

# ライブ放送のための分散型映像処理システムの設計と評価

義久智樹<sup>†1</sup> 川上朋也<sup>†2</sup> 石芳正<sup>†1</sup> 寺西裕一<sup>†3, †1</sup>

**概要:** 近年, USTREAM やツイキャスといった, 個人がインターネットを介してリアルタイムな映像配信を行えるインターネットライブ放送サービスが普及している. インターネットライブ放送では, ライブ放送を盛り上げるために映像効果を付加する処理を行うことがある. しかし, 個人が所有するパソコンやスマートフォンの計算能力では複雑な映像処理に時間がかかり, 放送者の所望のタイミングに映像効果を付加できなかつたり, 短時間で処理できる簡易な映像効果しか付加できないといった問題があった. 計算能力の高い計算機を用いることで映像処理にかかる時間を短縮できるが, 高価なため個人で導入しにくい. そこで, 本研究では, インターネットライブ放送のための分散型映像処理システムを提案する. 提案システムでは, サービス提供者が提供することを想定した計算能力の高い映像処理サーバを用意し, 放送者の映像撮影端末で撮影した映像を映像処理サーバに配信する. 映像処理サーバが処理した映像を配信することで, 複雑な映像処理にかかる時間を短縮しつつ, 放送者は映像効果を付加できる. 設計, 実装したシステムを用いて評価を行った結果, 提案システムを用いることで映像処理時間を短縮できることを確認した.

**キーワード:** ストリーミング配信, インターネット放送, ビデオオンデマンド, リアルタイム

## Design and Evaluation of a Distributed Video Processing System for Internet Live Broadcasting

TOMOKI YOSHIHISA<sup>†1</sup> TOMOYA KAWAKAMI<sup>†2</sup>  
YOSHIMASA ISHI<sup>†1</sup> YUUCHI TERANISHI<sup>†3, †1</sup>

**Abstract:** Recently, Internet live broadcasting services such as USTREAM, TwitCasting, and so on, in which individuals can deliver the real time video via the Internet have been popular. In Internet live broadcasting, video effects are often processed to make the live broadcasting exciting. However, the computational power of laptops or smart phones for consumers is not enough for complex video processing. So, the problems that live broadcasters cannot add video effects at their desirable timings or that available effects are restricted in only simple effects arise. Although the processing time can be reduced by using high computational power machines, they are expensive and difficult for individuals to adopt. Hence, in this paper, we propose a distributed video processing systems for Internet live broadcasting. Our proposed system equips high computational power servers installed by service providers. The video recorder clients of live broadcasters deliver their recorded video to the server and the servers deliver their processed video to the clients. The live broadcasters can add video effects while reducing the complex video processing time. As a result of our evaluations, we confirmed that our proposed system can reduce the video processing time.

**Keywords:** Streaming Delivery, Internet Broadcasting, Video on Demand, Real Time

### 1. はじめに

近年の映像配信技術の発達に伴い, USTREAM やツイキャスといった, 個人がインターネットを介してリアルタイムな映像配信を行えるインターネットライブ放送サービスが普及している ([1]). 例えば, USTREAM では, 個人のパソコンで USTREAM のホームページにブラウザでアクセスし, 放送を開始するボタンをクリックすることで, パソコンに接続されたカメラで撮影された映像を配信できる. ツイキャスでは, 主にスマートフォンからの配信を対象としており, 専用アプリを用いてスマートフォンのカメラで撮影された映像を配信できる. 視聴者は, インターネットブラウザや専用アプリの画面に列挙された配信中のインターネットライブ放送の中から興味のあるライブ放送を選択

し, 配信されている映像をブラウザや専用アプリで視聴する.

インターネットライブ放送サービスでは, 放送者が映像効果を付加する下記のような場合がある.

