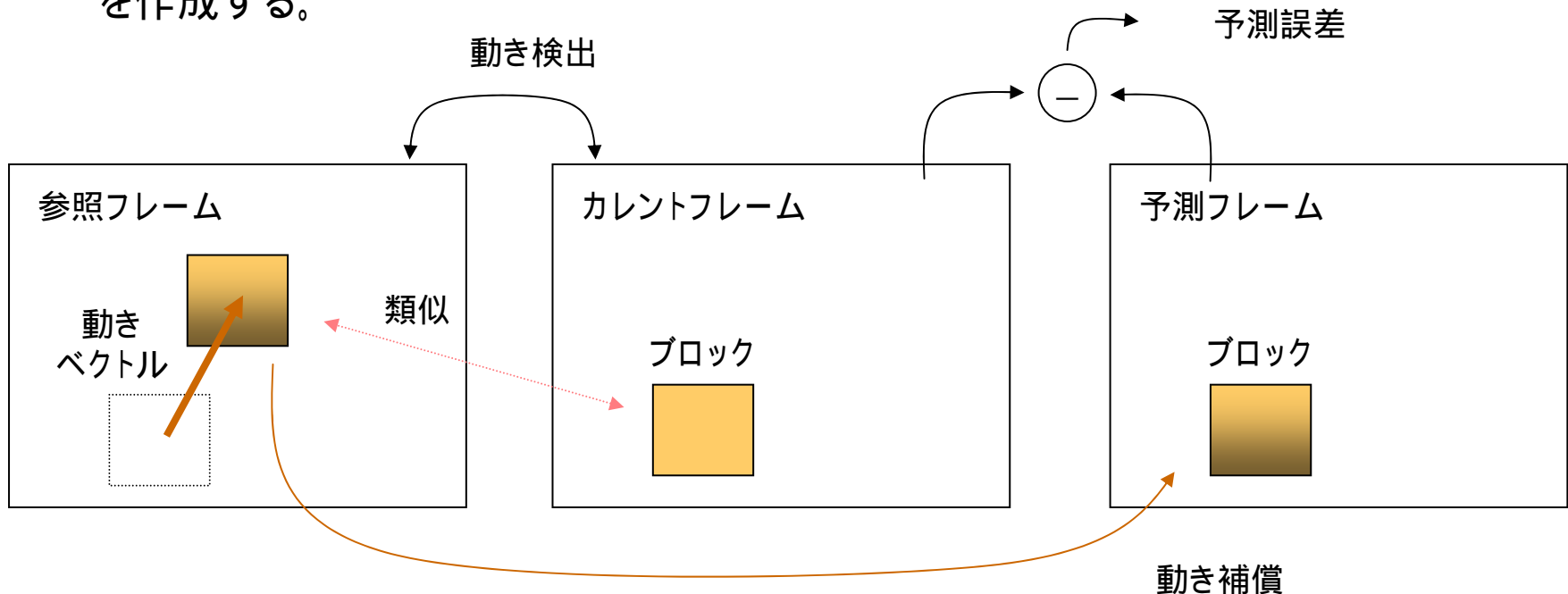
# 動き検出と動き補償 (1)

- 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:

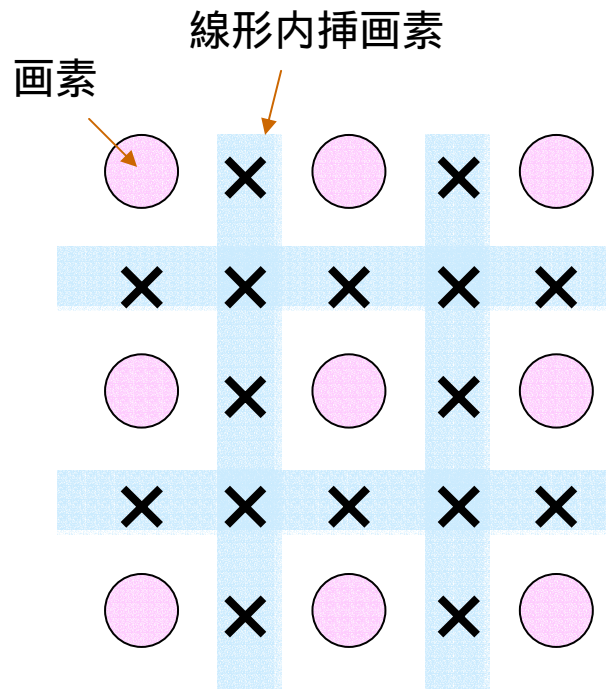
動き検出で求めた動きベクトルから、カレントフレームの予測画像 (予測フレーム) を作成する。



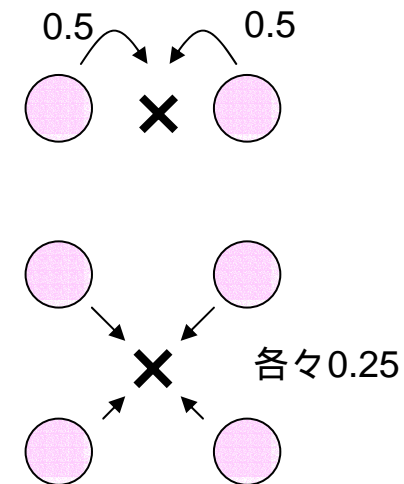
# 動き検出と動き補償 (2)

- 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:



(注) 1/4精度、1/8精度の効果はほぼ飽和

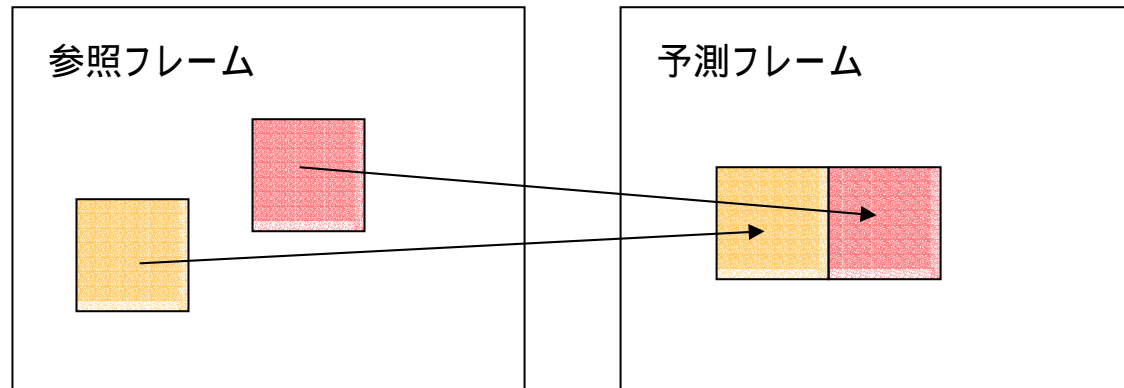


# 動き検出と動き補償 (3)

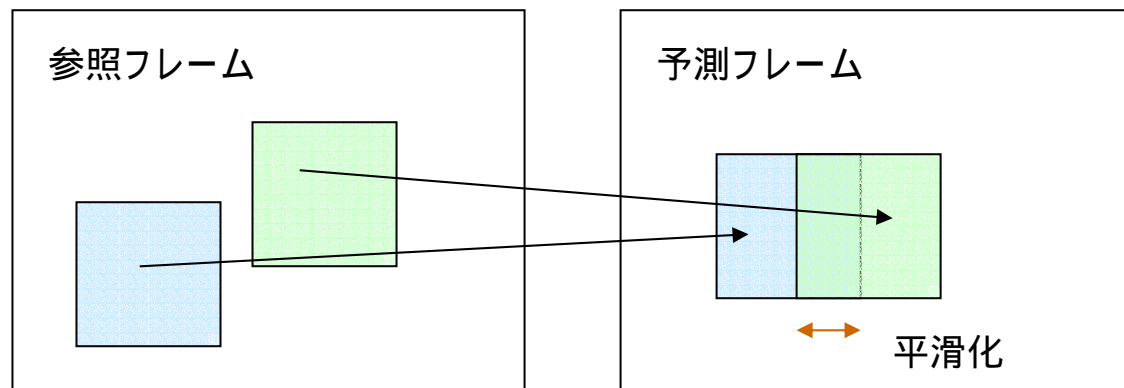
- オーバーラップ動き補償:

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



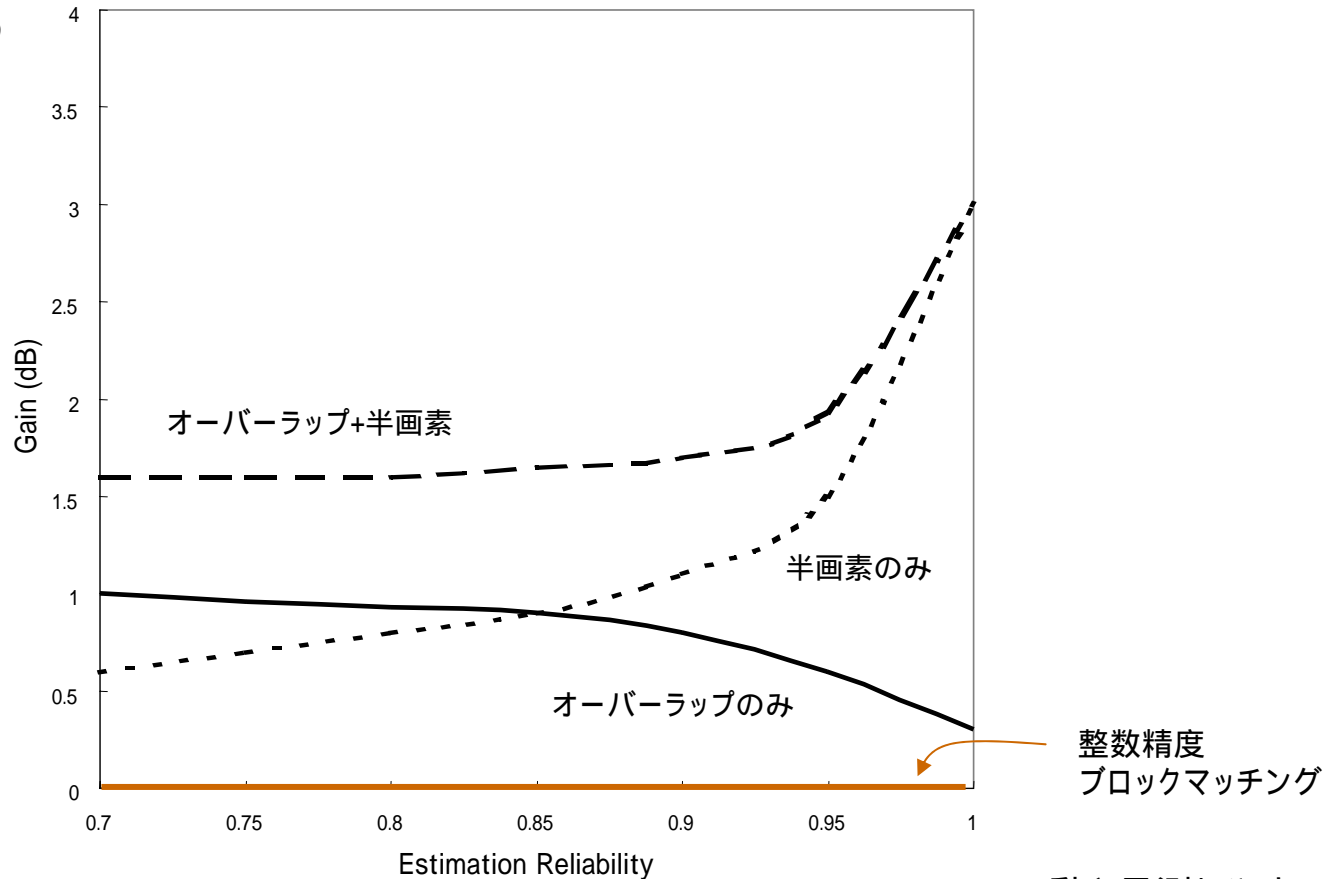
平滑化: 台形ウィンドウ、  
コサインウィンドウなど。

# 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

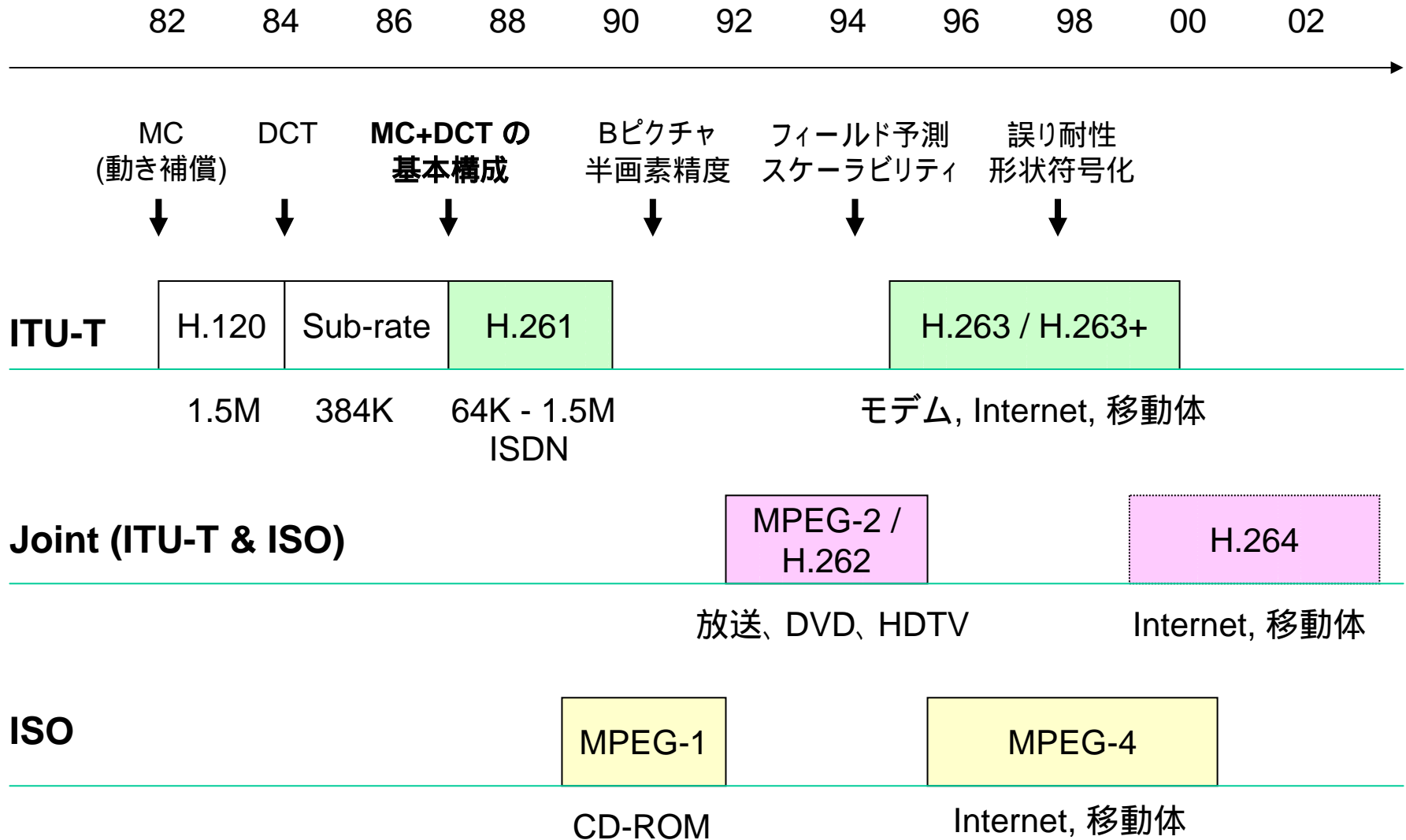
# 國際標準方式

# 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 (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							
H.264				-			

