

コンシューマ・デバイス論文

コンシューマ用途向け超低遅延 H.264 符号化制御アルゴリズムおよびシステム

溝添 博樹^{1,2,a)} 岡田 光弘^{1,b)} 小味 弘典^{1,c)} 佐々本 学^{1,d)} 羽鳥 好律^{2,e)}

受付日 2012年9月14日, 採録日 2013年1月17日

概要: 民生用途やスモールビジネス用途を見据え比較的低ビットレートで使用可能な H.264 フル HD 低遅延符号化制御アルゴリズムの検討を行った. ピクチャ内の局所的な符号量変動を抑制することによって低遅延化を画質の維持を両立させる最大持ち越しビット量制御の具体的な制御手法を提案した. さらに, 提案アルゴリズムをすでに開発済みの多目的 H.264 コーデックプラットフォーム上に実装し, ビットレート 8~10 Mbps で最短 10 ms の遅延時間を実現し, 画質への影響が最小限であることを確認した.

キーワード: 低遅延, レート制御, バッファ遅延, 持ち越し, H.264

Very Low-delay H.264 Coding Control Algorithm and System for Consumer Applications

HIROKI MIZOSOE^{1,2,a)} MITSUHIRO OKADA^{1,b)} HIRONORI KOMI^{1,c)}
MANABU SASAMOTO^{1,d)} YOSHINORI HATORI^{2,e)}

Received: September 14, 2012, Accepted: January 17, 2013

Abstract: This paper presents a new coding control algorithm for very low-delay H.264 full HD video coding with relatively lower bitrate focusing especially on consumer or small business applications. We propose a concrete controlling scheme for bitrate control algorithm of keeping the maximum carry over, which enables both low-delay and little degradation of picture quality by suppressing local bitrate fluctuation within a picture. We made experiments of the proposed algorithm on our versatile H.264 codec platform, and obtained the result of 10 ms minimum delay at a bitrate of 8 to 10 Mbps with least impact on picture quality.

Keywords: low-delay, rate control, buffering delay, carryover, H.264

1. はじめに

動画圧縮規格 H.264 [1] は圧縮効率の高さから多くのアプリケーションで利用されている. 一方, ビデオ通話や映

像伝送などの用途においては低遅延画像圧縮に対するニーズが高まっている.

放送局設備における業務用システムなどのハイエンド用途向けにはすでに H.264 や MPEG2 の低遅延エンコーダが実用化されている. しかしながらシステム価格が 100 万円程度以上 [2] と民生用途に用いるには高価である. さらに利用するビットレートが 30~200 Mbps と比較的高い. 家庭用のインターネット回線は FTTH (Fiber To The Home) のシェアが 50% を超え主流になっているが [3], ベストエフォート型サービスのため, 実効速度は平均 20 数 Mbps 程度である [4]. そのため 30~200 Mbps を安定的に送受信するためには帯域保証型回線を導入する必要があるが, たとえば 100 Mbps の帯域保証型回線は月額料金が 30 万円程

¹ 株式会社日立製作所横浜研究所
Yokohama Research Lab., Hitachi, Ltd., Yokohama,
Kanagawa 244-0817, Japan

² 東京工業大学大学院総合理工学研究科
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, Yokohama, Kanagawa
226-8502, Japan

a) hiroki.mizosoe.hf@hitachi.com

b) mitsuhiro.okada.uf@hitachi.com

c) hironori.komi.cq@hitachi.com

d) manabu.sasamoto.wa@hitachi.com

e) hatori@ip.titech.ac.jp

表 1 目標仕様

Table 1 Target specifications.

項目	仕様
コーデック遅延	33 ms (1 フレーム)未満
映像フォーマット	1920x1080i, 60 field/s
圧縮フォーマット	H.264/AVC
ビットレート	8-10 Mbps

度以上 [5] と高価であり、この点からも業務用システムを民生用途で用いることは難しい。

本研究においては特に民生用途や小規模ビジネス用途向けに着目して、比較的低いビットレートで利用することが可能な新たな低遅延フル HD 対応コーデックを開発した。ビットレートを制御する手法を新たに開発し、以前の研究で開発した H.264 コーデックハードウェア上で実際に動作させ、効果を確認した。

2. 目標仕様

今回、民生用途や小規模ビジネス用途を考慮して、表 1 に示す目標仕様を定めた。

遅延時間については、映像と音声の同期を考慮し、映像の End-to-End の全体遅延を音声と同等にすることを目指した。音声の遅延は IP 電話の品質クラス分類 [6] で規定されているクラス A の遅延時間 100ms を想定した。

全体遅延の内訳として、ネットワーク回線、映像の入出力処理、および圧縮伸長処理の各遅延がある。ネットワーク回線遅延については、FTTH のパケット遅延に関する実測結果 [7] より、ネットワークにおける各遅延要因の平均値の合計である 35 ms と仮定した。撮像・表示などの映像入出力処理遅延については、各 1 フィールドずつと仮定し、計 33 ms とした。したがって、圧縮伸長処理は、全体遅延の目標値 100 ms からこれらの値を差し引いた残りである 33 ms (1 フレーム) 以下を目標遅延と定めた。

映像フォーマットは、高画質でありデジタル HDTV 放送の普及にともない幅広く利用されている 1,920x1,080i を用いることとした。圧縮フォーマットは、MPEG2 に比べ 2 倍以上の高効率圧縮が可能で、ビデオオンデマンドやワンセグ放送など幅広く使用されている H.264/AVC 方式を用いることとした。

