

# 遠隔コラボレーション支援のための 映像伝送システムの開発とその遅延評価

村上 昂<sup>1</sup> 飯田 卓也<sup>2</sup> 片桐 滋<sup>1</sup> 大崎 美穂<sup>1</sup>

**概要:** 視聴覚メディアの対称性を確保することで遠隔地間コラボレーションの支援を目指すシステム「t-Room」のために、新たに Linux を OS とするミドルウェアの開発を進めている。本稿では、計算機内の処理やネットワーク通信に伴う遅延の大きさを接続地点間で補正する機能を持ち、新ミドルウェアにおける映像伝送機能を担当する、映像伝送システムの設計と実装結果、そしてその性能評価結果を報告する。評価は、映像データのサイズを変数としたシステムの伝送速度の定量的測定実験と、撮影映像のフィードバック再生における遅延と画質に関する主観評価実験によって行った。実験の結果、開発したシステムが一定の伝送品質を保ちつつ長時間動作することを確認することができた。

**キーワード:** マルチメディアシステム, 遠隔コラボレーション支援, t-Room, 映像伝送システム

## Development of Video Transmission System for Supporting Remote Collaboration and its Delay Evaluation

SUBARU MURAKAMI<sup>1</sup> TAKUYA IDA<sup>2</sup> SHIGERU KATAGIRI<sup>1</sup> MIHO OHSAKI<sup>1</sup>

**Abstract:** A remote collaboration support system called "t-Room" has been vigorously investigated, aiming to support collaboration works between remote cities by providing the sense of "being in the same room", i.e., symmetry in acoustic and visual media. To control this support system, we develop a new middleware on the Linux operating system. In this paper, we specially focus on our newly-developed video transmission system, which works as a subsystem for video transmission within the new middleware framework, report its design and implementation result, and show its evaluation experiment results. The evaluation consists of the quantitative measurement of transmission speed, which is done using video data size as control variable, and the mean opinion score test in terms of subjective assessment on the quality and delay of transmitted video data. Experiment results demonstrate that our video transmission system can run for a long time period with some acceptable transmission quality.

**Keywords:** Multi-media system, Remote collaboration support, t-Room, Video transmission system

### 1. はじめに

テレビ電話やテレビ会議システムなどの、遠隔地間における対話を支援するシステムが普及する中で、より効果的な情報共有や身体動作を伴う共同作業を支援することを目指す遠隔コラボレーション支援システムの研究開発が精力

的に行われている(例えば [1], [2])。こうした支援システムが目指す理想の1つは、遠隔地にいる利用者にあたかも同室にいるような感覚、同室感を提供することにある。しかし、多くのテレビ会議システムなどに見られるように、カメラ位置やディスプレイ位置の制約のために、利用者は、互いに対峙するような位置関係にあることを強いられ、メディアの対称性を確保できない状態、即ち互いが異なるモノを見聞きせざるを得ない状態におかれる。このメディアの対称性の崩れは、単にコラボレーションにおけ

<sup>1</sup> 同志社大学大学院 工学研究科  
Graduate School of Engineering, Doshisha University  
<sup>2</sup> 同志社大学 理工学部  
School of Engineering, Doshisha University

る自然性の低下を招くだけでなく、情報共有あるいは作業の質的かつ量的な低下も招く。

このメディアの対称性の崩れの克服を目指し、特有の部屋構造を持つ遠隔コラボレーション支援システム「t-Room」が提案されている [3]。t-Room は、大型のディスプレイとカメラ、スピーカ、マイクのメディア機器から成る壁（モリスと呼ばれる）によって囲まれた部屋構造を持つ。同型の t-Room どうしをネットワーク接続することでそれぞれの t-Room は仮想的に重なり合わされ、異なる t-Room にいる利用者は、原理的に同じ映像や音像を共有すること、即ちメディアの対称性を確保することができる。しかし、その質の高い実現のためには、例えば映像エコーの除去や正確な音像再生を行う必要があり、様々な研究開発が行われている [4], [5]。

t-Room どうしの接続、あるいは t-Room を構成するメディア機器の制御は、ディスプレイ上の映像再生を制御するディスプレイ・サーバ(ソフトウェア)やカメラ撮影データの取得を制御するカメラ・サーバ(ソフトウェア)などから成るミドルウェア「t-Room」(慣例的に本体と同名で呼ばれている)によって行われる。本ミドルウェアは、広く普及している商用 OS、Microsoft 社の Windows 上で作られている。確かに、この OS の利用は、t-Room の普及を促進させることが期待させる。しかし、その一方で、OS のバージョン・アップに伴い、研究開発上の要請とは直結しないミドルウェア「t-Room」の改編を余儀なくされたり、t-Room が前提とする実時間双方向通信にはこの OS のメディア処理速度は不十分であるなど [6], [7]、ミドルウェア「t-Room」の開発をこの商用 OS 上で行うことは必ずしも得策ではない。

