

# FPGA を用いた HDMI 向け低遅延映像同期システムの設計と実装

徳差 雄太<sup>1,a)</sup> 松谷 健史<sup>2</sup> 空閑 洋平<sup>2</sup> 中村 修<sup>3</sup>

受付日 2014年11月21日, 採録日 2015年5月9日

**概要:** 遠隔地間におけるより複雑なインタラクションのために、低遅延な映像通信システムがますます求められている。一般的な民生用カメラではクロックを同期するためのクロック入力がないため、送受信間で発生するクロックの誤差を吸収するバッファが必須となり、そのバッファリングによって通信遅延が増加していた。そこで、本論文では、民生用カメラおよびディスプレイを用いて、HD 画質の非圧縮伝送を行うための低遅延映像通信システムを市販の FPGA ボードを用いて設計実装した。送受信間の映像同期信号のジッタを最小限に抑えるための同期システムを FPGA 上に実現し、受信側のバッファを最小限に抑えている。本映像通信システムの通信遅延とそのジッタを高輝度 LED と照度センサを用いて計測したところ、カメラとディスプレイを除くシステム間における遅延を 1ms 以内に抑えることができた。これは同期機構を用いない市販の映像通信システムにおける遅延のたかだか 5%であり、この差は体感的にも有意である。

キーワード: FPGA, HDMI, 映像同期機構, 映像コミュニケーションシステム

## Design and Implementation of An FPGA-Based Low-Latency HDMI Video Synchronization System

YUTA TOKUSASHI<sup>1,a)</sup> TAKESHI MATSUYA<sup>2</sup> YOHEI KUGA<sup>2</sup> OSAMU NAKAMURA<sup>3</sup>

Received: November 21, 2014, Accepted: May 9, 2015

**Abstract:** A low-latency video communication system is a key enabler for sophisticated interactions between remote places. Because consumer video camera products typically do not have clock-input interfaces for the clock synchronization, a dedicated video frame buffer is required to absorb the clock skew between two systems, resulting in a longer communication latency. In this paper, we design and implement a low-latency uncompressed HD quality video communication system using an FPGA board for consumer video cameras and displays. The proposed video synchronization mechanism is implemented on the FPGA-based system in order to reduce the jitter between the sender and receiver even with a small frame buffer. The communication latency and jitter are evaluated using a super luminosity LED and an illuminance sensor. The results show that the communication latency (without those of camera and display themselves) is reduced down to 1ms, which is corresponding to only 5% of that in a commercially-available video communication system. This latency reduction is significant for interactions between users.

**Keywords:** FPGA, HDMI, video synchronization, video communication system

<sup>1</sup> 慶應義塾大学理工学研究科  
Keio University, Yokohama, Kanagawa 223-8522, Japan  
<sup>2</sup> 慶應義塾大学政策・メディア研究科  
Keio University, Fujisawa, Kanagawa 252-0882, Japan  
<sup>3</sup> 慶應義塾大学環境情報学部  
Keio University, Fujisawa, Kanagawa 252-0882, Japan  
a) tokusasi@arc.ics.keio.ac.jp

### 1. はじめに

IP ネットワーク上で使用される映像コミュニケーションシステムは、遠隔地とのビデオ通話、複数地点間を接続する遠隔会議、遠隔授業など様々な用途で利用されている。

さらに、人とマシン間を接続する遠隔ロボットの操作や遠隔医療のようなより複雑な遠隔地間の協調作業での利用が検討されている。このような遠隔地間のインタラクションの実現には、映像や音声の高品質化に加えて、インタラクションに直接影響を与えるカメラ、送信機、受信機、ディスプレイに関する総合的な遅延の削減が重要である。

インターネットを用いたこれまでの映像コミュニケーションシステムでは、帯域を節約するためにコーデックによる圧縮処理が行われることが多かった [9], [11]。一方、ネットワークの広帯域化にともない、ローカルエリアネットワーク内で 1 Gbps や 10 Gbps といった広帯域な通信の利用が可能となり、HD 品質の数 Gbps におよぶ非圧縮データの伝送が可能となった [14], [19]。さらに、民生用カメラやディスプレイの映像入出力として HDMI が採用され、HD 画質の非圧縮な映像や音声、副次情報を容易に扱えるようになった。

本論文では、インタラクションをとまなう映像アプリケーションへの応用を想定し、低遅延な映像伝送システムを広帯域な IP (Internet Protocol) ネットワーク上に民生用カメラとディスプレイを用いて実現する。従来の非圧縮な映像の IP 伝送システムにおいて、業務用カメラおよびディスプレイでは映像同期用マスタクロックと同期し、送受信間で映像同期信号の調整を図り、低遅延な映像通信を実現している。しかし、民生用カメラにはクロックを同期するクロック入力がないため、送受信間は異なるクロックソースで動作する映像同期信号で映像描画を行う必要がある。そのため、受信側ではクロックの誤差を吸収するための 1 フレーム分のバッファが必要であり、2.1 節で述べるように映像コミュニケーションシステムにおける遅延の要因となっていた。

本論文では、民生用カメラおよびディスプレイを想定し、HDMI 向けの低遅延な映像コミュニケーションを実現するために、非圧縮映像の IP 伝送用ハードウェアを設計、実装する。上述した遅延の要因となるバッファを削減するためには、送受信間の映像同期信号のジッタを最小限に抑え、より細かい時間粒度でクロック補正を繰り返す必要がある。そこで、受信側においてフレームごとに映像同期信号を補正し、同期を図る映像コミュニケーションシステムを実現した。実装には汎用の FPGA ボードを用い、本システムはこの FPGA ボード単体で動作する。本システムにおける送受信間の映像同期信号のジッタを評価した結果、遅延のジッタを 31.8 ms 程度にまで抑えることができた。また、映像コミュニケーション全体の遅延を 1 ms 以内に抑えることができ、これは LAN 環境において本同期機構を用いない Polycom HDX と比較して 5% の遅延である。この遅延差は有意であり、明らかに体感できるほどの低遅延化を実現できた。

本論文の構成は以下のとおりである。2 章では、映像同

期システムの関連研究を示す。3 章では、本論文で提案する映像同期システムの設計を論じる。4 章では、本同期システムを含む映像コミュニケーションシステムの実装を述べる。5 章では、本同期システムの性能評価を行う。6 章で本論文をまとめる。

## 2. 関連研究

本論文で想定する映像コミュニケーションシステムでは、遠隔地と 1 対 1 の映像コミュニケーションを行うものとする。図 1 に典型的な映像コミュニケーションシステムにおける遅延の内訳を示す [7]。映像コミュニケーションシステムにおける遅延は、入力デバイスであるカメラの Capture に相当する遅延、Encode 処理遅延、Internet におけるネットワーク伝搬遅延、Decode 処理遅延、表示デバイスである液晶ディスプレイでの Display (描画処理) 遅延に大別できる。ネットワーク処理に要する遅延を除いても 132 ms もの遅延が生じている。なお、映像コミュニケーションシステムにおける遅延は、実際にはコーデックや使用しているデバイス機器にも依存する。文献 [12] では、Google+ や iChat, Skype 映像システムにおける遅延は 180 ms から 270 ms 程度であると報告している。

