LJ-009

適応量子化による AVC/H.264 イントラフレームのフリッカ抑制
Adaptive Quantization Control for Reducing Flicker of AVC/H.264 Intra Frames

境田慎一 Shinichi Sakaida　**井口和久** Kazuhisa Iguchi　**合志清一** Seiichi Gohshi　**藤田欣裕** Yoshihiro Fujita

１．はじめに

AVC/H.264 方式は多くのツールの導入により MPEG-2 の約 2 倍の符号化効率を達成している．特にイントラフレーム符号化では，隣接画素からの画面内予測を行うため圧縮の効率が高い．しかし，イントラフレーム符号化ではこの画面内予測と粗い量子化に起因するフリッカが発生する問題が指摘されている[1]．文献[1]では，量子化・逆量子化処理を修正することでフリッカを抑制しているが，この方法は AVC/H.264 方式規格の範囲を超えている．本稿では，適応的な量子化制御とモード選択時のコスト関数変更により，規格の範囲内でフリッカを抑制する方法を提案する．

２．イントラ符号化におけるフリッカの原因

イントラ符号化における画面内予測と粗量子化によりフリッカが生じる原因について述べる．

2.1　画面内予測に起因するフリッカ

AVC/H.264 のイントラ符号化のデコード画像は次の式で表される．

$$R(O,P) = W(O-P) + P \qquad (1)$$

ここで，O は原画像，P は画面内予測画像を表す．W(O)は $T^{-1}Q^{-1}QT(O)$の略であり，T:直交変換，Q;量子化である．(1)式は，2 つのイントラフレームの同一位置にあるブロックの原画像がもし同一であっても，それぞれの予測画像 P_1,P_2が異なればデコード画像 $R(O,P_1)$と $R(O,P_2)$が異なることを示している．

2.2　量子化に起因するフリッカ

実際の自然動画像では，静止している物体を撮影しても連続するフレームで画像はわずかに変化している．そのため，フレーム間の僅かな差が量子化・逆量子化の処理により拡大されてフリッカとして検知されてしまうことがある．

図１に，フリッカが生じる例を示す．図１の横軸はフレーム(時間)，縦軸は信号レベルである．グラフは画面内のある 1 画素の，フレームに対する信号レベルの変化を表している．(a)は入力信号と予測信号の例である．(b)は(a)の残差信号である．入力信号が同じでも予測信号が異なれば残差信号が異なる．(b)を量子化・逆量子化した例が(c)である．量子化が粗い場合，残差信号の僅かな差が拡大される．逆量子化された信号に予測信号を加算してデコード信号(d)を得るが，(a)の入力信号にはないフリッカが生じている．

文献[1]の提案方法は，画面内予測の影響を除去するために，(1)式を(2)式のように変更している．

$$R(O,P) = W(O-W(P)) + W(P) = W(O) \qquad (2)$$

さらに，連続するフレーム間で微小な変化がある場合に，前のフレームと同じ量子化値を出力するように変更する．これによりフリッカの発生を抑えている．しかし，この方法は AVC/H.264 方式の規格を変えてしまうため，規格準拠のデコーダではデコードできない．

フリッカ妨害の現象は，イントラフレームが連続する場合だけでなく，チャンネルホッピングやランダムアクセスに対応するためにイントラフレームを周期的に挿入する場合でも生じる．特に低ビットレート符号化の際には，イントラフレームに続くインターフレーム中の多くのブロックがイントラフレームのブロックのコピーでデコード画像を再構成するスキップモードが使用される．このとき，図 2 に示すように静止部分のスキップモードのブロックはインターフレームが続く間は信号レベルの時間的な変化がないが，次に挿入されるイントラフレームでは前のイントラフレームとの間に上述した原因によるフリッカが生じるため，結果的に，挿入するイントラフレームの間隔でのフリッカ妨害として検知されてしまう．

