Vol.2012-CDS-5 No.16
Vol.2012-DCC-2 No.16
2012/10/17

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)]

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い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す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既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 8〜10M bps 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 amd System for Consumer Applications

Mizosoe Hiroki[1,2,a)]　Okada Mitsuhiro[1,b)]　Komi Hironori[1,c)]　Sasamoto Manabu[1,d)]
Hatori Yoshinori[2,e)]

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

動画像圧縮規格 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等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既に低遅延 H.264 エンコーダーが実用化されているが，コストの点や利用するビットレートが 30〜200 Mbps と比較的高い点から，民生用にそれらのシステムを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エア上で実際に動作させ，効果を確かめた．

1 （株）日立製作所 横浜研究所
Yokohama Research Lab., Hitachi, Ltd., Yokohama 244–0817, Japan
2 東京工業大学 大学院総合理工学研究科
Tokyo Institute of Technology, Interdisciplinary Graduate School of Science and Engineering, Yokohama 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

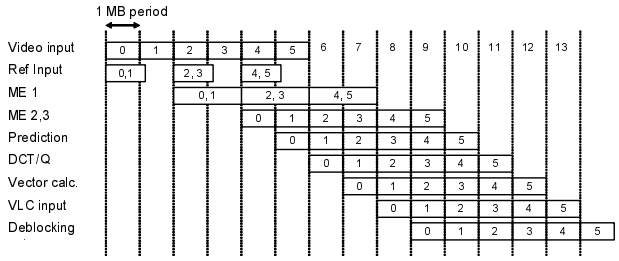

図 1　エンコーダパイプライン
Fig. 1　Encoder pipeline example.

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．目標遅延は 1 フレーム（33 ms）以下とした．低コスト化や民生用で利用可能なビットレートを考慮して, 8〜10 Mbps でフル HD 画質（1920 × 1080i, 60 フィールド／秒）を実現することを目標とした．

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．例えば冒頭に挙げたように，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取ると，画像並べ替えに伴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．例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等である．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．とりわけハードウエアで実装され，適切なパイプライン処理構造で設計されている場合に当てはまる．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 16 × 16 画素の画像であり，1 枚のフル HD 画像 (解像度 1920 × 1080) 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，1MB 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，目標の遅延値がミリ秒単位であることを考えるとはるかに小さい．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言える．無論これが当てはまるのは適切に設計されている場合のみである．なお図 1 に挙げた例は，筆者らの開発した H.264 コーデック [2] から取ったものである．

3.3 バッファ遅延

映像を符号化するに当たり，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伝送する場合，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ー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

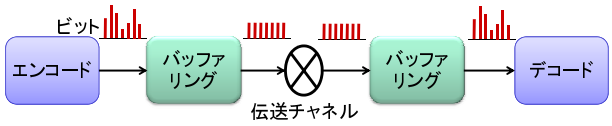

図 2 バッファによるビットレート平滑化
Fig. 2 Bit fluctuation smoothing buffer.

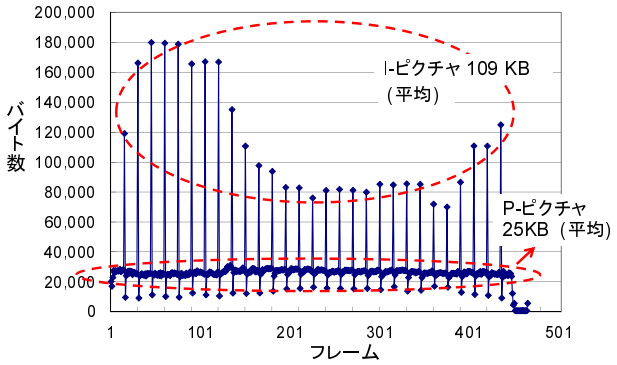

図 3 符号化映像シーケンスのビットレート変動
Fig. 3 Bitrate variation in a coded video sequence.

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．低遅延性能として 1 フレーム (33 ms) 未満を目標としていることを考えると，この値は大変大きい値である．

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はるかに限られているため，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現する上での鍵となる．

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，エラーの更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ピクチャに比べはるかに多く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．

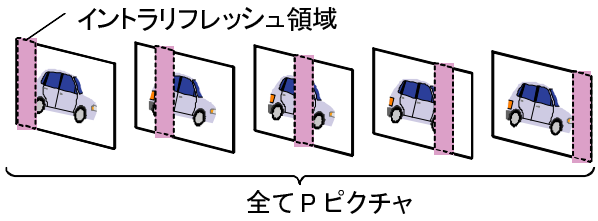

図 4 イントラリフレッシュ
Fig. 4 Refreshing Intra Macroblocks.