- 写ってる人を指して名前を表示したり, 物を囲んで商品名を表示するなど, 付加情報を表示するために, 写っている物体を検出する.
- 楽しい時に音符マークを表示したり, 驚いた時にびっくりマークを表示するなど, ライブ放送を盛り上げるために, あらかじめ撮影した映像を適切な位置に重畳表示する.
- 明るさを調整したり, 移動しながら自分を撮影して配信している場合に画像の中心以外をぼかすなど, ライブ放送しやすくするために, 撮影した画像に修正を加

<sup>†1</sup> 大阪大学

Osaka University

<sup>†2</sup> 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

<sup>†3</sup> 情報通信研究機構

National Institute of Information and Communications Technology

える。

これらの映像処理を短時間で行うことで、より多くの付加情報や重畳表示を行ってさらにライブ放送を盛り上げられたり、鮮明な画像処理を行ってライブ放送しやすくなったりする。しかし、現状では、個人が所有するパソコンやスマートフォンの計算能力に応じて、短時間で処理できる簡易な映像効果しか付加できなかった。例えば、物体検出に時間がかかるため、あらかじめ付加情報を表示させておいて、付加情報が表示されている位置に人や物が写るようにすることがあった。複雑な映像効果を付加すると、放送者の所望のタイミングに映像効果を付加できなかったり、映像配信のビットレートが下がるといった問題が発生する。計算能力の高い計算機を用いることで映像処理にかかる時間を短縮できるが、高価で個人で導入しにくい。

映像処理時間を短縮するために、映像撮影端末とは別の計算機で映像処理を行う分散型映像処理システムに関する研究が幾つか行われている。上記の例のように、インターネットライブ放送では、放送者の要求に応じて様々な映像効果を付加することがあり、「放送者任意のタイミングに映像効果を付加できる」「放送者が所望の映像効果を行える」といった機能が要求される。既存のシステムは、複雑な映像効果を放送者の要求に応じて付加することをサービスとして考えていなかった。このため、符号化や時計の追加といった既定の映像処理しかできず、これらのインターネットライブ放送の要求を満たせない。他の関連研究については2章で詳述する。

そこで、本研究では、インターネットライブ放送のための分散型映像処理システムを提案する。提案システムでは、サービス提供者が計算能力の高い映像処理サーバを提供する。また、放送者が準備した任意の映像処理を行うプログラムライブラリである映像効果ライブラリを送信する。プログラムライブラリとは、関数が実装されたバイナリファイル（WindowsのDLLファイルやJava VMのclassファイル）であり、プログラムで読み込み、実装された関数を呼び出すことで処理を行う。ライブ放送時には、放送者の映像撮影端末（パソコンやスマートフォン）で撮影した映像を映像処理サーバに送信する。映像効果を付加する場合には、放送者は映像処理を映像処理サーバに要求し、映像処理サーバは、要求に応じてあらかじめ送信された映像効果ライブラリを用いて映像効果を付加し、処理した映像を配信する。複雑な映像処理を計算能力の高い計算機で行うことで、映像処理にかかる時間を短縮しつつ、放送者は映像効果を付加できる。

以下、2章で関連研究について説明し、3章で本研究で提案する分散型映像処理システムの設計について述べる。4章で実装した提案システムについて説明し、5章で評価を行う。最後に6章で本稿をまとめる。

## 2. 関連研究

本章では、本研究に関連する研究について説明する。

映像撮影端末とは別の計算機で映像処理を行う分散型映像処理システムに関する研究が幾つか行われている。MediaPaaSでは、ライブ放送のためのPaaS(Platform as a Service)形式のクラウドコンピューティングサービスを目的とし、映像の符号化、再符号化、配信および画像処理をサービス提供者側の計算機で行えるシステムを提案している([2])。画像処理の内容は時刻を表示する、ロゴを表示するといったインターネットライブ放送中に一貫して行われる内容であり、放送中に放送者の要求に応じて行えない。文献[3]では、P2Pネットワークを用いてライブ放送を行う場合に、通信状況に応じて映像の再符号化を行う手法を提案しているが、放送者が明示的に再符号化を指定できない。また、静止画ではあるが、PictuARでは、撮影した静止画を動画サーバに転送して解析し、関連ある動画を検索している([4])。本研究で提案するシステムでは、放送者がプログラムした任意の映像処理を、要求に応じて行える点が異なる。

