オーディオビジュアル複合情報処理　30－12
（２０００．９．２２）

端末間のCPU能力差を考慮した ソフトウェアリアルタイム動画像通信に関する検討

宮地 悟史　　和田 正裕　　松本 修一

株式会社KDD研究所 画像通信グループ
〒356-8502 埼玉県上福岡市大原2-1-15
miyaji@kddlabs.net

動画像のリアルタイム双方向通信を全てソフトウェアで実現する時代を迎え，現存する様々な処理能力のコンピュータ同士が相互に通信する可能性が生じている．このとき，一方の端末の処理能力が他方に比べて著しく低い場合，処理遅延となってリアルタイム性が損なわれるばかりでなく，コンピュータの基本的な機能にも影響を与え，様々な障害が発生することとなる．本論文では，従来の動画像符号化アルゴリズムに通信端末相互の処理能力を考慮した符号化制御を新たに加えた方式について検討する．実験により端末間の処理能力が大きく異なる状況下においても安定したリアルタイム双方向動画像通信が可能となることを示す．

A Study on Software-Based Real-Time Video Communication System Considering Difference of CPU Performance between Terminals

Satoshi Miyaji　　Masahiro Wada　　Shuichi Matsumoto

KDD R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka-shi, Saitama 356-8502, Japan
miyaji@kddlabs.net

Recently, CPU performance has highly advanced level so that real-time video coding which generally requires a high computational power has become possible only by software. In a software-based video communication, however, terminals with different performance may communicate to each other. In this case, drawbacks such as picture frame loss, transmission delay and loss of intrinsic functions of a computer may occur at the terminal with lower performance. In this paper, a new control method for software-based real-time video communication is proposed to solve such communication problems.

1. はじめに

近年のパーソナルコンピュータ(PC)の性能向上により，動画像の圧縮符号化/復号処理を同時にかつリアルタイムに行う双方向動画像通信ソフトウェアが実用化されつつある．このようなソフトウェアでは，処理対象とする画像サイズはCIFやQCIFであり，フレームレートも毎秒 10〜15 フレームが上限となっていて，通常のテレビジョン信号に比べると見劣りするが，ユーザがインターネットを利用する際の通信回線が一般にはモデム(28.8kbit/s〜56kbit/s)や ISDN (64kbit/s, 128kbit/s)であることから，ソフトウェアの画像サイズやフレームレートの制約は伝送路のビットレートの制約とも合致しており，インターネット上でのリアルタイムアプリケーションとしては適当なものとなっている．

従来のハードウェアによる低ビットレート動画像符号化では，

- 入力画像の性質
- 符号化アルゴリズム
- 符号化ビットレート，フレームレート，画像サイズで与えられる画素当りのビットレート

のパラメータで品質が決定される．例えば，ある性質の動画像に対して，符号化アルゴリズムとピクセルレートが与えられると，画像品質は自ずと決定される．言い換えると，ある要求された品質に対するフレームレートが自ずと決定される．

これに対し，ソフトウェア，特にリアルタイムソフトウェアでは，処理に要する演算量が実行する CPU の能力を下回らなければならないという条件が加わり，これらのパラメータの関係が複雑となる．

画像符号化における必要演算量は，

- 符号化アルゴリズム
- ビットレート
- フレームレート，画像サイズで与えられる単位時間当りの画素数

で決定され，上記の品質を決定づけるパラメータに対して制約を加えることとなる．例えば符号化アルゴリズムと画像サイズが与えられた場合，符号化可能な最大フレームレートは，CPU 演算量の上限を超えない最大のフレームレート，あるいは上述の符号化品質の観点での最大フレームレートとの小さい方となる．

一般に，符号化アルゴリズムを簡略化すると演算量は低減されるが，符号化効率が低下する．このとき，適当な SNR の制約下で，符号化可能な最大フレームレートは，演算量による制約よりは，むしろ品質の点からの制約が支配的となる．一方，複雑な演算を用いて符号化効率を向上させると，最大フレームレートは演算量による制約の方が支配的となってくる．

