

# 'MPEG4 応用システム（1）

## 3 G-2 ソフトウェアによるリアルタイムエンコーダの実現

<sup>2</sup>千々谷真英 福井隆之 笠野章

<sup>3</sup>（株）東芝 東京システムセンター

### 1.はじめに

MPEG4 動画データをリアルタイムでエンコードするソフトウェアエンコーダの開発を行った。本エンコーダはPC上で動作し、エンコードされた動画データはインターネットを介して、クライアントPC上の専用プレイヤシステム<sup>(1)</sup>でストリーミング再生を行うことができる。

ソフトウェアでのリアルタイムエンコーダの実現にあたって、主に以下のような課題がある。

- ① ストリーミング再生を可能にするためのビットレート制御方法
- ② リアルタイムエンコード処理が間に合わない場合のエンコード破綻の回避方策

第2章では、エンコーダのシステム構成について述べる。

第3章では、上記課題への対策案について述べ、第4章でその効果について述べる。

第5章ではまとめと今後の課題について述べている。

### 2.システム構成

#### 2.1.システム全体構成

エンコーダ及びプレイヤシステムはインターネット等の既存のネットワークに接続されている（図1）。

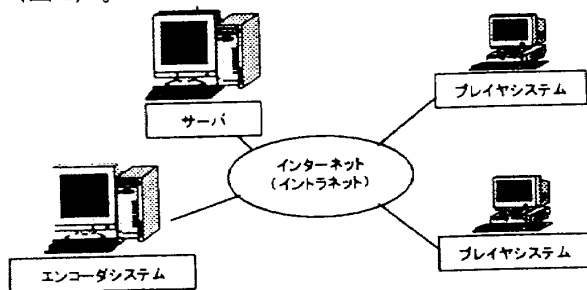


図1 全体構成

エンコーダはエンコードした動画データをサーバにアップロードする。プレイヤシステムはサーバ上の動画データをストリーミング再生することができる。

#### 2.2.エンコーダ構成

エンコーダのシステム構成を図2に示す。

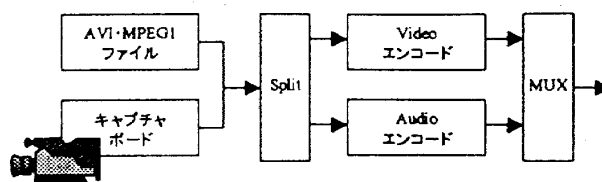


図2 エンコーダシステム構成

エンコーダは複数の処理ブロック：スレッドオブジェクトから構成されている。各スレッドオブジェクトはデータインプットを駆動イベントとして、処理を非同期で行う。

そのため、各処理は他の処理ブロックで一時的に遅延が発生した場合でも、自身の入力データがある限り処理を継続することができる。特にVideoエンコードの処理は、映像によって処理時間に揺らぎが生じてしまうが、この場合でもAudioエンコード処理は途切れずに続けることができる。

### 3.実現方法

#### 3.1.ビットレート制御

インターネットを介してストリーミング再生を行うためには、Video ストリームのビットレート制御が重要になる。特にリアルタイムエンコード時には、ダイナミックなビットレート制御が必要となってくるが、その場合主にフレームレートが犠牲になるため潤沢な動画エンコードと両立させるのは難しい。

本エンコーダではVideo ストリームをGOP (Group of Picture) 単位で一定量の符号量になるように

<sup>1</sup> MPEG4 application system(1) Development of realtime software encoder

<sup>2</sup> Masateru CHIJIYA, Takayuki FUKUI, Akira KASANO

<sup>3</sup> TOSHIBA Corporation

フレーム間隔・量子化パラメータをコントロールする。Video のフレームエンコードで発生した符号量の余剰分・不足分を GOP 全体に平滑化して制御することにより、フレーム間隔の変動を最小限に抑え、滑らかな動画エンコードを実現している。

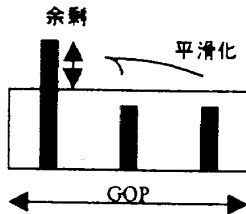


図3 GOP 内でのビットレート制御

もし GOP 中で符号量を制御しきれなかった場合には、過不足した符号量分は次の GOP に持ち越される。次の GOP では持ち越された符号量は必ず吸収するように強制的にフレーム間隔を調整する。こうすることによってトータルのビットレートを制御している。

### 3.2. エンコード遅延の回避

ソフトウェアによりリアルタイムエンコードを行う以上、PC の性能差、負荷変動等の因子によるエンコード遅延は十分に考慮する必要がある。そのためにエンコーダはマルチスレッドモデルとして設計されているが、リアルタイムエンコード時に遅延が限界に達した場合、これは根本的にマシン性能が満たされていないのだが、このような場合にもアプリケーションとしてはベストエフォートでエンコード処理を継続できるような対策があると有効である。本エンコーダでは Audio エンコード処理を最優先と考え、音声の途切れを出来る限り防ぐ方向で、エンコード遅延の回避策をとった。

具体的にはリアルタイムエンコード処理が間に合わなくなった場合、すなわち次にエンコードする予定時刻が、リアルタイムの時刻よりも遅れていた場合には、Video の GOP を強制的に切り、次の GOP 開始時刻をリアルタイムの時刻 (リアルタイム遅延=0) にスキップさせる。GOP 間でスキップした符号量はパディングデータを MUX が挿入することによって符号量を補正する。

## 4. 効果

今回のビットレート制御を入れることによって、単純にフレーム間引きによるレート制御と比較して、より滑らかな動画をエンコード結果として得ることができた。またストリーミング再生時にも、プレイヤーのほうで最初の 1GOP 分をバッファリングすることにより、途切れることなく再生するこ

とができた。

また、エンコード遅延回避策を導入することで、低性能なマシンでも、Video のフレーム間引きは発生するが音声は途切れない、ベストエフォート型のエンコードを実現することができた。

## 5. まとめ

MPEG4 動画データのソフトウェアエンコーダの開発を行った。主な特徴を以下に示す。

- 1) リアルタイムエンコード時により潤沢な動画になるようなビットレート制御方法
- 2) PC 性能不足によりエンコード処理が間に合わない場合でのベストエフォート型エンコード処理方法

今後は更に効果的なビットレート制御方法を検討すると共に、ソフトウェアエンコードのメリットを生かした応用システムの開発を目指す。

### 参考文献

- [1] 沖宗、堀内、笠野：MPEG4 応用システム (2) インターネットでの利用を考慮したプレイヤーシステムの開発、情報処理学会第 57 回全国大会講演論文集、3G-02