また、インターネットライブ放送のための映像配信システムに関する研究も行われている。文献[5]では、パノラマ撮影された映像から、あたかもカメラでズームやパンを行ったような映像効果を与えて配信できるシステムを実装している。文献[6]ではP2Pネットワークを用いてライブ放送を行う場合に、少し前から視聴（追っかけ再生）できるシステムを実装している。これらの映像配信システムでは、放送者が任意の映像処理を行えない点が本研究で提案するシステムとは異なる。

インターネットライブ放送において、映像配信の遅延時間を削減する幾つかの手法が提案されている。SmoothCache 2.0では、P2Pネットワークを用いてライブ放送を行う場合に、他のピアに映像データをキャッシュし、キャッシュしたピアから配信することで映像撮影端末にかかる通信負荷を低減させて遅延時間を削減する手法を提案している([7])。文献[8]では、遅延時間を理論上最小にするP2Pネットワーク上の配信経路決定手法を提案している。また、HD法では、1対1の通信と同時に、1対多の放送型配信を用いて同時に複数の視聴端末に映像データを送信することで、通信量を削減している([9])。提案システムにおいても、映像配信時にこれらの遅延時間短縮手法を適用できるが、本研究とは映像処理システムを対象としている点が異なる。

さらに、保存された映像データに対して映像処理を行うシステムに関する研究が行われている。文献[10]では、カメラで撮影した映像データを、高い計算能力がある計算機に転送して映像処理を行うシステムを提案している。文献[11]では、スマートフォン等の計算能力が低い映像撮影端

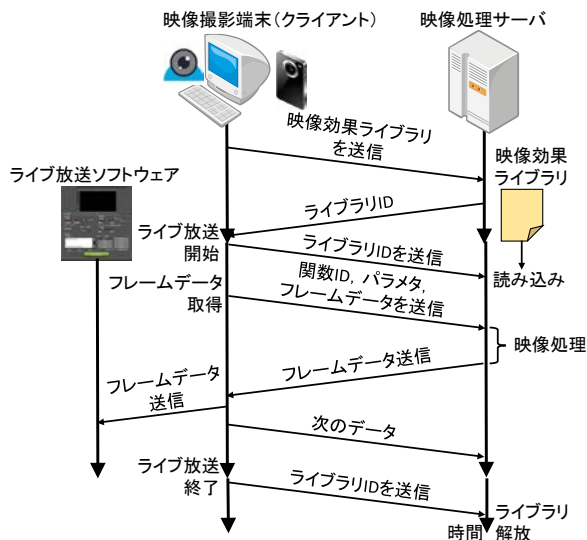


図1 システム構成と通信手順  
Figure 1 The system architecture and the communication process

末で撮影した映像を映像撮影端末に保存せずに直接、クラウドストレージ等の外部記憶装置に保存するシステムを提案している。これらのシステムは、保存された映像データを対象としており、インターネットライブ放送では利用できない。

### 3. 分散型映像処理システムの設計

本章では、提案する分散型映像処理システムの設計について説明する。提案システムでは、従来映像撮影端末で行ってきた映像処理を複数の映像処理サーバに分散させることを想定している。

#### 3.1 システム構成

分散型映像処理システムのシステム構成を図1に示す。映像撮影端末は、インターネットライブ放送の放送者が映像を撮影する計算機である。映像撮影端末は、接続されたカメラで映像を撮影でき、映像のデータをインターネットを介して他の計算機に送信できる。映像処理サーバは、インターネットライブ放送サービス提供者が設置することを想定した高い計算能力をもつ計算機であり、インターネットを介して映像データおよび映像効果ライブラリの送受信を行える。映像効果ライブラリは、放送者が用意した任意の映像処理を行えるプログラムライブラリであり、映像処理サーバのプログラムで読み込んで実装された関数を呼び出すことで実行できる。詳細は次節で説明する。例えば、放送者の所望の物体検出や重畳表示、画像処理を行う映像効果を実装した映像効果ライブラリが考えられる。ライブ放送ソフトウェアは、インターネットライブ放送サービス提供者が公開するライブ放送のためのソフトウェアであり、映像撮影端末から実行できる。インターネットブラウザを