このようにソフトウェア動画像符号化においては，符号化品質と演算量とが複雑な関係となり，最適なパラメータ導出が問題となる．

さらに今後，動画像通信のソフトウェア化が進むにつれ，従来のハードウェアを中心とした実装とは異なり，さまざまな処理能力の機器にインストールされ，あるいは携帯端末などに組み込まれ，これらが相互に通信されることが予想される．

ソフトウェアシステムにおいては，処理量の大きい動き補償型画像圧縮符号化/復号処理をリアルタイムで行うことは，常に CPU 処理能力最大限に近いところで動作していることが予想される．さらに現在，CPU の処理能力の進歩は目覚しく，現存する PC の処理能力の差は極めて大きいものと考えられる．その際，通信端末間の処理能力が大きく異なる場合，処理能力の低い端末においては様々な問題が発生することが考えられる．

本検討では，リアルタイム動画像双方向通信ソフトウェアに関するこれらの問題の中で，CPU 演算能力が異なる端末同士の通信における問題点に着目し，新たに通信端末間の処理能力を考慮した符号化制御を行う方式について検討する．

2. 端末間能力差による問題

一般的なリアルタイム双方向通信ソフトウェアにおいて，動画像圧縮符号化/復号処理が CPU の単位時間あたりに処理できる演算量を超えた場合，

- 符号化処理においては，所定の時間内に符号化処理が終了できないため，符号化可能

フレームレートが低下.

- 復号処理においては，復号処理が遅れるため画像表示の遅延，あるいは未処理ビットストリームの蓄積による受信バッファの破綻.

が発生する.

また双方向通信ソフトウェアにおいては，画像符号化/復号処理に加えてパケットのリアルタイム送受信処理が加わる. さらには，計算機の一般的な処理として，OS の動作を維持するための基本的タスクを実行する必要がある. 動画像符号化/復号処理により，CPU の演算能力をオーバーしてしまった場合，上述した符号化/復号処理に不具合を生じるだけでなく，パケット通信処理や，OS の基本的動作に障害を与えることとなる.

現在主流となっている GUI ベースのマルチタスク型 OS では，各プロセスへの CPU 時間の割当は OS により行われているものの，GUIメッセージをアプリケーション内で処理する必要があるため，上記のような CPU の処理量の著しい超過があった場合には，メッセージ処理を所定の時間内に行うことができず，マウスやキーボードの反応が遅れる，あるいは最悪の場合には無反応(ハングアップ)といった状態を引き起こす場合がある.

このような問題が発生する実例として，双方向動画像通信ソフトウェアを多数のユーザに配布した場合を考える. 配布されたソフトウェアは様々な演算能力の PC にインストールされ，相互に通信が行われることとなる. このとき，通信両者の CPU 能力に大きな差がある場合を想定し，次のような状況を仮定する.

- CPU 能力の高い PC 側では，符号化制御の結果，所定のフレームレートで符号化を行い，ビットストリームを出力すると同時に受信ビットストリームの復号処理が可能.
- CPU 能力の低い PC 側では，符号化処理と同時に上記 CPU 能力の高い端末から送られてくるビットストリームをリアルタイム復号処理ができない.

この場合，能力の低い PC 側では，符号化処理ならびに能力の高い PC から一方的に送られてくる高いフレームレートで符号化されたビットストリームの復号のため，CPU 能力をオーバーし，先に述べたように，符号化画像のフレームレートが低下する，受信画像フレームの表示が遅延する，受信バッファが破綻する，さらにはコンピュータの応答が無くなる，といった状態に陥る可能性がある. このとき，CPU 能力の低い PC 側では符号化可能なフレームレートが CPU 能力の高い側に比べて著しく低くなるため，能力の高い側での受信画像フレームが大きくスキップし，ビットストリームの復号に要する CPU 負荷はますます低くなる.

一般に，復号に要する演算量は符号化に比べて十分無視できると考えられているが，ソフトウェアに実装する際には符号化演算(特に動き検出)処理の最適化や簡素化を行うため，復号処理が符号化処理に比べて無視できるほど小さくはならず，実際に上記のような現象が起こる場合が十分に考えられる.