前述のように FTTH の実効速度は平均 20 数 Mbps 程度であり、また TV 向け VOD サービスでは回線品質として 12 Mbps 程度を想定しているものが多い [8], [9]。これらを考慮し、ビットレートとしては若干の余裕を見込んで 8~10 Mbps 程度を目標とした。

3. 低遅延コーデック実現に必要な要因

3.1 ピクチャ符号化構造

一般的な画像圧縮アプリケーションにおいては、IBBPBBPBBPBB…というピクチャ符号化構造が用いら

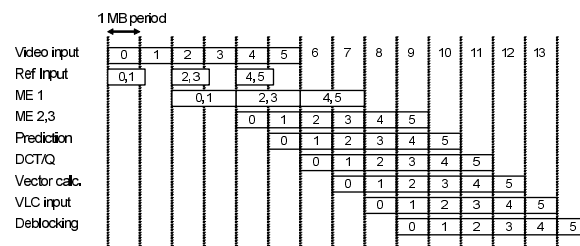


図 1 エンコーダパイプライン

Fig. 1 Encoder pipeline example.

れる場合が多い。ここに I, P, B はピクチャの符号化タイプを示しており、それぞれイントラ (Intra) 符号化 (符号が 1 枚の画像内部で完結)、予測 (Predictive) 符号化 (過去に符号化した画像を参照画像として利用し、その差分を符号化)、および双方向 (Bi-directional) 予測符号化 (過去および未来の画像を参照画像として利用) を表す。しかし、このうち B ピクチャは時間的に未来の画像を参照するため、B ピクチャを利用する場合は画像の並べ替え操作が必要となる。画像の並べ替えは遅延が増大する要因となる。たとえば冒頭にあげたように、よく用いられるピクチャ符号化構造として参照画像 (I または P ピクチャ) の間に B ピクチャが 2 枚ずつ挟まれる構造を例にとると、画像並べ替えにともなう遅延はエンコード、デコードとも各 3 フレームずつ、合計で 6 フレーム (200 ms) にもなる。このため、低遅延符号化アプリケーションにおいては B ピクチャを用いない IPPPP…のようなピクチャ符号化構造が利用される場合が多い。

3.2 処理遅延

処理遅延とは、エンコードやデコードなどの処理 (演算) そのものに要する時間のことである。たとえば映像のエンコードに際して以下のような処理が必要である。動きベクトル探索、直交変換、量子化、エントロピー符号化、デブロッキング処理などである。しかし、このような各処理を実行するのに必要な処理遅延は実はほとんど無視できるほど小さい。とりわけハードウェアで実装され、適切なパイプライン処理構造で設計されている場合にあってはまる。その理由について以下に説明する。図 1 にエンコーダのパイプラインの一例を示す。縦のスロット 1 つ分は 1 マクロブロック (MB) 期間を表している。1 つの MB は 16x16 画素の画像であり、1 枚のフル HD 画像 (解像度 1,920x1,080) の場合、8,160 個の MB から構成されている。映像のフレームレートが 30 frame/s の場合、1 フレームは 33 ms となるため、1 MB 期間は約 4 マイクロ秒となる。図 1 の場合、パイプライン処理の段数の合計は 10 MB であり、時間に換算すると約 40 マイクロ秒となる。この値は、ミリ秒単位である目標の遅延値に比べ数百分の 1 であり、したがって処理遅延はほとんど無視できるほど小さいといえる。もちろんこれがあてはまるのは適切に設計されている場合のみ

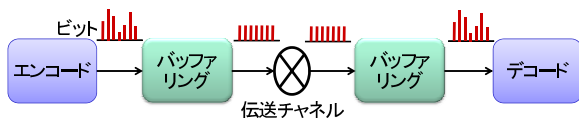


図 2 バッファによるビットレート平滑化

Fig. 2 Bit fluctuation smoothing buffer.

である。なお図 1 にあげた例は、筆者らの開発した H.264 コーデック [10] からとったものである。

3.3 バッファ遅延

映像を符号化するにあたり、1 枚のピクチャを符号化するのに必要なビット数はピクチャごとに異なっている。その理由としては、映像のシーンや画像ごとに絵柄の複雑さが異なること、ピクチャ符号化タイプのようなパラメータがピクチャごとに異なること、エントロピー符号化の統計的性質から来る要因などがある。最後のエントロピー符号化とは、頻繁に登場する情報ほどより短いビット長の符号を割り当てるようにすることで、情報を表現する効率を向上させる手法である。

一方、符号化されたビットストリームを送信する場合、伝送チャネルの容量帯域は有限であるのが通常である。そのため図 2 に示すようなバッファを挿入して、伝送チャネルにビットストリームを送出する前にビットレート変動を平滑化する必要がある。同様に、デコーダ側においても、各フレームのデコードを行う瞬間に必要な量のビットをデコーダにタイムリに供給するためにバッファが必要である。デコーダ側のバッファの大きさは少なくともエンコーダ側と同じ大きさが必要であり、結果としてバッファ遅延は全体合計で 2 倍となる。

