

画質多重伝送方式による インターネット動画サーバとビューア

山本吉伸, 仁木和久
通商産業省工業技術院電子技術総合研究所
{yoshinov,niki}@etl.go.jp

本稿では, インターネット上で利用できる動画サーバと専用ビューアについて報告する. 我々の動画サーバは, Motion-JPEG 画像を HTTP で送出するため, 一般的な市販の WWW ブラウザで映像を見ることができる. また, 画像中から必要な部分を異なる品質で伝送する機能及びその機能を有効に利用するためのパターン認識機能を組み込んでいる.

Animated Image Server and Viewer on the Internet by using Multi-Quality Image Transmission Method

Yoshinobu YAMAMOTO, Kazuhisa NIKI
Electrotechnical Lab.

In this paper, we describe an animated image server and viewer on the Internet. The animated image can be watched on popular WWW browser in terms of transmitting Motion-JPEG images by HTTP. Furthermore, our server realizes multiple quality image transmission method that consists of partial images transmitting function and pattern recognition function.

1. はじめに

動画をインターネット／イントラネット環境で通信したいという要求は強い。しかしながら動画として快適な映像であると感じられるに十分な回線容量はまだ確保できていないのが現状である。確かに通信環境は向上している。イントラネットではEthernetで10Mbps～100Mbps(実効で最大2Mbps～20Mbps)の通信容量を期待できるようになった。しかし多くのユーザが利用すれば一人当たりの通信容量は当然少なくなる。インターネットではなおさらのことである。一般的にはイントラネットの100分の1から1000分の1以下の通信速度しか期待できない。さらに近年のノート型、サブノート型コンピュータの性能向上は、次世代の動画通信システムの環境が携帯電話やPHSを介したものになることを強く示唆している。技術革新の状況から、携帯端末も将来的には2Mbps程度までは実現できると考えたとしても、その時代に求められるであろう画質の水準に対して充分ではないと予想される。

本研究では、常に安定した高通信帯域が保証できないインターネット環境であっても、動画によって可能な限り快適なヒューマンコミュニケーションを実現できる環境の構築を目的として、サーバ[WACS]とクライアント[WACC]の設計と実装を行った。

2. WACS/WACC

WACS は、画質多重伝送機構、疑似輻

輻検出機構、Server-Push 機構を備えた、動画 (Motion-JPEG) を送出する HTTP(Hyper Text Transport Protocol)サーバであり、WACC は専用クライアントである。

我々が想定している環境(高速回線と低速回線が混在し、さらに回線状態が常に変化するような環境)で最大のパフォーマンスを得るために、WACS のデザインには画質多重伝送機構をはじめとするいくつかの工夫が取り入れられている。

2.1 画質多重伝送

図1は、画質多重伝送方式を説明する映像の例である。顔の付近が高精細で、それ以外の領域では粗い画質になっていることがわかる。(特にコミュニケーション時の)人間の視線(興味)の対象は、限られている。それは相手の顔表情であったり、指の形であったり、相手の部屋に入ってきた人物であったり、背景の後ろで動いている物であったりする。



図1 画質多重伝送

そこで、特定の領域だけを高精細かつ／または高fps(frame per second)で送信し、その他の領域は低精細かつ／または低fpsで送信してもよいと考えられる。

WACS は、映像に任意の領域を設定し、その各領域の画像圧縮率や fps を変更できるように設計されている。

この技術は用途が広い。例えば複数の超高画質カメラで球場全体や空全体を捉えていたとしよう。そのすべてのデータを通すのに十分な通信回線が確保できない場合、全体像は比較的低画質で送り、受信者が見たいと思う領域だけを集中的に送信することができるのである。類似の研究として、例えば Kasai らは周辺視野の認識率が低いという認知特性を利用し、注視点から外れていくにしたがって階層的に画質を変化させるアイデアをシミュレーションで評価している [1]。WACS では領域ごとに fps を変更することも可能であり、アプリケーションによっては Kasai らの方法よりも通信帯域を節約できる。さらに受信側の画面の大きさなどの情報には依存しないシステムであるので、放送型通信や送受信が一体になっていないシステムであっても適用可能である。井口らは WACS 同様に多くのパラメータが調節可能なシステムを実装して報告している [2] が、受信者側は優先的に送信を希望する領域を複数選べないことや、領域が静的であるなどの実装上の制約がある。

2.2 画像認識モジュール

画質多重伝送方式では、小さな領域を選択すればその分だけ通信容量を節約できることになるが、アプリケーションの利用状況によっては快適な利用を阻害することになりかねない。快適利用を阻害する危険は大きくわけて二種類ある。一

番目は、撮影対象の移動である。カメラが部屋全体をとらえているような状況であればほとんど問題ないが、撮影対象の近くに設置されているような状況では、撮影対象物自身がわずかにずれるだけで受信者の定めた任意領域からはずれてしまうことになる。毎回受信者が領域を設定し直したり、送信者側に動かないように強制しては快適なコミュニケーションを阻害してしまう。