まるごとのIピクチャを定期的に挿入する代わりに用いられる手法の一つに，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ピクチャの一部分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かなり抑制することが可能である．

また，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（[3], [4], [5]）．

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と同様ピクチャの内部にもビット割り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 程度に留まっていた．

更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り当て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り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 [6]．

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

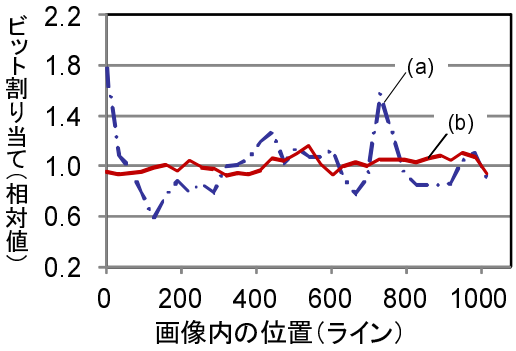

図 5 ピクチャ内の符号量割り当て
Fig. 5 Bit allocation within a picture.

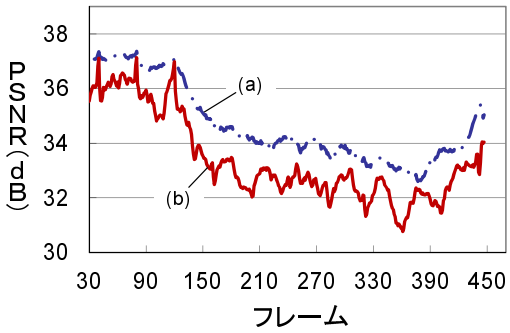

図 6 ビット量均一化の画質比較
Fig. 6 Picture quality comparison.

割り当ての例を示す．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り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. 低遅延符号量制御手法

5.1 持ち越しビット量

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．

バッファ遅延を削減して 1 フレーム未満の低遅延を実現する試みとして，Lee および Song[7] による手法の提案があるが，全て I ピクチャーだけで構成されたシーケンスを対象としており，全体のビットレートが高くなってしまうという課題があった．そのような制約なしに低遅延化を実現するため，筆者らは「持ち越しビット量の最大値の遵守」という手法によりアプローチしてきた [8]．これについて

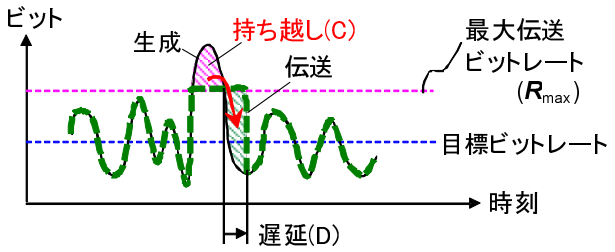

図 7 持ち越しビット量と遅延
Fig. 7 Carry over vs. Delay.

本節と次節で述べる．

まず最初に「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伴う時間の経過（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，縦軸は単位時間当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 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直ちに送出される．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直ちに送出されず，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留まり，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．

5.2 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，持ち越しビット量をある限界量以内に留まるよう注意深く制御すれば，持ち越しビットの存在を敢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．

$$D = \frac{C}{R_{max}} \tag{1}$$

ここに D は遅延 [s]，C は持ち越しビット量 [bit]，R_{max} は伝送チャネルの最大伝送ビットレート [bps] である．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．

$$C_{max} = D_{max} \cdot R_{max} \tag{2}$$

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

$$C \leq C_{max} \tag{3}$$

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．

5.3 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ダ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等に応じた種々の方式が用いられている．

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．

本手法の利点は，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関わらず，制御を行うことができる点である．これにより，既存の様々の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．

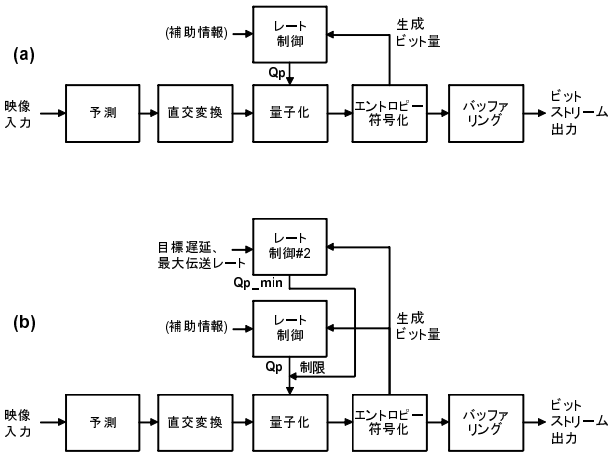

図 8 低遅延ビットレート制御手法
Fig. 8 Bitrate control method. (a) conventional, (b) proposed.