インターネット放送で有効  
+ 符号量制御 (後述)

**ISO/IEC MPEG-4**

# MPEG-4 の特徴

- 機能拡張 (誤り耐性とオブジェクトベース符号化)

誤り耐性ツール (アダプテーション)	:	シンプルプロファイル
形状符号化ツール (シーン合成)	:	コア・メインプロファイル
スプライト符号化	:	メインプロファイル

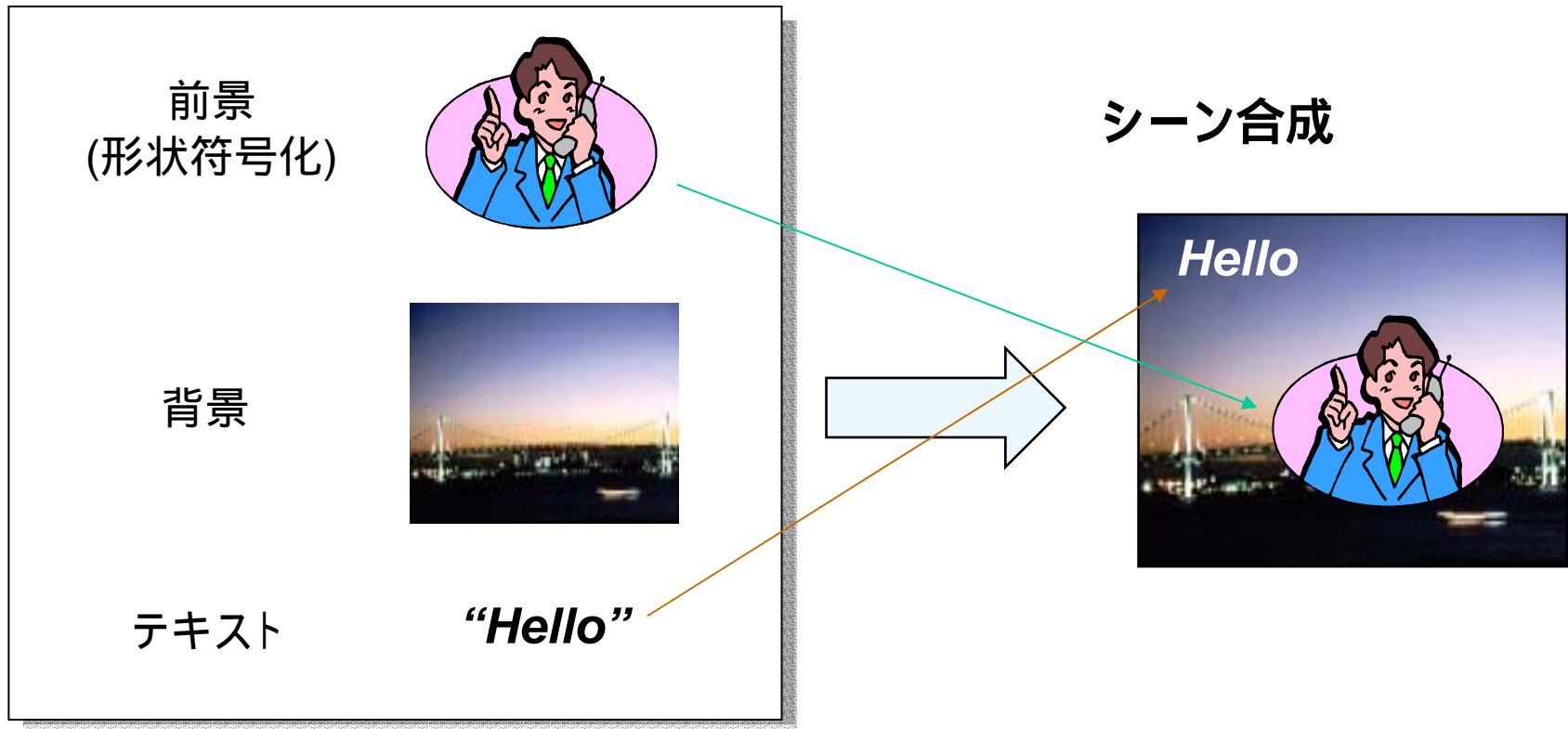
---

静止画像符号化 (Wavelet 変換)	:	ハイブリッドプロファイル
顔画像・胴体アニメーション	:	ハイブリッドプロファイル
メッシュ符号化	:	ハイブリッドプロファイル