上記の問題を軽減するため、著者らは、新たにオープンソース OS の Linux 上にミドルウェア「t-Room」を開発すること目指してきた。本稿は特に、その新ミドルウェアの一部である映像伝送システムの設計と実装、その動作の評価結果を報告するものである。以下、まず開発する映像伝送システムに求められる映像同期の必要性を解説し、続いて、映像伝送システムの設計、実装を詳述する。最後に、開発したシステムの動作評価に関して、処理遅延に直結するシステム内部のデータ滞積量を尺度とする定量的な性能調査結果と、利用者による遅延と画質に関する主観評価実験の結果を報告する。

## 2. 映像再生における同期の崩れとその解決

### 2.1 2 地点間の映像再生に求められる同期性

図 1 に 2 地点間(地点 A 及び地点 B)で映像伝送を行う様子を示す。両地点において、利用者は同期的コラボレーション作業を行うことを想定する。図中、縦軸は時間の経過を表し、地点 A と地点 B 間で映像データが送受信される(図中、破線)。この時、地点 A の映像は通信遅延(+

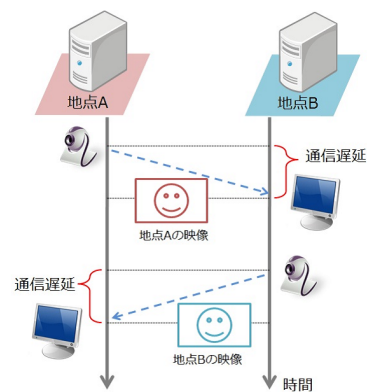


図 1 遅延を伴う 2 地点間の映像伝送。

Fig. 1 Lag-affected video transmission between two cities.

地点 A における処理遅延)を伴って地点 B に届く。地点 B の利用者は遅れた地点 A の映像にタイミングを合わせて行動する。そして、その地点 B の利用者の行動は、更に通信遅延を伴って地点 A に届く。こうして、時間の経過と共に伝送される映像における遅延は増大し、両地点における利用者間の行動の同期性は崩されてしまう。

2 地点間における同期の崩れに対する根本的な解法は、言うまでもなく処理遅延や通信遅延を軽減することである。しかし、そうした遅延の完全な解消は困難であり、遠隔地間の映像伝送には不可避的になんらかの遅延が伴ってしまう。従って、映像伝送サーバの開発における現実的な解は、その動作に伴う遅延量を利用者に許容される範囲にとどめることになる。

一方、実際の t-Room 接続においては、映像データのみならず音データの伝送も行われ、映像データと音データは互いに一定の時間ずれ内におさまる同期が求められる [8]。また、映像や音のメディアを用いる同期的作業においては、遅延がその作業を著しく阻害することも知られている [9], [10]。さらに、文献 [2] におけるシステムとは異なり、t-Room においては、原則的に自己映像が他地点の利用者映像上に重畳されることはないものの、映像オブジェクトを対象とした同期作業を求められることはあり得る。このような視覚的同期作業における遅延の克服のためには、例えば文献 [9] のような、利用者の知覚レベルにおける遅延量の低減を目指す試みもなされている。このような種々の環境状況を考えるとき、t-Room における 2 地点間の映像伝送サーバは、まず、接続地点間における遅延量を容易に制御できる機能を持つことが重要であることがわかる。

### 2.2 多地点間の映像再生に求められる同期性

t-Room の開発においては、多(3以上)地点の t-Room を同時接続し得ることを想定している。その最も簡単な事例として 3 地点間の接続の様子を図 2 に図解する。一般に、接続ポイント間の通信遅延量は異なる。図では、地点 A を基準として、地点 B の映像は 10 ms の遅延を伴い、地

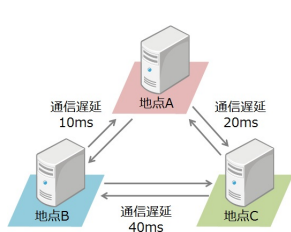


図 2 遅延を伴う  
3 地点間の伝送 .

Fig. 2 Lag-affected transmission among three points.

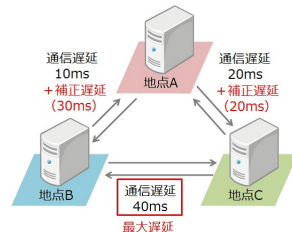


図 3 補正遅延制御を用いた  
3 地点間の接続 .

Fig. 3 Lag-corrected transmission among three points.

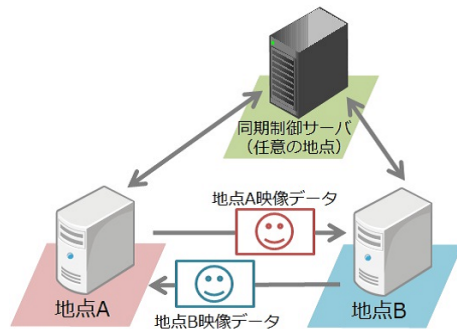


図 4 提案システムの構成例 .

Fig. 4 A configuration example of proposed system.

