

ユーザカスタマイズが可能なレート制御方式

2V-7

伊藤 智祥, 佐藤 潤一, 山口 孝雄, 丸野 進
松下電器産業（株） 先端技術研究所

1.はじめに

インターネットを利用した映像や音声の伝送では、端末負荷や伝送品質の変化に応じた符号化レート制御が行われている。本提案では、伝送帯域等の変動に応じたレート制御のアルゴリズムを記述したスクリプトを受信側端末（復号化側）から送信側端末（符号化側）に送信することで、ユーザの好みに応じた符号化アルゴリズムの変更を実現する。

2.ユーザカスタマイズが可能なレート制御

従来、符号化レート制御アルゴリズムは送信側で固定であった[1]。従って、伝送帯域の変動に応じて、フレームレートをある一定値に固定したいとか、ある水準より画質を下げないようにしたいというユーザの要求を反映することは容易ではない。従来の方法を用いて受信側の要求を満たしたい場合、利用者が伝送の状態を監視しながら手操作で伝送方法を制御する必要があり、ユーザの操作が煩雑となっていた（図1）。また、文献[2]の手法では、フレームレートや量子化ステップ、画像サイズに優先度を与え、エージェントが優先度に基づいた伝送品質制御を行うため、ユーザの操作がある程度軽減できる。しかしながら、単に優先度を指定するだけでは、伝送環境（端末や伝送路の負荷）の変化に応じて、ユーザの好みを反映させたレート制御方法を記述するのは難しい。例えば、従来法では、符号化レートが低くても高レートのレート制御と同じアルゴリズムが使用される。

本提案では、音声・映像伝送を開始する前に、伝送帯域の変動や受信端末の負荷、コンテンツの種類に応じたレート制御アルゴリズムを記述したスクリプトを送信端末に送信する。このスクリプトに伝送帯域や端末負荷が変動した際の符号化アルゴリズムの動作内容をユーザの好みにあわせて記述することで、伝送環境の

変化に応じた符号化レート制御アルゴリズムの変更が自動的に行われる。

以上で説明したスクリプトの例を示す。図2は伝送帯域によって、レート制御アルゴリズムを切り替える例である。伝送帯域の変動に応じて、画質を安定させるためにフレームレートを一定にしたり、また量子化ステップを一定にするように設定している。

図3は受信側端末の負荷に応じたアルゴリズムの切り替えの例である。受信端末の負荷を表す値として、受信端末側で復号されたフレーム数を用いる[3]。復号されたフレームが少ない場合には、フレームレートを下げ、復号されずに破棄されるフレーム数を抑え、量子化ステップを小さくすることで画質が向上するように設定することが可能である。

3.実験と結果

本提案方式を評価するためのシステム構成を図4に示す。本システムでは伝送品質を監視するためにRTP/RTCPを利用し、映像符号化方式としてMPEG4を利用している。また、符号化レート制御アルゴリズムを変更させるために、受信端末にスクリプトの送出部、送信端末にスクリプトの受信部と解釈部を備えている。

実験では図2に示すスクリプトを用いて行い、図中の下線部分の動作を検証した。図5は伝送帯域が128Kbps以上の場合、図6は伝送帯域が64Kbps未満の場合の処理結果である。

図5,図6の上段が送出レートの変化を表す。破線で示されているのがネットワークの利用可能帯域である。中段がフレームレートの変化を表した図であり、下段が量子化ステップの変化を表した図である。また、各図に点線で記されているのはスクリプトによる制御を行わない場合の結果である。図5の結果から、スクリプトに記述されているとおり、フレームレートの最大値が10fpsに制限されており、また、量子化ステップの値が8で固定になっている。一方、図6の結果から、フレームレートが3fpsになるよう制御しており、制御

されていない場合に比べてフレームの変化の幅が少なく、画質が安定している。実験により、スクリプトに記述されている符号化レートの制御アルゴリズムが正しく動作していることを確かめた。

4.まとめ

スクリプトをエンドユーザが記述することで、符号化レート制御アルゴリズムをユーザ・カスタマイズ可能にした。今後は、検証を進め、ユーザ・オリエンテッドな音声・映像伝送システムを構築していく予定である。

