

動画並列エンコーダシステムにおける簡易圧縮の改善

中根 翔[†] 三浦 康之[†]

湘南工科大学[†]

1. はじめに

PC、デジタル家電の普及に伴い、PC が高性能化し、ソフトウェアによる動画の編集や圧縮が可能になり、個人レベルで動画を扱うことが多くなった。また、PC の普及に伴い、PC の低価格化により安価に並列処理が可能になっている。以上の背景から、動画の圧縮にかかる時間を削減し、動画の編集者の負担を減らすシステムの研究および開発を進めている [1][2]。それら一連の研究の一環として、LAN で伝送出来るようなデータの簡易圧縮法の検討を進めている [2]。現在検討中の手法は、単純差分と RLE、ハフマン符号化を組み合わせたもので、少ない CPU への負担で数分の一の圧縮が可能となる。本稿では、RLE データの画素成分と長さ成分に分けてそれぞれに対してハフマン符号化することにより、簡易圧縮の圧縮率の向上を図る。

2 並列ビデオエンコーダシステム

図 1 に、並列ビデオエンコーダシステムの構成を示す。本システムは、PC 端末から LAN を使いマルチコア PC に動画を送って圧縮を行い、PC 端末に戻すものである。このシステムは通信網に特別線を使わずに、それぞれのクラスタ、利用者の PC を Gigabit Ethernet で結合している。利用者は、クラスタマシンに符号化する動画を送り、クラスタマシンは、符号化処理が終了し次第、利用者の端末へ戻す。

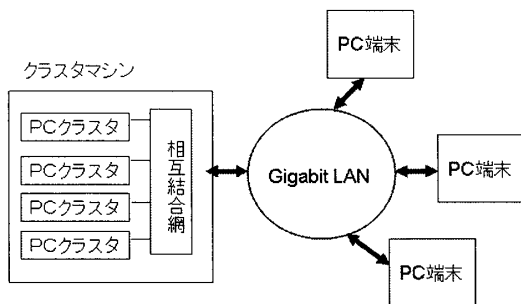


図1 並列ビデオエンコーダシステム

An Improvement of Simple Compression Method for Parallel Encoding Environment

[†] Sho Nakane, Yasuyuki Miura, Shonan Institute of Technology

3 簡易圧縮アルゴリズム

図 2 に、簡易圧縮アルゴリズムの流れを示す。最初に今の画像と次の画像の 2 枚の画像を YUV 形式に変換し単純差分で画素の差を取り、その値を、RLE (三種類の R L E) を使いその後ハフマン符号化を行う。

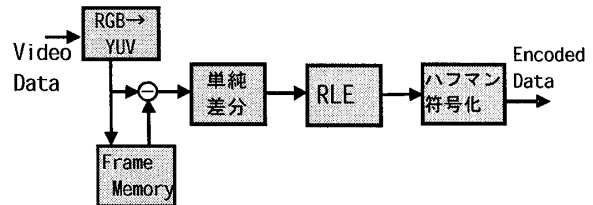


図2 簡易圧縮アルゴリズムの流れ

以下、各ブロックの詳細を説明する。

3.1 RGB→YUV 変換

YUV とは、輝度信号と青色成分の差(U)、輝度信号と赤色成分の差(V)の 3 つの情報で色を表す形式である。座標 (x, y) の各画素に対して、下式に従って処理を行った。

$$\begin{aligned} Y(x, y) &= 0.2990R(x, y) + 0.5870G(x, y) \\ &\quad + 0.1140B(x, y) \\ U(x, y) &= -0.1648R(x, y) - 0.3316G(x, y) \\ &\quad + 0.5000B(x, y) \\ V(x, y) &= 0.500R(x, y) - 0.4187G(x, y) \\ &\quad - 0.0813B(x, y) \end{aligned} \quad (1)$$

3.2 単純差分

単純差分は、時間的に隣接する画素間の相関が高いことを元に、情報量を削減する処理である。下式に基づき、着目フレームの座標 (x, y) における画素 $I(x, y)$ と、直前のキーフレームの画素 $I_b(x, y)$ との差分の画素 $I_d(x, y)$ を、以下の式により求める。

$$I_d(x, y) = I(x, y) - I_b(x, y) \quad (2)$$

このようにすると、情報量が少なくなり、差分した後の画素の値は近くなるためこの後の処理が容易になる。

3.3 RLE

RLE は、データ列において同じ値が連続する時に、連続するデータ長と値そのものの組み合わせを示すことによりデータ長を短縮する可逆圧縮である。

RLE には、データが連続していない部分では符号化後のデータが大きくなる欠点があるため、それを軽減するために今回は以下の三種類の RLE を使う。

3.3.1 SRLE

SRLE (Switched Run Length Encoding) は連続している部分と連続していない部分に分けて考える。まず値が連続している所まで探し、連続していない値の数を頭に出し、後ろに連続していない値をそのまま出力し連続している値の数を数えてその後ろに出力する。