点 C の映像は 20 ms の遅延を伴って、ディスプレイ表示されると仮定されている。その結果、地点 A における地点 B の映像と地点 C の映像との重畳映像は同期が崩れたものとなる。同様の現象は他の 2 地点においても発生する。また、各地点間の遅延量の相違に起因し、各地点の利用者はそれぞれに異なる時刻の映像を見ることにもなる。当然のことながら、こうした同期の崩れは、利用者の同期的コラボレーションを阻害する。なおここで、各地点における映像表示には、前小節で述べた 2 地点間の同期のずれもさらに加わり、利用者間の同期的作業は一層困難なものとなる。

上述の困難を軽減する解の 1 つとして、地点間（ネットワーク経路上）における最大遅延量によって各経路上の遅延量を置き換える方法を考える [11]。図 2 に示す地点間の遅延を、この手法によって補正する様子を図 3 に図解する。即ち手法は、3 地点間で最も大きな遅延である地点 B-地点 C 間の 40 ms を基準とし、他の地点間の遅延量も同量になるよう遅延を追加する。結果的に、全ての地点間における通信遅延量は同一となり、遅延は大きくなるものの、各地点における利用者は全て同期がとれた映像を共有することが可能となる。

### 3. 映像伝送システム

#### 3.1 概要

前節で述べた同期の崩れに対処する機能、即ち、接続地点間の遅延量に応じて映像表示に伴う遅延量を容易に制御でき、かつ地点間（経路上）の最大遅延量を用いて全地点

における映像表示の同期をとる機能を有する映像伝送システムの設計と実装を行う。

本システムは、映像サーバと同期制御サーバから構成される。図 4 に 2 地点間接続時の構成例を示す。映像サーバは、映像の（カメラやディスプレイとの）入出力や映像データの（映像サーバ間の）送受信など、映像伝送における主要な役割を果たす。一方、同期制御サーバは、多地点接続における同期制御（最大遅延への補正）や映像サーバどうしの通信情報管理などの、補助的役割を果たす。映像サーバは t-Room 設置地点の全てにそれぞれ 1 台ずつ設置し、同期制御サーバは任意の 1 地点に 1 台のみ設置する。

いずれのサーバの開発も、オープンソース OS である Linux (Fedora14) 上で行う。プログラミング言語は C++ である。また映像処理に関しては、これもオープンソースの OpenCV を用いる。

効率よく映像データ（動画）の伝送を行うためには、例えば MPEG などによるデータ圧縮を採用することが望ましい。しかし、t-Room そのものが研究開発段階にあり、映像を構成する各静止画（フレーム）に対する様々な画像処理の調査検討が行われることを想定し、この想定とデータ圧縮による処理量削減とのバランスを考慮し、本システムでは映像を Motion JPEG で表現するものとする。また送受信の安定性を考慮し、映像データの送受信プロトコルは TCP とする。

本映像伝送システムは文献 [11] の遠隔合奏支援システムの構成を参考にしているが、両者にはリングバッファ内のデータ単位（ブロック）の考え方に相違がある。映像データは撮影も表示も基本的に動画の構成要素であるフレーム単位で扱われるため、本映像伝送システムではバッファのブロック長をフレームサイズと同一とする。一方、音データは固定長（例えば 512 バイト）のブロック単位で切り出され、それが順次、送信、受信、そしてリングバッファ経由の出力へと受け渡される。そこでは、映像のフレームのように一定量のデータをまとめて処理をすることはない。

#### 3.2 映像サーバの構造

図 5 に映像サーバの機能的構成を図解する。本サーバは、カメラから撮影データを取得し、それを受信地点の映像サーバに送信する機能と、映像データを受信し、それをディスプレイに出力する機能とを併せ持つ。より具体的には、カメラから撮影データを取得し、それを送信する機能は Input Thread として、送信されてくるデータを受信する機能は Receive Thread として、そして受信した映像データをディスプレイに出力する機能は Output Thread としてそれぞれ実装する。また、Receive Thread と Output Thread のデータの受け渡しは、リングバッファ（1 フレームを転送単位の 1 ブロックとする 256 ブロックからなる）を介して行う。一般に、ネットワークからの受信とディス

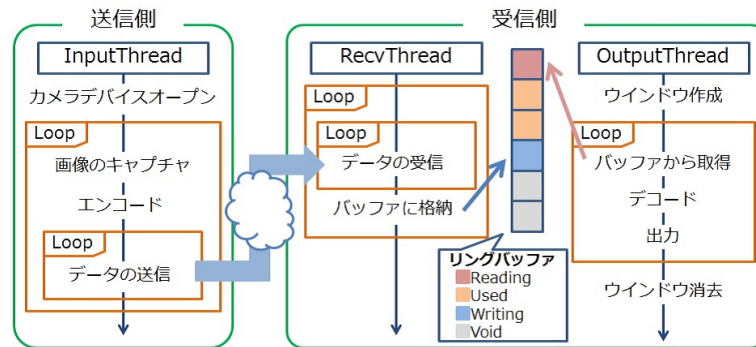


図 5 映像サーバの構造 .

Fig. 5 Architecture and data flow of proposed video server.