### 2.1 映像コミュニケーションシステムにおける遅延の分類

上述のとおり、映像コミュニケーションに影響を与える遅延の要素は以下のように分類できる。

- (1) 入力デバイスによる Capture 遅延
- (2) Encode 処理におけるフレーム間圧縮でのバッファリング遅延
- (3) OS レベルでのネットワークプロトコルスタック処理の遅延
- (4) Decode 処理における遅延
- (5) ネットワーク (Internet) 伝搬遅延およびパケットロスによる遅延
- (6) Display 描画およびカメラ入力の映像同期信号のずれによる遅延

(1) は映像入力デバイスであるカメラに起因する遅延である。カメラの撮影素子であるイメージセンサは多数の受

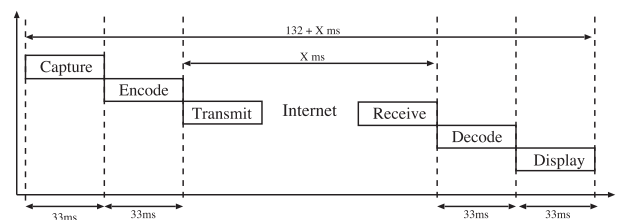


図 1 映像コミュニケーションシステムの遅延 (文献 [7] の図 1 より引用)

Fig. 1 Latency on video communication systems (Copied from Fig. 1 in Ref. [7]).

光素子によって構成されており、それぞれの受光素子は光エネルギーの明暗に対する電荷を発生する。RGB 変換、YC 変換や色差変換、スケーラ処理を経て出力インターフェイスに出力される。

(2) は送信側マシンにおけるコーデックに起因する。とくに遅延の要因となるのは H.264 などで行われるフレーム間圧縮である。数フレーム分の差分を予測するために数フレーム分のバッファを要し、そのバッファリングによって遅延が生じる。具体的には、1 フレーム分のバッファを満たすために 60 fps で約 16.6 ms, 30 fps で約 33.3 ms の遅延が生じる。

(3) はネットワークプロトコルスタック処理に起因する遅延である。文献 [1] は、Linux Kernel v3.5 系のネットワークプロトコルスタック処理の遅延について、UDP の場合に数 10 us から数 ms のジッタが発生すると報告している。このようなジッタは、ソフトウェアでプロトコルスタックを処理しているため、他のプロセスとのスケジューリングや割り込みに依存して生じていると考えられる。

(4) は受信機のコーデックのデコード処理に起因する遅延である。(2) と同様にコーデックを利用することで、デコードにおける処理の遅延が発生している。

(5) はネットワーク上のスイッチやルータ、物理ケーブル長に起因する遅延である。とくに輻輳が発生した場合、ネットワーク遅延やジッタの原因となる。

(6) はディスプレイの表示機構に起因する遅延である。ディスプレイ内において、映像入力端子である HDMI から入力された信号には、画像音声信号処理、スケーラ処理、TCON 基盤への映像出力といった処理が行われ、そのための遅延が生じる。

本論文では、上述した (2), (3), (4) の遅延を削減する手法を議論する。(1), (5), (6) はそれぞれカメラ、ネットワーク、ディスプレイに固有の遅延であり、これらの遅延を削減することは本論文の範囲を超える。図 2 に本論

文で目指す理想的な遅延のタイムチャートを示す。X は LAN を利用したときのネットワーク伝搬遅延であり、S は X を除くシステム遅延である。本論文では (2) および (4) で解説したコーデックを用いない非圧縮伝送し、(3) で解説したネットワークプロトコルスタックをハードウェアで処理し、パイプライン処理することで、遅延を削減する。さらに (1) と (6) の映像同期信号のタイミングを考慮に含めるフレームの同期を行う機構を実装することで、低遅延を実現する。したがって、固有遅延である (1), (5), (6) を考慮に含めつつ、S が最小となるように全体が低遅延となる設計を目指す。

## 2.2 映像コミュニケーションシステムの遅延評価手法

文献 [17] では、送受信側のフレーム数をカウントすることで、送受信間で発生した遅延を計測している。これによって映像コミュニケーションにおける終端デバイスであるカメラやディスプレイを含む総合的な遅延を計測できる。しかし、計測の精度がシステムの映像フレームレートに制限され、30 fps および 60 fps の場合の精度はそれぞれ 33.3 ms および 16.6 ms に制限される。

一方、本論文における評価手法として、LED と照度センサを用いた光計測器を使用する。送信機側のカメラレンズに LED を固定し、受信側のディスプレイの左上に照度センサを固定する。FPGA の General Purpose I/O ピンに LED と照度センサを接続しておき、LED が点灯してから照度センサが感知するまでのクロックサイクル数をカウントする。FPGA の動作周波数が 100 MHz のとき、1 クロックサイクルあたり 10 ns の十分な計測精度でカメラから送信機、受信機そしてディスプレイまでの伝送遅延を計測することが可能である。

## 2.3 映像クロック同期化技術

放送局内の映像機器は、マスタクロックを用いて同期される。しかし、放送局間で映像を共有して編集し、放送する場合は、放送局それぞれが独立したマスタクロックを用いるため、送受信間で映像クロックにおける位相のずれが生じる。このずれに起因して、受信側の映像を一時的に保存するフレームバッファの過不足が発生する可能性がある。そこで、文献 [15] と [16] で用いられている映像クロック同期化技術では、受信側で映像パッケージがフレームバッファに格納される度合いを観測し、クロック差を検出、さらに映像クロック制御パッケージを生成し、送信側にフィードバックする。送信側では、受信側からのフィードバックに基づいて補正された映像クロックをカメラに与えることで、理想的な映像クロック同期を実現している。

しかし、上記の文献のように受信側からのフィードバックに基づいて送信元カメラの映像出力のクロック周波数を補正可能な環境は、業務用カメラに限られており、用途も

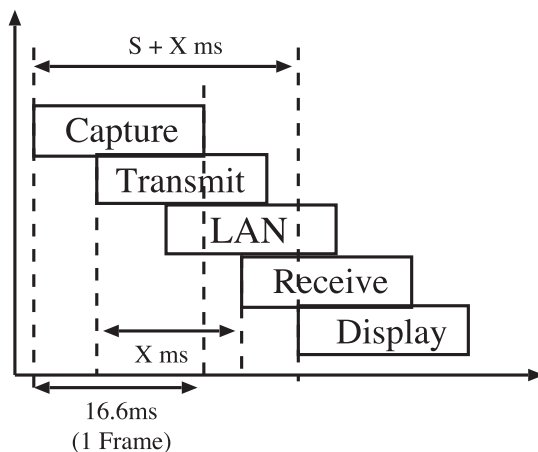


図 2 本研究で目指すシステム遅延のタイムチャート  
Fig. 2 Time chart of ideal latency in this paper.