従来法

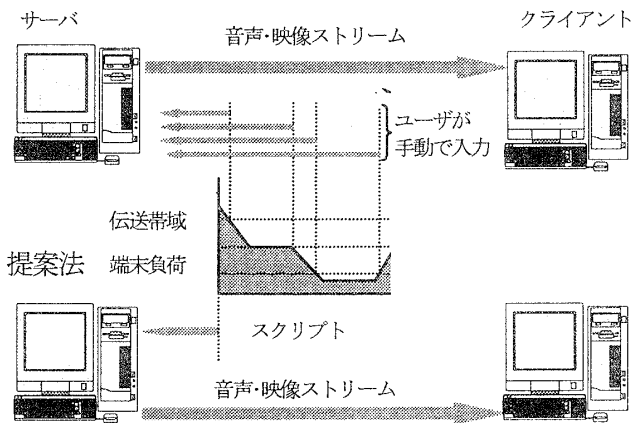


図1：従来法と提案法の比較

```

if($bitrate >= 128000){ 伝送帯域が128Kbps以上の場合
    $qp = 8;           量子化ステップを8に固定
    $max_frame_rate = 10; 最大FRを10fpsに設定
    $image_size = $cif;  画像サイズをCIFに設定
} elseif($bitrate < 128000 && $bitrate >= 64000){
    $minqp = 16;       量子化ステップの最大値が16
    $maxqp = 12;       量子化ステップの最小値が12
} else{
    $frame_rate = 3;   フレームレートを3に固定
    $i_frame = 5;     フレーム内符号化フレームの挿入間隔を5枚ごとに設定
}
    
```

図2：伝送帯域に応じたスクリプトの例

```

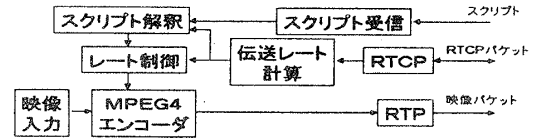
if($dispframe >= 10){復号されたフレームが10枚以上
    $frame_rate = 10;   フレームレートを10に固定
    $i_frame = 30       フレーム内符号化フレームの挿入間隔を30枚ごとに設定
} elseif($dispframe < 10 && $dispframe >= 3){
    $max_frame_rate = 5; 最大フレームレートを5fps
    $minqp = 24;         量子化ステップの最小値が24
    $maxqp = 8;          量子化ステップの最大値が8
} else{
    $max_frame_rate = 3; 最大フレームレートを5fps
    $maxqp = 12;         量子化ステップの最小値が12
    $i_frame = 5;       フレーム内符号化フレームの挿入間隔を5枚ごとに設定
}
    
```

図3：端末負荷に応じたスクリプトの例

参考文献

[1] "Video Codec Test Model, TMN5", Telenore Research, <http://www.nta.no/blukere/DVC/tmn5/tmn5.html>.
 [2] 藤田他, 「やわらかいビデオ会議システムの協調プロトコルの拡張と評価」, 情報処理学会論文誌, Vol.39, No.2, pp.178-187(1998).
 [3] 山口他, 「処理環境変化に耐性のある音声・動画同期再生方式」, 映像情報メディア学会誌, Vol.52, No.7, pp.1055-1061(1998).

送信側端末



受信側端末

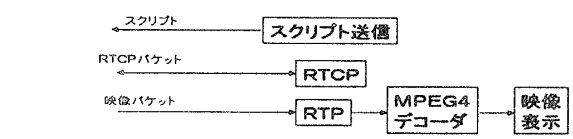


図4：システム構成

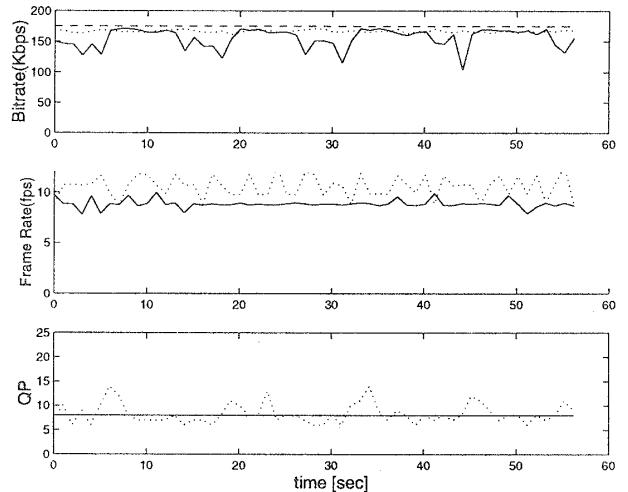


図5：伝送帯域が175Kbpsのときの動作

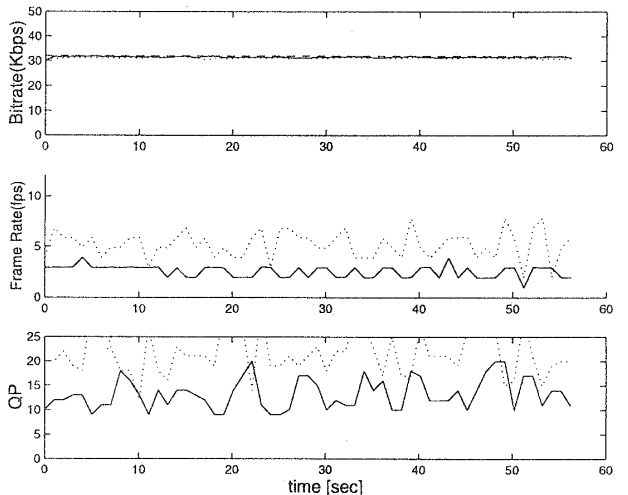


図6：伝送帯域が32Kbpsのときの動作