

H.26L における非対称マクロブロック分割の効果の検証と実装方法の 一検討

木全 英明 北原 正樹 八島 由幸

日本電信電話株式会社 NTT サイバースペース研究所 〒239-0847 横須賀市光の丘 1-1

E-mail: kimata.hideaki@kitahara.masaki, yashima.yoshiyuki@lab.ntt.co.jp

あらまし 映像符号化方式 H.26L ではマクロブロックを 2 分割または 4 分割に等分割する分割パターンを定義している。これらの領域毎に動き補償や参照画像の選択を行うことができるため、分割数が多いほど予測効率が良いが、分割領域毎に動き情報等を符号化する必要がある、低ビットレートでは逆に冗長になる場合がある。この冗長度を減らす目的で、縦 16 横 4 画素や縦 16 横 12 画素といった非対称の 2 分割パターンを定義する方法が提案されている。本稿ではこれらの非対称分割パターンの効果を検証し、実装方法の提案を行う。

キーワード 映像符号化, H.26L, JVT, 分割パターン, MPEG-4 part 10, H.264

A Study on Asymmetric MB Partitioning with regard to its efficiency and syntax implementation on H.26L

Hideaki Kimata Masaki Kitahara and Yoshiyuki Yashima

NTT Cyber Space Laboratories, NTT Corporation 1-1 Hikari-no-oka, Yokosuka, 239-0847 Japan

E-mail: kimata.hideaki@kitahara.masaki, yashima.yoshiyuki@lab.ntt.co.jp

Abstract In H.26L, the segmentation pattern is defined where a macroblock is divided equally into 2 or 4 rectangular blocks. Since motion vectors and reference pictures are selected per a block, prediction efficiency is improved more when the number of blocks is increased. However in low bitrate, such many motion vectors and many indices representing a reference picture could be redundant. For the purpose of reducing this redundancy, an additional partitioning method where a macroblock is divided into 2 asymmetric blocks such as 16x4 or 16x12. In this paper, efficiency of such an asymmetric partitioning is evaluated, and a new implementation method is proposed.

Keyword Video coding, H.26L, JVT, segmentation, MPEG-4 part 10, H.264

1. はじめに

映像符号化方式 H.26L (MPEG-4 part 10 / H.264) では縦横 16 画素のマクロブロックを 2 分割または 4 分割に等分割する分割パターンを定義している。4 分割する場合には、更にブロックを複数の小ブロックに分割する木構造となっている。H.263 と比較し、2 分割パターンは、縦横 16 画素のマクロブロック単位よりも動き補償の予測効率を上げ、4 分割パターンよりも動きベクトル等のオーバーヘッドを低減する効果がある。更にマクロブロックを 2 分割または 4 分割したブロック毎に参照画像の選択を行うことができる。従って分割数が多いほど予測効率が良くなる。

更に 2 分割における予測効率を向上する目的で、縦 16 横 4 画素や縦 16 横 12 画素といった非対称の 2 分割パターンを定義する方法が提案されている。確かに 2 分割パターン全体として予測効率を向上することが可能となるが、一方で分割パターンが多いと反ってオーバーヘッドが多くなる問題がある。

本稿ではこれら非対称 2 分割パターンの効果を検証し、4 分割パターンと共存する実装方法の提案を行う。

2. H.26L の分割パターン

まず H.26L の CD[1] に記載されているマクロブロック分割パターンとそのパターン情報のエントロピー符号化方法を概説する。CD では算術符号化する方法も定義されているが、ここでは可変長符号化方法について検討する。図 1 に示すように、マクロブロック全体として、分割しないモード (16x16 ブロック) と、2 分割するモードと、4 分割するモードが定義され、2 分割モードでは横 16 縦 8 画素のブロック (16x8 ブロック) に分割するモードと横 8 縦 16 画素のブロック (8x16 ブロック) に分割するモードが定義されている。これらの分割パターン情報は表 1 に示したテーブルを用いて符号化する。これにより指定された分割単位毎に、参照画像を選択することが可能である。

4分割する場合には、更に各 8x8 ブロック毎に、分割しないモード(8x8 ブロック)と、横 8 縦 4 画素のブロック(8x4 ブロック)に分割するモードと、横 4 縦 8 画素のブロック(4x8 ブロック)に分割するモード、横 4 縦 4 画素に分割するモードが定義されている。これらの分割パターン情報は表 2 に示したテーブルを用いて符号化する。

3. 非対称分割パターン

分割数を増やすことにより、予測効率を上げることができる。しなしながら、分割されたブロック毎に動きベクトルや参照画像指定情報を符号化するために必要なオーバーヘッドが、予測効率の向上による予測誤差の低減効果よりも影響が大きい場合もあり、この場合にはマクロブロック当りのトータル符号量が増加する。