このとき各端末における CPU の使用状態は，一方は限界を超え，他方は十分余裕のある状態となり，総合的な通信品質が大きく低下する. この状態を避けるためには，CPU の能力を使い果たすことのないよう画像符号化/復号処理を制御することが必要となる.

符号化側では，

- フレームレートを落とす
- 符号化アルゴリズムを簡略化する

ことで演算量の制御が可能である. 一方受信側では，H.263 や MPEG 等一般的な MC-DCT 方式の場合，フレーム間予測を用いているため，画像フレーム復号を一方的に間引くことはできない.

復号処理は受信ビットストリームにしたがって行う必要があるため，アルゴリズム変更による処理量削減の可能性も低い. また，画像表示処理のみを間引くことは可能であるが，処理量削減の寄与は低い. したがって，受信側の演算量の制御は符号化側において行うのが望ましい.

さらに，演算能力の制約により符号化フレームレートを落とす場合，フレームレートが低下した分，画像符号化のための情報量割り当てに反映させて，画像品質を向上されることが可能である.

このような背景の下で，本検討では双方向動画像通信における演算量制御方式を提案する．以下に詳細を述べる．

3. 演算量制御方式の詳細

3.1 端末の演算量

まず，端末で必要な演算量について述べる．各端末において，画像圧縮符号化に要する演算量，復号に要する演算量，および通信処理や OS が使用する演算量が必要となる．必要演算量は，符号化/復号のアルゴリズムや，使用するCPUの能力(キャッシュ容量，パイプライン構成，並列演算機能の有無など)により異なる．これら必要演算量の合計は，CPU の持つ単位時間あたりに処理可能な演算量を下回る必要がある．

必要演算量に関して，記号を次のように定義し，動作の条件を示す．

n	:	端末番号(n=1 or 2)
En	:	1フレーム符号化処理に必要な演算量
Dn	:	1フレーム復号処理に必要な演算量
Mn	:	その他の処理に必要な1秒あたりの演算量
Cn	:	CPU の 1 秒あたりの演算量
FEn	:	符号化フレームレート
FDn	:	復号フレームレート

このとき端末 1, 2 において

$$E_1 \cdot FE_1 + D_1 \cdot FD_1 + M_1 \le C_1 \quad (1)$$

$$E_2 \cdot FE_2 + D_2 \cdot FD_2 + M_2 \le C_2 \quad (2)$$

が必要条件となる．また，デコーダでは符号化フレームレートにしたがって復号が行われるため，

$$FE_1 = FD_2 \quad (3)$$

$$FE_2 = FD_1 \quad (4)$$

となる．

本方式では，式(1)，(2)を満たすよう，両端末の符号化フレームレートを制御し，演算量制御を行うこととする．

符号化フレームレートの制御は 1 秒あたりの符号化処理フレーム数を制御することとなり，これにより演算量の制御が行える．また式(3)，(4)により，符号化フレームレートは受信側に対する復号フレームレートとなるため，同時に復号時の演算量を制御することとなる．

H.263 や MPEG-4 など一般に低ビットレート画像符号化では，可変フレームレートの方式を用いている．本方式では，演算量制御結果をこのような可変フレームレート符号化方式のフレームレート上限値に反映させることで演算量の最大値を制御する．さらに，画像符号化方式として符号化前処理を有する低ビットレート画像符号化方式[1]を使用することにより，フレームレート低下分は効果的にフレーム割り当て情報量に反映させることができ，画像品質が向上する．

3.2 制御の実際

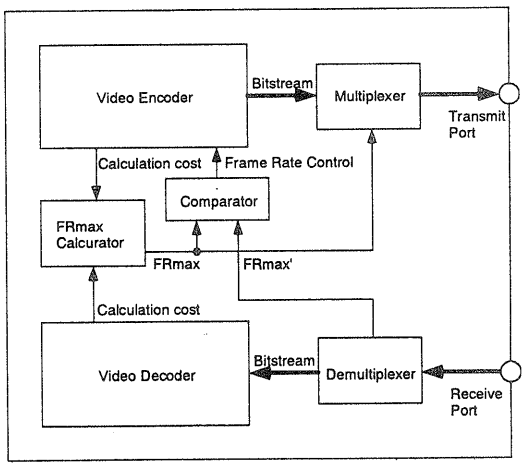

図 1: 端末の構成

端末の構成を図 1 に示す．各端末において，以下に述べる処理を行う．以下，端末 1(n=1)を例に説明する．端末 2(n=2)については，以下の説明において式(1)と式(2)，および式(3)と式(4)が入れ替わる．

