[招待講演] MPEG-4 AVC | H.264 の概要と標準化動向

鈴木輝彦†

†ソニー（株）S&S アーキテクチャセンター 〒141-0032 東京都品川区大崎 1-11-1
E-mail: †teruhiko@av.crl.sony.co.jp

あらまし 昨年 12 月に MPEG と ITU-T で次世代動画像符号化方式の標準化を行う、Joint Video Team が設立された。ITU-T で標準化が進められてきた H.26L をベースとして現在規格化が進められている。JVT で規格化された符号化方式は MPEG では MPEG-4 part 10 (Advanced Video Coding) ITU-T では H.264 として標準となる予定である。本発表では、JVT で規格化が進められている MPEG-4 AVC|H.264 の概要と最新の標準化動向を報告する。

キーワード MPEG-4, H.264, ビデオ符号化

The overview of MPEG-4 AVC | H.264 and its standardization

Teruhiko SUZUKI†

† S&S Architecture Center, Sony Corp. 1-11-1 Osaki, Shinagawa-ku, Tokyo, 141-0032 Japan
E-mail: †teruhiko@av.crl.sony.co.jp

Abstract JVT (Joint Video Team) was established by ISO and ITU-T to standardize next generation video coding technology in the last December. The standardization has been started based on the "H.26L", which was standardized in ITU-T SG16. The output of JVT will be H.264 in ITU-T and MPEG-4 part 10 (Advanced Video Coding) in ISO. In this document, the overview of H.264 and the recent standardization activity are reported.

Keyword MPEG-4, H.264, Video Compression

1. Introduction

昨年 12 月に MPEG と ITU-T で次世代動画像符号化方式の標準化を行う、Joint Video Team が設立された。ITU-T で標準化が進められてきた H.26L をベースとして現在規格化が進められている。JVT で規格化された符号化方式は MPEG では MPEG-4 part 10 (Advanced Video Coding) ITU-T では H.264 として標準となる予定である。本発表では、JVT で規格化が進められている MPEG-4 AVC|H.264 の概要と最新の標準化動向を報告する。本稿では規格の名称を H.264 として以降解説する。

2. H.264 の技術概要

2.1. Codec の構成

H.264 は既存の規格(H.263 や MPEG-4)の 2 倍の符号化効率を達成することを目標に標準化が進められている。Codec の構成自体は全く新しい物ではなく、従来の動き補償＋DCT 変換を用いた構成が引き続き採用されている。全くの新規技術により、2 倍の符号化効率を達成するのではなく、個々のツールを最適化し、その蓄積により達成している。

以下の技術要素は既存技術と共通の技術である。

- 16x16 Macroblocks
- Conventional 4:2:0
- Block motion displacement
- Motion vectors over picture boundaries
- Variable block-size motion
- Block transforms
- Scalar quantization
- I, P and B picture types

Figure 1 に H.264 の構成を示す。

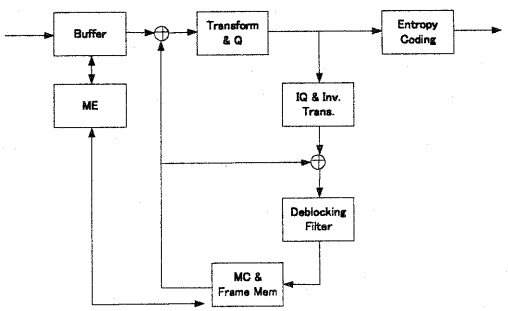

Figure 1: Block Diagram of H.264

2.2. 動き補償

H.264 においては、一つの Macroblock を、Figure 2 に示すように 16x16, 16x8, 8x16, 8x8 の MC Block に

分割することが可能である。更に、8x8 ブロックに関しては、8x8, 8x4, 4x8, 4x4 の Sub MC Block に分割することが可能である。それぞれの MC Block は独立した動きベクトルを持つ。動きベクトル(MV) の精度は 1/4 pel 精度であり、6 tap のフィルタにより補間される。Figure 2 に H.264 の MC Block を示す。