図 3 に符号化映像シーケンスのビットレート変動の例を示す。横軸はフレーム番号、縦軸は各フレームごとのデータ量（バイト数）を示す。このシーケンスは B ピクチャを用いない IPPP…のピクチャ符号化構造であり、15 フレームごとに I ピクチャを挿入している。平均して I ピクチャは 109 KB、一方で P ピクチャは 25 KB のデータ量である。この差の 84 KB を均すためのエンコーダ側におけるバッファ遅延は、ビットレートを 8 Mbps と仮定した場合、84 ms となる。デコーダ側にも同じ大きさのバッファが必要なため、必要なバッファ遅延は合計で 168 ms 以上となる。低遅延性能の目標値である 1 フレーム (33 ms) 未満に比べ数倍の値である。

放送局設備などの業務用システムの場合は比較的高い伝送帯域が利用できるためビットレートの変動もある程度までは許容できる。しかしながら、民生用途向け低遅延コーデックの場合は利用できる伝送帯域が限られているため、このようなビットレートの変動が伝送遅延に与える影響はより深刻である。したがって、いかにしてバッファ遅延を抑えるかが特に民生用途向けシステムにおいて低遅延を実

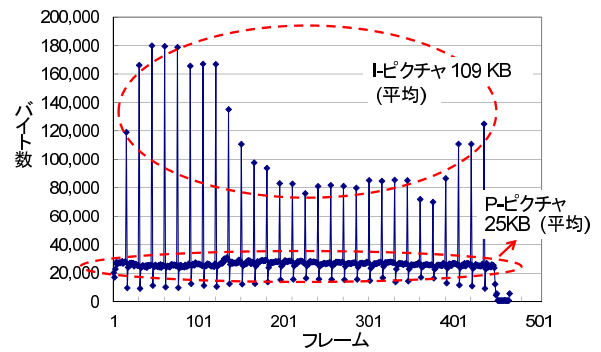


図 3 符号化映像シーケンスのビットレート変動

Fig. 3 Bitrate variation in a coded video sequence.

現するうえでの鍵となる。

4. ビットレート変動と画質

4.1 ピクチャ間の符号量変動

低遅延コーデックは実時間処理のアプリケーションで用いられる場合が多いため、伝送の間にビットエラーが発生することがある。ビットエラーが発生した場合、エラーのさらなる伝播や蓄積を防ぐために I ピクチャを用いることができる。I ピクチャは過去のピクチャの情報を参照しないため、エラーの影響で崩れた画像は I ピクチャ挿入によりリセットされる。定期的に I ピクチャを挿入することは、ビットストリームがエラーに対する耐性を持つのに有効である。

しかし、図 3 に示したように I ピクチャは通常 P ピクチャに比べ数倍のビット数を必要とする。したがって定期的に I ピクチャを挿入することは符号化ビットストリームにおいてビット数の大きな変動をもたらし、結果的にビットレートの変動につながる。

そこで I ピクチャ 1 枚分全体を定期的に挿入する代わりに用いられる手法の 1 つに、イントラスライスまたはイントラマクロブロック (MB) を用いて P ピクチャの一部をイントラモードとして符号化する手法 (イントラリフレッシュ) がある。これを図 4 に示す。イントラモードで符号化する部分は図の縦長の短冊上の領域 (イントラリフレッシュ領域) で表され、ピクチャごとに領域の位置を変化させる。エラーが発生した際、イントラリフレッシュ領域が移動することにより所定のフレーム数が経過するとエラーがクリアされ、シーケンスがエラーから復帰する。イントラリフレッシュを用いることにより定期的に I ピクチャ全体を挿入する場合に比べてピクチャ間のビットレートの変動を抑制することが可能である。

4.2 ピクチャ内の符号量変動

エンコードの結果により生成されるビットの量は 1 枚のピクチャの内部でも画像の局所的な複雑さの程度に応じて変動する。したがってピクチャ間で符号量の変動があるの

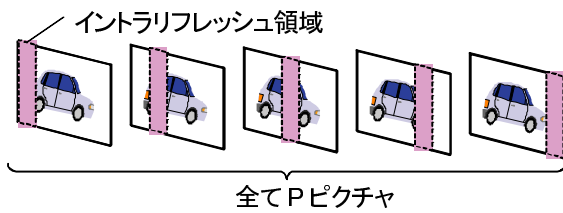


図 4 イントラリフレッシュ
Fig. 4 Refreshing intra macroblocks.

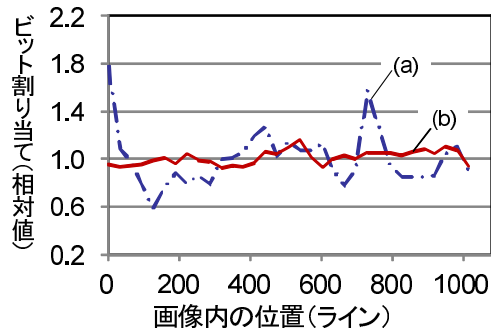


図 5 ピクチャ内の符号量割当て
Fig. 5 Bit allocation within a picture.