端末 1 において，自分自身のフレーム符号化に要する演算量 E_1 と，フレーム復号に要する演算量 D_1 と，その他に必要な演算量 M_1 とから式(1)を確実に満足する符号化/復号フレームレートを一旦決定する．通常は符号化フレームレートと復号フレームレートは同一，すなわち両端末での受信されるフレームレートは同一であることが望ましいため，FE_1=FD_1

として初期フレームレートを決定する.

初期フレームレート決定後, 端末1においてCPUの負荷状態をモニタし, CPU 負荷があらかじめ設定した目標値(ターゲット CPU 負荷)に満たない場合にはフレームレート FE_1 を一定ステップ増加させる. 一方, CPU負荷が目標値以上の場合には, フレームレート FE_1 を一定ステップ減少させる. ここで新たに決定したフレームレート FE_1 は, 現段階ではまだ実際の符号化制御へは反映させず, 候補最大フレームレートとして, ビットストリームに多重して端末2へと伝送する.

同時に, 端末2で同様にして決定された候補最大フレームレート $FE_2(= FD_1)$ が, 受信ビットストリームから多重分離して取得される. ここで, 両端末間のフレームレートに大きな差が生じないよう, 両端末とも FE_1 と FE_2 の小さい方の値を確定最大フレームレート FE として, 実際の符号化処理に反映させる.

実際に符号化処理に反映させた後, 再度CPU負荷状態をモニタし, 同様の候補最大フレームレートの決定, 端末 2 の候補最大フレームレートの受信, 確定最大フレームレートの決定の処理をリアルタイムで行う.

以上の処理は端末2でも同様である.

このようにして算出されたフレームレート FE は, 両端末において式(1)および式(2)を満足することとなる. この FE を符号化フレームレートの最大値として画像エンコーダのフレームスキップ制御部へと逐次入力する. 本演算量制御の結果フレームスキップが増大させられた場合は, 画像エンコーダのビット割当制御により, フレームの割り当てビット数を効果的に増大させることができ, 画像品質が向上する.

4. 実験による検証

提案方式の有効性を確認するために, 実験を行った. 実験時の条件を表1に示す. また, 参考として実験に使用した2台のPCのベンチマーク結果を表2に示す.

4.1 実験結果

先に述べた2台の端末間で, コーデックソフトウェアを動かし, 測定を行った. 低速側 PC での測定結果を図2～図5に示す.

なお, 通信中の各端末におけるCPU使用率の取得については, パフォーマンスカウンタ Microsoft Windows API(Pdh***)を使用した.

CPU 使用率(図2, 図3)については, 制御なしの場合, 通信開始直後から急激に立ち上がり 100%の状態で飽和している. ただし, この状態において高速側 PC の CPU 使用率は平均約 40%であった. 低速側 PC では, 高速側から送られてくる毎秒 10 フレームで符号化されたビットストリームを最大限その速度で復号を試み, かつ送受信パケット処理, ビデオキャプチャ処理, 画像符号化処理を同時に行っている状態である. 一方制御ありの場合には, CPU 使用率がターゲット値(今回は 80%)となるよう演算量制御が行われているのがわかる.

また, 受信フレームの復号表示遅延について図4, 図5に示す. 高速PCからは, 10フレーム/秒のビットストリームが送られてくるため, 可能な限り 100mSec 間隔で復号表示を試みるが, CPU 負荷が限界に達しており, 多数のフレームが処理遅れとなり, また遅れを取り戻すだけの処理能力に余裕が無いため, 画像復号遅延が蓄積してしまう. 制御なしの場合(図4)100フレームで5秒の遅延となっている. 一方, 本制御が行われている場合には(図 5), フレームの表示遅延の蓄積が解消され, リアルタイム通信が支障無く行えることがわかる.