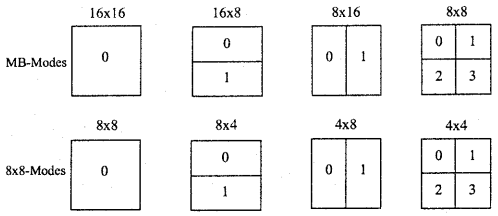

Figure 2: 動き補償の際のブロック分割

2.3. 動きベクトル符号化

Figure 3 に符号化するブロックとそれに隣接するブロックを示す。E が当該 MC Block、A,B,C,D は隣接する MC Block である。

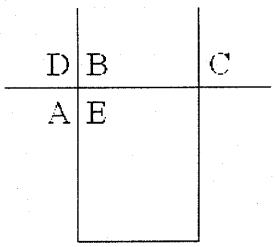

Figure 3: 隣接する MV Block

MC Block E に対する動きベクトルの予測値は、A,B,C に対する動きベクトルの Median が用いられる。

2.4. フレーム内符号化

フレーム内符号化（イントラ）を行う際、それ以前に復号が修了している上と左隣のブロックの画素値から、符号化ブロックの画素値を予測する空間予測が採用されている。予測のモードには 4x4 mode と 16x16 mode の 2 種類が定義されている。

2.4.1. Intra 4x4 mode

Intra 4x4 mode においては 4x4 block 単位で予測を行う。 Figure 4 に示す a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p から構成されるブロックを周囲の A - Q の画素を用いて予測を行う。

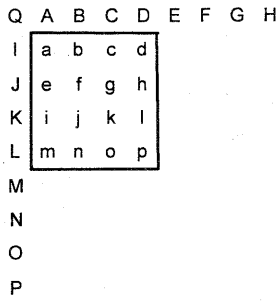

Figure 4: Intra prediction に使用される画素

予測を行う際に、Figure 5 にしめされる 8 つの方向に画素を内挿して、予測画素を生成する。

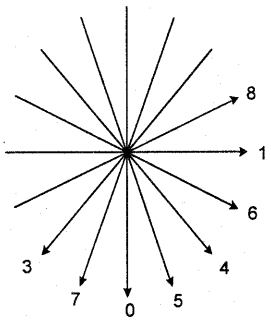

Figure 5: Intra prediction の 8 つのモード

Intra 16x16 mode では 16x16 画素のマクロブロック単位で予測を行う。Intra 16x16 予測には以下の 3 つのモードが定義されている。

Mode 0: vertical prediction

隣接する上のブロックの画素から予測を行う。

Mode 1: horizontal prediction

隣接する左ブロックの画素から予測を行う。

Mode 2: DC prediction

隣接する上および左のブロックの画素を用い、平面近似に予測を行う。

2.5. DCT変換と量子化

MPEG や H.263 といった既存の規格では実数精度の 8x8 block size の DCT 変換を採用してきた。しかし、H.264 では整数精度の 4x4 block size DCT 変換が採用された。これにより DCT の精度を規定する必要がなくまた各社デコーダ間のミスマッチも生じないと

いう利点がある。

DCT 変換は以下の式で表せる。

$$H_{kn} = H(k,n) = c_k \sqrt{\frac{2}{N}} \cos\left((n+1/2)\frac{k\pi}{N}\right)$$

$$x = H^{-1}X = H^T X$$

4x4 変換の場合、変換行列は以下の通りとなる。

$$H = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ c & s & -s & -c \\ 1 & -1 & -1 & 1 \\ s & -c & c & -s \end{pmatrix}$$

$$c = \sqrt{2}\cos(\pi/8), s = \sqrt{2}\sin(\pi/8)$$

これを整数精度に近似すると係数は以下の式で表せる。

$$Q(k,n) = round(\alpha H(k,n))$$

ここで、直交変換となり一様な Norm をもつ変換行列となる、最小の α は 2 である。α=2 はアダマール変換であり、高い Energy Compaction が得られない。次に小さい α は 26 であり、H.26L の TML 9 まではこの変換行列が採用されていた。しかしながら、α=26 の場合、32 bit 精度の演算が必要となってしまう。そこで、H.264 では 16 bit 演算で実現可能にするために、直交変換ではあるが、Norm が一様でない行列（α=2.5）を採用した。