プレイへの出力などの入出力処理の処理速度は異なり、さらに OS の管理下におかれるこれらのスレッドの動作は時間的なずれを避けえない。リングバッファを介したデータ受け渡しは、これらのスレッド間の時間的ずれを吸収する役割を果たす。またさらに、この受け渡しは、前小節で述べた同期制御にも用いる。即ち、補正遅延の追加は、このバッファにデータを滞積させることで実施する。

### 3.3 映像品質の制御属性

本映像伝送システムあるいは映像サーバでは、映像のサイズ（撮影用カメラから得られる映像データのキャプチャサイズと等価）と JPEG 圧縮時の Quality 値の 2 つを映像品質を制御する変数とする。サイズも Quality 値も大きいほど映像の質は高いものの処理に伴う遅延が増え、サイズも Quality 値も小さいほど質は低下するものの処理遅延は低減する。これらの 2 つの変数の適切な設定値は本システムの実装に依存するものであり、システムの動作実験を通して見出す必要がある。

なお、上記の 2 つの変数によって遅延量が変動するため、今回の実装では映像の FPS (Frame Per Second) 値は制御変数としては用いないものとする。上記の 2 種の変数を用いる制御の結果として、FPS 値は自動的に決定される。

## 4. 動作確認実験

### 4.1 背景と目的

本稿では、3 地点以上の多地点接続における同期制御機能の動作確認に先立ち、映像サーバ単体の動作確認とその処理遅延量の評価に焦点を合わせる。そこでまず、映像サイズ（キャプチャサイズ）と JPEG の Quality 値を制御変数として映像サーバ内のリングバッファの滞積量を測り、本サーバの動作の安定性や処理遅延の大きさを調査した。

### 4.2 映像サーバの実装条件及び実験環境

提案する映像サーバを実装するコンピュータやメディア機器等の仕様と実験に用いる通信環境の詳細を以下に記す。

映像サーバ 映像の入出力を行うサーバの仕様を表 1 に示す。本実験では、表 1 に示す仕様に従う 2 台のサーバを実装、利用した。

表 1 映像サーバの仕様 .

| 機種     | Dell XPS 8500                         |
|--------|---------------------------------------|
| OS     | Fedora 14                             |
| CPU    | Intel®Core™i-7-3770 Processor 3.40GHz |
| メモリ    | 8GB DDR3 SDRAM                        |
| ビデオカード | NVIDIA®GeForce®GT 640                 |
| コア数    | 4                                     |

入力デバイス 入力デバイスとして Web カメラ、Logicool®HD Pro Webcam C920 を用いた。フル HD 1080p (1920 × 1080) で、最大 FPS 値は 30 であった。  
 出力デバイス 出力デバイスとして、t-Room 内に備え付けてあるディスプレイを使用した。画面解像度 1360 × 768 の 40 インチの液晶ディスプレイであった。  
 通信環境 サーバ本体の性能評価に焦点を合わせ、ネットワークの通信遅延が評価の妨げになることを回避するため、LAN 環境において測定を行った。LAN はギガビットイーサネットであり、その通信速度は最大で 1Gbps であった。

### 4.3 実験手順

2 台の映像サーバ間で双方向の映像伝送を行い、受信側のバッファ滞積量を測定した。キャプチャサイズを “1920 × 1080”, “1600 × 896”, “1280 × 720”, “1024 × 576”, “640 × 360” の 5 種とし、また Quality 値を 100, 80, 60, 40, 20 の 5 種とし、それらを組み合わせた 25 条件において、それぞれ 1 時間分の滞積量を測った。なお、入力映像に一貫性を持たせるため、被写体は t-Room 内の中央に吊るしたぬいぐるみ (図 6) とした。部屋の明るさやカメラ位置などの環境条件は全て固定した。

### 4.4 実験結果：バッファ滞積量の測定結果

キャプチャサイズ毎に Quality 値を変動させた際のバッ



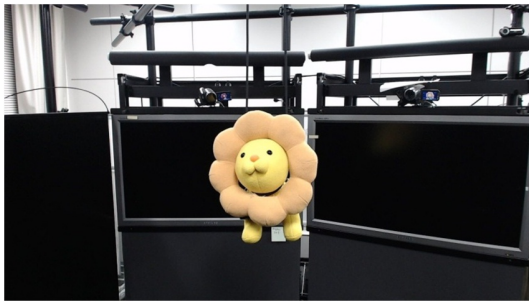


図 6 撮影対象 .

Fig. 6 Shooting subject.

ファスタ積量の変化を、キャプチャサイズの降順に図 7 から図 12 に示す。縦軸はバッファ積量を表し、横軸は観測の経過時間を表す。なお、図 10 は、キャプチャサイズ“1280 × 720”における Quality 値が 80-20 のケースを抜粋したものである。

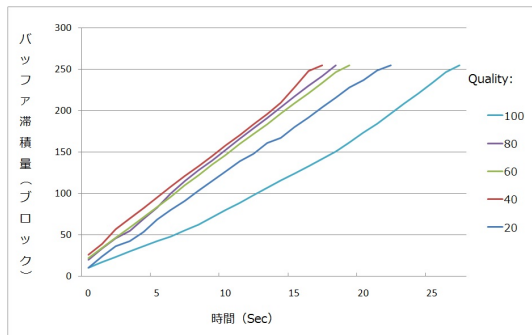


図 7 映像サイズ“1920 × 1080”におけるバッファ積量 .

Fig. 7 Amount of buffered data for "1920 × 1080".