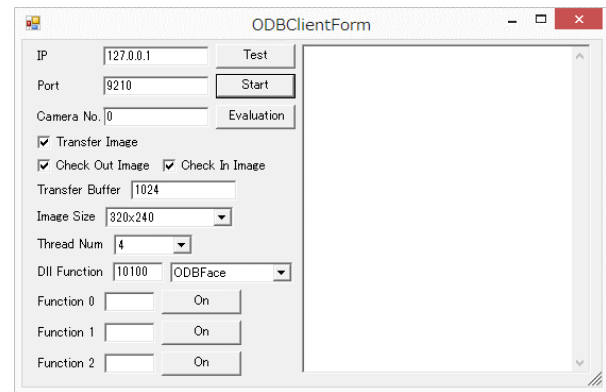


図2 映像撮影端末ソフトウェアのスクリーンショット  
Figure 2 A screen shot of the video recording node software

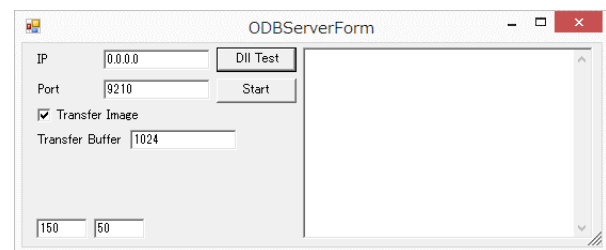


図3 映像処理サーバソフトウェアのスクリーンショット  
Figure 3 A screen shot of the video processing server

介して実行するライブ放送ソフトウェアもある。

近年のクラウドコンピューティング環境の普及に伴い、インターネットライブ放送サービス提供者が高性能な計算機を設置し、放送者が利用する上記のようなシステム構成は現実的な構成といえる ([2])。

#### 3.2 要求機能

本研究では、以下をインターネットライブ放送のための分散型映像処理システムへの機能要求と考える。

- 放送者の任意のタイミングに映像効果を付加できる：1章で述べた例のように、ライブ放送中に一貫して付加される映像効果とは別に、付加情報を表示したり、放送を盛り上げたりするために、放送者の任意のタイミングで映像効果することがある。例えば、放送者の知人が写っている間、名前を表示させたり、放送者が楽しいと思ったときに音符マークを表示したりすることが考えられる。これらを実現するためには、放送者の任意のタイミングに映像効果を付加できる必要がある。
- 放送者が所望の映像効果を付加できる：ロゴを表示したり時計を表示するといった簡易な映像効果ではなく、放送者の所望の任意の映像効果を付加することで、よりライブ放送を盛り上げたり、ライブ放送しやすくなる。例えば、放送者独自の画像アイコンを表示させてライブ放送を盛り上げたり、放送者の所望の明るさに映像を調整したりすることが考えられる。これらを実現

するためには、放送者が所望の映像効果を付加できる必要がある。

### 3.3 映像効果ライブラリ

本研究で提案する分散型映像処理システムでは、放送者が所望の映像効果を付加できるように、放送者が映像効果を実装した映像効果ライブラリを用意する。映像効果ライブラリはプログラムライブラリであり、様々な映像効果を実装できる。ライブ放送中に映像効果ライブラリを映像処理サーバに送信すると、通信帯域が低下するため、ライブ放送前に映像処理サーバに送信する。複数の放送者が送信した映像効果ライブラリを識別するため、ライブラリ ID を設定する。放送者は、映像効果を付加したいタイミングに、ライブラリ ID で指す映像効果ライブラリに実装されている複数の関数の中から、付加したい映像効果を示す関数 ID を選択する。関数 ID が示す関数の引数として渡すパラメータも渡せる。

映像処理サーバは、放送者からあらかじめ映像効果ライブラリを受信して保存しておく。映像効果ライブラリの読み込み時間を削減するため、映像処理サーバは、ライブ放送が開始されると、プログラムで放送者が送信した映像効果ライブラリを読み込む。映像撮影端末からフレームデータを受信時にライブラリ ID が指定されていると、映像効果ライブラリに実装されている関数の中から対応する関数呼び出し、放送者の所望の映像効果を付加する。ライブ放送が終了すると、読み込んでいた映像効果ライブラリを開放する。