二番目の問題は、撮影対象以外の変化の察知である。一番目の問題とは逆に、カメラが部屋全体をとらえているような状況で予想される問題点である。指定した領域だけが動画でその他の領域が静止画になる方式では、部屋の中に起こった変化、例えば部屋の中に入って来た人物など、を知ることができない。

画質多重伝送の利点を活かしつつ、これらの欠点を解決するため、WACS には画像認識モジュールが組み込まれている。現在、画像認識モジュールには(1)矩形領域の自動追尾モジュールと(2)変化領域の自動検出モジュールがある。

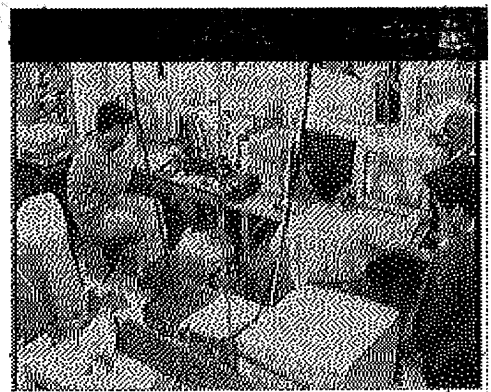


図 2 元映像

2.2.1 矩形領域の自動追尾モジュール

WACS では、最初にクライアント側から指定された領域を、相関トラッキング法によって追跡することができる。図 1 中で、高画質領域に指定されている頭部が左右に振れても自動追尾し、常に顔の領域部分を高画質領域として送出させることができる。なお、本モジュールはサーバ側クライアント側どちらにも実装可能であるが、(1)通信回線が信頼できない環境でも追尾するにはサーバ側での処理が必要であることに加え、(2)ノートパソコンなどの非力なシステムでも稼働できるようにビューアをできる限り軽くする必要があるので、サーバ側に実装されている。

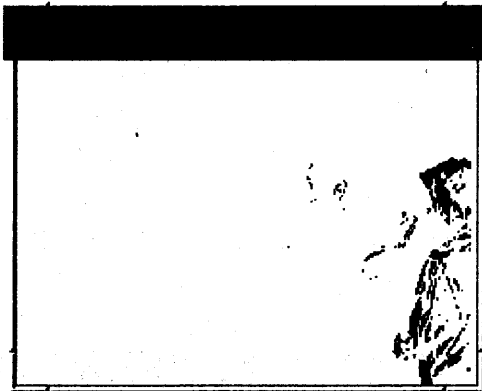


図 3 差分映像

2.2.2 変化領域の自動検出モジュール

サーバ側でキャプチャした映像情報の差分を常に監視するモジュールである。

図 2 は、右端に人が入ってきた状態である。図 3 は、その直前の画像との差分をとったものを示している。差分をとって得られる変化量があらかじめ与えておいたしきい値を越えたときに、その変化

のあった領域をクライアントに送出する。なお、このモジュールは慶應義塾大学理工学部安西研究室の斉藤氏が試作を進めているものを利用して実験している。

2.3. Motion-JPEG の採用

WACS は、回線の伝送容量が著しく低下したときにも対応しなければならない。最低回線速度は 1kbps 程度まで想定しなければならない。最悪時には動画をあきらめなければならないかもしれない。また、Ethernet 上で TCP(UDP)を利用する場合、運が悪ければパケットの到着そのものが確実ではないことが起こり得る。そのため、例えば MPEG のように動画の性質を利用した(キーフレーム中心の)圧縮方式ではキーフレームの欠落の危険があり、結果として伝送情報量の低下を招く可能性がある。

そこで本システムでは、画像圧縮に Motion-JPEG を採用した。Motion-JPEG は JPEG を時系列でならべた構造で、どのフレームが到着しても通常の JPEG 圧縮された静止画として扱うことができる。しかも JPEG は、近年の CPU パワーであればソフトウェアだけでもある程度はリアルタイムの圧縮伸長が可能であり、ハードウェアを実装する場合でも JPEG 用の圧縮伸長チップは安価である。

さらに、JPEG には画像によって圧縮後のサイズが異なるという興味深い性質がある。圧縮後のサイズを監視するだけで画像中に大きな変化があったことを察知できる可能性を意味しており、QoS(Quality of Services)として利用できると考えられる。

2.4. 輻輳／サーバ過負荷検出

動画サーバは、ネットワークの状況に応じてデータの送出を停止、再開をコントロールするべきである。画像の転送には送信側にも相当の負荷がかかっており、可能な限り無駄な送出を避けたいからである。もちろん、複数のネットワークを経由してデータを送出することを想定しなければならないので、受信端末からの反応時間を検出する方式は不適切であるしオーバーヘッドが大きすぎる。そこで、送出側の書き出し状況を調べる方法で疑似的に輻輳検出を行なう。WACS は映像送出の際、socket への書き出し部分を fork して子プロセスとしている。その fork のとき、以前の子プロセスがまだ残っていれば(= socket への書き出しがブロックしているかサーバが過負荷状況にあれば)、送出操作が完了していないことを意味するので、自分自身は直ちに exit する。これはいわゆるネットワークの輻輳検出とは異なるが、実用上は有効である。また、インターネットではそれ以上のことは原理的に困難である。