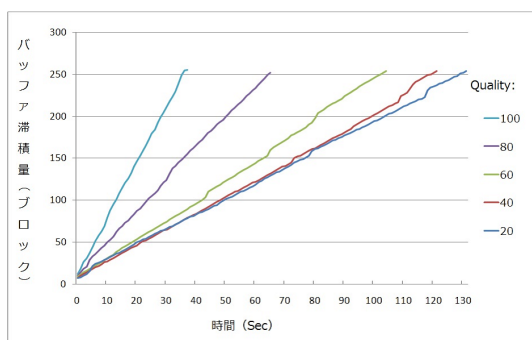


図 8 映像サイズ“1600 × 896”におけるバッファ積量 .

Fig. 8 Amount of buffered data for "1600 × 896".

理想的には、Receive Thread がバッファに 1 ブロック分のデータを書き込む毎に、そのデータは Output Thread によって取り出され、ディスプレイに出力されることが望まれる。リングバッファの滞積量が大きくなればディスプレイ表示に至るまでの遅延が大きくなる。

図 7 と図 8, 図 9 より, “1920 × 1080” と “1600 × 896”

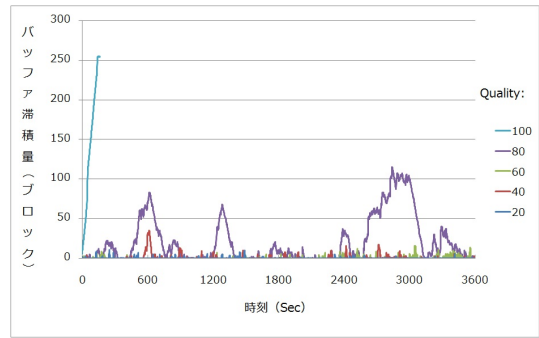


図 9 映像サイズ“1280 × 720”におけるバッファ積量 (Quality 値: 100-20) .

Fig. 9 Amount of buffered data for "1280 × 720" (Quality:100-20).

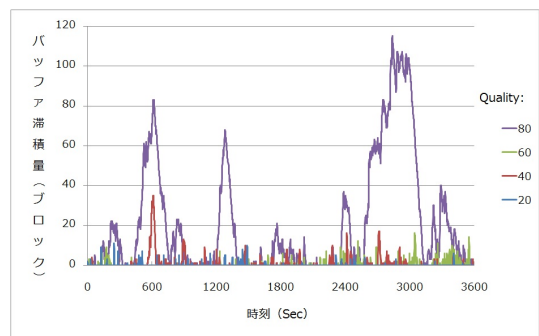


図 10 映像サイズ“1280 × 720”におけるバッファ積量 (Quality 値: 80-20) .

Fig. 10 Amount of buffered data for "1600 × 896" (Quality:80-20).

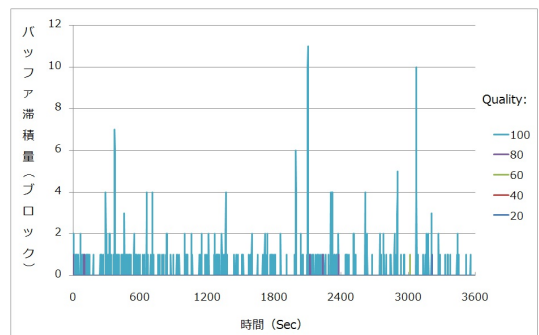


図 11 映像サイズ“1024 × 576”におけるバッファ積量 .

Fig. 11 Amount of buffered data for "1024 × 576".

における全ての Quality 値, 及び “1280 × 720” における Quality 値が 100 の場合では、滞積量がデータ伝送開始と同時に急激に上昇し、短時間のうちにバッファサイズを越えてしまった。即ち、これらの条件下では本サーバを長時間利用することは困難なことになる。また “1280 × 720” において 1 時間の継続的測定が可能であった Quality 値が 80 から 20 の場合の結果を拡大的に図示する図 10 によれば、これらの条件下においては滞積量が時間とともに大幅に変動する、即ち遅延量が大きく変動するものの、継続的利用が可能であることを読み取ることができる。一方、映

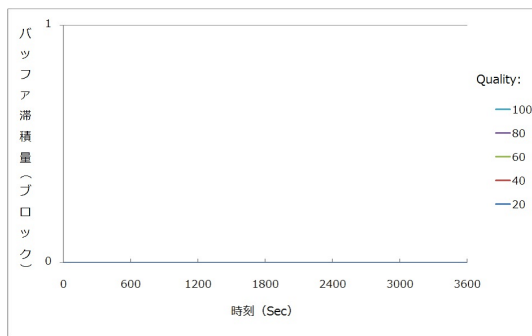


図 12 映像サイズ “640 × 360” におけるバッファ滞積量 .  
 Fig. 12 Amount of buffered data for “640 × 360”.

像サイズが比較的小さな “1024 × 576” では全ての Quality 値に対して滞積量はせいぜいでも 10 程度 (10 フレーム程度) であり, “640 × 360” では全ての Quality 値において滞積量は 0, 即ちリングバッファに滞積することに伴う遅延はない良好な結果が得られた .