### 3.4 映像撮影端末

インターネットライブ放送では、通信量を削減するために映像の幾つかのフレームのデータをまとめて他の計算機に送信することがあるが、送信するフレームの取得を待つため、遅延時間が長くなる。遅延時間が長くなると、放送者の任意のタイミングに映像効果を付加しにくくなる。そこで、提案する分散型映像処理システムでは、1 フレーム毎に映像処理サーバにフレームデータを送信する。映像効果を付加する場合には、同時にライブラリ ID とパラメータも送信する。

フレームデータの送信後、映像処理サーバで映像処理されたフレームデータを受信すると、ライブ放送ソフトウェアに送信し、インターネットで映像配信する。続けて次のフレームデータを映像処理サーバに送信する。ライブ放送が終了すると、映像処理サーバに通知する。

### 3.5 映像処理サーバ

インターネットライブ放送サービスのほとんどでは、放送者を識別するために、ライブ放送開始前にサービスにログインすることが多い。映像処理サーバが直接インターネットで映像配信すると、これらのログイン情報を映像処理サーバが保持する必要がある、セキュリティ等の問題が発生する可能性がある。そこで、提案する分散型映像処理サ

ーバでは、映像処理サーバは映像処理を施したフレームデータを映像撮影端末に返信し、映像撮影端末がライブ放送ソフトウェアを用いて映像配信する。ログイン等の配信上の制約がなければ映像処理サーバから直接ライブ放送を行うことも考えられる。

映像処理サーバは、映像撮影端末からフレームデータを受信すると、関数 ID が指定されている場合には映像効果を施して映像撮影端末に返信する。ライブ放送が終了すると、読み込んでいた映像効果ライブラリを解放する。

## 4. 分散型映像処理システムの実装

3 章で設計した分散型映像処理システムを実現するために、映像撮影端末用の映像撮影端末ソフトウェアと、映像処理サーバ用の映像処理サーバソフトウェアの実装を行った。

### 4.1 映像撮影端末ソフトウェア

Windows OS で動作する映像撮影端末を対象として Visual Express 2013 C++を用いて実装を行った。カメラからの映像の取得には、画像処理でよく用いられている画像処理ライブラリ Open CV を用いた ([12])。通信には socket 通信を用い、パケット欠損を防ぐために TCP でデータの送受信を行った。ライブ放送ソフトウェアへのフレームデータの送信には、ソフトウェアで疑似的にカメラとして動作するソフトウェアカメラドライバを用いた。ライブ放送ソフトウェアのほとんどは、カメラから直接映像取得することを想定して設計されており、ソフトウェアカメラドライバを用いることで、ほとんどのライブ放送ソフトウェアで実装した映像撮影端末ソフトウェアを利用できる。映像のサイズは 640x480, 320x240, 160x120, 80x60 から選択できるようにした。

映像撮影端末ソフトウェアのスクリーンショットを図 2 に示す。映像処理サーバの IP アドレス、ポート番号、撮影するカメラの番号を指定して Start ボタンを押すと、映像処理サーバとの通信を開始する。Function 0~3 のボタンが用意されており、ボタンを押している間のみ、これらの映像効果を与えられる。右側には通信のログが表示される。他はテスト用のボタンや入力ボックスである。

### 4.2 映像処理サーバソフトウェア

Windows OS で動作する計算機を対象として Visual Express 2013 C++を用いて実装を行った。映像効果ライブラリは Windows のプログラムライブラリ (DLL) 形式で作成でき、アップロードについては、FTP や SCP といったファイル転送方式でアップロードすることとした。映像撮影端末ソフトウェアからライブ放送開始の通知を受け取ると、映像効果ライブラリを動的に読み込む。通信には socket 通信を用いた。映像処理には Open CV を利用できるように、映像処理サーバソフトウェアでも Open CV を利用できるようにした。

映像処理サーバソフトウェアのスクリーンショットを図3に示す。映像処理サーバのIPアドレス、利用するポート番号を指定して Start ボタンを押すと、映像撮影端末ソフトウェアからの通信待機を始める。他はテスト用のボタンや入力ボックスである。