# 形状符号化 (1)

- オブジェクト合成

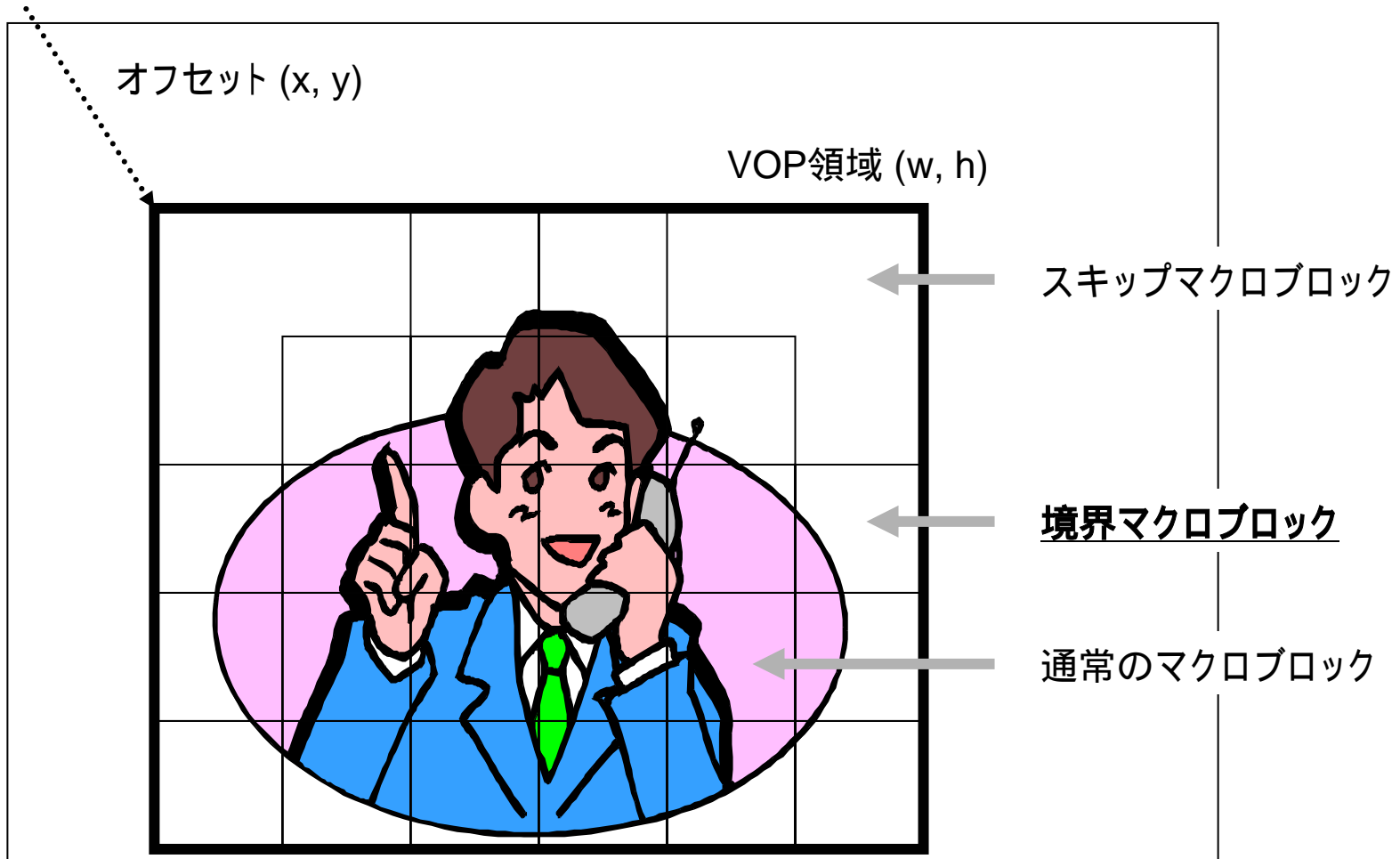


(注) 形状取得方法 (領域分割方法) は標準化の対象外

# 形状符号化 (2)

- 境界マクロブロック

通常のフレーム (CIF, QCIF, ...)

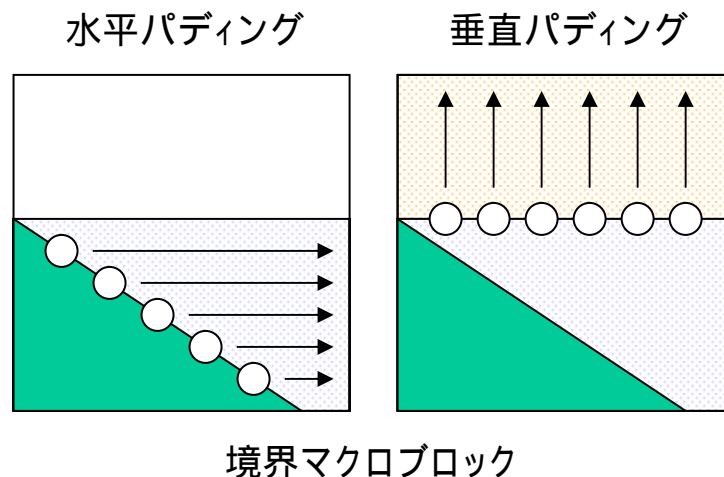


# 形状符号化 (3)

## • 境界マクロブロックにおけるパディング処理

### 境界MBの動き検出・動き補償

- (1) 形状範囲外をパディング
- (2) ポリゴンマッチング
- (3) 予測画像作成

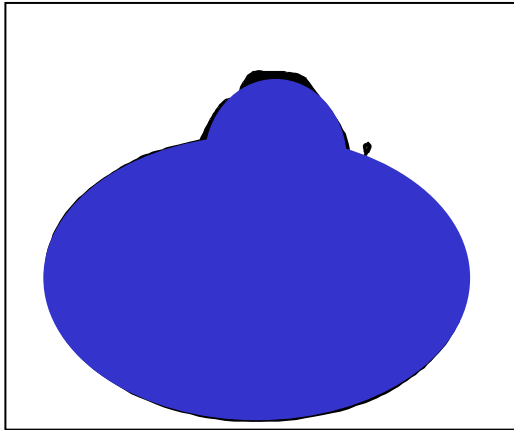


### 境界MBのテクスチャ符号化

- (1) I-ピクチャ: ブロック内平均値でパディングしたブロックに DCT
- (2) P-ピクチャ: 形状範囲外を0でパディングしたブロックに DCT

# 形状符号化 (4)

- 形状の符号化



(1) バイナリ符号化

2値画像 (0,1) として符号化

(2) グレイスケール符号化

(0, 255) の画素とみなして符号化 (DCT)

