

解像度とビットレート の関係を用いて適切なエンコード 設定へ導くシステムの提案

Proposal system that appropriate encoding settings using relation between movie size and bitrates

基盤ソフトウェア学講座 0312011011 井上 優希

指導教員： 澤本 潤 杉野 栄二 瀬川 典久

1. はじめに

動画エンコードを行う際、ゲーム機のさらなる進化や HDTV 以上の画質を誇るテレビ、動画の普及などにより動画の圧縮率と画質の保持はさらに重要となっている。

今のところ、「YouTube」への動画投稿の制限は緩く、エンコード設定に関する壁が少ない一方で「ニコニコ動画」への投稿ではビットレートの値の制限、解像度の制限が著しく厳しい。その為に一般会員、プレミアム会員問わずに現状ではまだまだ画質を求めて投稿するには厳しいところがある。

その為、動画の内容によってエンコードする際にある程度必要なエンコード設定がある。「fps(Frame Per Second(フレーム数/秒)、フレームレート)」や「アニメ」「実写」、さらに動きの激しい動画と動きの静かな動画では必要な設定が異なる。

本研究では、「ビットレート」と「解像度」の関係から見て比較的容易に適切なエンコード設定に導き、できる限り画質を保持した動画を作成する為の支援システムの提案を行う。

2. 提案システムについて

2.1. 「解像度」と「ビットレート」の関係

動画を「ニコニコ動画」に投稿するとなると、「解像度」と「ビットレート」の関係は動画容量、画質を考える際に重要となる。その時の「解像度」に適切な「ビットレート」を与えないと、多すぎた時は画質は良いものの、容量が膨んで制限以上になり、再生負荷が無駄に大きくなるデメリットがある。逆に少ないと容量は小さいものの「ブロックノイズ」が目立つようになり、視聴に難ができてしまう。

今回の提案システムでは「解像度」と「ビットレート」の関係から適切なエンコード設定にすることで、容易に十分な画質を保ちつつも程よい容量の動画を作成することが可能となる。また、「ニコニコ動画」への投稿を前提とし、利用するエンコーダーは「x264」での mp4 (H.264, AAC) 形式への動画エンコードのみとする。

図1は提案するシステムを用いてエンコードする際の簡単な流れを表した図である。



図1 動画のエンコードから動画サイトへ投稿するまでの簡単な流れ

2.2. エンコーダーについて

今回扱う「x264」は「H.264」のみの映像フォーマットを扱っているものの、圧縮率が高いながらも画質の保持が優れているため、多くのところで対応している。「ニコニコ動画」に mp4 形式で投稿する際は「H.264」と「AAC(音声フォーマット)」から mp4 コンテナに収納し、動画として投稿することがほとんどである。この事から、今回提案するシステムは「x264」を扱う前提でエンコードに利用する。

2.3. システムへの要求

今回提案するシステムでは、以下の機能が求められる。

- 「動画の入出力」
- 「解像度の入力」
- 「動画内容 (アニメ、実写) を選ぶ項目」

3. 実験

3.1. 実験目的

今回の実験では、「解像度」と「ビットレート」の関係を見る為に「動画の劣化具合」や「各項目の劣化する値の目安」を「SSIM(動画の劣化具合を見る値)」から計り、目安を見る事を目的としている。これを割り出すことで動画をエンコードする際に各項目で適切な値を見ることができる。

3.2. 実験方法

実験で扱う PC のスペックは以下の通りである。

- Intel(R)Core(TM)i7-3612QM CPU@2.10GHz (CPU-Z ver1.62(64bit) より)

実験方法として、まずは以下の動画を用意する。

「用意した動画の情報」

- 使用する動画
 - ・アニメ (動的シーン, 静的シーン)
 - ・実写 (動的シーン, 静的シーン)
- 再生時間:00:01:00.00
- フレームレート:30fps
- 解像度:1280x720

次に、「x264」で各パターン毎にエンコードを行う。

「解像度のパターン (動画のリサイズを行う)」

- 1280x720 - 864x486 - 640x480

「ビットレートの値のパターン (値は固定)」

- 3000kbps - 2500kbps - 2000kbps
- 1500kbps - 1000kbps - 500kbps

これら合計 18x4 パターンを各パターンごとの結果を表の作成、グラフ化させてまとめる。

3.3. 実験結果

本論文では結果の一部である「アニメ (動的シーン)」を「解像度:864x486」でエンコードを行ったときの各ビットレートの値の結果を図 2, 図 3 に表した。

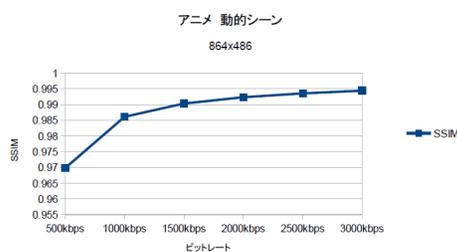


図 2 「解像度:864x486」エンコード結果 グラフ

ビットレート	SSIM	容量	エンコ時間
500kbps	0.9697788	4.488MB	00:02:47.40
1000kbps	0.9860888	8.193MB	00:03:30.80
1500kbps	0.9903063	11.872MB	00:03:56.50
2000kbps	0.9922938	15.549MB	00:04:10.70
2500kbps	0.9935307	19.219MB	00:04:24.20
3000kbps	0.9943959	22.892MB	00:04:35.70

図 3 「解像度:864x486」エンコード結果表

3.4. 考察

今回の実験では余分ながらも、SSIM 以外にもエンコードした動画の容量とエンコード時間も計った。SSIM の値の変化はビットレートの値の上昇に伴って変化は少なくなっている一方で、動画容量とエンコード時間はある程度決まった値ごとに増加、遅延している。また、SSIM の値が「約 0.97(500kbps)」と「約 0.99(2000kbps)」ではブロックノイズが目立ち具合に大きく差が生じ、以下の図 4 のシーンの 1 コマから比較しても見た目で差が出ている。



[1] SSIM=約 0.97(500kbps) [2] SSIM=約 0.99(2000kbps)

図 4 アニメ (動的シーン) の 1 コマ

4. おわりに

本論文では「解像度」と「ビットレート」の関係から、エンコードする際の適切な設定に導くためのシステムを提案した。エンコードは慣れていない人にとっては「ニコニコ動画」への投稿が「厳しい容量制限」、「ビットレート制限」という壁が高いため思い通りの画質へのエンコードが難しい。そこで、今回提案したシステムが少しでもエンコードに慣れていない人でも快適な動画のエンコードができるようになれば、動画作成者と視聴者の互いが快適になるだろう。

参考文献

- 1) 見神 広紀, 宮本 孝道, 木村 啓二, 笠原 博徳:H.264/AVCエンコードのマルチコアプロセッサにおける階層的並列処理, 情報処理学会研究報告. 計算機アーキテクチャ研究会報告 (2010)
- 2) x264 Settings:

http://mewiki.project357.com/wiki/X264_Settings

- 3) ニコニコ動画まとめ Wiki:

<http://nicowiki.com/>