5. まとめ

ソフトウェアによる双方向リアルタイム動画像通信に関して, CPU 処理能力を考慮した演算量制御法についての検討を行った.

通信端末間で CPU 処理能力に大きく差がある場合, 能力の低い側での PC の動作は能力の高い側で符号化されたビットストリームの復号処理に支配された形となり, 送受フレームレートのアンバランスや, 低速 PC の不安定動作を招くという問題
があった.

本検討では, 双方のCPU使用状態をリアルタイムにモニタリングしながら, 常に両端末の動作が破綻

を来たさないよう，かつ送受信の通信品質が同等となるよう演算量制御を行う手法を提案した．

演算速度に差がある2台のPCを用いて実験を行った結果，本方式による制御を用いることにより，上記問題点が解決され，本方式の有効性が確認された．

日頃御指導いただく株式会社ケイディディ研究所の秋葉代表取締役所長に感謝いたします．

参考文献

[1] 宮地，松本：“統合的時空間制御による超低ビットレート動画像符号化方式”，映情学誌，Vol.51, No.10, pp.1706-1714, 1997

表1：実験時の条件

項目	仕様
高速側PC	Pentium-III 500MHz Desktop
低速側PC	Mobile Pentium MMX 233MHz
伝送路	Ethernet LAN
OS	Microsoft Windows 2000
画像符号化	Advanced Pre-Coding [1]
画像サイズ	160×112
パケット間隔	60mS
パケットレート	48kbit/s
ターゲットCPU使用率	80%
ターゲットフレームレート	10 frames/s

表2：ベンチマーク結果(HDBench)

CPU	整数演算（×500Loops/sec）
Pentium-III 500MHz	20118
Mobile Pentium 233MHz	5446

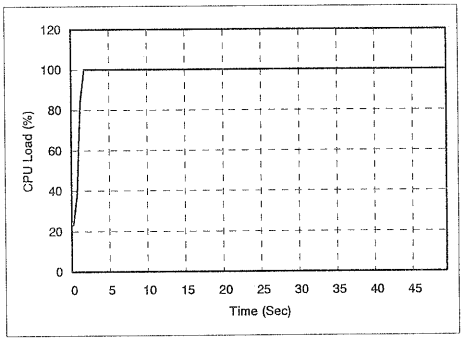

図2：CPU使用率(制御なし)

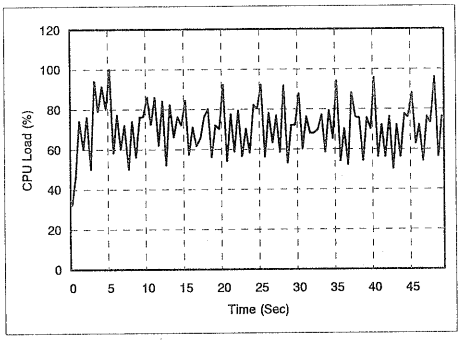

図3：CPU使用率(制御あり)

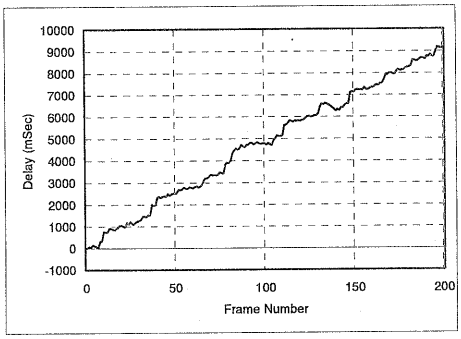

図4：フレーム復号遅延(制御なし)

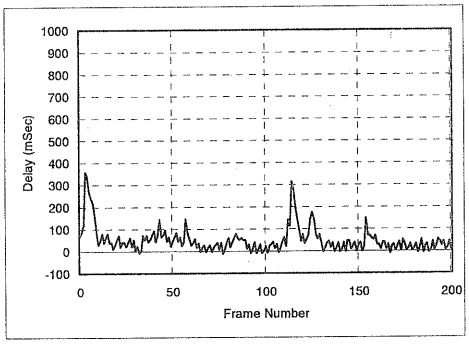

図5：フレーム復号遅延(制御あり)