と同様ピクチャの内部にもビット割当ての変動が存在する。

上述のようにビットレート制御によってピクチャ間の符号量変動を小さくする手法は多く発表されているが、それらによる低遅延化は数フレーム、100 ms 程度にとどまっていた。

さらなる低遅延化の観点からは、上述のバッファ遅延を削減して遅延を最小化するために、ピクチャ内のビット割当てでもできるだけ均一化することが理想的である。筆者らはマクロブロックライン単位でビット割当てを制御することによってピクチャ内を平滑化し遅延量の削減を図った [11]。

しかしながら、単純なビット量の均一化を行うと画質の低下の原因になることがある。図 5 にピクチャ内の符号量割当ての例を示す。横軸は 1 枚の画像の中での位置 (ライン番号) を、縦軸は 2 MB ラインごとのビット量 (相対値) を示す。一点鎖線 (a) はすべて P ピクチャとしてイントラリフレッシュを用いてエンコードした場合を、実線 (b) は (a) の条件に加えピクチャ内での符号量割当てを単純に均一化する処理を追加した場合を示す。

図 6 にそれらに対応する各場合の画質を PSNR (ピーク Signal/Noise 比) で比較したものを示す。単純なビット量均一化の場合は約 1~2.5 dB の PSNR 値の低下が見られる。

5. 関連研究

低遅延符号化向けに、ビットレート制御によってピクチャ間の符号量変動を小さくする手法が多数発表されている [12], [13], [14]。これらは絵柄の複雑度に応じて各フ

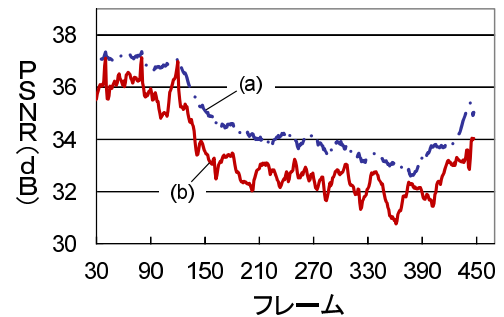


図 6 ビット量均一化の画質比較
Fig. 6 Picture quality comparison.

レームのビット割当て量を調整することにより、ピクチャ間の符号量変動を制御しようとするものである。しかしながら 1 フレーム分に相当するビット量もしくはそれ以上の大きさのバッファサイズをターゲットとしているため、エンコードとデコードを合わせると最低でも 2 フレーム以上遅延する。

そこで、バッファ遅延をさらに削減して 1 フレーム未満の低遅延を実現する試みとして、Lee ら [15] によるレート制御手法の提案がある。現在および過去フレームの絵柄の複雑度と MB ごとの複雑度の情報に基づいた最適なビット配分予測と、MB ロウ内の子実算比較による量子化パラメータ調整により、バッファサイズを 1/3 フレームまで削減した低遅延化を実現している。

しかしながら、上記は低遅延に特化したレート制御アルゴリズムになっている。一般に、レート制御は圧縮を高画質で行うための鍵となる重要な技術であるため、製品や技術の差別化のためアプリケーションに応じた様々な最適化や工夫を各社ごとに行っている場合が多い。しかしながら、仮に通常遅延のアプリケーション向けにそのような高画質ビットレート制御手法を構築していたとしても、上記はそれを活かして低遅延化させることができない。

また、上記低遅延手法はオール I フレームを対象にしており、全体的にビットレートが高くなってしまったという課題があった。

そこで本研究では、通常遅延用の高画質化レート制御手法がある場合に、それを活かしつつさらに 1 フレーム未満の低遅延化を実現すること、および I だけでなく P フレームや、イントラリフレッシュ付き P フレームに対しても適用可能であることを目的とした低遅延化手法を提案する。

6. 低遅延符号量制御手法

4 章で述べたように単純なビット量均一化は画質の低下の原因になることがある。したがって低遅延化のためにバッファ遅延を削減しようとする場合には、画質ができる限り保たれるよう注意を払う必要がある。

以下、6.1 節および 6.2 節で「持ち越しビット量の最大値の遵守」[16] による低遅延化手法について述べ、6.3 節

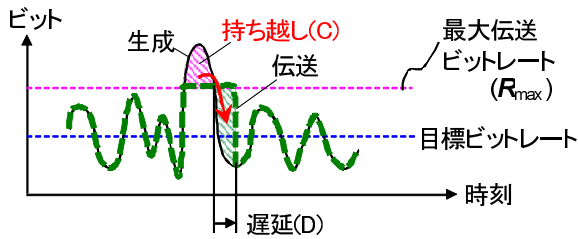


図 7 持ち越しビット量と遅延
Fig. 7 Carry over vs. delay.

で具体的な制御方法について説明する。

6.1 持ち越しビット量

まず「持ち越しビット量」について図 7 を用いて説明する。横軸はエンコードにともなう時間の経過（フレーム番号もしくはピクチャ内でのライン番号など）を、縦軸は単位時間あたりの符号量を示す。細い実線は、横軸の各時刻においてシーケンスの一部もしくはピクチャの一部をエンコードした結果、各時点で生成される符号量を示す。この生成符号量の平均値は図に示す目標ビットレートの値であると仮定する。しかしながら、3.3 節で述べた理由により局所的には変動が存在する。

生成されたビットは伝送チャンネルに送出される。もし局所的な生成量が最大伝送ビットレート以内であればビットはただちに送出される。しかし、もし生成量が最大伝送ビットレートを一時的に超えた場合はただちに送出されず、超過した部分は「持ち越しビット」としてバッファに蓄えられ、しばらく後に送出される。太い破線が実際に伝送される符号量を示す。持ち越しビットはすべてが送出されるまでの間バッファにとどまり、これがバッファ遅延となる。