放送業務用に限られる。本論文で想定している民生用カメラは映像クロックの入力端子を備えておらず、送信側カメラの映像出力用のクロック周波数を補正する手法は民生用カメラには適用できない。したがって、本論文で提案する手法では、送信側の映像出力のタイミングに合わせて IP パケットを生成し、受信側での IP パケット到着タイミングに基づいて受信側のピクセルクロックを修正する。これによって受信側に必要なバッファを 1 フレーム未満に抑えることもできる。

### 3. 映像同期信号の同期機構：RV-SYNC

本章では、クロック入力を備えていない民生用 HD カメラ向けに低遅延な映像通信を実現する映像同期システムである Remote Video Synchronization (以下, RV-SYNC) を提案する。

#### 3.1 ライン単位処理

映像データを IP を用いて伝送するため、Ethernet や IP ヘッダなどのオーバーヘッドを含めて映像パケットのサイズを決定する必要がある。以下では、映像データを伝送するにあたり適切なパケットサイズを試算する。ただし、後述する副次情報は、パケットサイズに依存するため試算には含まない。UDP ヘッダを含むネットワーク制御情報は 64 バイトである。ここで  $x$  を UDP データグラムサイズとするとき、1 パケット送信すると  $(64 + x)$  バイト分の帯域を消費する。1GbE (Gigabit Ethernet) を想定しているため、 $p$  パケット送信したときのビットレートが 1 Gbit 以下になる必要がある。また、1 秒間に伝送する映像データの総量 ( $xp$ ) は、1 秒あたりのフレーム数 (60 fps)、1 フレームあたりのピクセル数 ( $1,280 \times 720$ )、1 ピクセルあたりのデータ量 (16 bit カラー深度) によって決まる。

$$\begin{cases} 8(64 + x)p \leq 10^9 \\ xp = 1280 \times 720 \times 2 \times 60 \end{cases}$$

上記の連立方程式を計算すると、 $x \geq 491.247$  となり、UDP データグラムサイズを 492 バイト以上にする必要がある。

本システムでは、映像フレームにおけるライン単位の処理 [18] を目指す。本論文におけるラインとは、映像フレームにおける水平ラインを示す。つまり、1 ラインの処理時間内にパケット送信を完了する必要がある。映像のパケット化では、1 ライン、もしくは、その整数分の 1 など区切りの良いサイズでパケット生成すると効率が良い。また、パケットごとに副次情報として、フレーム内におけるパケット映像データの位置情報を表す Index、データフォーマット、解像度情報が必要となる。しかし、パケットサイズが小さいほど、映像パケットの位置情報を示す Index の数が抑えられて副次情報サイズを小さくできる。以上から、1

パケットに 1 ラインの半分のピクセルに相当する 640 ピクセル (1,280 バイト) を格納することにした。

#### 3.2 送受信間のピクセルクロックの制御問題

送信機の映像処理に使用するクロックソースはカメラ内蔵のクロックを用いる。受信機では、ディスプレイと同期するピクセルクロックを生成する。このクロック周波数として CEA-861 で定義されている値が一般的に使用される。本論文では、 $1,280 \times 720$  60 Hz Progressive の映像モードを想定し、74.25 MHz のピクセルクロックを使用する。しかし、送受信機間で異なるクロックソースを使用しているため、以下にあげる 2 つの問題が懸念される。

- (1) カメラによるピクセルクロックが仕様と異なる可能性がある。
- (2) 送受信間で動作する映像同期信号が異なるため、受信側で FIFO バッファの過不足が生じる。

そこで、本論文で対象とする解像度モードのピクセルクロック周波数が CEA-861 のパラメータに適合しているかを検証した。カメラの映像出力インターフェイスである HDMI では、4 レーンのうち 1 レーンをクロック伝送用ラインとして使用している。このクロック伝送用ラインは、映像の解像度に適するピクセルクロックを送信機側へ伝送している。ここで、カメラの内蔵クロックから生成されたクロックを FPGA ボードに与え、カメラからのクロックを計測する。この計測で使用するカメラは Sony 社 HXR-NX70J [2] である。また、この計測で確認する周波数値は、CEA-861 で標準化されている  $1,280 \times 720$  60 Hz Progressive での動作周波数である 74.25 MHz とする。

HDMI で伝送される TMDS (Transition Minimized Differential Signaling) 信号は、デコーダにてピクセルクロックや映像同期信号、映像データに変換される。FPGA ボードが備えている 100 MHz の基準クロックを用いて 1 秒をカウントし、ピクセルクロックのクロックサイクル数を求める。複数回の試行の結果、Sony HXR-NX70J の HD カメラが  $1,280 \times 720$  Progressive 60 fps で使用するピクセルクロックは約 69.68 MHz であった。この値は 74.25 MHz から約 5 MHz もずれている。送信側でこのピクセルクロックを使用し、受信側で 74.25 MHz を使用した場合、受信 FIFO バッファは頻繁に不足の状態が発生し、安定した描画ができない。

さらに、送信機から伝送する映像データの走査位置と、受信機で描画しようとしている走査位置のタイミングは同期されていない。これは送受信間で使用している映像同期信号が異なり、映像を描画するタイミングも独立しているからである。このため、つねに受信側で FIFO バッファの過不足が発生する。したがって、(1) および (2) の問題を解決するために、送受信間で映像フレームの描画を開始し始めるタイミングを同期する手法を検討する。これを実現す

表 1 CEA-861 映像タイミングパラメータ (1,280×720 Progressive 60 Hz)

Table 1 CEA-861 video timing parameters (1,280 × 720 Progressive 60 Hz).

Parameter	Value
Vsync Pulse Width	5
Vsync Front Porch	20
Vsync Back Porch	5
Hsync Pulse Width	40
Hsync Front Porch	220
Hsync Back Porch	110

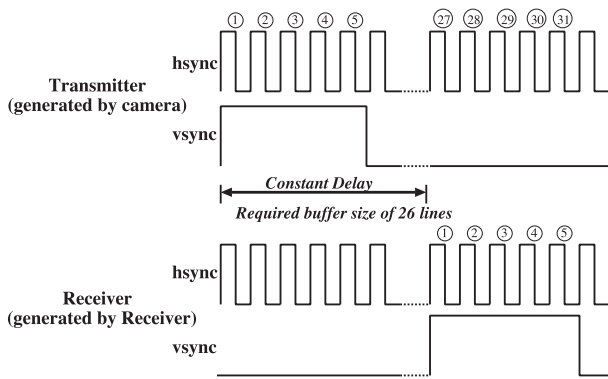


図 3 送信機と受信機の映像同期信号のずれ

Fig. 3 Delay of video synchronization signals between transmitter and receiver.