#### 4.3 通信方法

映像撮影端末ソフトウェアからフレームデータを送信し続けると、映像処理サーバソフトウェア側でフレームデータの開始位置を発見できずフレームデータを受信できないことがある。そこで、実装した映像撮影端末ソフトウェアでは、フレームデータを送信する前に、フレームデータ内にほとんど現れない値として、0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00 を2回送信することとした。これは、フレームデータとしては白と黒が連続して現れることに相当し、カメラで撮影した映像には現れにくいと考えられる。このヘッダーに続いて、フレームの縦幅、横幅、チャンネル数、および評価用の送受信開始時刻を付加してフレームデータを送受信することとした。

### 5. 評価

本章では、実装した分散型映像処理システムの性能評価を行う。

#### 5.1 評価環境

評価には、映像撮影端末として、Windows 8.1, CPU が Intel Core-i7@2GHz (2 コア)、メモリ8G バイトのラップトップパソコンを用いた。映像撮影端末にはカメラが内蔵されており、無線 LAN (IEEE802.11g)を介して FTTH 接続サービスでインターネットに接続できる。ライブ放送に用いる計算機として一般的なパソコンと考える。映像処理サーバには、Windows 8.1, CPU が Xeon@2.6GHz (16 コア)、メモリ32G バイトのラックマウントコンピュータを用いた。映像撮影端末と比べるとコア数が8倍であり、計算能力の高い計算機といえる。著者らの大学内に設置されており、有線でインターネットに接続されている。本評価では、映像処理に関する評価を行うため、ライブ放送ソフトウェアは用いていない。映像効果ライブラリは、Intel の並列計算ライブラリ TBB ([13]) で並列化された Open CV を利用して開発した。

評価では、映像処理の一例として1章で述べた初めの例を想定し、映像中の人の顔を検出する映像処理を用いた。Open CV の顔検出機能では、入力画像のサイズを変更してテンプレートマッチングによる顔検出を行うことで、テンプレートと大きさの異なる顔も検出できる。サイズの変更には Scale Factor と呼ばれる値が用いられ、Scale Factor の割合ずつ縮小して検出を行う。例えば、Scale Factor が5の場合、5%ずつ縮小して入力画像のサイズが0になるまで検出を行う。サイズの異なる入力画像を並列に処理することで、計算能力の高い計算機を用いると早く検出を行える。

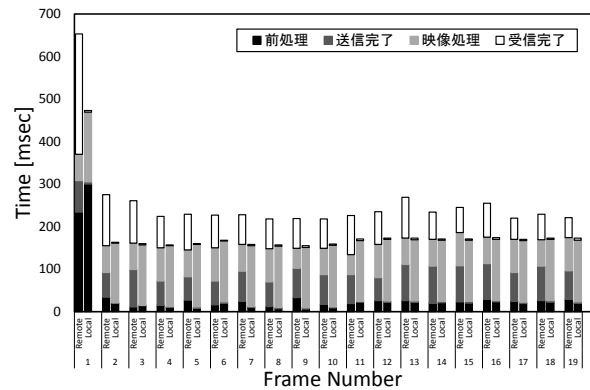


図4 フレームの送受信にかかる時間

Figure 4 Times for transferring frames

#### 5.2 フレームデータの送受信にかかる時間

実装した分散型映像処理システムの性能を評価するため、本システムを用いずに映像撮影端末で映像処理する場合とフレームデータの送受信にかかる時間を比較した。フレームデータの送受信には、前回の受信完了から次に送信を開始するまでの、カメラから映像を取得するなどの前処理、映像撮影端末から映像処理サーバにフレームデータ等を送信する送信完了までの時間、映像処理サーバで映像処理を行う時間、映像撮影端末が映像処理サーバからフレームデータ等を受信する受信完了するまでの時間が含まれる。