2分割パターンは、この問題の一つの解法であるが、CD に定義されたように等分割された領域の形状と、実際の画像内容とは異なる場合がある。このような場合には4分割パターンで分割することになる。

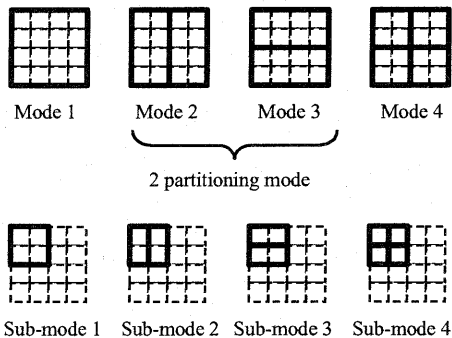


Fig.1. Illustration of MB partitioning of H.26L

Table1. VLC codes for MB partitioning

MB mode	Code
Mode 1	1
Mode 2	010
Mode 3	011
Mode 4	00100
Intra4x4	00101
Intra16x16, ...	00110,

Table2. VLC codes for 8x8 block partitioning

Sub mode	Code
Sub-mode 1	1
Sub-mode 2	010
Sub-mode 3	011
Sub-mode 4	00100
Intra4x4	00101

そこで2分割パターンを等分割ではなく、それぞれ非対称のブロックに分割する方法が提案されている[2]。まず提案されている非対称分割パターンの紹介をし、簡単な実験結果を踏まえて、その性能を評価する。

3.1. 非対称分割パターンの例

関口らが提案している非分割パターンは図 2 のようなものである。ここで、彼らが提案した際の比較対象は図 3 のようなもの[3]であったため、CD に記載されている分割パターンとは異なる。2分割されたブロック毎に、参照画像を選択することができる。

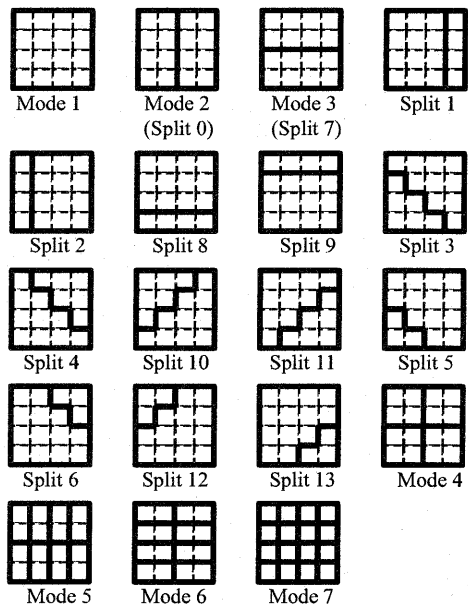


Fig.2. Illustration of previously proposed asymmetric MB partitioning of H.26L

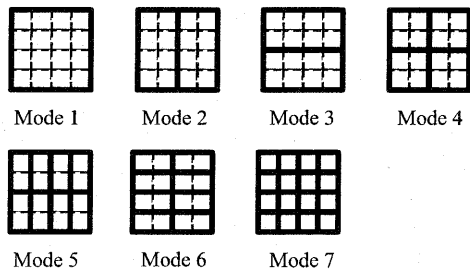


Fig.3. Illustration of previously defined MB partitioning of H.26L

3.2. 非対称分割パターンの効果

計算機シミュレーションにより、図2に示す非対称分割パターンの効果を確認する。4分割パターンとの共存を前提として評価を行うため、表3に記載の符号化モードテーブルを使用して符号化してみる。ここで表3-1で”Split Type 1”または”Split Type 2”となる場合には、更に表3-2で示した Split を符号化する。表3-1で”Split Type 1”となる場合には Split として0から6までの値を選択し、表3-1で”Split Type 2”となる場合には表3-2では Split として7から13までの値を選択する。表4に実験条件を示す。参照画像数は1とした。符号化モードの決定には式(1)に示すコストを評価に使用する。このコスト J が最も小さくなるように符号化モードを決定する。 R は符号量を示し、 $r(i, j)$ と $o(i, j)$ はそれぞれ復号画像と原画像を示す。

$$J = SSD + \lambda \times R$$

$$SSD = \sum (r(i, j) - o(i, j))^2 \quad (1)$$

図4に画像 Foreman の符号量削減率を示す。この図から、表3に記載の符号化モードテーブルでは、従来方法に比べて改善はほとんどみられず、逆にデグレードが見受けられる。Q=16の時の、分割パターンの選択数を図5に示す。この結果から従来の等分割ブロック(16x8, 8x16)の選択数が低下し、代わりに16x16ブロックと8x8ブロックの選択数が増加していることがわかる。また”Split Type 1”と”Split Type 2”では、従来の等分割ブロック(16x8, 8x16)と、縦方向または横方向に非対称に分割するブロック(16x4, 16x12,