る機構を次節で説明する。

### 3.3 映像同期機構

本節では、送信側の映像同期信号を IP パケットの到着時間に同期させる機構を設計する。CEA-861 における映像同期信号のパラメータでは、各信号の仕様が決められている。表 1 に 1,280 × 720 60 Hz Progressive における映像同期信号のパラメータを示す。

それぞれのパケットに、映像フレームにおける位置を識別する 12 bit 長の Index 番号を付加する。また、映像ピクセルの深度である 16 bit 長ごとに、パケットに付加された Index 番号 (12 bit) およびピクセルデータ (16 bit) を格納するために 28 bit 幅の FIFO バッファを設ける。FIFO の Depth はピクセルデータが 25 ラインを格納できる容量 (Depth : 32,768) とする。受信側の MAC (Media Access Control) の処理を終えたデータを受信 FIFO バッファに格納する。

図 3 は、カメラとディスプレイ間を流れる理想的な映像同期信号のタイミングを表している。送信側カメラは HDMI を介して FPGA 上の送信機に映像を送信している。また、送信機はカメラから入力された映像を送信 FIFO バッファ (Depth : 2,046) に 1 ラインごとにバッファリングし、1/2 ラインごとにパケット化して送信している。一方、受信機では、FPGA 上の受信機でピクセルクロック

を生成し、独自のタイミングで映像同期信号が動作する。RV-SYNC では、パケット到着のタイミングでパケット内に含まれる index (12 bit) を識別し、先頭パケットであれば映像同期信号を再生成する。具体的には、先頭パケットの到着タイミングで映像同期信号をリセットし、垂直同期信号 (vsync) を立ち上げて補正する。したがって、映像フレームの先頭パケットを受信してから、描画開始まで 25 ライン分のブランキング期間 (Vsync Pulse Width 5 ライン, Vsync Front Porch 20 ライン) があり、この期間に受信した映像データは受信 FIFO バッファに格納される。その結果、図 3 に示すように送受信機間で生じる映像同期信号の遅延を 26 ライン程度 (送信側 1 ライン, 受信側 25 ライン) に抑えることができた。

### 3.4 パケットロス発生時の映像同期信号の復旧

RV-SYNC では、UDP パケットの到達タイミングによって受信側で映像同期信号を再生成している。パケットロスなどにより映像フレームの開始パケットを受信できなかった場合、受信側で映像同期信号が生成されない。また、映像同期信号が途中で途絶えると、受信機とディスプレイの同期が切れて、ディスプレイへの安定した映像出力が維持できない。したがって、パケットロス発生時に映像同期信号を復旧する仕組みが必要である。

そこで RV-SYNC では映像フレームの開始ピクセルを含むパケットが遅延して到着した場合、もしくはパケットロスが発生した場合、表 1 に示すように垂直同期信号を一定に保つために Back Porch 期間 (水平ライン 5 本分) が終わり次第、垂直同期信号をアクティブにすることにした。これにより、パケットロスが発生したときでも、映像同期信号は一定した動作を維持することができる。

### 3.5 理想的な映像描画のタイミング

RV-SYNC では受信バッファ量を最小限に抑えるために、送受信間で映像描画のタイミングを合わせている。理想的には、1/2 ラインを含む映像パケットが受信機に到着した直後にその 1/2 ラインを描画できれば合理的であるが、ここでは 1 フレームを描画する前にフレームごとに映像同期信号の修正を行うため、映像同期信号を安定させるために受信側に 25 ライン分の遅延が生じる。

これを確認するために、RV-SYNC による HDMI 映像入力と送信のタイミング、受信と HDMI 映像出力のタイミングのシミュレーションを行った。論理回路シミュレータ Icarus Verilog によるシミュレーション結果を波形表示ツール gtkwave を使用して表示した。図 4 に送信機のパケット送出タイミングを示す。図中の vde (Video Data Enable) 信号は、映像データのアクティブ期間を示している。TXEN は Ethernet の Data Enable 信号を示している。Line1 がアクティブのとき、ピクセルデータは送信用

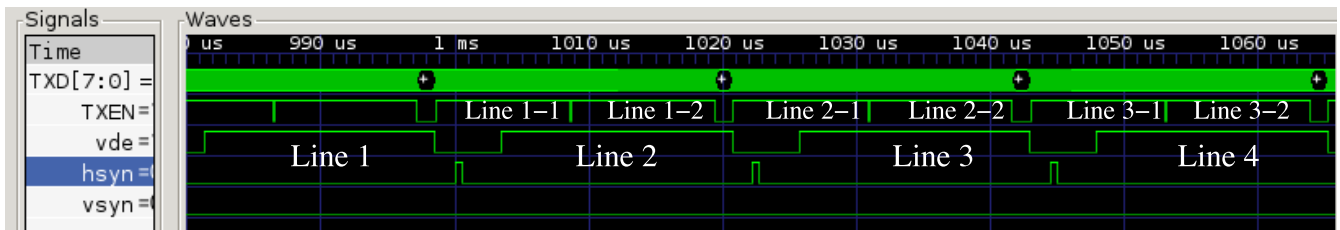


図 4 送信機の packets 送信タイミング  
Fig. 4 Timing of packet transmission on transmitter.

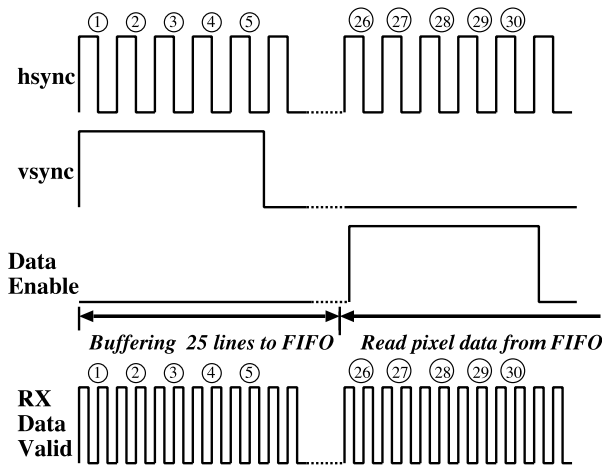


図 5 受信側で発生する RV-SYNC による映像同期信号の遅延  
Fig. 5 Delay of video synchronization signals at receiver side with RV-SYNC.