**(参考) (R, G, B, A)、(Y, U, V, A) フォーマット**

A: アルファマップ (コンピュータグラフィックス用語)

A = 0: 透過、形状無し (transparent)

A = 255: 形状あり (opaque)

A = 1 ~ 254: アルファブレンディング (前景と背景の混合)

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

# H.264の特徴

- 圧縮効率の改善 (MPEG-4 の 70%)

動き補償予測、イントラ予測の改善 (多モード化)

可変ブロックサイズ (Sub-macroblock prediction)

1/4・1/8 画素精度 (1/4, 1/8 pel interpolation)

複数参照ピクチャ (Multiframe prediction)

アダマール変換 (SAT: sum of absolute transformed differences)

適応イントラ予測 (Adaptive intra prediction)

適応ブロックサイズDCT (Adaptive block-size transform)

エントロピー符号化の改善

CABAC (Context-based Adaptive Binary Arithmetic Coding)

---

符号化レイヤとネットワークレイヤの分離

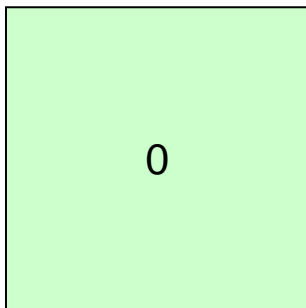
VCL: Video Coding Layer

NAL: Network Adaptation Layer

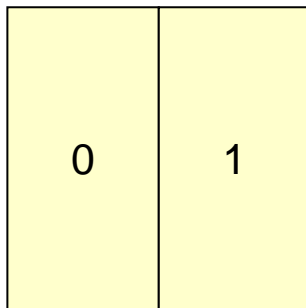
RTP Packetization

# 可変ブロック動き補償

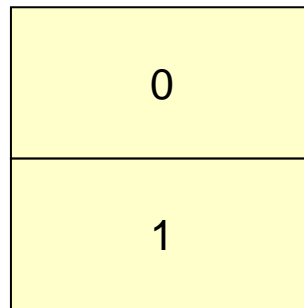
mode 1  
16x16 block  
1 vector



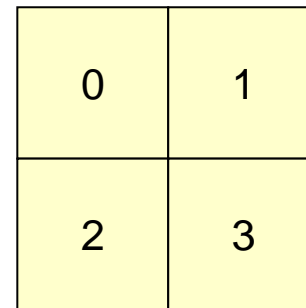
mode 2  
8x16 block  
2 vectors



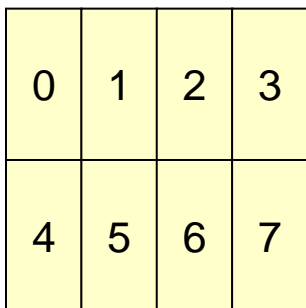
mode 3  
16x8 block  
2 vectors



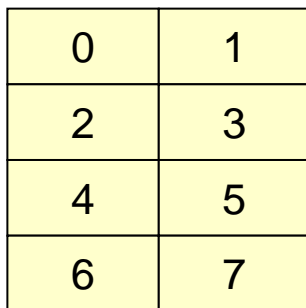
mode 4  
8x8 block  
4 vectors



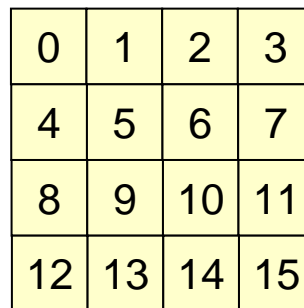
mode 5  
4x8 block  
8 vectors



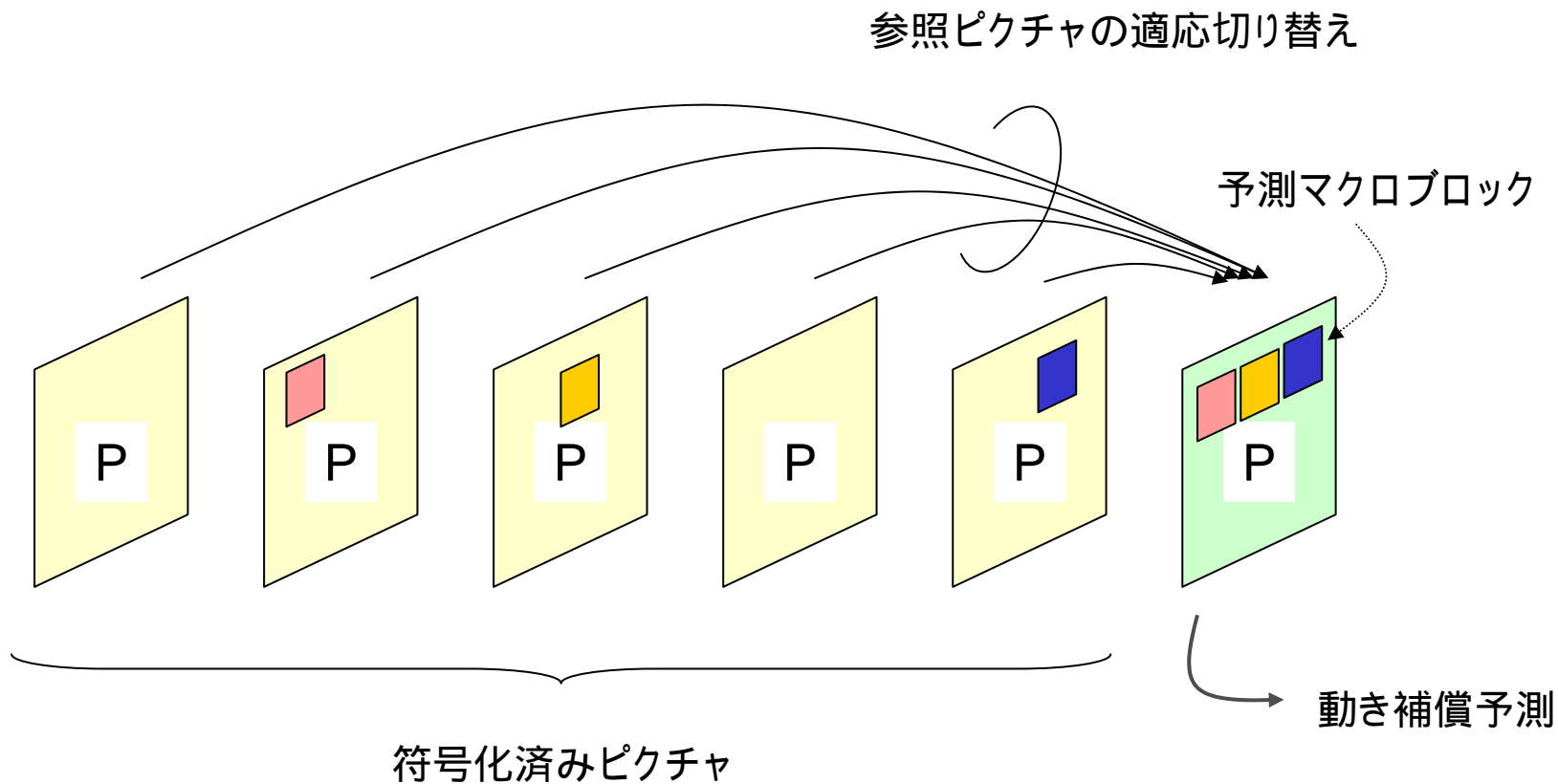
mode 6  
8x4 block  
8 vectors



mode 7  
4x4 block  
16 vectors



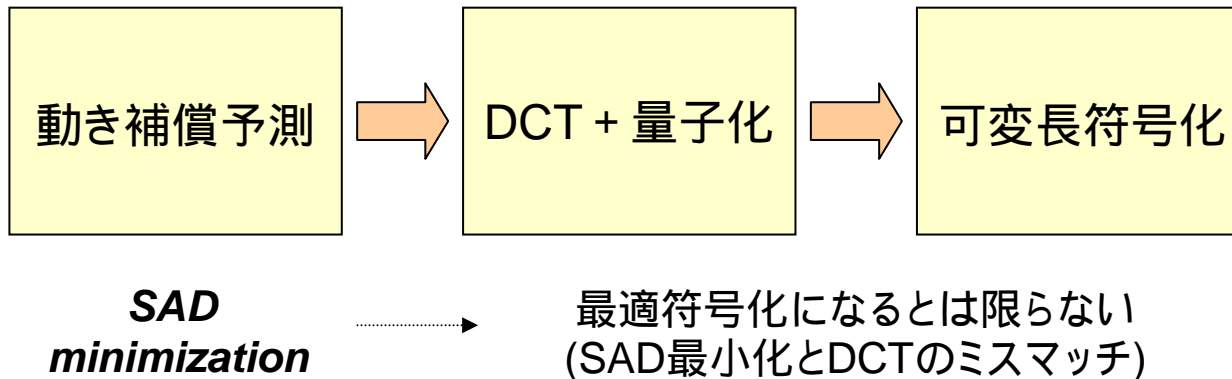
# 複数参照ピクチャ



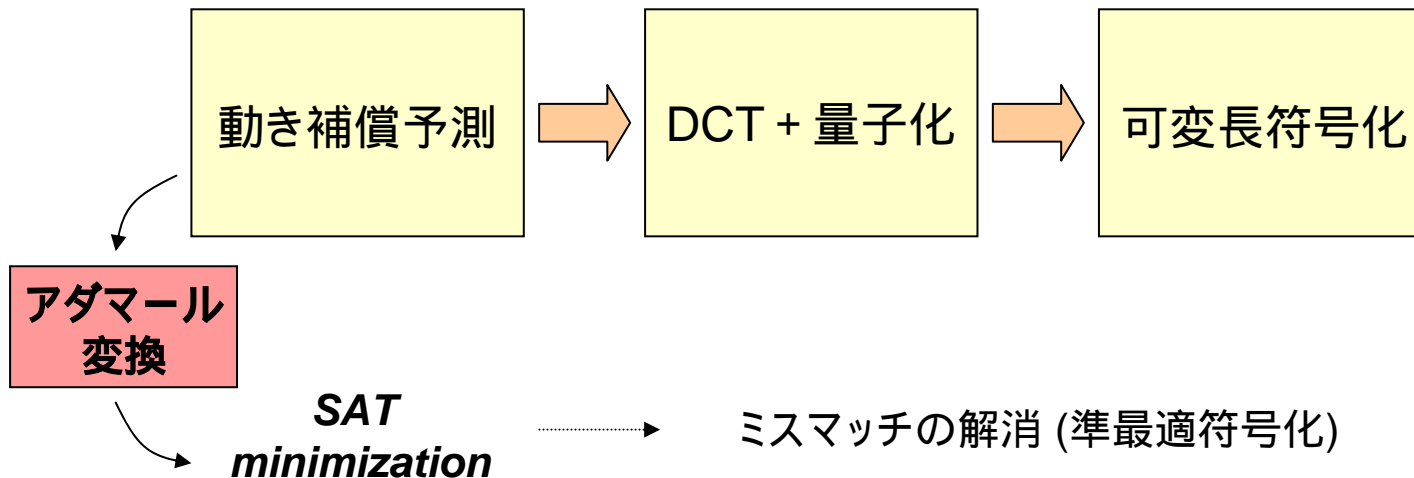


# SAT (アダマール変換)

SAD minimization (空間領域の予測誤差最小化):



SAT minimization (変換領域の予測誤差最小化):