なお、前述のように生成符号量の平均値は目標ビットレートに等しいと仮定しているため、持ち越しビット量が継続的に蓄積され続けてバッファが破綻するという状況は発生しない。

6.2 最大持ち越しビット量の遵守

エンコードの際の生成符号量の変動とその結果の持ち越しビット量はバッファ遅延をもたらす。しかしながら、逆に、持ち越しビット量がある限界量以内にとどまるよう注意深く制御すれば、持ち越しビットの存在をあえて許容してもよい。このようにして、ビット量の変動を意図的に容認し、画質ができるだけ保たれるようにすることとした。

持ち越しビット量の制御方法についてより詳細に説明する。図 7 において持ち越しビットにより生じる遅延 D [s] は次のように表される。

$$D = \frac{C}{R_{max}} \quad (1)$$

ここに D は遅延 [s]、 C は持ち越しビット量 [bit]、 R_{max} は

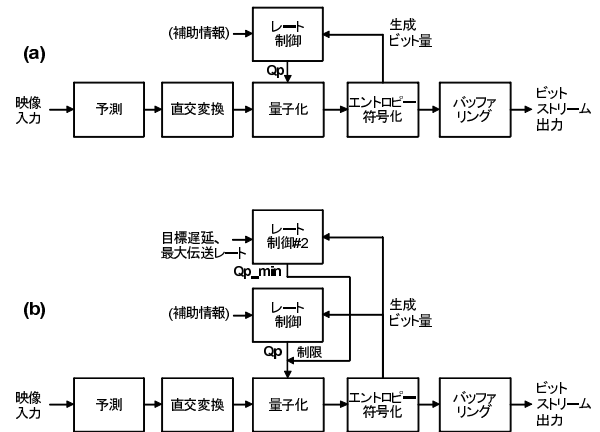


図 8 低遅延ビットレート制御手法

Fig. 8 Bitrate control method. (a) conventional, (b) proposed.

伝送チャンネルの最大伝送ビットレート [bps] である*1。最大持ち越しビット量 C_{max} は、目標の最大遅延 D_{max} [s] を用いて次のように表される。

$$C_{max} = D_{max} \cdot R_{max} \quad (2)$$

C_{max} は持ち越しビット量の最大許容値である。したがって持ち越しビット量 C が

$$C \leq C_{max} \quad (3)$$

をエンコードのビット生成の各時点において満たされるように制御を行う。持ち越しビット量 C の最大値を式 (3) に制限するが、その値未満の場合はできるだけ変動の発生を許容するように制御する。

6.3 持ち越しビット量制御手法

本研究では、エンコードの際に上述の持ち越しビット量 C の最大値を制限する具体的な方法について新たに提案する。以下、図 8 を用いて説明する。図 8 (a) は持ち越しビット量の制御を行わない場合の一般的なエンコードにおけるビットレート制御を示したものである。通常、量子化部、エントロピー符号化部、レート制御部の間のフィードバックループが存在する。符号化済みビット量に関する情報がレート制御に戻される。その後レート制御部は、その時点までに生成された符号量の情報や、必要に応じ符号化対象領域の画像の複雑さなどの補助的な情報に基づいて、量子化パラメータ Qp の値を決定する。量子化部はレート制御部から指示された量子化パラメータ Qp の値に基づいて量子化処理を行う。上記のレート制御部におけるレート制御のアルゴリズムは用途や要件などに応じた種々の方式が用いられている。

*1 持ち越しビット量 C による遅延自体は上述のように D 期間で消化されるが、その結果 D 期間本来の発生ビット量は D 期間以後に持ち越される。したがって実際には D およびその後続期間のビット量の総和が R_{max} のビットレートですべて消化されてはじめて持ち越しの影響が完了する。

図 8(b) に提案手法を示す。第 2 のレート制御部を追加で設け、持ち越しビット量の制御を行う。符号化済みのビット量に関する情報がエントロピー符号化部から供給される。さらに最大伝送ビットレート (R_{max})、目標遅延 (D_{max}) などの情報も供給される。第 2 のレート制御部はそれらの情報に基づいて量子化パラメータ Qp_min の値を求める。量子化パラメータ値 Qp_min は、持ち越しビット量が最大伝送ビットレートと目標遅延から定まる所定値を超えないようにするための Qp の最小値である。このようにして Qp の下限は Qp_min によって制限されるが、その他の場合は Qp の値は画質をできるだけ保つための変動が許容される。

本提案のように、第 1 のレート制御部の外側に第 2 のレート制御部を設けたことにより、第 1 のレート制御部のアルゴリズムにかかわらず、制御を行うことが可能になる。これにより、既存の様々なレート制御方式と組み合わせて、低遅延化のための制御を追加することが可能になる。

7. 実機検証

7.1 H.264 コーデック検討プラットフォーム

本研究においては筆者らの開発した H.264 コーデックプラットフォーム [10] を検討に利用した。本コーデックプラットフォームは主として民生用途向けの適用を考慮して開発したものである。図 9 にエンコーダ部のブロック図を示す。エンコードの各処理を行う専用のハードウェアブロックと、2つの制御用プロセッサ MPU および SPU (Main/Sub Processing Unit) からなっている。