FIFO に蓄積される。1 ライン分のピクセルデータが FIFO に格納されたタイミングで、Ethernet 送信モジュールは映像データの packets 化を開始する。1 ラインを 2 packets に分割するため、Line1-1 packet と Line1-2 packet の間の IFG (Inter Frame Gap) はその最小値である 12 クロックとして設計している。図 4 のタイミングチャートより、1 ラインのビデオアクティブ期間に 2 packets の送信が確認できる。

図 5 に受信側で発生する RV-SYNC による映像同期信号の遅延を示す。受信側では、1 ライン目の packets 到着のタイミングで映像同期信号を修正できており、この修正したタイミングで垂直同期信号がアクティブになる。この結果、Vsync Pulse 幅 5 ライン分、Vsync Front Porch 20 ライン分の計 25 ライン分遅延してから 1 ライン目の描画を開始する。

#### 4. FPGA ボードへの実装

本論文では、RV-SYNC を用いた低遅延映像コミュニケーションシステムのプロトタイプとして、Xilinx 社の FPGA である Spartan-6 上で動作する HDMI-TS (HDMI Transport System) を開発した。使用した FPGA ボードは Digilent 社 Atlys ボードであり、Gigabit Ethernet のポートを 1 つ、HDMI Input 端子を 1 つ、HDMI Output 端子

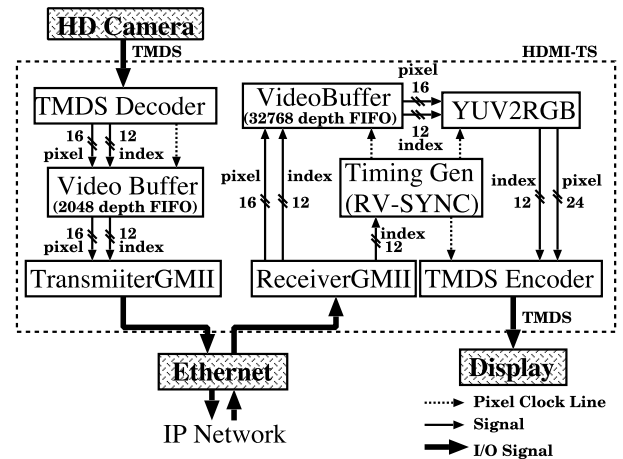


図 6 HDMI-TS のブロックダイアグラム  
Fig. 6 Block diagram of HDMI-TS.

を 2 つ備えている。

HDMI-TS は、送信機と受信機の両方の機能を搭載している。ボード上の入力用および出力用 HDMI コネクタには、HDMI ケーブルを用いてカメラおよびディスプレイをそれぞれ接続し、RJ-45 コネクタには UTP ケーブルで IP ネットワークに接続している。図 7 に実装したシステムの動作風景を示しており、スイッチングハブを経由した IP ネットワーク上で HD 画質 (1,280 × 720 60 fps) の非圧縮映像伝送を行っている。なお、IP ネットワークを利用したシステムであるため、ネットワークに接続された汎用 PC 上のソフトウェアでも映像データの送受信自体は可能である。ただし、通信相手が汎用 PC 上のソフトウェアの場合、本論文で提案するハードウェア上の機構である映像同期機構は利用できない。

HDMI-TS のブロックダイアグラムを図 6 に示す。具体的には以下の項目を実装する。

- (1) ネットワークプロトコルスタックのハードウェア実装
- (2) 映像フレームのラインごとのパイプライン処理
- (3) RV-SYNC による映像同期信号の同期機構

#### 4.1 HDMI-TS の仕様

HDMI-TS の仕様を表 2 に示す。民生用 HD カメラでは、HDMI 出力のカラーフォーマットとして YUV422 が一般的に使用されている。映像データは、このフォーマット



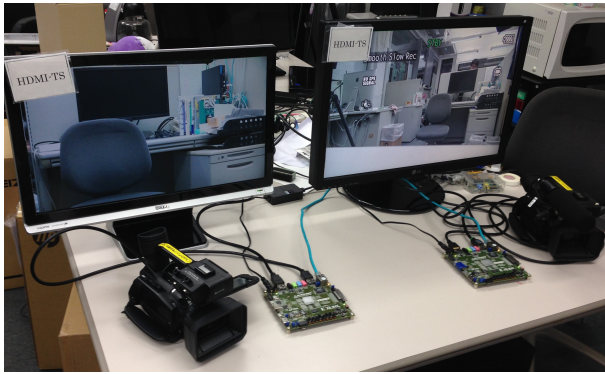


図 7 HDMI-TS 動作風景

Fig. 7 Appearance when HDMI-TS is running.

表 2 HDMI-TS の仕様

Table 2 Specification of HDMI-TS.

映像入力	HDMI
映像出力	HDMI
対応解像度	1,280 × 720 Progressive
カラーフォーマット	YUV422
フレームレート	60 fps
ビットレート	約 885 Mbps
ネットワーク	1000BASE-T
伝送方式	UDP/IP

そのまま非圧縮で IP 伝送され、受信側で HDMI を介してディスプレイに表示される。この際、HDMI の物理層である TMDS において、AUX（音声や副次情報を扱う）を有効にしていないため、カラーフォーマットは RGB 形式で表示される。したがって、受信側では YUV 形式から RGB 形式へのカラー変換を行っている。

#### 4.2 TMDS デコーダおよびエンコーダ

本 HDMI-TS では、映像入出力に HDMI を採用する。HDMI は TMDS の信号規格によりエンコードおよびデコード処理を行う。Xilinx 社が提供している TMDS のエンコーダ・デコーダのリファレンスアプリケーション [3] をもとに TMDS デコーダとエンコーダの設計、実装を行った。

#### 4.3 Ethernet モジュール

カメラから入力された映像は TMDS でデコードされた後、映像データがビデオバッファに入力される。このバッファはピクセルクロックと Gigabit Ethernet で使用されるクロックの緩衝として使用される。バッファリングされた映像データは GMII (Gigabit Media Independent Interface) モジュール内に実装された UDP プロトコルスタックに基づいてパケット処理が行われる。

#### 4.4 論理合成結果

本システムでは、Xilinx 社の FPGA である Spartan-6

表 3 論理合成結果

Table 3 Result of synthesis.

項目	使用数	全体数	占有率
Slice	759	6,822	11%
Slice Registers	1,604	54,576	2%
Slice LUTs	1,895	27,288	6%
BlockRAM	57	116	49%

XC6SLX45 上に実装した。論理合成に使用したツールは Xilinx ISE 14.4 である。本システムは、GMII モジュールでは 125 MHz で動作し、映像処理ではピクセルクロックである 74.25 MHz で動作する。本実装の論理合成結果を表 3 に示す。既存の MAC の IP core を使用せず、映像通信専用の Ethernet MAC を実装したため、全体的なスライスの使用割合が少なく、ロジックの使用割合が 11%となっている。また、送受信両方にバッファとして合計 26 ラインを格納できる FIFO を利用しているため、使用した BlockRAM の使用割合が高く 49%となっている。

使用した BlockRAM の割合は、送信機側で 0.9 割、受信機側で 9.1 割を占めている。送信機側では、カメラから入力されるピクセルを 1 ラインごとにパケット化するために 1 ライン分のピクセルおよびピクセルの位置を示す Index をバッファしている。送信側ではバッファサイズは 1 ライン分で十分である。受信機側の FIFO バッファは、ネットワーク処理と映像処理の緩衝と同期の役割を担っており、25 ライン分のピクセルとピクセルの位置を示す Index を格納している。さらにバッファサイズを増やすことで、パケットが到達するゆらぎを吸収できると考えられる。本システムで使用した FPGA では、最大で現在の 2 倍程度まで FIFO バッファのサイズを拡大することが可能で、パケットの到達のタイミングを吸収できると考えられる。