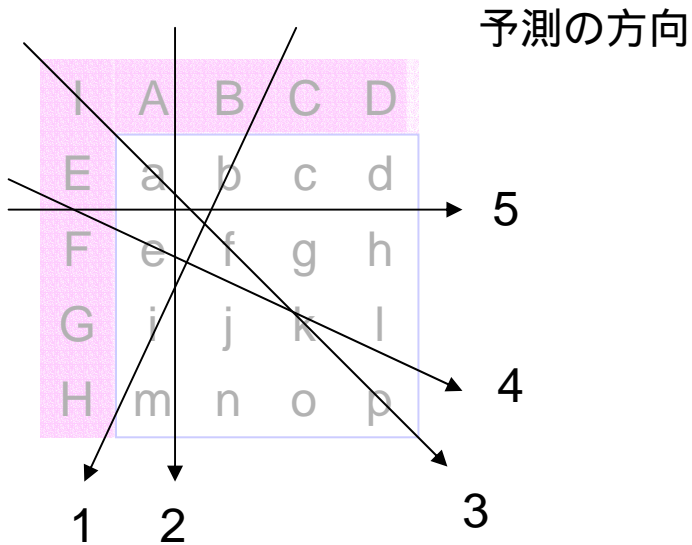


SAD: sum of absolute differences, SAT: sum of absolute transformed differences

# 多モード・イントラ予測

I	A	B	C	D	← 符号化済み画素
E	a	b	c	d	
F	e	f	g	h	← 未符号化画素
G	i	j	k	l	
H	m	n	o	p	

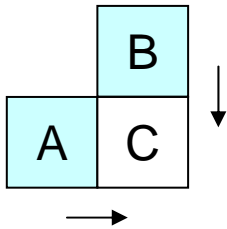
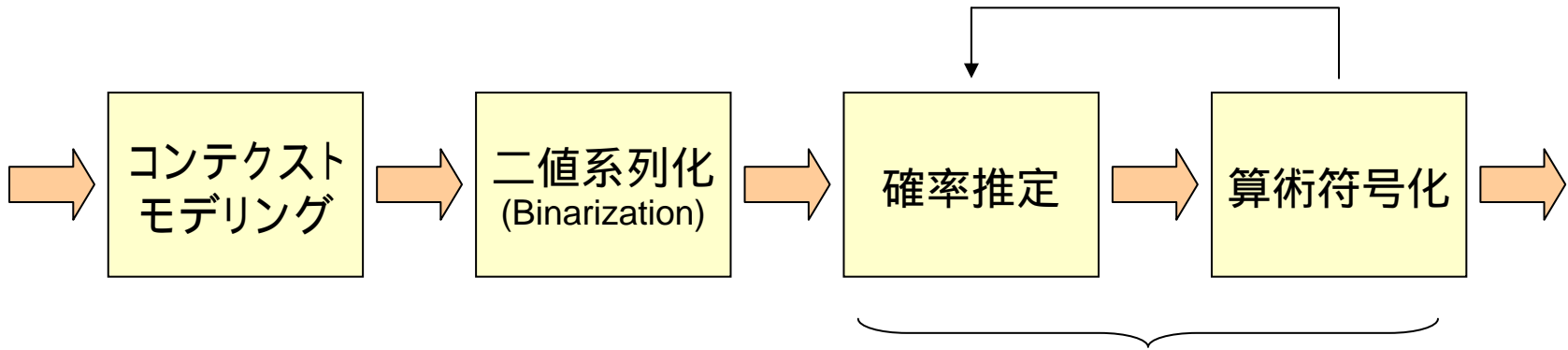
- mode 0: DC prediction
- mode 1: Vertical/Diagonal prediction
- mode 2: Vertical prediction
- mode 3: Diagonal prediction
- mode 4: Horizontal prediction
- mode 5: Horizontal/Diagonal prediction



例: mode 0:  
 予測値 =  $(A+B+C+D+E+F+G+H) / 8$

mode 1:  
 $a = (A+B) / 2$   
 $e = B$   
 $b = i = (B+C) / 2$   
 $f = m = C$   
 $c = j = (C+D) / 2$   
 $d = g = h = k = l = n = o = p = D$

# CABAC



A, B: 符号化済みブロック

符号化済みブロックの結果に応じて確率モデルを切り替え

C: 符号化対象ブロック

Binarization の例 (unary)

index	binarization
0	1
1	01
2	001
3	0001
4	00001
5	000001

選択された確率モデルに従い、二値系列を算術符号化

さらに、入力結果に従って確率モデルを更新

vs. ハフマン符号 (従来)

# 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%
アダマール変換	~ 5%
複数参照ピクチャ	~ 5%
R-D最適化 (後述)	10 ~ 15%
総計	30%以上

# 最近の話題

- Rate-Distortion Optimization
- Multiple Distortion Coding
- Wyner-Zip Coding

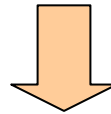
# R-D最適化 (1)

## Lagrange の未定乗数法 (Rate-Constrained X)

$$\text{minimize } J = D + \lambda \cdot R$$

最適化問題

ビデオ圧縮の多モード化への対応



ひずみとレート (D, R) の設定方法に応じてさまざまな応用:

- 動き補償における予測モード・動きベクトル選択 (圧縮効率)
- 量子化におけるマクロブロックの INTRA/INTER モード選択 (圧縮効率)
- 量子化におけるマクロブロックの INTRA/INTER/SKIP モード選択 (誤り対策)
- 複数参照フレームにおける参照フレーム選択 (圧縮効率)
- ストリーミングにおけるパケット送出スケジューリング (FastStreaming)

# R-D最適化 (2)

の設定例 (DCT係数の量子化):

(1)  $J$  の最小化

$$\frac{dJ}{dR} = \frac{dD}{dR} + \lambda \quad \longrightarrow \quad \lambda_{MODE} = -\frac{dD}{dR}$$

(2) レートひずみ関数

$$R(D) = a \log_2 \left( \frac{b}{D} \right)$$

(3) ひずみと量子化ステップサイズの関係式

$$D = \frac{(2Q)^2}{12} = \frac{Q^2}{3}$$

$$\lambda_{MODE} = c \cdot Q^2 \quad (\text{H.263})$$

$$\lambda_{MODE} = c \cdot 2^{\frac{Q}{3}} \quad (\text{H.264})$$

$$c = \frac{\ln 2}{3a} \approx 0.85$$

# R-D最適化 (3)

## 使用例

従来方式:

【動き補償予測】

動きベクトルの選択:

$$D_{motion} \rightarrow \min$$

動き補償予測誤差が最小となるモードと動きベクトルを選択

【量子化】

符号化モードの選択:

$$D_{mode} \rightarrow \min$$

量子化誤差が最小となるモードを選択

RD 最適化:

【動き補償予測】

動きベクトルの選択:

$$J_{motion} = D_{motion} + \lambda_{motion} \cdot R_{motion} \rightarrow \min$$

コスト (動き補償予測誤差とベクトルオーバーヘッドの和) が最小となるモードと動きベクトルを選択

【量子化】

符号化モードの選択:

$$J_{mode} = D_{mode} + \lambda_{mode} \cdot R_{mode} \rightarrow \min$$

コスト (量子化誤差とヘッダ情報、ベクトル情報等の各種オーバーヘッドと可変長符号の和) が最小となるモードを選択

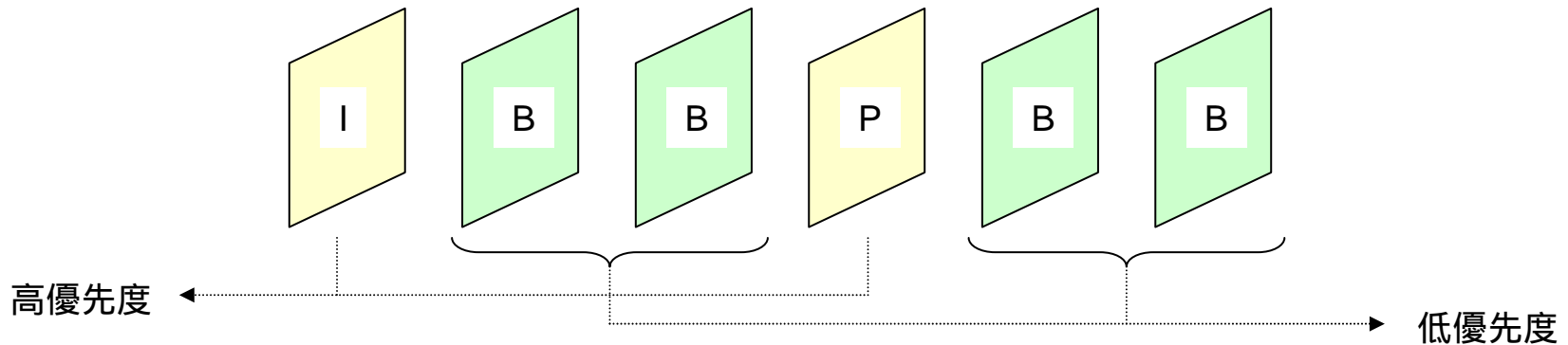
課題: 演算量の増加をどのように抑えるか?



# Multiple Description Coding

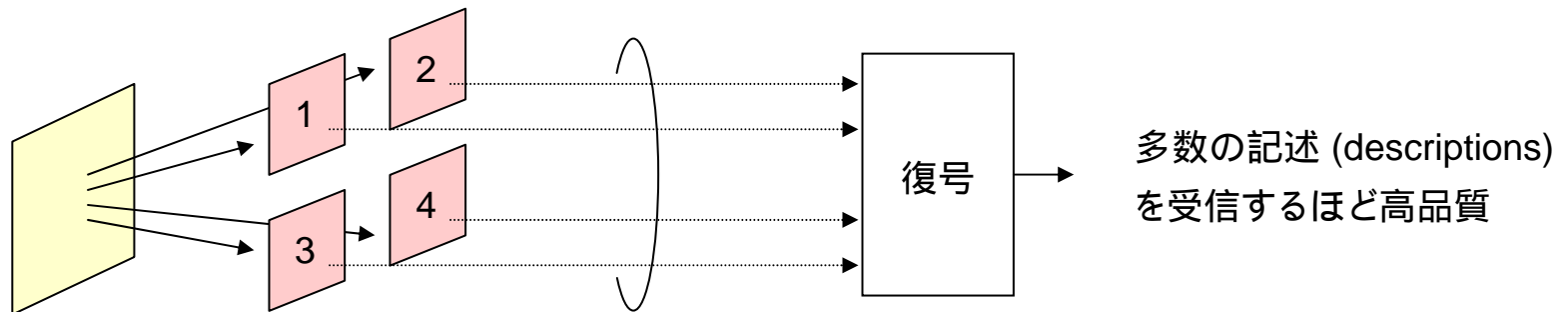
## スケーラブル符号化 (従来):

(例) Temporal scalability: 各種ストリーミング



## Multiple Description Coding:

例: 空間サンプリング、複数コーデック、...

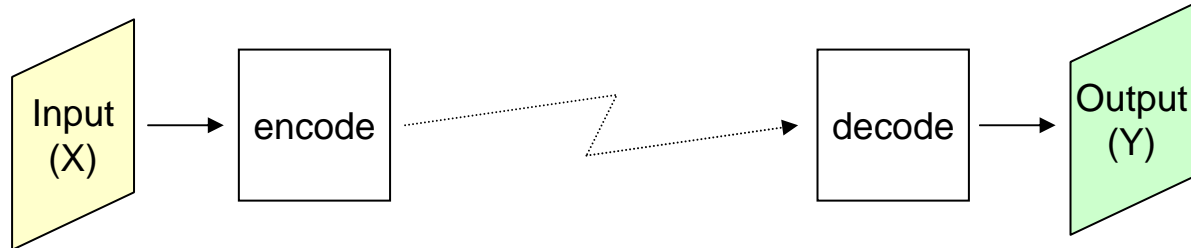


同一優先度 + マルチパス伝送

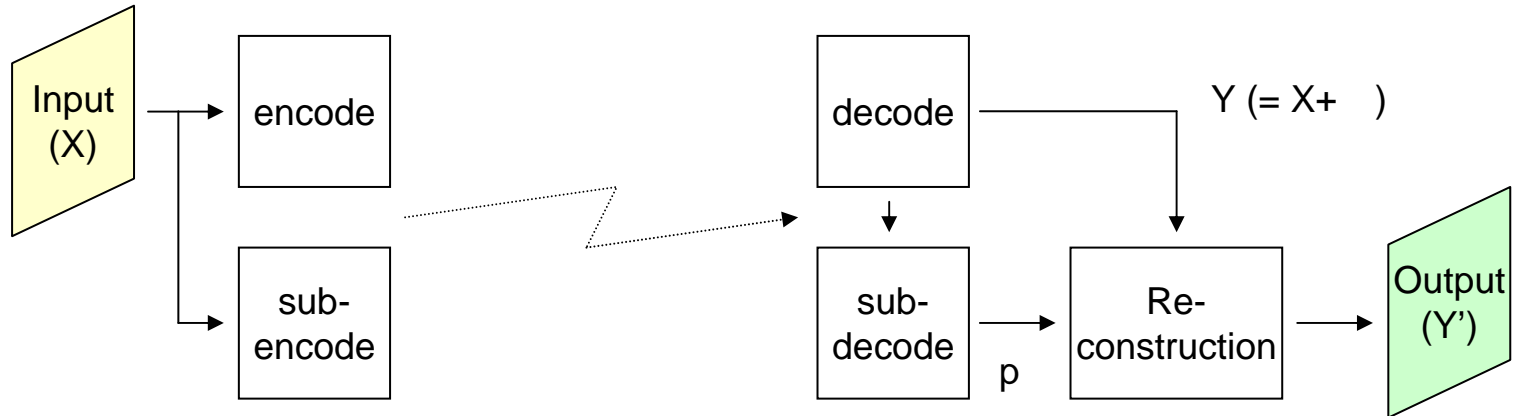
課題: 「記述」をどう定めるか?

# Wyner-Ziv Coding

従来の符号化:



Distributed Source Coding (Wyner-Ziv Coding):



MAP推定:  $Y' = \arg \max P(X | Y, p)$

課題: 補助情報  $p$  (sub-information) をどのように定めるか?