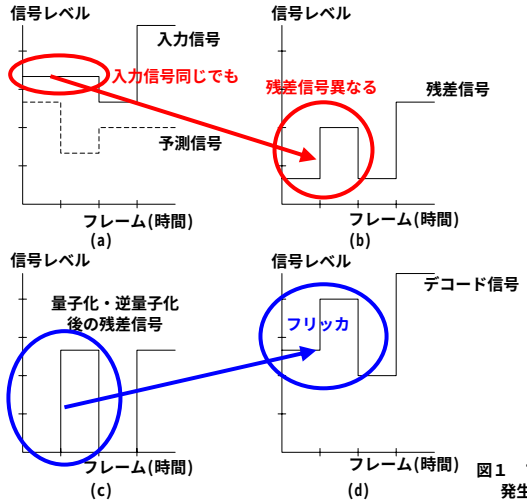

図１　フリッカの発生する例

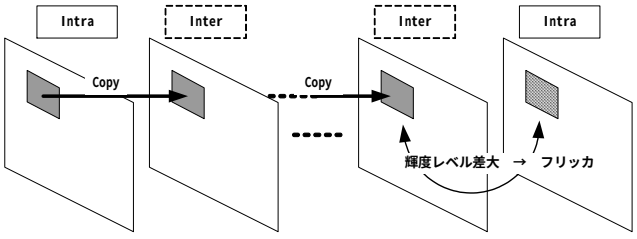

図2　イントラフレーム挿入間隔でのフリッカ

３．提案するフリッカ抑制方法

発生するフリッカを抑制するため，既に符号化済みの前のイントラフレームと現在のイントラフレームの静止部分のレベル差ができるだけ小さくなるように符号化処理を変更する．処理の流れを図 3 に示す．画面内予測モードおよびブロック分割モードに対しては後述するモード決定時のコスト関数を変更する．さらに，量子化パラメータを前のイントラフレームとの差がなるべく小さくなるように制御することでフリッカを抑制する．

3.1 処理対象マクロブロックの決定

提案手法を適用するマクロブロックを以下の手順で決定する．

日本放送協会,NHK

すべてのフレームをイントラ符号化する場合と，周期的に挿入するイントラフレームを処理の対象とし，インターフレーム内のイントラブロックは対象としない．

既に符号化済みの前イントラフレームの原画像のマクロブロックと，符号化する現イントラフレームの原画像の同位置のマクロブロックの類似度が高い場合にのみ処理を施す．

類似度 r は次の式で定義する．

$$r = SAD\,(O(i,j,t) - O(i,j,t-T)) \quad (3)$$

ここで，i,j はフレーム内のマクロブロック位置，O(・)は原画像を表す．t はフレーム(時間)を表し，T はイントラフレーム挿入の周期を表す．SAD は差の絶対値和である．

類似度 r が閾値以下の場合，該当位置のマクロブロックは前のイントラフレームのマクロブロックと類似度が高い静止部分と判断して以降の処理を行う．

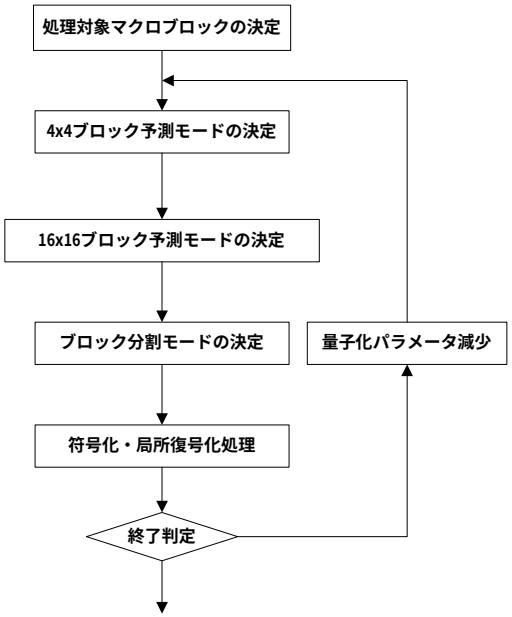

図3　提案手法のフロー

3.2 画面内予測モードのコスト関数変更

AVC/H.264 のイントラ符号化は，マクロブロックを 4x4 画素ブロックあるいは 16x16 画素ブロックに分割して行われる．そして，輝度信号の場合 4x4 画素ブロックでは 9 通り，16x16 画素ブロックでは 4 通りの画面内予測のモードが定義されている．また，色差信号は 4 通りの予測モードが定義されている．AVC/H.264 の参照ソフトウェア JM[3]では，レート歪み最適化による予測モード決定が可能である．下記の式のコスト関数を予測モードごとに計算し，これが最小になる予測モードを選択するものである．

$$J_{4x4} = D_{4x4} + \lambda R_{4x4} \quad (4)$$

ここで D_{4x4} は歪み，R_{4x4} はそのモードを選択した場合に必要となる情報量を表し，λ はラグランジュ係数である．参照ソフトウェアでは，4x4 ブロックの予測モード決定にこのコスト関数が使われている．16x16 ブロックについては(5)式に示すように歪み D_{16x16} のみをコスト関数として予測モードを決定する．

$$J_{16x16} = D_{16x16} \quad (5)$$

本提案では，このコスト関数に対して既に符号化済みの前イントラフレームのマクロブロックと現マクロブロックとの差を表す項 E および F を下記に示すように追加する．これら E,F を追加したコスト関数を最小化することにより，前の符号化済みイントラフレームの同位置マクロブロックとの差ができるだけ小さくなるように予測モードを選択できるようになる．なお，α, β は重み付けのための係数である．

