

画像データに音データを埋め込むことによる動画圧縮技術の検討

Study of video compression technology by embedding the sound data to the image data

○外山紘之^{*1}, 細野裕行^{*2}*Hiroyuki Toyama^{*1}, Hiroyuki Hosono^{*2}

Abstract: This study is purpose to improve the video compression rate. We performs processing for embedding audio data into image data, add this processing to the part of the algorithm of JPEG compression. In this paper, we confirmed that the compression rate of Motion-JPEG of 30 seconds can be improved further 3%.

1. まえがき

デジタルデータの大容量化にともない、圧縮技術は必要不可欠となっている。一般的な動画圧縮方式は、映像ファイルと音ファイルを別々に圧縮し、ひとつにまとめたものである。そのため、動画データを再生する際に音ズレが生じる可能性がある[1]。

映像ファイルは JPEG 圧縮が基本で構成されているものが多数存在し、その 1 つに MotionJPEG が挙げられる。そこで、本報告では MotionJPEG 動画データを用いて画像データに音データを埋め込み圧縮率を更に高める方法を提案する。これにより、音データ分の容量が削減され、音ズレの低減にも資する。また、ここでは音が付いている映像を動画と定義する。音が付いていないものはそのまま映像と呼ぶ。

2. 利用データ

[映像]

- ・解像度：1920×1080 pixel
- ・色：RGB256 段階
- ・フレームレート：29.97 fps
- ・再生時間：30 s
- ・ファイル形式：MotionJPEG

[音]

- ・サンプリング周波数：48,000 Hz
- ・量子化ビット：24 bit
- ・チャンネル数：2 ch
- ・ファイル形式：PCM

サンプリングされた音データの数の単位を個と定義する。

以上の条件より、フレームレートが約 30 fps、1 秒間の音データが 96,000 個であることから、1 フレームの画像に 3,200 個ずつ音データを埋め込む。

3. アルゴリズム

JPEG 画像に音データを埋め込むアルゴリズムを示す。また、圧縮時のカラーフォーマットは YCrCb の

4:1:1 を用いる。量子化テーブル、ハフマンテーブルは JPEG 規格に則って利用している。

[動画の圧縮]

- (1) 8 個ずつのデータを離散コサイン変換(DCT)する。
- (2) DCT 係数を全て 127 倍する。
- (3) 小数点以下を四捨五入する。
- (4) JPEG のエンコードをおこなう[2]。このとき、交流(AC)係数の色差成分は符号化しない。
- (5) AC 係数の色差成分の上端 1 行と下端 1 行の 1 列目から 100 列目までのブロックにおいて、ジグザグスキャン順の 3 番目から 10 番目に 8 個の音データを上書きしていく。これにより上端ブロックに 800 個、下端ブロックに 800 個のデータを埋め込むことができる。Cr, Cb 成分それぞれにおいて同処理をおこなう。
- (6) AC 係数の色差成分も符号化する。
- (7) (1)~(3)の処理を全フレームの画像でおこなう。

ハフマンテーブルの AC 係数の絶対値の最大値は 1024 である。また、音の DCT 係数の最大値は 8 であることから、音の DCT 係数が AC 係数の最大値を上回らないように量子化するために 127 倍している。

音を埋め込む場所は、再生した際に劣化に気づきにくい位置がよいと考えたため、上端と下端に埋め込む。

以上のアルゴリズムにより、映像と音をひとつに合成することで元々の音の容量分を削減できると考えられる。この処理には圧縮率を高める以外にも利点がある。従来の動画方式では、動画再生時に映像データと音データの時間を同期させて再生するものである。しかし、本アルゴリズムでは画像データに音データが埋まっていることから既に同期している状態と同じである。これにより、音ズレに強くなる。また、基本は JPEG コーデックを利用しているため大幅なプログラムの変更の必要がない。

4. 結果

まず、実際に埋め込んで抽出した音データの変化を見る。ここで、ビット精度は 24 bit であったが、埋め込む際に、前処理をおこなうためビット精度が 14 bit に下がる。表 1 に 2 つの音データの時間に対する振幅誤差の比較結果を示す。