本コーデックは図 1 に示す非常にフレキシブルなパイプライン構造を持っている。エンコードの各ステージで 1 マクロブロック (MB) ごとの処理を行う。パイプライン全体の動作は MPU により制御されている。SPU は、エントロピー符号化ブロック (ハードウェア) の制御と、ビットストリームの上位階層のヘッダの処理を行う。MPU、SPU のソフトウェアプログラムの交換によりエンコード動作の振舞いはフレキシブルに変更可能である。それにより本コーデックプラットフォームはアプリケーションごとの様々なニーズを満たすことが可能である。3.2 節で述べたように本コーデックプラットフォームを用いた場合、パイプライン自体の処理遅延は非常に小さく、目標遅延時間から考えると無視できるレベルである。

7.2 実装および検証

6 章で述べた低遅延符号化制御手法をコーデックプラットフォームの制御プロセッサに実装し、実機で検証を行った (図 10)。なお実装にあたっては符号化タイプはすべて P ピクチャとし、イントラリフレッシュを用いた。

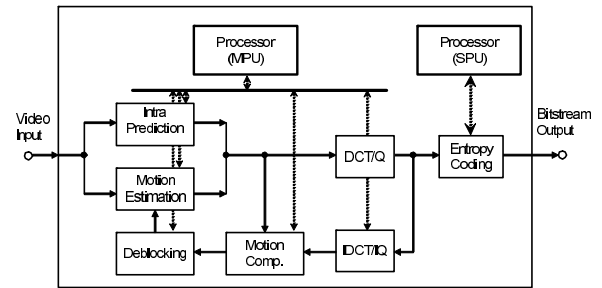


図 9 エンコーダのブロック図

Fig. 9 Block diagram of encoder core.

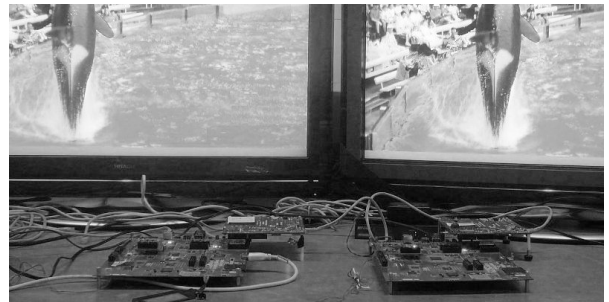


図 10 H.264 コーデックプラットフォームによる実機検証

Fig. 10 H.264 Codec platform testing the algorithm.

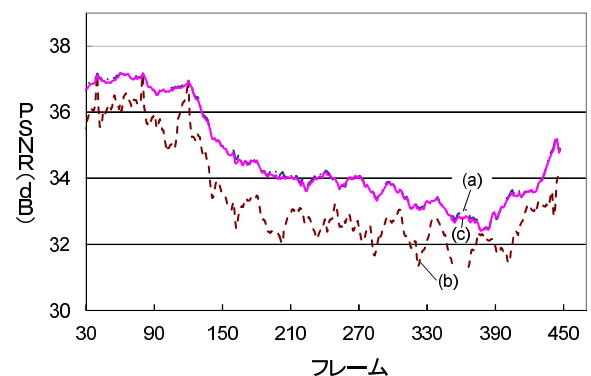


図 11 PSNR 比較結果

Fig. 11 PSNR comparison.

8. 実験結果

図 11 に実験結果の PSNR を示す。エンコードにはテストシーケンス (Square Garden) を用い、目標ビットレート 8 Mbps、最大伝送ビットレート 10 Mbps、目標 (最大) バッファ遅延 5 ms の条件で、以下の 3 種類のレート制御を比較した。(a) 従来のレート制御 (一点鎖線)、(b) 単純ビット量均一化 (破線)、(c) 提案手法 (実線) である。提案手法 (c) は、単純なビット量均一化 (b) のように画質が低下することなく、従来のレート制御 (a) とほぼ同じ画質を保っている。

表 2 に上記を含むその他のテストシーケンスの平均 PSNR を示す。同表において従来手法は持ち越しビット量制御のない通常の符号化を示す。いずれのシーケンスも提案手法適用による画質の低下はなく従来と同等の画質を

表 2 平均 PSNR 結果

Table 2 Average PSNR result.

シーケンス	目標ビットレート [Mbps]	最大伝送ビットレート [Mbps]	平均 PSNR [dB]	
			従来手法	提案手法
Square garden	8	10	34.7	34.5
Whale show	8	10	28.6	28.4
Ceremony	10	12	29.4	29.3

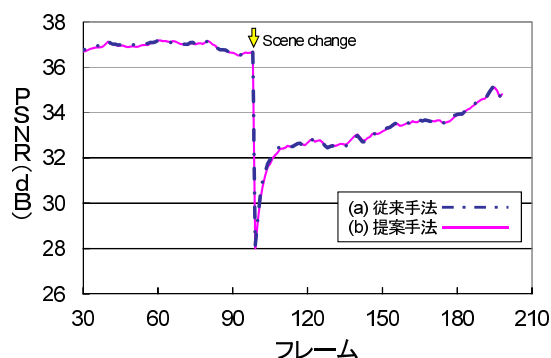


図 12 シーンチェンジ時の PSNR 比較結果 (Square Garden)

Fig. 12 Effect on PSNR at scene change (Square Garden).