上記のような整数精度の DCT 変換を採用した場合、変換自体に Gain が生じる。この変換による Gain を考慮して量子化スケールが決められている。

また、各量子化レベルの間隔は、量子化パラメータ QP が 1 増加すると 12%増加し、また QP が 6 増加するとスケール値が 2 倍になるように設計されている。MPEG や H.263 の量子化では単純に DCT 係数に対するスケール値が等間隔になっており、量子化係数と歪に特に一定の関係が存在しない。H.264 の量子化レベルは各レベル間での歪増加量が一定になるように設計されており、より符号量制御が行い易くなっている。

また、色差の量子化スケールは輝度の量子化スケールからテーブル参照によって決定されるが、色差の量子化スケールの方が輝度よりも小さくなるように設計されており、より色差の劣化を抑えるようにしている。

Intra 16x16 mode で符号化されたブロックは、最初に 4x4 DCT を行った後、各ブロックから DC 成分のみを取り出し、4x4 DC Block を構成する。この 4x4 Block にさらにアダマール変換を施し、より効率を高めている。

色差信号も同様にして 4x4 DCT 変換を行った後 2x2 Block を構成し、これに 2x2 のアダマール変換を行う。

変換された DCT 係数は以下の Zig Zag Scan により 1 次元に変換され、エントロピー符号化される。

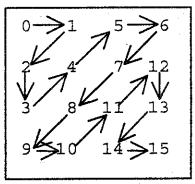

Figure 6: Zig Zag Scan

2.6. エントロピー符号化

H.264 では、Syntax element を VLC および CABAC と呼ばれる算術符号化の 2 種類の方式によって符号化している。Syntax element は例えば、SKIP MB の個数(RUN)、MB の種類(MB_Type)、8x8 ブロックの種類、動きベクトルの差分、Quant の差分、CodedBlockPattern、イントラ予測モード、変換係数 などがある。

2.6.1. VLC

変換係数以外の Syntax element には、Exp-Golumb code を用いて可変長符号化を行う。変換係数の符号化には、CAVLC(Context-Adaptive VLC)により符号化される。周囲の DCT 係数から context を計算し、context に応じて VLC を切り替えて符号化する。MPEG-2 など既存の符号化方式では DCT 係数を Zig Zag Scan して 1 次元の信号にした後、信号の LEVEL, 非ゼロ係数の間隔 RUN を組み合わせて 2 次元 VLC を用いて符号化される。また、ブロックの終了を示すフラグ EOB を別途符号化する。(MPEG-4 では EOB も含めた 3 次元 VLC を使用)

CAVLC では RUN, LEVEL は独立に符号化され、また DCT 係数も高周波から低周波へと、従来とは逆方向にスキャンされる。また、EOB を符号化せずに、係数の数を符号化する。

2.6.2. CABAC

エントロピー符号化には上記の VLC の他に、CABAC (Context Adaptive Binary Arithmetic Coding) という算術符号化が採用されている。

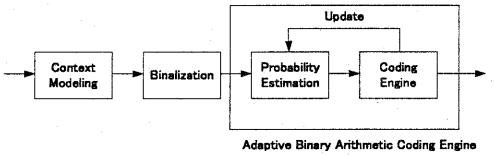

Figure 7: CABAC Overview

Figure 7 に CABAC の構成を示す。Context modeling 各シンボルを符号化する際の確率モデルである。各 Syntax Element に Context が定義されており、この Context に応じて確率テーブルを切り替えて算術符号を行う。