Table 1. The error rate of sound

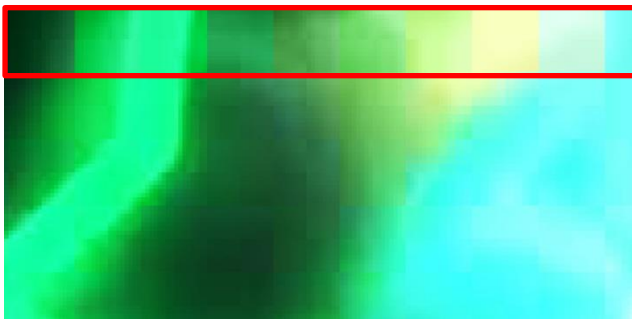
平均誤差[%]	最大誤差[%]
0.16	0.85

数値的には誤差 1 %未満であった。

次に本アルゴリズムを用いた画像を検討する。音データを抜き出した後の画素に従来手法との変化が見取れる。図 1 (a), (b)に処理後の画像と拡大画像を示す。



(a). Original picture of after processing



(b). The expansion picture of red frame of (a)

Fig1. The picture after processing

(b)の枠の部分が音を抜き出した後の画素ブロックの行である。他の領域と比べるとブロックノイズが顕著に出ている。(3')で音を抜き出した後に、0に修正したため高調波成分が減る。その結果、ブロック内の画素間の色差変化が少なくなったためブロックノイズが強調されたと考えられる。

最後に通常の JPEG 画像、前処理をした PCM 音を足し合わせた容量を基準として、本処理結果の圧縮率を求める。このとき JPEG の量子化テーブルの画質を 50 %とする。画質の 50 %とは JPEG 規格のテーブルであり、50 %より高画質の場合は、

$$\text{量子化テーブル値} \times (1 - \text{画質}) \quad (1)$$

であらわされる。逆に低画質にする場合は(1)式とは異なる式により計算するが、本稿では省略する。本アルゴリズムの条件は、再生方式がステレオであったが、この他にモノラルの場合もおこなった。音を埋め込む場所はモノラルの場合、上端 1 行のみに埋め込む。両方の再生方式のときの圧縮率をそれぞれ求める。本手法と従来の方式の比較を表 2 (a), 本手法で 2 種類の動画を処理した結果を(b)に示す。

Table 2. Compression result

(a). Capacity

再生方式	容量[kByte]	
	映像+音	本手法
モノラル	105,778+2,520	106,556
ステレオ	105,778+5,040	107,379

(b). Rate

再生方式	圧縮率[%]	
	Sample1	Sample2
モノラル	98.4	98.3
ステレオ	96.9	96.4

再生方式を変えると、モノラルのときよりステレオのときのほうが圧縮率は小さくなった。埋め込む音データの容量が増えたときに圧縮率が小さくなること示された。Sample1 の削減された数値は 3.1 %であり、元々の動画データに含まれる音データの比率はステレオで 4.6 %であった。これより、全ての音データ容量を削減することはできなかった。

5. まとめ

新しい動画圧縮方式を示した。動画によっては副音声がついている場合や多チャンネルのものも存在する。今回の結果から、音のデータが増えるほど圧縮率は小さくなると考えられる。

今後、埋め込み位置の違いによる圧縮率の変化、音データを増やした場合や動画形式を変更したときの人間の知覚面での劣化を検討する。そこで、劣化の定量的評価をおこなう予定である。

6. 参考文献

- [1] 亀山渉・金子格・渡辺裕：「そこが知りたい最新技術 オーディオ・ビデオ圧縮入門」, インプレス R&D, 2007 年
- [2] 越智宏・黒田英夫：「JPEG&MPEG 図解でわかる画像圧縮技術」, 日本実業出版社, 2003 年
- [3] 小野定康・鈴木純司：「わかりやすい JPEG/MPEG2 の技術」, オーム社, 2001 年