結果を図4に示す。Scale Factor は5とした。横軸がフレームの番号、縦軸がそれぞれの時間を示す。Remote は映像処理サーバを用いた場合、Local は用いなかった場合である。初めの前処理に時間がかかっているのは、ソケットの作成やカメラの初期化を行っているためである。Remote では、インターネットを介してフレームデータ等を映像処理サーバに送信しているため、Local と比べて送信、受信に時間がかかっている。映像処理サーバは映像撮影端末より計算能力が高いため、映像処理にかかる時間は Remote の方が短くなっている。映像処理時間とは、映像データを受信して映像処理を開始してから、終了するまでの時間である。利用するスレッド数を変化させて映像処理時間を計測した。Remote の方が Local より合計時間が長くなっているが、これは Scale Factor が5で映像処理の高速化が大きいためである。

この結果より、初めのフレームの送受信の後であれば、これらの時間に大きな違いがないことが分かる。以降、初めのフレームの送受信後の10フレーム分の平均値を評価値として用いる。

#### 5.3 映像処理時間

実装した分散型映像処理システムの性能を評価するため、本システムを用いずに映像撮影端末で映像処理する場合と映像処理時間を比較した。映像処理時間とは、映像データを受信して映像処理を開始してから、終了するまでの

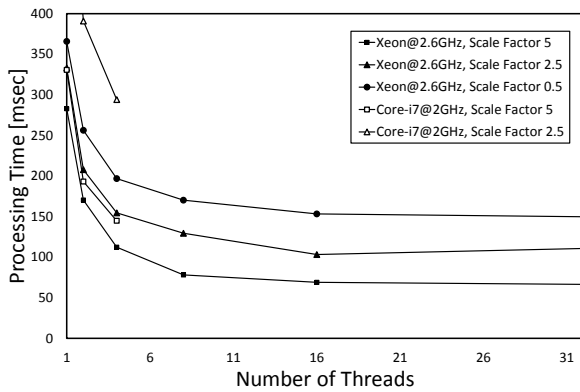


図5 映像処理時間

Figure 5 The elapsed time for video processing

時間である。利用するスレッド数を変化させて映像処理時間を計測した。

結果を図5に示す。横軸はスレッドの数、縦軸が映像処理にかかった時間である。用いた Core-i7 の最大スレッド数は4スレッドであるため、スレッド数が4までしか描いていない。Core-i7 で Scale Factor が 0.5 の場合は1秒ほどかかっており、グラフの視認性のため表示していない。同じ Scale Factor において、Xeon の処理時間が短くなっていることが分かる。スレッド数が同じでも Xeon の処理時間が短いのは動作周波数が主な原因と考えられる。Scale Factor が小さいほど処理する画像の数が増えるため、処理時間が長くなっている。

この結果より、評価環境において、映像撮影端末よりも映像処理サーバで顔検出を行う方が処理時間を短くできることが分かる。

#### 5.4 ターンアラウンド時間

Scale Factor を変化させてターンアラウンド時間を計測した。5.2節の評価結果の合計値に相当する。

結果を図6に示す。横軸が Scale Factor であり、小さいほど処理負荷が大きくなる。縦軸はターンアラウンド時間である。このグラフより、Scale Factor が小さく処理負荷が大きい場合には、映像処理サーバを用いることで映像処理時間を短縮できてターンアラウンド時間を短くできるが、大きい場合には通信時間が増加するため、映像撮影端末で映像処理した方がターンアラウンド時間が短いことが分かる。Scale Factor を大きくすると、顔検出の精度が悪くなるため、応用に合わせて適切な計算機で映像処理を行う必要がある。

#### 6. おわりに

本研究では、インターネットライブ放送のための分散型映像処理システムを提案した。提案システムでは、放送者はあらかじめ任意の映像処理を行う映像効果ライブラリを映像処理サーバに送信する。ライブ放送時には、放送者の要求に応じて、映像処理サーバが映像効果ライブラリに基づいて映像効果を付加し、処理した映像を配信する。評価