保っている。

次にシーンチェンジの際の影響について検討した結果を説明する。テストシーケンス Square Garden から 100 フレームずつのシーンを時間的に離れた箇所から 2 つ抽出し結合した。これを用いた従来手法と提案手法の PSNR 比較結果を図 12 に示す。エンコードの条件は図 11 の場合と同様である。PSNR 値は両手法ともシーンチェンジの発生する箇所ですぐ大きく落ち込むが、両者の間の落ち込み度合いに大きな差は見られない。また同じシーケンスにおいて、持ち越しビット量によって生じる遅延を図 13 に示す。図 13 (a) は従来手法 (持ち越しビット量制御のない通常の符号化) であり、シーンチェンジの発生する箇所において鋭いピークが生じており、目標遅延を大きくオーバーしている。図 13 (b) の提案手法の場合は、シーンチェンジにおいてやはりピークは発生するものの、遅延の最大値は目標遅延である 5 ms 以内に収まっている。

同様の実験を異なるテストシーケンス Square Garden と Whale Show を結合したものに対しても行った。その PSNR 比較結果を図 14 に示す。やはりシーンチェンジの発生する箇所ですぐ両手法とも PSNR 値がいったん大きく落ち込むが、落ち込みの度合いに両者の間で大きな違いは見られない。なおシーンチェンジ以外の箇所でも本手法を適用した場合のほうが若干 PSNR が下がっているがこれは量子化パラメータ Qp_min 制御の影響で生成ビット量がわずかに低くなっているためである。また、本シーケンスにおいて、持ち越しビット量によって生じる遅延を図 15 に示す。図 15 (a) の従来手法では、シーンチェンジに際し鋭

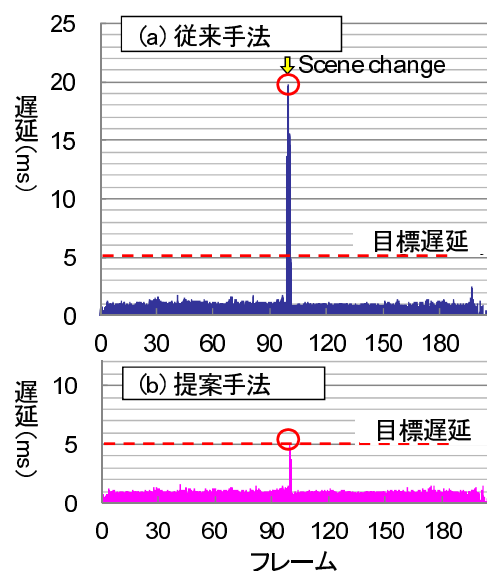


図 13 シーンチェンジ時の遅延比較結果 (Square Garden)

Fig. 13 Effect on the delay at scene change (Square Garden).

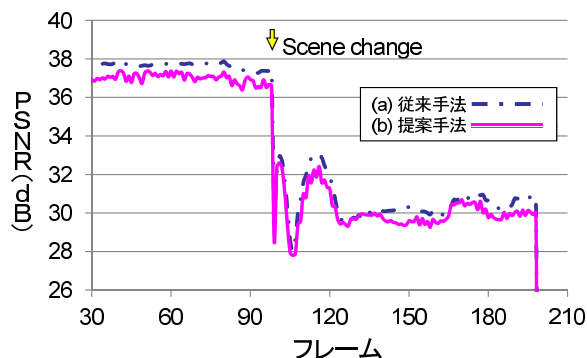


図 14 シーンチェンジ時の PSNR 比較結果 (Square-Whale)

Fig. 14 Effect on PSNR at scene change (Square-Whale).

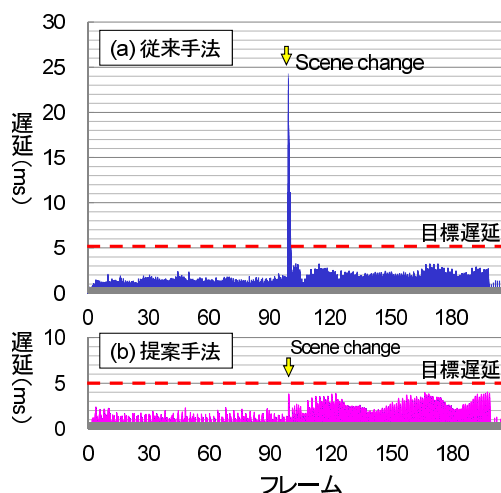


図 15 シーンチェンジ時の遅延比較結果 (Square-Whale)

Fig. 15 Effect on the delay at scene change (Square-Whale).

いピークが生じているが、図 15 (b) の提案手法の場合は、シーンチェンジにおけるピークは抑制され、遅延の最大値は目標遅延である 5 ms 以内に収まっていることを同様に

表 3 シーンチェンジ時の平均 PSNR 結果

Table 3 Average PSNR result with scene change.

シーケンス	目標ビットレート [Mbps]	最大伝送ビットレート [Mbps]	平均 PSNR [dB]	
			従来手法	提案手法
Square garden (100 フレーム× 2 シーン結合)	8	10	34.8	34.6
Square Garden + Whale show (各 100 フレームずつ)	10	12	33.8	33.3

表 4 実験結果

Table 4 Experimental results.