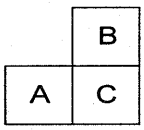

Figure 8: 隣接ブロック

例えば、Cのブロックを符号化する際、隣接する簿ロックA，Bの状態に応じて Context が決定される。確率テーブルはスライスの先頭で初期化される。

各 Syntax Element は Unary Code によりバイナリ化され(MB Type は例外で Table によりバイナリ化される)、算術符号化が行われる。各シンボルを符号化した後、確率モデルは更新され、適応的に符号化される。

2.7. Deblocking Filter

MC+DCT 変換を採用する符号化方式の場合、Block 境界で、歪が生じる。この劣化を低減するために、既存方式の場合、Post Filter を用いる。Block 境界にフィルタをかけ、歪を平滑化する。

H.264 では、画像の 4x4Block 境界に Filtering を施し、Block 分割符号化方式に特有の Block 歪を低減することを目的に、動き補償の Loop 内に Deblocking Filter が導入された。Filtering された画像は出力画像として用いられるだけでなく、そのフレーム以降の MC 参照画像としてフレームバッファに保存されるため（いわゆる In-Loop Filter)、参照画像が Block 歪のない滑らかな画像となり、符号化効率や主観画質を向上させる。

Block 境界（4画素）に適用するフィルタの強度を Boundary Strength(Bs)として定義し、その値に応じて Block 境界毎に最も適した強度のフィルタをかけることが可能となっている。この Bs 値は、その境界に filter をかけるかどうかの判定と、filter をかけた際の画素値変動の最大値を定義するために用いられる。

各境界における Filtering において修正される画素は、最大で境界に隣接する2画素とその隣の画素の最大4画素。4画素のうちどの画素が Filtering で修正されるかは、Bs 値と画素そのもののなだらかさを閾値として決定される。基本的には、隣接画素の差が小さい場合（なだらか、または明確にエッジではない場合）に Filtering により値が修正されるようになっている。

2.8. Network Adaptation Layer

H.264 は Network への ITU-T で標準化が開始された経緯から、Network への配信をサポートするよう、codec のデザインが行われてきた。Codec の構成を以下に示す。

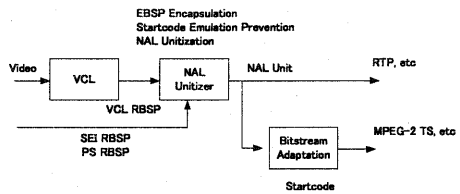

Figure 9: JVT Codec Structure

Codec は Slice layer 以下の Bitstream を符号化・復号する VCL (Video Coding Layer) とそれ以上の High level syntax を扱う NAL (Network Abstraction Layer) から構成される。NAL は VCL が生成する Stream を様々な Network の Payload にマッピングする。

MPEG などの既存の方式では、ビデオの符号化 Stream は全ての情報を1本の Bitstream に多重化し、Bitstream 単独でデコード可能なように規格が策定されていた。しかし、H.264 では Packet Network の伝送を想定し、Video Bitstream それ自体と、それをデコードするために必要な Configuration 情報、その他の情報を別々に伝送可能なように規格が決められている。

H.264 の Bitstream は以下の要素から構成される。

VCL RBSP: VCL では Slice Layer 以下の Video Stream を生成する。この VCL の出力を VCL RBSP (Raw Byte Sequence Payload) と呼ぶ。

Parameter Set RBSP: Picture header 以上の High level syntax で Video Stream のデコードに必要な情報を含む。Header を数種類テーブルにしてあらかじめ伝送し、Slice header ではそのテーブル番号を参照する。

SEI (Supplemental Enhancement Information) RBSP: User data などを伝送するもの。デコードに直接関連のない情報（バッファ情報も含む）は SEI Message として伝送される。

VCL RBSP, PS RBSP, SEI RBSP はそれぞれ NAL Unitizer で、Startcode emulation 対策が施され、EBSP (Encapsulated Byte Sequence Payload) と呼ばれる Stream が生成され、NAL unit と呼ばれる単位に分割さ