#### 4.5 Thread 優先度の制御による追加実験とまとめ

高質な映像を得るためには, 映像サイズ (キャプチャサイズ) も Quality 値も大きいことが望まれる . 前小節の結果から, これらの制御変数に関して本実装で求め得る最も望ましい設定は “1280 × 720 (Quality 値 = 80)” であると判断できる . しかし, 長時間利用における遅延量は大きく変動し, 安定的な利用には十分とはいえない .

上述の点を精査するため, Output Thread の優先度を高くすることが及ぼす滞積量, 言い換えれば遅延量軽減への影響を調査した . 具体的には, Linux における priority 設定を最大にし, 他の動作環境や手順はこれまでのものと同一とした . 3 回の測定を行い, その結果を図 13 に示す .

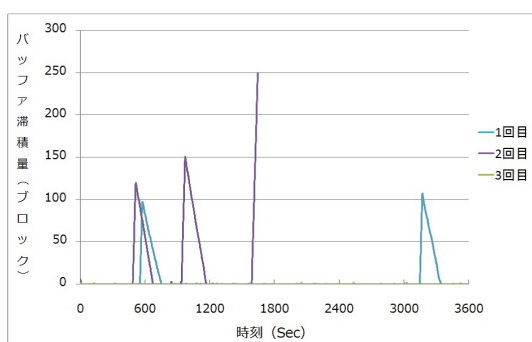


図 13 出力最優先時のバッファ滞積量 .  
 Fig. 13 Amount of buffered data with setting Output Thread top priority.

図中, 1 回目の測定の滞積量は, 100 前後まで急上昇し, すぐに下降, その後はほぼ 0 に留まった . 2 回目の測定では, 1 回目と同様の現象が生じ, バッファサイズを越えてしまった . 一方, 3 回目の測定では, 常にほぼ 0 に留まった . 図 10 と比べ, Output Thread の動作優先度を上げる

ことで, 滞積量が常に変動することは軽減されたが, 試行毎に異なる現象が現れ, この条件による安定な利用はやはり困難であることがわかった . 3 回目の試行に見られるような理想に近い結果を得ることができたことから, 試行毎に異なる結果に至る原因を解明することで, 本条件による長時間利用も可能であるように思われる . この点は, 今後の課題である .

以上の結果から, 本映像サーバを用いた長時間通信に適用可能な映像サイズと Quality 値との組み合わせは明らかとなった . ここで, 各フレーム映像に高い画質を求める上ではいずれの変数も大きいにこしたことはない . しかし, いずれの変数値も大きくなれば, 仮にバッファ滞積量が大きくならなくても, サーバ内における処理遅延やネットワーク通信遅延は大きくなり, 動画としての質が高くなるとは限らない . 従って, 動画映像の伝送を行う映像サーバに求められるこれらの変数の適切な設定値は, 画質と遅延に関する総合的な評価によって選ばれる必要がある . こうした観点に立ち, 次の節では, 安定した長時間利用が可能な範囲で設定できる比較的大きな変数値を採用し, それらの組み合わせ条件下における映像サーバの性能を主観的に評価し, 望ましい変数値設定の組み合わせを明らかにする . 採用した変数値の組み合わせは, “1280 × 720” かつ Quality 値 = 60, 40, “1024 × 576” かつ Quality 値 = 100, 80, 60, 40 の 6 条件である .

#### 4.6 システム内の処理遅延の測定

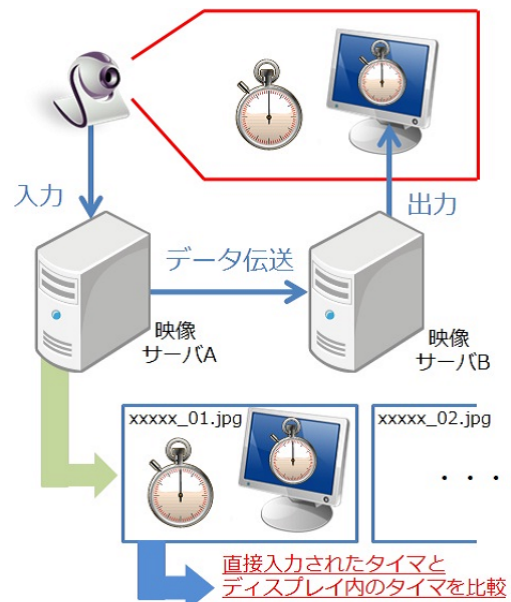


図 14 システム内遅延の測定方法 .  
 Fig. 14 Method of measuring processing delay in proposed server.