項目	結果
コーデック遅延*	10 ms (min.)
映像フォーマット	1920x1080i, 60 fields/s
圧縮フォーマット	H.264/AVC HP@L4.0
ビットレート	8-10 Mbps

*伝送路の遅延は含まない

確認した。

なお、上記の遅延量はエンコーダ内のバッファ遅延であるが、デコーダ側においても同じだけのバッファ遅延が生じる。そのため、エンコーダ側とデコーダ側を合計した全体の遅延は 10 ms 以内となる。

シーンチェンジ時の平均 PSNR 結果を表 3 にまとめる。また、以上の実験結果を表 4 にまとめる。

9. まとめ

民生用や小規模ビジネス用に着目し比較的低いビットレートで適用可能な低遅延フル HD コーデックを新たに開発した。低遅延実現にあたっての要因を分析し、なかでもバッファ遅延の削減に注目した。画質の劣化を最小限に抑えて目標遅延を実現するため、持ち越しビット量を最大値を抑える新たなビットレート制御アルゴリズムを開発した。本アルゴリズムを 7.1 節で述べた H.264 コーデックプラットフォームに実装し、実機を用いて検証した結果、8~10 Mbps で 10 ms の最小遅延を実現し、目標仕様を達成できることを確認した。

参考文献

- [1] ITU-T H.264, Advanced video coding for generic audio-visual services (Nov. 2007).
- [2] 超低遅延 MPEG-2 映像伝送システム HLD-2000, 入手先 (<http://www.ipros.jp/product/detail/2000015647/>) (参照 2012-11-30).
- [3] インターネット白書 2012, インプレスジャパン (2012).
- [4] 実積, “固定インターネット回線品質に対する消費者意識の分析: ネットワーク中立性議論との関連に着目して”, 情報通信学会誌, Vol.29, No.1, pp.19-28 (2011).
- [5] KDDI 国内イーサネット専用サービス, 入手先 (http://www.kddi.com/business/ethernet_senyo/charge.html) (参照 2012-11-30).

- [6] 総務省: IP ネットワーク技術に関する研究会 報告書, (Feb. 2002).
- [7] 吉田 薫, 藤井資子, 菊池 豊, 山本正晃, 永見健一, 中川郁夫, 江崎 浩: ユーザ視点に基づいたブロードバンドインターネット環境における遅延・パケットロスの傾向分析, 電子情報通信学会論文誌 B, J91-B(10), pp.1182-1192 (2008).
- [8] アクトビラ サービスガイド, 入手先 (<http://actvila.jp/guide/>) (参照 2012-11-30).
- [9] T's TV ご利用ガイド, 入手先 (<http://t-s.tv/public/pc/guide/>) (参照 2012-11-30).
- [10] Mizosoe, H., Yoshida, D. and Nakamura, T.: A Single Chip H.264 / AVC HDTV Encoder/Decoder/Transcoder System LSI, *IEEE Trans. Consumer Electronics*, Vol.53, No.2, pp.630-635 (2007).
- [11] Mizosoe, H., Okada, M., Komi, H., Sasamoto, M. and Hatori, Y.: Very low-delay H.264 codec for consumer applications, *IEEE International Conference on Consumer Electronics*, pp.65-66 (Jan. 2012).
- [12] Ribas-Corbera, J. and Lei, S.: Rate Control in DCT Video Coding for Low-Delay Communications, *IEEE Trans. Circuit and Systems for Video Tech.*, Vol.9, No.1, pp.172-185 (1999).
- [13] Jiang, M. and Ling, N.: Low-Delay Rate Control for Real-time H.264/AVC Video Coding, *IEEE Trans. Multimedia*, Vol.8, No.3, pp.467-477 (2006).
- [14] Liu, Y., Li, Z.G. and Soh, Y.C.: A Novel Rate Control Scheme for Low Delay Video Communication of H.264/AVC Standard, *IEEE Trans. Circuit and Systems for Video Tech.*, Vol.17, No.1, pp.68-78 (2007).
- [15] Lee, Y.G. and Song, B.C.: An Intra-Frame Rate Control Algorithm for Ultralow Delay H.264/Advanced Video Coding (AVC), *IEEE Trans. Circuit and Systems for Video Tech.*, Vol.19, No.5, pp.767-752 (2009).
- [16] 溝添博樹, 岡田光弘, 小味弘典, 佐々本学, 羽鳥好律: 民生用途向け超低遅延 H.264 コーデック, 映像情報メディア学会技術報告, Vol.36, No.7, pp.21-24 (2012).



溝添 博樹

1991 年京都大学理学部卒業。同年(株)日立製作所入社。1999 年ワシントン大学大学院修士課程修了。映像処理および符号化とその応用に関する研究に従事。映像情報メディア学会会員。



岡田 光弘

2004 年東京理科大学理工学部電気工学科卒業。2006 年同大学大学院修士課程修了。同年(株)日立製作所入社。画像の高効率符号化に関する研究に従事。



小味 弘典

1992 年大阪府立大学電気工学科卒業。
1994 年同大学大学院修士課程修了。
同年（株）日立製作所に入社。画像圧縮伸張技術の研究開発に従事。1999～2000 年南カリフォルニア大学客員研究員。映像情報メディア学会会員。



佐々本 学

1985 年大阪工業大学工学部電子工学科卒業。同年（株）日立製作所入社。デジタル映像応用システムの研究開発に従事。



羽鳥 好律 （正会員）

1971 年東京大学工学部卒業。1971 年国際電信電話（現、KDDI）入社。工学博士。2003 年より東京工業大学大学院教授。電子情報通信学会，映像情報メディア学会，画像電子学会フェロー，IEEE 会員。