### 5. 評価

本論文で提案した映像同期機構である RV-SYNC を実装した HDMI-TS を用いて LAN 環境において映像通信を行い、総合的な遅延を計測する。受信側は異なるクロックソースで動作する映像信号を使用しているため、映像フレーム最大 1 フレーム分の遅延が発生する可能性があった。

評価項目は以下の 2 点である。

- (1) 提案手法 RV-SYNC により映像同期信号のジッタを改善できているかどうか確認する。
- (2) 本システム HDMI-TS により、カメラからディスプレイを含む総合的な遅延を見積もり、遅延が改善されているかどうか確認する。

#### 5.1 遅延評価環境

2.2 節で述べたとおり、送受信間の遅延を示すフレーム差をカウントする手法では、フレームレートが 60 fps および

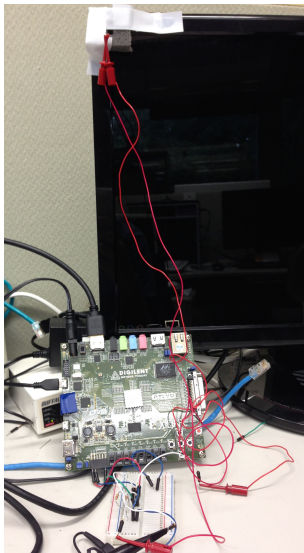


図 8 ディスプレイ遅延計測の外観

Fig. 8 Appearance when display processing delay is measured.

30 fps のとき、計測精度はそれぞれ 16.6 ms および 33.3 ms であった。遅延評価手法として以下の 2 つの候補が考えられる。

- (1) 高速度撮影によるフレーム差の計数
- (2) 受光素子による遅延計測回路

(1) では、送信側のディスプレイと受信側のディスプレイの両画面を計測用カメラで撮影し、送受信間のフレーム数の差を検出する。映像伝送実験で遅延を計測する際に、この手法が用いられてきた [17]。しかし、この手法は視覚的にフレームごとの差異を判断するため、定量的な計測が難しい。

(2) では、受光素子の 1 つである照度センサを用い、照明の光を発してから照度センサが認識するまでの遅延を FPGA による計測回路で測定する。送信側のカメラに白色高輝度 LED を設置し、受信側のディスプレイの該当箇所にも照度センサを設置する。この LED と照度センサは FPGA ボードの GPIO に接続しておき、LED が点灯してから照度センサが認識するまでのクロックサイクル数を求める。また、計測回路内に状態を持たせることで、複数回におよぶ計測が可能となり、(1) と比べて、定量的な映像遅延の計測がしやすい。

本論文では、総合的な遅延を計測に加えて、遅延のジッタも定量的に計測する。したがって、上述の (2) で示す計測回路を実装し、総合的な遅延を計測することにした。

図 8 に示すように、カメラのレンズにおいて映像走査の始点となるポイントに LED を設置する。同様に、ディスプレイの走査開始位置である左上に照度センサ [6] を設置する。遅延計測のブロックダイアグラムを図 9 に示す。図の破線部は LED の光の進行および遅延を表しており、実線部は伝送ケーブル内の遅延を表している。

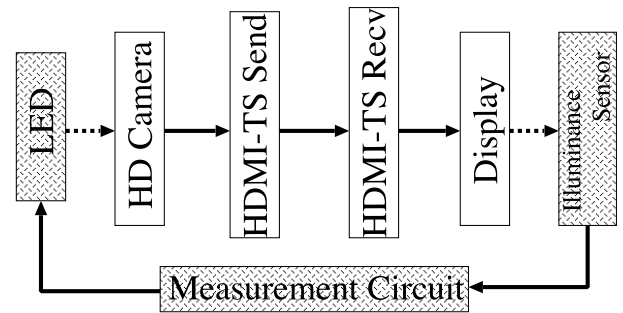


図 9 遅延評価環境

Fig. 9 Delay measurement environment.

### 5.2 計測器補正

本計測器では、FPGA 上に実装されたロジックにより高輝度白色 LED が点灯してから、照度センサが反応するまでの時間を計測する。この時間には、ロジックが点灯開始の信号を送ってから、LED が点灯するまでの LED 反応時間と照度センサが反応して電気信号に変換するまでの照度センサ反応時間が含まれている。このことから、計測結果に誤差が生じる可能性があり、計測器の補正が必要である。

そこで照度センサと LED を直接つなぎ、計測器の誤差を計測する。480 回の試行により、平均  $9.6 \mu\text{s}$ 、標準偏差  $0.5 \mu\text{s}$  となった。終端デバイスを含む映像コミュニケーションシステムの遅延を計測する際、関連研究の結果を考慮すると、ms オーダでも評価可能であると考えられる。しかし、内部の遅延を検討する際に、 $\mu\text{s}$  オーダの計測精度が求められる可能性があるため、内部の遅延を検討する際にこの補正値を適用する。

### 5.3 評価対象

HDMI-TS および既存の映像コミュニケーションシステムにおける総合的な遅延について評価、比較する。代表的な映像コミュニケーションシステムとの比較として、ここでは Skype, Polycom 社 HDX8000, webRTC フレームワークを利用したアプリケーション, DVTS [8], [10] を比較対象とした。これらは LAN 環境で利用できるアプリケーションである。なお、Skype は通話先を検索するためにインターネット上のスーパーノードに接続する必要があるが、通話においては同じ LAN 内の通信と考えられるため LAN 環境での映像コミュニケーションと見なす [13]。

表 4 に比較に使用するシステムを示す。システムによって、対応している解像度や映像フレームレート、コーデックが異なる。さらに専用のカメラがあるなど、すべての条件をそろえて評価することは難しい。また、商用システムの内部遅延を外部から計測すること自体困難である。

そこで、図 10 に本評価で使用される遅延の定義を示す。まず最初に、映像コミュニケーションシステムを使用せずにカメラとディスプレイをビデオインターフェイスで接続する（もしくは、直接カメラとディスプレイを直結できな



表 4 評価対象

Table 4 Evaluation targets.

