

PC画面のビデオキャプチャによる圧縮の効率化 Compressed Efficiency of Improvement for PC-Screen Video Capture

進 隆治† 森 秀樹† 上原 稔†

Ryuji Shin Hideki Mori Minoru Uehara

1. はじめに

e-ラーニングや会議・論文発表などの映像は、講演者・発表者のみが動くだけでそれ以外の対象物にはほとんど動きがない。つまり背景は静止画もしくはそれに等しい動画とすることができる。よって被写体と背景を別々のものとして符号化を行えば処理効率が高まるが、現在の技術では被写体と背景を効率よく分離するには計算量が多く広く普及されるレベルにない。

2. 目的

図1のようにすでに複数の映像に分離されていると前提(対象物が特定の空間内しか動かない)した場合の効率について調べ、考察と検証を行う。

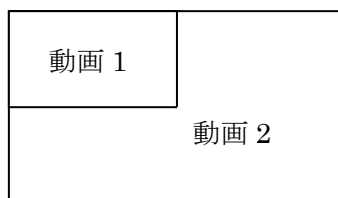


図1 分離した2つの空間

3. オブジェクト符号化

映像信号中の被写体や背景といった特定の意味を持つ構造単位ごとに信号を分離して符号化する技術。分離された信号を組み合わせると1つの映像とするためデジタルコンテンツの編集・加工が柔軟に行える[2]。

被写体の画像だけでなく輪郭情報も一緒に符号化されるため計算処理が多い。本研究では一般のパソコンで圧縮処理することを目的とするためオブジェクト符号化を使つての方式は採用しがたい。

4. GOP(Group of Picture)

GOPは時間軸上に並んでいるフレームをグループ化した構造。I・P・Bピクチャと呼ばれる3つのピクチャから構成される。

- ・I-ピクチャ ー現在の画像
- ・P-ピクチャ ーIピクチャから予測
- ・B-ピクチャ ーI・Pピクチャ双方のデータから予測

GOPは、画像を圧縮する際に前後の画像を参照し現在の画像を予測、その予測との誤差をデータとするため大きなデータの削減となる[1]。

GOP配列の長さは決まっておらずエンコーダーの設定によって決定される。基本設定では18フレーム、Iピクチャ1つにつきPピクチャ5個、Pの間にBピクチャ2個ずつ入る。

基本的なGOP配列は以下の通りである
「IBBPBBPBBPBBPBBPBB」

5. 動画像比較方法

画質の比較方法の1つとして同じ動画・同じ時間の1部分の画面を拡大して見比べてみる方法がある。比較する最も適した場面はIピクチャが挿入された部分である。IピクチャはP・Bピクチャよりも多くの情報量を必要とするからである。

図2に示されるように低ビットレートの文字は隣り合うビットの色が同じ、もしくは近い色になってぼやけて見える。



図2 100bit/sec(左)、500bit/sec(右)

6. ビットレートと GOP との関係調査

図 1 のような 2 つの異なる動画が表示されるサンプルで、1 つの動画とした場合と別々の動画として分離した場合とでそれぞれエンコードを行う。

サンプルとして Microsoft Producer の映像を動画として保存し比較を行った。

ビットレートは 100kbit/sec と 500kbit/sec また、GOP 構造を基本設定 (I=1,P=5,B=2) と (I=1,P=10,B=2)に変更したものを比較した。

7. 結果と考察

7.1 結果

(1)Producer の Power Point 部分

ビットレート・GOP 配列を変更して比較しても画質の差がない。

(2) Producer の目次部分

GOP 配列を長くすると最初の I-ピクチャのときは文字をはっきりしていたが時間が経つにつれぼやけてきた。特に 100kbit/sec のとき、この現象は顕著に出た。

(3) Producer の動画部分

100kbit/sec と 500kbit/sec との画質の差ははっきりと現れた。

GOP 配列を長くすると動き検出に失敗し時々画面の動きが止まってしまう現象がおきた。

・評価を以下に示す

(1)の部分は情報量を少なく、GOP 配列を長くしても正しい表示が可能。

(2)は文字なので、動きがないはずなのに動きが検出され I-ピクチャから次の I-ピクチャになるまで徐々に文字がぼやける現象が起き、GOP の特徴を確認できた。

(3)では予測した前後の画面が同じであると誤認識したため動きが止まる現象が起きた。

7.2 考察

(1)と(2)は当初は同じ文字であるため同じ処理をすればよいと考えていたが(2)では誤認識が起き、別々の処理が必要である。必ずしも GOP 配列を長くすればよいというわけではない。

誤認識を避けるため動き検出精度を低く設定し、動きの検出を行わないようにしたが文字のぼやけは大きくなった。動きを検出し予測に失敗したことになる。

つまり、画面のぶれを素直に再現するのではなく一定の範囲内の動きの検出を 0 にするという補正をかけて処理することが望ましい。

ただし、この処理を動画部分に適用するとコマ落ちした動きになることが予測される。そこで 8×8 ブロック、もしくは 16×16 ブロックごとの動き検出のベクトル移動データをもとに 1 つの動画を複数の動画に分離し、それぞれの動画に適切な設定値を代入することで解決される。

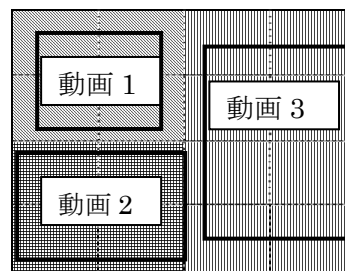


図 3 ブロックごとによる分離

8. まとめ

処理効率を上げるために GOP 配列を長くすればよいというわけではなかった。P・B-ピクチャの予測に用いられる動き検出にはぶれなどの余計な動きを読み込まない柔軟性が必要である。今後、画面のぶれ補正によるコマ落ちについて検討を行う。

参考文献

- [1]「MPEG 理論と実践」加藤 孝・鈴木 雅也 著、NTT 出版、2003 年
- [2]「動画像の高効率符号化」小野 定康・村上 篤道・浅井 光太郎 著、オーム社、2005 年