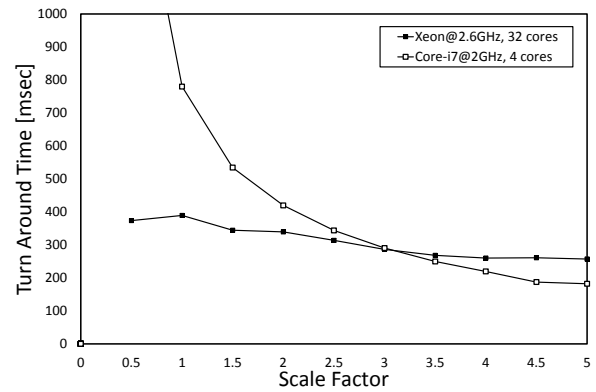


図6 ターンアラウンド時間

Figure 6 The turn around time

の結果、複雑な映像処理を計算能力の高い計算機で行うことで、映像処理にかかる時間を短縮しつつ、放送者は映像効果を付加できることを確認した。

今後、複数の映像処理サーバから利用する計算機を効果的に選択したり、映像撮影端末の計算能力で十分な映像処理は映像撮影端末で行ったりすることを考えている。

**謝辞** 本研究の一部は、NICT・大阪大学共同研究「大規模分散コンピューティングのための高機能ネットワークプラットフォーム技術の研究開発」および、文部科学省科学研究費補助金・基盤研究(B)(15H02702)、挑戦的萌芽研究(26540045)の研究助成によるものである。ここに記して謝意を表す。

#### 参考文献

- 1) Sodagar, I.: The MPEG-DASH Standard for Multimedia Streaming Over the Internet, IEEE Multimedia, Vol. 18, No. 4, pp. 62-67 (2011).
- 2) Cheng, B.: MediaPaaS: A Cloud-Based Media Processing Platform for Elastic Live Broadcasting, Proc. of 2014 IEEE International Conference on Cloud Computing (CLOUD), pp. 713-720 (2014).
- 3) Wu, J.-C., Huang, P. H., Yao, J.J., Chen, H.H.: A Collaborative Transcoding Strategy for Live Broadcasting Over Peer-to-Peer IPTV Networks, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 21, Issue 2, pp. 220-224 (2011).
- 4) スマートフォン向け AR (拡張現実) サービス <http://www.nttcom.co.jp/ar-saas/>
- 5) Gaddam, V.R., Langseth, R., Stensland, H.K., Gurdjos, P., Charvillat, V., Griwodz, C., Johansen, D., and Halvorsen, P.: Be Your Own Cameraman: Real-Time Support for Zooming and Panning into Stored and Live Panoramic Video, Proc. 2014 ACM Multimedia Systems Conference (MMSys), pp. 168-171 (2014).
- 6) Gotoh, Y., Yoshihisa, T., Taniguchi, H., and Kanazawa, M.: Brossom: A P2P Streaming System for Webcast, Journal of Networking Technology, Vol. 2, Issue. 4, pp. 169-181 (2011).
- 7) Roverso, R., Reale, R., El-Ansary, S., and Haridi, S.: SmoothCache 2.0: CDN-Quality Adaptive HTTP Live Streaming on Peer-to-Peer Overlays, Proc. 2015 ACM Multimedia Systems Conference, pp. 61-72 (2015).
- 8) Dai, J., Chang, Z., and Chan, S.-H.G.: Delay Optimization for Multi-Source Multi-Channel Overlay Live Streaming, Proc. 2015 IEEE International Conference on Communications (ICC), pp. 6959-6964

(2015).

- 9) Yoshihisa, T. and Nishio, S.: A Division-Based Broadcasting Method Considering Channel Bandwidths for NVoD Services, IEEE Transactions on Broadcasting, Vol. 59, Issue 1, pp. 62-71 (2013).
- 10) Gibbon, D. and Begaja, L.: Distributed Processing for Big Data Video Analytics, IEEE ComSoc MMTC E-Letter, Vol. 9, No. 3, pp. 29-31 (2014).
- 11) Ting, W.-C., Lu, K.-H., Lo, C.-W., Chang, S.-H., and Liu, P.-C.: Smart Video Hosting and Processing Platform for Internet-of-Things, Proc. of 2014 IEEE International Conference on Internet of Things (iThings), pp. 169-176 (2014).
- 12) OpenCV, <http://opencv.org/>
- 13) Thread Building Blocks, <https://www.threadingbuildingblocks.org/>