	HDMI-TS	Skype	webRTC	DVTS	Polycom
解像度	1,280 × 720	1,280 × 720	1,280 × 720	720 × 480	1,920 × 1,080
fps	60	30	30	30	60
コーデック	非圧縮	VP8	VP8	非圧縮	H.264
カメラ	HXR NX70J	iSight	iSight	DCR-PC350	EagleEye III
ディスプレイ	Dell ST2210	MacbookPro	MacbookPro	トリニトロン管	KDL-40EX500

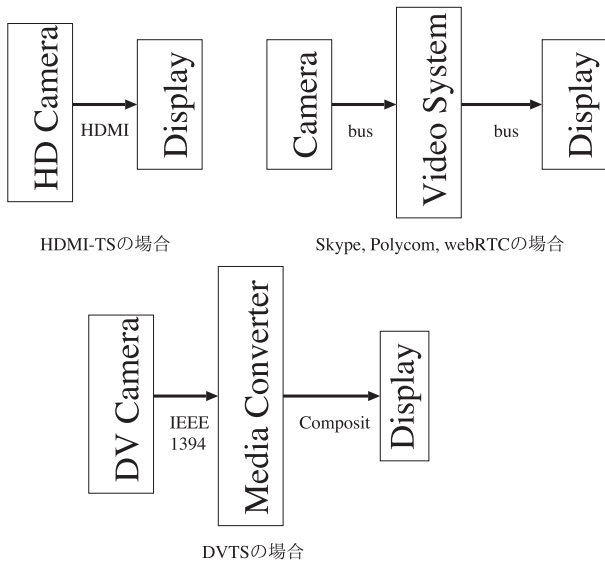


図 10 各システムにおける直結遅延の測定環境

Fig. 10 Direct delay measurement environment for each system.

表 5 映像コミュニケーションシステムの遅延計測結果 (us)

Table 5 Measurement result of latency on video communication systems.

	HDMI-TS	DVTS	Polycom
合計遅延 (A)	69,879	342,665	353,654
直結遅延 (B)	69,013	128,458	333,880
通信遅延 (A-B)	866	214,207	19,774

い場合にシステムを仲介し、プレビュー表示を行う) 場合を考える。このとき発生する映像の遅延を直結遅延と定義する。一方、映像システムを利用して伝送を行った場合の遅延を合計遅延と定義する。

#### 5.4 遅延評価

カメラやディスプレイ固有の遅延に依存しない、各映像コミュニケーションシステム間で発生する映像通信の遅延の計測結果を表 5 に示す。合計遅延は、カメラからディスプレイを含んだ映像の IP 伝送時間を示している。直結遅延は、カメラとディスプレイを直結したときの伝送時間を示している。計測では、合計遅延および直結遅延をそれぞれ 400 回計測し、その最大値を使用している。通信遅延は、カメラとディスプレイを含まない送受信機の端末間の

表 6 映像コミュニケーションシステムの合計遅延のジッタ (ms)

Table 6 Jitter of total latency on video communication systems.

	HDMI-TS	Skype	webRTC	DVTS	Polycom
最大値	69.9	252.4	843.1	342.7	333.9
最小値	38.1	143.0	394.3	274.3	248.4
差分	31.8	109.4	448.8	68.4	105.2

遅延を示しており、合計遅延 (A) と直結遅延 (B) の差で表される。

通信遅延は、コーデック、ネットワークプロトコルスタック、ネットワーク伝搬遅延を合計した遅延に相当する。本実装では、通信遅延が理想的には 26 ライン分の遅延 (578 us に相当) になるように設計した。一方、実際に測定した結果、本実装である HDMI-TS は 866 us であった。これは直結遅延 (カメラおよびディスプレイの内部遅延) の測定誤差によるものと考えられる。表 5 に示すとおり、ディスプレイおよびカメラを含まない HDMI-TS の遅延 (A-B) は合計遅延に比べて圧倒的に小さく、1 ms 以内に抑えることができた。

Polycom はコーデックとして H.264 を使用しており、フレーム間圧縮のフレーム予測により数フレームをバッファしている。また、ネットワークプロトコルスタックの遅延の影響により遅延は約 20 ms であった。DVTS は約 214 ms の遅延であった。この結果より、本実装である HDMI-TS は Polycom の 5% の遅延にまで削減できたといえる。この差は体感的にも有意である。

#### 5.5 ジッタ評価

表 6 は、映像コミュニケーションシステムのカメラからディスプレイまでの合計遅延のジッタを示している。それぞれの項目について 400 回の遅延計測を行い、最大値と最小値を示している。差分は、最大値と最小値の差を示しており、合計遅延のジッタに相当する。また分散を計算したところ、HDMI-TS では 45.7、DVTS では 240.3、Polycom では 228.0 となり、HDMI-TS では分散が小さいことが示された。

本実装である HDMI-TS は、遅延のジッタは 31.8 ms であった。これはカメラの 60 Hz の周期で行われるフレームスキャン (16.6 ms に相当) および同周期で行われるディス

プレイ内のフレームリフレッシュ (16.6 ms に相当) が原因して発生するジッタであると考えられる。DVTS は送受信機でコーデックによる処理を行っていない。DV フォーマットは DV 機器内部にてフレーム内圧縮に基づいてエンコードおよびデコードされるため遅延のジッタは 68.4 ms であり、Polycom よりも小さいと考察できる。

Skype, Polycom では、フレーム間圧縮に基づいてエンコードおよびデコードされるため、数フレームのジッタがあると考えられる。それぞれの遅延のジッタは、Skype が 109.4 ms, Polycom が 105.2 ms であった。この結果より、HDMI-TS は遅延のジッタを 31.8 ms にまでに抑えることができた。これは上述したカメラのフレームスキャンおよびディスプレイのフレームリフレッシュのジッタである 33.3 ms に近似している。これとは別に最大 1 フレーム分のジッタが発生する可能性があったが、ジッタなく伝送できていると考察できる。

## 5.6 議論

本論文では、1GbE での LAN 環境を想定して、提案同期機構 RV-SYNC のプロトタイプを設計実装した。対象ネットワークを 1 Gbps から 10 Gbps, 40 Gbps に広帯域化した場合でも、使用した同期機構をそのまま移植することで利用可能である。

映像コミュニケーションとして、1 Gbps から 10 Gbps へ広帯域化することで 4K (約 6.9 Gbps), 40 Gbps では 8K (約 31.8 Gbps) に匹敵する解像度の非圧縮伝送が可能となる。解像度が大きくなるにつれて、1 フレームあたりの転送量が増え、1 フレーム分のバッファリングの量が増える。本システムは、受信側に 25 ライン程度 (4K 解像度に対応する場合、82 ライン程度 [4]) を格納できるバッファサイズが必要である。そのため、4K や 8K に解像度を拡張させる場合、安価な FPGA 内のリソースではバッファサイズが足りないため、DRAM 上にバッファを設ける必要がある。

また、1 Gbps の帯域幅にさらに音声パケットを追加することは、映像パケットのタイミングが固定であり、映像データが帯域の 9 割を占めていることから困難である。しかし、カラーフォーマットを YUV422 から YUV411 に変換するなど空間的圧縮を適用することで 1 Gbps の帯域を活用できる可能性もある。一方で、10 Gbps に広帯域化することで非圧縮映像に加えて音声パケットも伝送することが可能となる。

本システムで使用した FPGA ボードでは 1GbE のネットワークインターフェイスを搭載しており、これに律速されて非圧縮による伝送が可能な解像度は  $1,280 \times 720$  60 fps が限界である。専用機器への実装を検討する場合、上述したように 10GbE のインターフェイスや 40GbE のインターフェイスを設計し、BlockRAM をより多く内蔵するハイエンドな FPGA を採用することで、今後普及してくる超高