2.4. 汎用性の高いプロトコルと Server-Push 方式の採用

プロトコルは可能な限り軽く、汎用的なものが望ましい。UDP は軽量であるし RTP は優れた特性を持っているが両者ともゲートウェイを通過できないことが多い。そこで TCP を用いることにした。さらに、単に TCP を利用するだけでなく、多くのアプリケーションと接続可能になるという意味から、HTTP を用いている。これにより、例えば Netscape Navigator

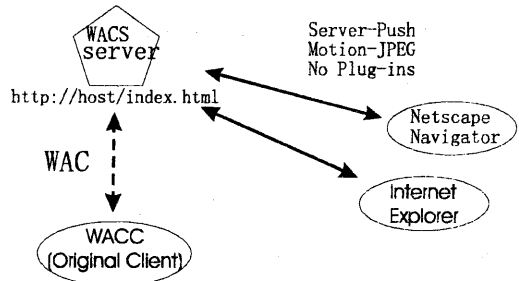


図 4 動画サーバとクライアント

等のビューアにも対応可能になる(図 4)。

このとき、Navigator には plug-in を付加して Motion-JPEG を表示するという方式も考えられるが、各種のコンピュータに併せて plug-in を用意しなければならないことから、汎用性に欠けることになる。そのために plug-in を用いない伝送方法を実現することが望ましい。そこで本システムでは server-push 方式を利用した[3]。通常、JPEG 映像を連続して見ようとする、一コマ毎に HTTP コネクションを張り直すために大変遅くなってしまふ。server-push 方式であれば、一セッションあたり一回だけコネクションを成立させれば Navigator 上でも動画として表示させることができる。

2.5 専用ビューア

専用ビューア(クライアント)WACCは、サーバから送られてくる映像中の任意の領域を設定できるようになっており、かつ受信した任意領域の映像を適切な領域に表示する機能(「WinW 機能」)を有している。

ユーザは、最初に WACC 上に映されたサーバからの画像中の任意の領域をマウスで指定するだけでよい。その後は指定領域だけが動画として伝えられる。現在

の実装では、各種パラメータはキーボードより入力している。

3. まとめ

おおむね実用的レベルでの安定運用が可能になっているが、ソフトウェア実装上の課題は残されているように感じられる。

a. 矩形領域は任意の箇所を複数指定することができるが、各領域毎に JPEG への圧縮を行うためにサーバには高負荷がかかる。10M の Ethernet であっても、JPEG 圧縮しないで送出した方が高速になる場合もあり、定量的な評価が今後の課題である。

b. 認識モジュールは、画像全体に激しい変化がないことなどを前提としているので知識処理などを加えなくても実用的な認識率を保つことができる。ただし、相当の計算機パワーを消耗するため、大きく動く小さな領域（たとえば手先）をトラッキングするような状況には向かない。Pentium150MHz のサーバでの実験では、手の先部分をトラッキングして通信量を減らす努力をするより、手の部分を含んだ大きめの領域を指定してしまった方が全体を俯瞰しやすく、結果として効率がよい(トラッキングの効果が低い)ようである。リアルタイムトラッキング専用のハードウェアを導入するか、あるいは部屋全体を映している状況で人物全体をトラッキングする方が好ましい。

c. WACS で想定している変化領域の自動検出手法では、部屋全体が頻繁に変化する環境(人の出入りが頻繁であるような環境など)に適していない。この場合、デー

タ量を削減できないのである。このモジュールをどのようなタイミングで利用するのが最適であるのか、定量的評価が必要である。

d. 井口らは文献[2]で『(全体の画像は低画質で送信、相談のポイントとなる範囲は高画質で送信することによって)全体のイメージを理解しながら、相談のポイント部分を詳しく観察することができる。』と指摘している。本システムでも同様の点に着目した。図 2 でもわかるように、領域外が低画質でもほとんど違和感なく利用できる。さらに領域内だけを表示するのに比べ、広い領域の絵を見ることによって「除き穴」の不快感[4]を低減することに貢献している。WinW 機能はほんの小さな事項かもしれないが、そこから受ける印象には大きな違いがあることを指摘しておきたい。

4. 参考文献

[1]Hiroyuki KASAI, Mei KODAMA, Keiji MAEDA, and Hideyoshi TOMINAGA, "A Visual Information Model with Hierarchical Image Coding", Proc. Of multimedia Japan 96, pp.102-105, 1996.

[2]井口信和, 高坂知子, 内尾文隆, 津田達, "マルチメディア型遠隔技術相談システムにおける QoS コントロール機能", 情報処理学会論文誌, Vol. 17-2, pp.7-12, 1996.

[3]http://home.netscape.com/assist/net_sites/pushpull.html

[4]山本吉伸, 仁木和久, "コミュニケーション基礎実験計画とその環境設計", 情報処理学会論文誌, Vol.60-7, pp.49-56, 1995