3.3.2 Pack Bits

Pack Bits も値が連続している部分としない部分に分けて考える。連続している値の数を負数で出力しその後値を出力する。連続していない値はその数を整数で出力しそのまま出力する。

3.3.3 RLE- n

RLE- n は n 個以上連続している値があった場合に RLE をし、値を n 個置きその後連続している数を出力する。連続しない値はそのまま出力する。

3.4 ハフマン符号

ハフマン符号では、出現率の高いデータに短い符号を、出現率の低いデータに長い符号を割り当てることで、データの効率化を行う。今回は、静的ハフマン符号化を使用した。

4 アルゴリズムの改善

RLE は、データそのものの (この場合は、画素) 成分 (以降画素成分と呼ぶ) と、データの長さの成分 (以降、長さ成分と呼ぶ) がある。これらはそれぞれ、値の登場頻度が異なるので、「画素成分」と「長さ成分」を分割して、別々にハフマン符号化する。そのようにすることによって、ハフマン符号化の効率が向上し、圧縮率の改善が可能と考えられる。

5 実験

5.1 実験条件

3 種類の RLE のうち、Packbits と SRLE に対して「ハフマン符号化の改善」のプログラムを作成し、従来法と比較した。同じ画像の閾値を変えて圧縮し、圧縮率を比較した。画像の大きさを 720×480 とした。非圧縮の状態における画像のサイズは、1013KB である。

5.2 実験結果

表 1 に、Packbits による改善前と改善前と改善後の圧縮後のデータ量の比較を示す。また、表 2 に、SRLE による改善前と改善後の圧縮後のデータ量の比較を示す。表 1、および表 2 に示すように、いずれの場合でも改善後において圧縮率が改善している。Packbits と SRLE を比較した場合、SRLE の方が、改善率が高くなっている。

6 まとめ

本稿では、簡易圧縮のアルゴリズムの改善法の提案をした。RLE データの画素成分と長さ成分に分けてそれぞれに対してハフマン符号化する。2 種類の画像を使い従来法と比較した。同じ画像を、閾値を変えて圧縮し、圧縮率を比較した。結果として 1~10%ほど改善出来た。

参考文献

- [1] 柏木文徳, 三浦康之, 渡辺重佳, Grid Computing における MPEG-4 エンコーダの並列化に関する検討, 第 137 回情報処理学会 DPS 研究会, pp.91-96, 2008.11.
- [2] Yasuyuki Miura and Shogo Yamato, Simple Compression Method for Parallel Encoding Environment of Video Image, Proc. of the 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing

表 1 Packbits の実験結果

| 閾値 | | 0 | 1 | 2 | 4 | 8 |
|---------|-----|-------|-------|-------|-------|-------|
| 画像 1 | 改善前 | 174.6 | 120.6 | 80.2 | 53.0 | 32.0 |
| | | KB | KB | KB | KB | KB |
| 改善後 | | 169.4 | 114.1 | 79.4 | 48.7 | 28.7 |
| | | KB | KB | KB | KB | KB |
| 画像 2 | 改善前 | 340.7 | 302.4 | 259.7 | 202.0 | 141.5 |
| | | KB | KB | KB | KB | KB |
| 改善後 | | 323.1 | 278.6 | 237.3 | 182.6 | 128.0 |
| | | KB | KB | KB | KB | KB |

表 2 SRLE の実験結果

| 閾値 | | 0 | 1 | 2 | 4 | 8 |
|---------|-----|-------|-------|-------|-------|-------|
| 画像 1 | 改善前 | 257.2 | 188.0 | 127.6 | 70.0 | 34.5 |
| | | KB | KB | KB | KB | KB |
| 改善後 | | 229.5 | 167.3 | 113.8 | 63.2 | 31.4 |
| | | KB | KB | KB | KB | KB |
| 画像 2 | 改善前 | 361.7 | 330.5 | 290.2 | 219.6 | 134.7 |
| | | KB | KB | KB | KB | KB |
| 改善後 | | 311.7 | 282.2 | 243.7 | 184.3 | 115.5 |
| | | KB | KB | KB | KB | KB |