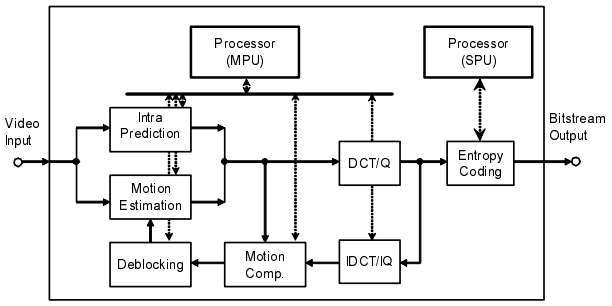

図 9 エンコーダのブロック図
Fig. 9 Block diagram of Encoder core.

6. 実機検証

6.1 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 [2] 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エアブロックと，2 つの制御用プロセッサ MPU および SPU（Main/Sub Processing Unit）から成っている．

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）毎の処理を行う．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（ハードウエア）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 のソフトウエアプログラムの交換によりエンコーダ動作の振る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毎の様々なニーズを満たすことが可能である．3.2 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

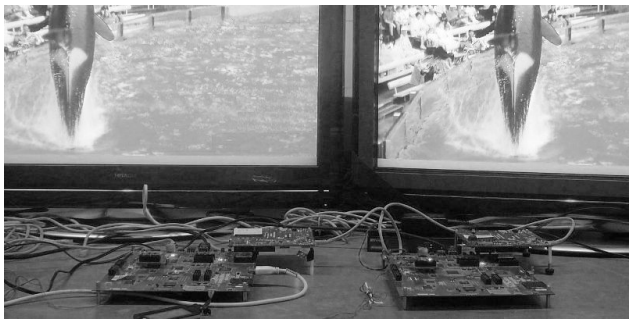

図 10 H.264 コーデックプラットフォームによる実機検証
Fig. 10 H.264 Codec platform testing the algorithm.

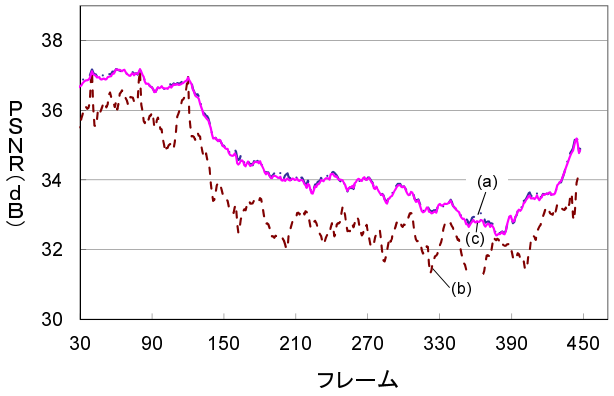

図 11 PSNR 比較結果
Fig. 11 PSNR comparison.

ると無視できるレベルである．

6.2 実装および検証

5 節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）．なお実装に当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．

7. 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．いずれも提案手法適用による画質の低下はほとんどなく従来とほぼ同じ画質を保っている．

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一旦大きく落ち込むが，両者の間に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無い通常の符号化）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．

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

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両手法とも PSNR 値が一旦大きく落ち込むが，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．

8. 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，中でも

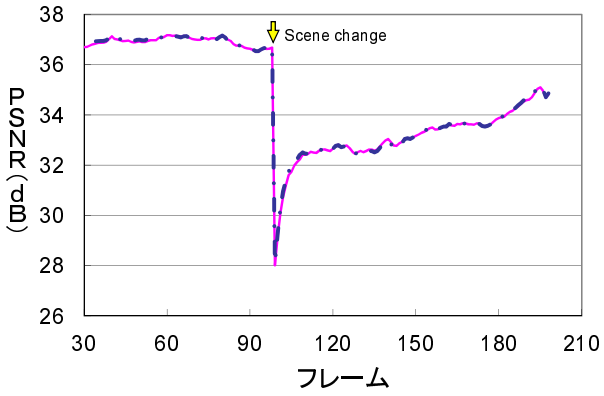

図 12 シーンチェンジ時の PSNR 比較結果 (Square Garden)
Fig. 12 Effect on PSNR at scene change. (Square Garden)