前小節で調べたリングバッファ中のデータ滞積は, その量に応じた伝送遅延を引き起こす . しかし当然ながら, 仮

にバッファ内滞積が0であっても、システム内のデータ処理に起因する処理遅延が発生する。この遅延量を測るため、バッファ内滞積がほぼ常に0である“1024 × 576 (Quality 値 = 40)”の条件下で図 14 に示すような計測実験を行った。図中、映像サーバ A はタイマそのものと映像サーバ B のディスプレイを同時に撮影し、そのデータを映像サーバ B に伝送する。映像サーバ B のディスプレイ上に映る、タイマそのものとディスプレイ内のタイマとして重ねあわされて撮影されるタイマとの時刻の差が処理遅延量となる<sup>\*1</sup>。

以上の方法で計測した結果、“1024 × 576 (Quality 値 = 40)”の条件における処理遅延は平均でおよそ 70ms であることがわかった。他の 5 条件では、基本的にこの条件の場合よりもデータサイズが大きくなることが予想される。従って、この遅延量が本伝送システムのおおよその最小処理遅延量ということになる。

## 5. 主観評価実験

### 5.1 実験内容

実際に映像伝送を行い、利用者によってその表示映像の品質に関する主観評価を行う実験の詳細を記す。

#### 実験環境

使用する計算機やマルチメディア機器、通信環境は 4.2 におけるものと同一であった。

#### 通信遅延のシミュレーション

映像表示に至るまでの処理遅延には伝送サーバ内の処理遅延にネットワークに起因する通信遅延も含まれる。この通信遅延を実験的に制御するため、開発したサーバのリングバッファに制御遅延量に相当する架空フレームデータを滞積させて映像表示に遅延を発生させる方法を考案し、実装した。例えば FPS= 30 の場合、30 フレームの架空データをリングバッファに滞積させることで 1 秒の遅延を発生させることができる。

#### 実験参加者

実験参加者は、情報系学科の 10 名の大学生及び大学院生であった。

#### 評価項目

実験参加者には、遅延を感じる程度と画質の劣化を感じる程度を、それぞれ「5: かなり感じる」、「4: 少し感じる」、「3: どちらとも言えない」、「2: あまり感じない」、「1: 全く感じない」の 5 段階で評定させた。

#### 映像伝送制御条件

前小節末尾に示した映像サイズと Quality 値に関する 6 条件から成る映像そのものに関する制御条件に、通信遅延を模擬する 12 条件を掛け合わせ、合計 72 の制御条件を設定した。ここで、12 の通信遅延量は、2 フ

レーム単位で 0 フレーム (通信遅延なし) から 24 フレーム (FPS= 30 の場合、0.8 秒の遅延発生に相当) までリングバッファ滞積量を増やして作成した。

#### 評価方法

映像サーバが撮影する被写体を実験参加者自身とした。参加者は、映像伝送システムが搭載された t-Room に入り、そのカメラの前で手振り動作 (図 15) を行い、ディスプレイにフィードバック的に再生される映像中の自身の動きを目視し、その遅延と画質に関する評定を回答した。無作為に順番づけられた上述の 72 の制御条件のそれぞれに従った映像サイズと Quality 値、通信遅延量の設定を行った。その条件下で実験参加者は、15 秒間の手振り動作を行い、それに続く 15 秒のインターバル間に評定した。なお、実験参加者の疲労を回避するため、合計 40 分に及ぶ実験の全プロセスは前半と後半の 20 分ずつに分けて行った。



図 15 主観評価実験の様子。

Fig. 15 An example scene of subjective assessment.

### 5.2 結果と考察

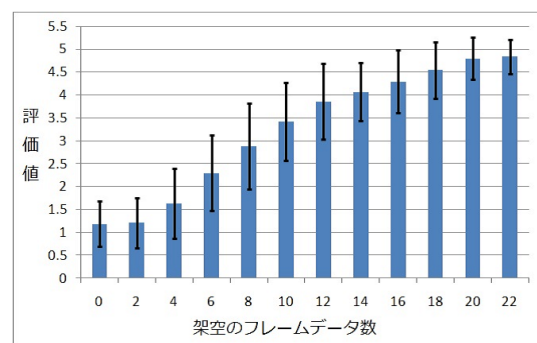


図 16 制御遅延量を変数とする評価値。

Fig. 16 Opinion scores with different delay amounts.

遅延と画質に対する主観評価値の平均値と標準偏差をそれぞれ図 16 と図 17 に示す。

図 16 は、縦軸は評価値の平均を表し、横軸は架空のフレームデータ数、即ち制御遅延量の大きさを表す。図より、架空のフレーム数を増やすにつれ、評価値が高くなっていく (遅延を感じやすくなっている) ことがわかる。平均オ

<sup>\*1</sup> なお、図中のディスプレイ上のタイマ映像は、理解の便宜のために簡単化している。実際には多数のタイマ映像が重ねあわされることに注意されたい。



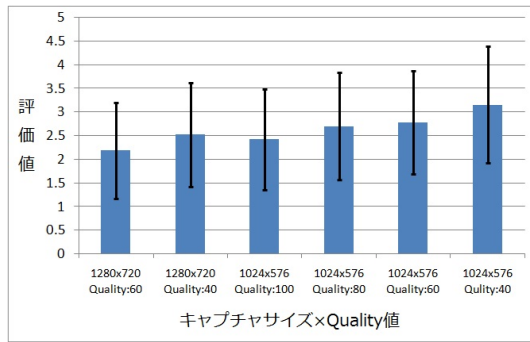


図 17 設定条件毎の評価値 .

Fig. 17 Opinion scores for different control conditions.

ピニオン評定の評価基準に従うと、MOS= 2.5 に対応する 8 フレームを超える滞積で遅延を感じない (評定の 2 及び 1) 評定者はほぼ半数に減り、MOS= 3.5 に対応する 12 フレーム分を超えると遅延を感じない (評定の 2 及び 1) 評価者はわずか 10 %程度になることになる .