Table3-1. VLC codes for MB mode of asymmetric partitioning

MB mode	Code
Mode 1	1
Split type 1 (Split 0 - 6)	010
Split type 2 (Split 7 - 13)	011
Mode 4	00100
Intra4x4	00101
Intra16x16, ...	00110,

Table3-2. VLC codes for sub mode of asymmetric partitioning

Sub mode	Code
Split 0 / 7	1
Split 1 / 8	010
Split 2 / 9	011
Split 3 / 10	00100
Split 4 / 11	00101
Split 5 / 12	00110
Split 6 / 13	00111

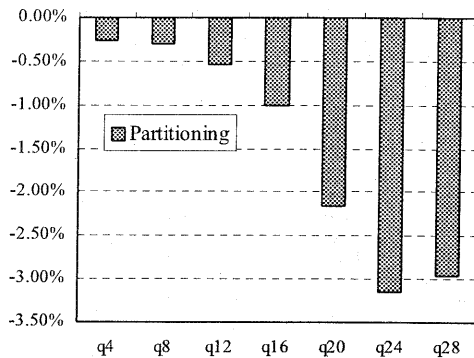


Fig.4. Bit saving of asymmetric partitioning defined by Table 3

Table4. Simulation Condition

Images	Foreman, Stefan, Mobile, Coat
Framerate	30fps
# of Reference	1, 2
Pictures	
MC accuracy	1/4 pel
Software	JM1.9

4x16, 12x16)とが殆どを閉めていることがわかる。

図4から、従来の16x8ブロックまたは8x16ブロックの符号によって2分割することを示す方法では符号化効率が低下してしまうことがわかったため、従来の16x8ブロックと8x16ブロックの符号を変更せずに、非対称2分割モードを指定する方法を検討することにする。

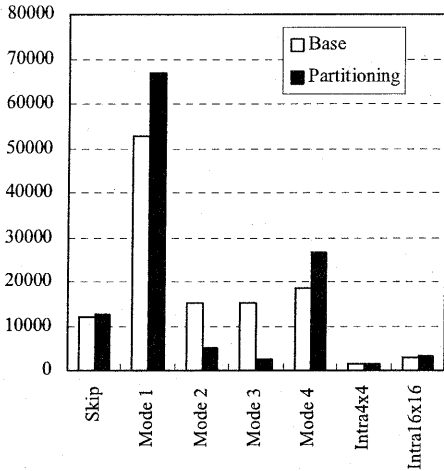
4. 非対称分割パターンの符号化モード提案

4.1. 直接統合する場合

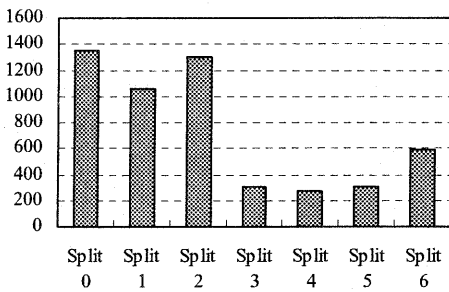
まず4つの非対称分割パターンを符号化モードに直接追加する方法(DirectInと呼ぶこととする)を提案する。UVLCの枠組みで構成した、この方法における符号化モードテーブルを表5に示す。2分割パターンを固めて定義できるため、テーブル構成は自然な格好となるものの、8x8ブロック(4分割モード)の符号長が長くなる欠点がある。

4.2. 間接定義する場合

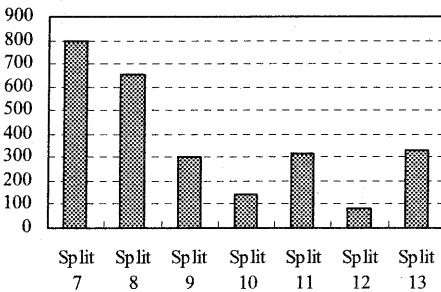
次に4つの非対称分割パターンを別の補助的な符号化モードとして定義する方法(SplitOutと呼ぶこととする)を提案する。UVLCの枠組みで構成した、この方法における符号化モードテーブルを表6に示す。ここで表6-1で”Split”となる場合には、更に表6-2で示した Split を符号化する。非対称分割パターンを



(a) MB mode



(b) Split Type 1



(c) Split Type 2

Fig.5. Number of selected MB mode defined by Table 3

別に定義するため、8x8ブロック(4分割モード)の符号長は従来と同じである。しかし、非対称分割パターンの符号長がDirectInよりも長くなる欠点がある。

Table5. VLC codes for MB mode of DirectIn

MB mode	Code
Mode 1	1
Mode 2	010
Mode 3	011
Split 1	00100
Split 2	00101
Split 8	00110
Split 9	00111
Mode 4	0001000
Intra4x4	0001001
Intra16x16, ...	0001010,