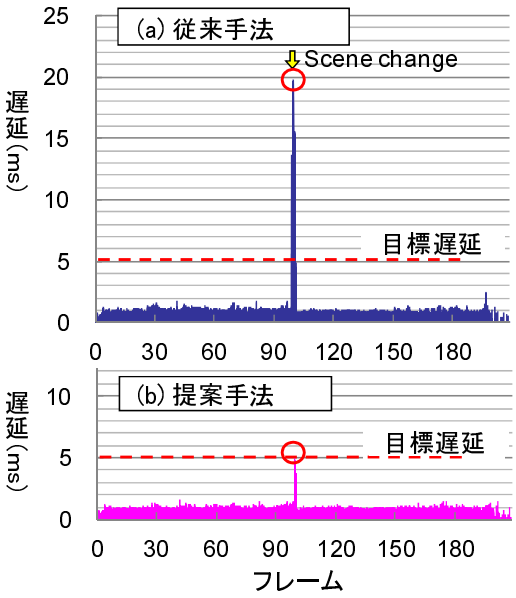

図 13 シーンチェンジ時の遅延比較結果 (Square Garden)
Fig. 13 Effect on the delay at scene change. (Square Garden)

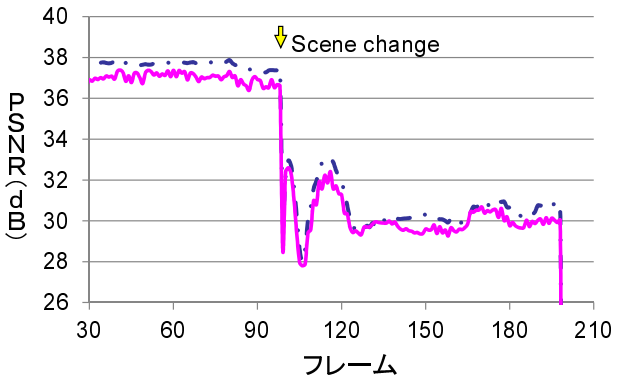

図 14 シーンチェンジ時の PSNR 比較結果 (Square–Whale)
Fig. 14 Effect on PSNR at scene change. (Square–Whale)

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 6.1 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．

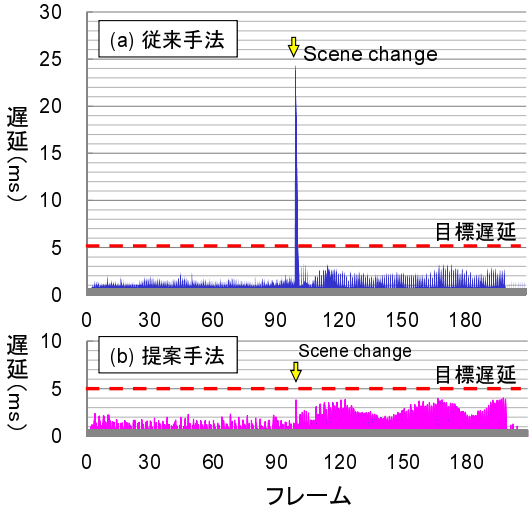

図 15 シーンチェンジ時の遅延比較結果 (Square–Whale)
Fig. 15 Effect on the delay at scene change. (Square–Whale)

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

参考文献

[1] ITU-T H.264: Advanced video coding for generic audio-visual services, Nov. 2007.
[2] H. Mizosoe, D. Yoshida, and T. Nakamura, “A Single Chip H.264 / AVC HDTV Encoder/Decoder/Transcoder System LSI,” *IEEE Trans. on Consumer Electronics*, Vol. 53, No. 2, pp.630-635, May 2007
[3] J. Ribas-Corbera and S. Lei, “Rate Control in DCT Video Coding for Low-Delay Communications,” *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 9,

No. 1, pp.172-185, Feb. 1999.
[4] M. Jiang and N. Ling, "Low-Delay Rate Control for Real-time H.264/AVC Video Coding," *IEEE Trans. on Multimedia*, Vol. 8, No. 3, pp.467-477, June 2006.
[5] Y. Liu, Z. G. Li, and Y. C. Soh, "A Novel Rate Control Scheme for Low Delay Video Communication of H.264/AVC Standard," *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 17, No. 1, pp.68-78, Jan. 2007.
[6] H. Mizosoe, M. Okada, H. Komi, M. Sasamoto and Y. Hatori, "Very low-delay H.264 codec for consumer applications," *IEEE International Conference on Consumer Electronics*, pp.65-66, Jan. 2012.
[7] Y. G. Lee and B. C. Song, "An Intra-Frame Rate Control Algorithm for Ultralow Delay H.264/Advanced Video Coding (AVC)," *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 19, No. 5, pp.767-752, May 2009.
[8] 溝添，岡田，小味，佐々本，羽鳥，"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.