図 17 では、縦軸は評価値の平均を表し、横軸はキャプチャサイズと Quality 値による設定条件を示している . 条件間の差異は小さかったものの、映像サイズも Quality 値も大きい “1280 × 720 (Quality 値 = 60)” が最も評価が良く、映像サイズも Quality 値も小さな “1024 × 576 (Quality 値 = 40)” の評価が最も低い結果となった .

本実装において、FPS 値は自動的に定まる設定になっていたが、結果的にいずれの条件下における伝送においても全て FPS= 30 であった . 従って、架空の 1 フレーム分の対応する遅延量はおよそ 33ms となり、例えば上述の 8 フレームの場合の総遅延量は少なくとも 330ms 程度になる . これよりも小さな遅延に抑えることで、本システムは半数程度の利用者に遅延を感じさせない伝送を行うことが期待できる .

画質に関する最良と最悪の評価結果 (図 17 中の両端の場合) に対する遅延評価結果を精査した (図 18 参照) . 図から、画質の評価が分かれても遅延評価にはあまり大きな差異がないことがわかる . 従って、本システムを動作させる場合の条件として、少しでも画質の良い “1280 × 720 (Quality 値 = 60)” を採用することが推奨されることがわかる .

## 6. おわりに

Linux 上で動作する t-Room のための新しいミドルウェアの開発を目指し、映像伝送システムの設計と実装、そしてその動作実験を行った . その結果、本映像伝送システムを安定的に用いるために必要な映像サイズ等の制御条件が明らかになり、またシステムの処理遅延量、またシステムが提供する映像の画質と遅延に関する主観評価結果も明らかになった .

謝辞 本研究を進めるにあたり、システムの設計から開

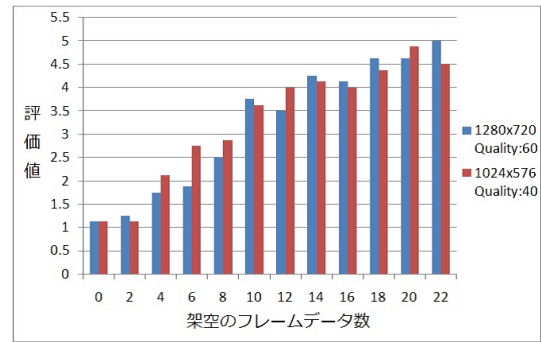


図 18 最良と最悪の画質に伴う遅延の評価 .

Fig. 18 Opinion scores for the best and worst video quality evaluations.

発に至るまで数多くの御助言を頂きました山口毅氏と竹森幸輝氏に深く感謝致します .

## 参考文献

- [1] Bly, S. A., Harrison, S. R. and Irwin, S. *Media spaces: bringing people together in a video, audio, and computing environment*, Communications of the ACM, 36, 1, pp.28-46 (1993).
- [2] 森川治: 超鏡: 魅力あるビデオ対話方式を目指して, 情報処理学会誌, 41, 3, pp.815-822 (2000).
- [3] Hirata, K., Harada, Y., Takada, T., Aoyagi, S., Shirai, Y., Yamashita, N., and Yamato, J.: *The t-Room: Toward the Future Phone*, NTT Technical Review, Vol. 4, No. 12, pp. 26-33 (2006).
- [4] 中村譲, 片桐滋, 大崎美穂: 遠隔協働支援システム「t-Room」における映像オブジェクト抽出法の改良, 信学会信学技報 ITS2011-56, IE2011-132, pp.325-330 (2012).
- [5] Pablo Nava, G., Hirata, K., and Miyoshi, M.: *A loudspeaker design for sound image localization on large flat screens*, Acoust. Sci. & Tech., 31, 4, pp.278-287 (2010).
- [6] 井ノ口浩平: 遠隔コラボレーション支援システム「t-Room」におけるメディア間同期について: 音声遅延量について, 同志社大学 理工学部 情報システムデザイン学科 卒業論文 (2012).
- [7] 李榮宰: 遠隔コラボレーション支援システム「t-Room」におけるメディア間同期について: 映像遅延量について, 同志社大学 理工学部 情報システムデザイン学科 卒業論文 (2012).
- [8] 鏝沢勇, 滝川啓, 大久保栄, 渡部義郎: 衛星通信を利用した画像会議におけるエコー及び伝搬遅延の影響, 信学論, J-64-B, 11, pp.1281-1288 (1981).
- [9] Stuckel, D., and Gutwin, C.: *The Effects of Locallag on Tightly-Coupled Interaction in Distributed Groupware*, Computer Supported Cooperative Work, pp.447-456 (2008).
- [10] 入江洋介, 青柳滋己, 高田敏弘, 平田圭二, 梶克彦, 片桐滋, 大崎美穂: t-Room のための遠隔合奏支援システムの構築, 情報処理学会, グループウェアとネットワークサービス研究会, 2009-GN-73 (23) (2009).
- [11] 竹森幸輝: ローカル・ラグ制御を用いた遠隔合奏支援システムの開発, 同志社大学大学院 工学研究科 情報工学専攻 修士論文 (2012).