Table6-1. VLC codes for MB mode of SplitOut

MB mode	Code
Mode 1	1
Mode 2	010
Mode 3	011
Mode 4	00100
Split	00101
Intra4x4	00110
Intra16x16, ...	00111,

Table6-2. VLC codes for sub mode of SplitOut

Sub mode	Code
Split 1	00
Split 2	01
Split 8	10
Split 9	11

5. 実験と考察

提案した DirectIn と SplitOut の符号化効率を評価した。参照画像数 1 の場合について、図 6 に画像 Foreman における符号量削減率を、図 7 に画像 Stefan における符号量削減率を示す。平均 PSNR はどちらの方法も、非分割パターンを用いない方法と同じであった。どちらの方法においても、いずれの Q においても非対称 2 分割パターンを使わない場合よりも符号化効率が高い。特に Q が大きい場合に符号化効率がよいことがわかる。この結果から、4 分割パターンにおけるオーバーヘッドを効率良く低減できていることがわかる。なお参照画像を 2 にした場合も同様な傾向を示した。

また概して Q が大きい場合には DirectIn の方が符号化効率が高く、Q が小さい場合には SplitOut の方が符号化効率が高いことがわかる。図 8 に画像 Foreman で Q=16 の時の、分割パターンの選択数を示す。DirectIn の方が非対称分割パターンの選択される数が多いことがわかる。逆に SplitOut の方が当分割パターンの選択される数が多いことがわかる。これらの原因は、非対称 2 分割パターンの符号長が DirectIn の方が SplitOut よりも短く、また SplitOut の方が 4 分割パターンの符号長が短いためであると考えられる。

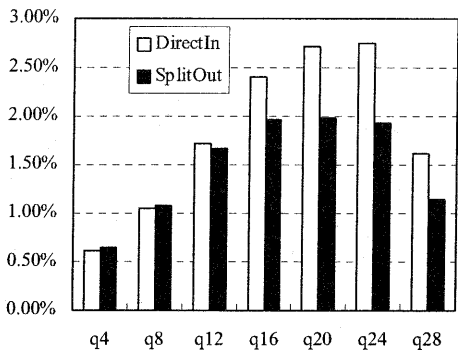


Fig. 6. Bit saving of DirectIn and SplitOut (Foreman)

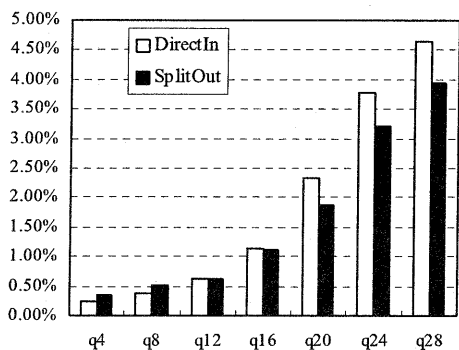


Fig. 7. Bit saving of DirectIn and SplitOut (Stefan)

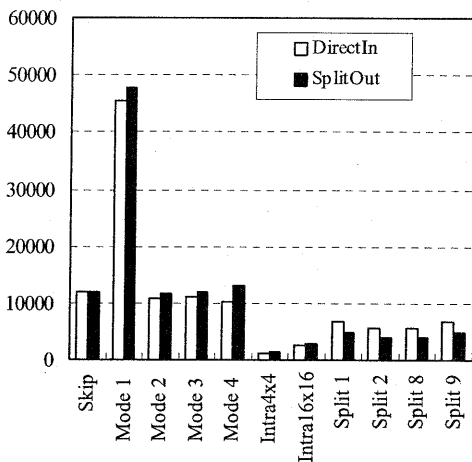


Fig. 8. Number of selected MB mode of DirectIn and SplitOut

- [3] G.Bjontegaard, "H.26L Test Model Long Term Number 8(TML-8) draft0", VCEG-N10, ITU-T SG16 Q.6 Meeting, 2001.

6. まとめ

縦方向または横方向に非対称 2 分割パターンを適用することによって符号化効率が向上することを確認した。Split 1,2,8,9 については直接 MB モードとして符号化する方が Q が大きい時の符号化効率が高いことを確認した。

今後は他の非対称 2 分割パターンも含めた場合の符号の構成について検討をすすめる。

文 献

- [1] "Joint Committee Draft (CD)," JVT-C167, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Meeting, 2002.
- [2] Shun-ichi Sekiguchi, Sadaatsu Kato, Satoru Adachi and Minoru Etoh, "Low-overhead INTER Prediction Modes," VCEG-N45, ITU-T SG16 Q.6 Meeting, 2001.