4x4 ブロックの場合

$$J_{4x4} = D_{4x4} + \lambda R_{4x4} + \alpha E \quad (6)$$

$$E = SAD\,(L_{4x4}(i,j,t) - L_{4x4}(i,j,t-T)) \quad (7)$$

16x16 ブロックの場合

$$J_{16x16} = D_{16x16} + \beta F \quad (8)$$

$$F = SAD\,(L_{16x16}(i,j,t) - L_{16x16}(i,j,t-T)) \quad (9)$$

ここで，L_{4x4}(・)は 4x4 ブロックのローカルデコード画像，L_{16x16}(・)は 16x16 ブロックのローカルデコード画像である．現 4x4 ブロックおよび現 16x16 ブロックのローカルデコード画像は，予測モードごとに(1)式のデコード画像生成の処理を施して計算する．4x4 ブロック,16x16 ブロックそれぞれについて(6)式，(8)式の最小化で予測モードの候補を選択する．

3.3 ブロック分割モードのコスト関数変更

本提案では，予測モードの決定と同様に，マクロブロック分割モード決定時にも，(6)式や(8)式と同様に既に符号化済みのイントラフレームのマクロブロックと現マクロブロックとの差を表す項 G を γ を重み付け係数として加えてコストを最小化することにより，イントラフレーム間の同位置マクロブロックのローカルデコード画像の差ができるだけ小さくなるように分割モードを決定する．

$$J_{block} = D_{block} + \lambda R_{block} + \gamma G \quad (10)$$

$$G = SAD\,(L(i,j,t) - L(i,j,t-T)) \quad (11)$$

3.4 量子化パラメータの制御

以上のコスト関数変更によるフリッカ抑制と合わせて，粗い量子化の原因に対処するために量子化パラメータの適応的な制御を行う．3.2 および 3.3 節で述べた処理によって決定したモードで，現マクロブロックの符号化およびローカルデコード処理を行い，前の符号化済みイントラフレームのマクロブロックとの SAD を計算する．この値が閾値以下になっている場合にはこのマクロブロックの符号化処理を終了する．閾値以上の場合は，量子化パラメータを 1 減らして量子化を細かくし，再び 3.2 および 3.3 節のモード決定処理を行う．以上の処理を SAD が閾値以下になるまで繰り返すことで，粗量子化に起因するフリッカを抑制する量子化パラメータを決定する．

4．実験

提案手法のフリッカ抑制効果を確認するため，参照ソフトウェアに組み込んで実験を行った．実験条件は表 1 に示す通りである．

表 1　実験条件

符号化ソフトウェア	JM7.3
テスト画像	ITE システム評価用標準動画像 No.30 人混み(SIF サイズ)　(図 4)
フレーム数	450
Intra フレーム周期	15 フレーム間隔(B ピクチャなし)
エントロピー符号化	CABAC

フリッカを測定する客観量として，文献[2]で定義されているフリッカ量をベースにした(12)式を用いる．ここでは，

差の計算を処理対象マクロブロックの決定条件と合わせるために SAD に変更した．

$$S = \underset{i,j,t(SAD(O(i,j,t),O(i,j,t-T))<\varepsilon)}{AVG} SAD(R(i,j,t)-R(i,j,t-T), O(i,j,t)-O(i,j,t-T)) \quad (12)$$

AVG は平均値，R(・)はデコード画像を表す．ε は原画像のマクロブロック類似度を表す係数で，3.1 節で述べた処理対象マクロブロックと同一になるようにし，今回の実験では ε=10 とした．(12)式のフリッカ量 S が大きいほどフリッカの量が多いことを示す．

図 4 に示す今回用いたテスト画像は，画面上方の看板部分と，下方の道路の部分が平坦な静止領域であり，フリッカの妨害が現れやすい．図 5 は，テスト画像の第 15 フレーム目のイントラフレームにおいて，3.1 節で説明した処理対象マクロブロックを表示したものである．灰色のマクロブロックがコスト関数変更と適応量子化処理の対象であり，上方看板と下方道路の一部が含まれている．