れる。

RTP に伝送する場合には、NAL unit 単位で RTP payload にマッピングされる。

MPEG-2 TS などのシリアル伝送には、NAL unit を繋ぎ合わせ1本の Stream にし、Startcode を付加する。1本の Bitstream にした後、MPEG-2 TS などの Payload にマッピングされる。

H.264 は MPEG-2 などのように Bit 単位で扱われず、Byte 単位で処理が行われる。Stream も byte stream と呼ばれる。Startcode は Bitstream adaptation のみに使用され、RTP などのようなパケット伝送には使用されない。しかしながら、Startcode emulation 対策は全てのネットワーク伝送において行わなければならない。

3. Profile

現在のところ、H.264 では Baseline, Main, Profile X の3つの Profile が定義されている。Baseline profile は B-picture を含まない低遅延が要求されるアプリケーション用の Profile であり、TV会議などに用いられることが想定されている。Main Profile は B-picture や算術符号化をサポートし、Broad cast や高画質なエンターテイメント用の Profile である。Profile X はエラー耐性のツールなどを含む、Streaming を主にターゲットとした Profile である。これまでの MPEG の規格では上位の Profile は下位のプロファイルを包含するようにオニオンリング状に定義されてきた。しかしながら、現在のところ、3つの Profile でサポートするツールが異なるため、オニオンリング状の Profile 構造とはなっておらず、各プロファイル間の互換はない。

Level は、現在5つ定義されている。QCIF, CIF, SDTV, HDTV, SHDTV 向けのレベルが定義されている。また、この Level の間の Sub Level (HHR, 720p 等)も定義されている。

4. 標準化動向

H.264 は ITU-T Q.6/SG16 (VCEG - Video Coding Experts Group) で、H.26L として標準化が開始された。1999 年の 8 月には最初の Test Model である TML1 が策定された。

2001 年に MPEG から "Call for Proposal for coding efficiency tool" の提案募集があり、H.26L および数社がこれに応募して、正式な主観評価を行った。主観評価の結果、ほぼ全てのテストで H.26L がトップの結果が得られた。この結果を受け、2001 年 12 月に次世代符号化方式の標準化を行う、 Joint Video Team が正式に設立され、ITU-T と MPEG が共同で標準化を行うことが決定された。ITU-T と MPEG は過去に MPEG-2 | H.262 の標準化を共同で行った経緯がある。その際は、成果となる標準は Common text となり、一字一句同一の規格となった。しかしながら、今回 JVT により標準化される技術は Common text とはならず、技術的内容が同一な Technically align な標準となる。

2002 年7月に FCD が発行され、10 月のジュネーブ会合で ITU-T で投票作業開始が予定されている。

2002 年の MPEG 淡路会合で MPEG 標準では FDIS となり、3月に IS となる予定である。

ITU-T で投票が開始されてから、MPEG の標準となるまでに時間差があり、この間に技術的に差が生じる可能性がある。10 月以降技術的変更が生じた場合、2 月に再度 Harmonization の会合を開催する予定である。

JVT は今回の H.264 version 1 だけのために時限付きで設立されたものであるが、既に version 2 を標準化しようという動きはある。しかしながら、version 2 を作るという正式な決定はまだ合意されていない。

5. まとめ

JVT では次世代符号化方式として既存技術の2倍の符号化効率改善を目指して標準化活動が行われてきた。この目標は現在のところほぼ達成されていると思われる。規格化された符号化方式は MPEG では MPEG-4 part 10 (Advanced Video Coding) ITU-T では H.264 として標準となる予定である。10 月のジュネーブ会合で、ITU-T では投票作業が開始され、MPEG では 12 月に DIS となる予定である。

文　献

[1] MPEG-2 Video, ISO/IEC IS 13818-2
[2] MPEG-4 Video, ISO/IEC IS 14496-2
[3] H.263, ITU-T Rec. H.263