解像度の映像の非圧縮伝送に対応できるシステムを実現できる。またハイエンドな FPGA における BlockRAM を最大限活用することで、パケットの到達タイミングのジッタを回避することも可能である。

## 6. まとめ

本論文では、インタラクションをとまなう映像アプリケーションに応用できる低遅延な映像伝送システムを FPGA 上に設計実装した。民生用カメラやディスプレイでは、送受信間の映像同期信号の同期が取れないため、送受信機に 1 フレーム分のバッファを設ける必要があり、遅延の原因となっていた。本論文では、IP ネットワーク上で送受信間の映像同期を実現するために、映像パケットの到着タイミングで受信側の映像同期信号を再生成する同期モジュールを設計した。送受信間の映像同期信号のジッタを評価した結果、31.8 ms にまでに抑えることができた。また、システムにおける遅延は 1 ms 以下に抑えることができた。

なお、RV-SYNC を備えた HDMI-TS である本設計は Digilent 社が主催する FPGA 設計コンテストにおいて、国内大会で優勝し、国際大会へ出場するなど実装の完成度においても高く評価された [5]。

## 参考文献

- [1] Brandeburg, J.: A way towards lower latency and jitter, *Proc. Linux Plumbers Conference* (2012), available from <http://www.linuxplumbersconf.org/2012/wp-content/uploads/2012/09/2012-lpc-Low-Latency-Sockets-slides-brandeburg.pdf> (accessed 2015-01-25).
- [2] Sony Corporation: HXR-NX70J, available from <http://www.sony.jp/nxcam/products/HXR-NX70J/> (accessed 2015-01-25).
- [3] Bob Feng: Xilinx Application Note XAPP495: Implementing a TMDS Video Interface in the Spartan-6 FPGA, available from <http://www.xilinx.com> (accessed 2015-01-25).
- [4] HDMI: HDMI specification version 1.4.
- [5] Digilent Inc.: Digilent Design Contest 2014.
- [6] 新日本無線株式会社 (NewJRC): NJL7502L, 入手先 <http://semicon.njr.co.jp/jpn/PDF/NJL7502L-J.pdf> (参照 2015-01-25).
- [7] Mody, M., Swami, P. and Shastry, P.: Ultra-low Latency Video Codec for Video Conferencing, *Proc. Electronics, Computing and Communication Technologies*, pp.1–5 (2014).
- [8] Ogawa, A., Kobayashi, K., Sugiura, K., Nakamura, O. and Murai, J.: Design and Implementation of DV Stream over Internet, *Proc. Internet Workshop*, pp.255–260 (Feb. 1999).
- [9] Ren, Z., Liu, M., Ye, C. and Shao, H.: The Real Time Video Transmission System Based on H.264, *Proc. Web Information Systems and Mining*, pp.270–274 (Nov. 2009).
- [10] Stream WG: DVTS Project, available from <http://www.sfc.wide.ad.jp/DVTS/> (accessed 2015-01-25).
- [11] Wiegand, T., Schwarz, H., Joch, A., Kossentini, F. and Sullivan, G.J.: Rate-Constrained Coder Control and

Comparison of Video Coding Standards, *IEEE Trans. Circuits and Systems for Video Technology*, Vol.13, No.7, pp.688-703 (July 2003).

- [12] Xu, Y., Yu, C., Li, J. and Liu, Y.: Video Telephony for End-consumers: Measurement Study of Google+, iChat, and Skype, *Proc. Internet Measurement Conference*, pp.371-384 (Nov. 2012).
- [13] Zhang, X., Xu, Y., Hu, H., Liu, Y., Guo, Z. and Wang, Y.: Profiling Skype Video Calls: Rate Control and Video Quality, *Proc. INFOCOM*, pp.621-629 (Mar. 2012).
- [14] 原田啓司, 丸山 充: 非圧縮 HDTV over IP 伝送技術 (i-Visto), 電子情報通信学会総合大会講演論文集, Vol.2006, No.2 (Mar. 2006).
- [15] 清水健司, 原田啓司, 南 陽: R&D ホットコーナー i-Visto による非圧縮 HDTV 映像の多地点間同期伝送に関する共同実験, *NTT 技術ジャーナル*, Vol.17, No.3, pp.85-88 (Mar. 2005).
- [16] 釘本健司, 小倉 毅, 君山博之, 川野哲生, 清水健司, 丸山 充: 非圧縮 HDTV-IP 伝送におけるフィードバック制御の検討: 応答性の高いストリーミングサーバの実装と評価, 電子情報通信学会技術研究報告 IE, 画像工学, Vol.106, No.243, pp.13-18 (Sep. 2006).
- [17] 田中健二, 櫻田武嗣, 杉浦一徳, 町澤朗彦, 中川晋一: 沖縄 IT シンポジウムにおける沖縄-幕張間 DV 会議システム遅延測定, 電子情報通信学会技術研究報告 IN, 情報ネットワーク, Vol.101, No.413, pp.21-25 (Nov. 2001).
- [18] 徳差雄太, 松谷健史, 空閑洋平, 村井 純: 低遅延により自然な遠隔コミュニケーションを実現する映像配信システムの提案, マルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム論文集, pp.911-917 (July 2013).
- [19] 油谷 暁, 垣内正年, 藤川和利, 猪俣敦夫, 香取啓志, 眞鍋佳嗣, 千原國宏: 非圧縮 4K 超高精細映像のためのインターネット伝送実験: 電子情報通信学会技術研究報告 IA, インターネットアーキテクチャ, Vol.109, No.208, pp.55-58 (Sep. 2009).



徳差 雄太

2014 年慶應義塾大学環境情報学部卒業。現在, 同大学大学院理工学研究科修士課程在籍中。



松谷 健史 (学生会員)

2006 年慶應義塾大学環境情報学部卒業。2008 年同大学大学院政策・メディア研究科修了。同年慶應義塾大学後期博士課程入学。



空閑 洋平

2009 年慶應義塾大学政策・メディア研究科修了。2015 年同大学大学院博士 (政策・メディア)。同年 5 月より現職 (慶應義塾大学大学院政策・メディア研究科特任助教)。インターネット計測技術の研究に従事。



中村 修

1983 年慶應義塾大学工学部卒業。工学博士。1990 年から東京大学大型計算機センター助手を経て 1993 年慶應義塾大学環境情報学部助手となり, 現在, 慶應義塾大学環境情報学部教授。

1987 年から WIDE プロジェクトにてインターネットの研究開発をに携わる。2003 年からは AutoID Lab Japan 副所長となり, ネットワーク型 RFID をはじめ各種センサネットワークの研究開発にも携わり, 2009 年からは, 藤沢地域の WiMAX の運用会社, オープンワイヤレスプラットフォーム合同会社の技術協議会委員長となり無線インフラを含めたインターネット関連の研究開発も行う。2014 年からは W3C/KEIO のサイトマネージャに就任し, Web 関連の標準化活動も行う。ACM 会員, 電気情報通信学会会員, ISOC 会員。