図 4　テスト画像（人混み）

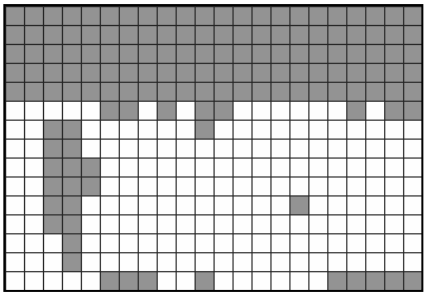
図 5　処理対象マクロブロック

4.1 コスト関数変更による効果

画面内予測モードとブロック分割モード決定のためのコスト関数を変更したことによる効果を調べた．コスト関数(5)式，(7)式，(9)式の重み係数 α，β，γ は予備実験の結果から，順に 10, 10, 500 とした．提案手法の比較対象として，JM7.3 を固定量子化で実行した結果を用いた．

4.2 量子化制御による効果

コスト関数の変更に加えて，量子化パラメータ値を制御することによる効果を確認した．符号化済みの前イントラフレームとの SAD の閾値は予備実験結果から 1 に設定した．

4.3 実験結果と考察

図 6 に実験結果のフリッカ量を PSNR を変数としてグラフ化したものを示す．コスト関数の変更により，PSNR に関わらず JM7.3 よりも常にフリッカ量が抑制されている．量子化パラメータの制御も伴うことにより，さらにフリッカ量を抑制可能である．

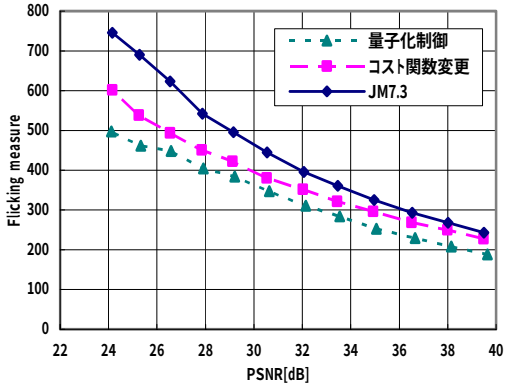

図 6　フリッカ量

次に図 7 にビットレートと PSNR を示す．コスト関数を変更する方法は，前イントラフレームとの誤差がなるべく小さくなるようなコスト定義をしているため，JM7.3 と比べると客観的な符号化効率は僅かながら低くなる．また，量子化パラメータの制御による手法は，マクロブロックによっては量子化パラメータが小さくなるため，ビットレートが JM7.3 より若干高めになるが，原画像との歪みは大きくならず PSNR 値が下がることはない．主観画質を比較すると，提案手法の効果によりフリッカの現れやすい静止領域のフリッカが抑制されていることが確認でき，トータルの画質としては良好であった．

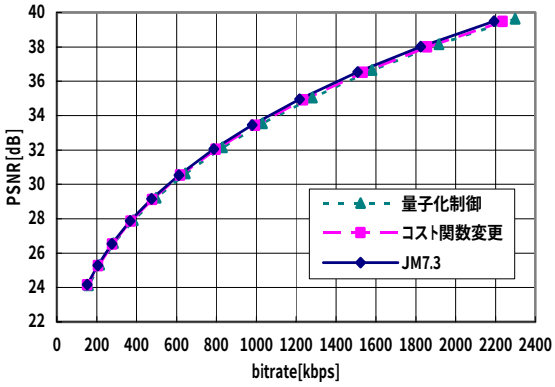

図 7　ビットレートと PSNR

5．まとめ

AVC/H.264 のイントラフレーム符号化におけるフリッカ妨害を抑制するために，イントラブロックの分割モード，予測モード決定のコスト関数を修正するとともに，前のイントラフレームとの差分を考慮して量子化パラメータを制御した．本提案により，AVC/H.264 符号化規格を変更することなく，周期的に目立ったフリッカを抑制することが可能になった．今後は低レートでのフリッカ抑制の向上のためにコスト関数の重み付け係数の最適化を検討する．

参考文献

[1] 井口他，”H.264 符号化におけるイントラモードのフリッカ低減方法”，FIT2003, J-040, 2003
[2] X. Fan et al., “Flicking Reduction in All Intra Frame Coding”, JVT-E070, 2002
[3] http://bs.hhi.de/~suehring/tml/