

# HDD 録画型ハイビジョンカメラを用いた 講義ビデオ自動撮影加工システムの開発

永井孝幸<sup>1\*</sup>

<sup>1</sup> 熊本大学 総合情報基盤センター

**概要:** 収録時のカメラ操作が不要な簡易な講義収録方式として、固定ハイビジョンカメラと仮想カメラワークを用いた方式が知られている。この方式と自動収録システムを組み合わせることで大規模な講義収録が可能になると考えられるが、既存の自動収録システムの多くはハイビジョン撮影に対応していない。そこで今回、市販の HDD 録画型フルハイビジョンカメラと小型サーバを組み合わせた低コストな講義自動撮影加工システムを開発した。簡易な外付け回路によりビデオカメラの制御と撮影データのキャプチャを全て USB 経由で行えるようにし、小型 Linux サーバによる自動撮影・キャプチャを可能とした。撮影スケジュールは iCalendar 形式対応のスケジュール管理ソフトで編集し、ネットワーク経由で配信する。

## Automation of lecture recording by AVCHD camcorder

Takayuki Nagai<sup>1</sup>

<sup>1</sup> Center for Multimedia and Information Technologies, Kumamoto University

**Abstract:** A possible solution for large-scale lecture recording is the combination of recording with fixed high-definition camcorder and post-processing of camerawork generation that traces regions of interest. In this paper, we describe the automatic lecture recording system we developed with AVCHD camcorder and micro Linux server. In AVCHD camcoders recorded videos are stored as files on their filesystem; video capture now becomes an easy task of copying files from USB storage to the server. With a simple auxiliary circuit, the server can control recording and capturing videos via only USB interface. Recording schedule is easily maintained through iCalendar-ready application.

## 1 はじめに

講義ビデオの撮影を省力化するために、これまでに様々な研究が行われている。(1) 可動式三脚を用いた撮影方式 [1][2] にはじまり、(2) 複数カメラ切替による自動撮影方式 [3] や (3) 固定カメラと仮想カメラワークの組み合わせによる撮影方式 [4][5][6][7] も登場している。

講義撮影を大規模に行うことを考えた場合、人手による撮影方式では撮影スタッフの person 費だけでなく、スタッフの手配・研修・管理の手間も発生する。このため継続的に撮影を行うことは容易でなく、組織の実情に応じて運用方法も含めた様々な工夫が必要となる [9][10]。一方で自動撮影システムを用いる場合は人手の問題は解決されるものの、一般に専用の可動式カメラ・制御ユニットの設置を伴うために高価であり、多くの講義

室に設置することはコスト面から困難である。そのため、継続的な収録が容易な反面、自動収録可能な講義が実質的にはごく一部の科目に限られてしまうという問題がある。

収録時のカメラ操作が不要な簡易な講義収録方式として、固定ハイビジョンカメラと仮想カメラワークを用いた方式 [5] [6][7][8] が知られている。この方式はカメラ操作が不要であるため省力化・自動化に向いており、また市販ハイビジョンカメラの低価格化が進んでいることから導入コストも比較的安価であり、自動収録システムを組み合わせることで大規模な講義収録が可能になると考えられる。しかし、既存の自動収録システムの多くはハイビジョン撮影に対応していない。

そこで今回、市販の HDD 録画型フルハイビジョンカメラと小型サーバを組み合わせた低コストな講義自動撮影加工システムの開発を行った。撮影スケジュールの管理を容易におこなえるようにするため、iCalendar 形式に対応した汎用のカレンダーソフトによる撮影ス

\*連絡先: 熊本大学総合情報基盤センター  
〒 860-8555 熊本市黒髪 2-39-1  
E-mail: tnagai@cc.kumamoto-u.ac.jp

スケジュール管理を可能にするためのツールも合わせて作成した。

以下、本報告書の構成は次の通りである。まず2章では現在一般に利用可能なハイビジョン映像の収録方法について述べる。3章で今回開発した講義ビデオ自動撮影・加工システムの構成について述べた後、4章で撮影カメラの制御方法、5章で収録素材の自動加工システムについて述べる。最後に実装したシステムの稼働状況について6章で述べる。

## 2 ハイビジョン映像の収録方法

ハイビジョン映像の自動収録・加工を低コストで行うには映像信号の伝送・キャプチャが最初の課題となる。この章では現在一般に利用可能なハイビジョン映像の収録方法について述べる。

現在市販されている民生用ハイビジョンカメラはHDV規格[11]またはAVCHD規格[12]に準拠しており、ハイビジョン映像の入出力用インタフェースとしてIEEE1394, HDMI[13], USBが利用可能である。これらの機器とPCを組み合わせるとハイビジョン映像を収録する1つの方式はハイビジョンカメラから出力されるHDVストリーム、あるいはHDMI信号を取りこぼしなくキャプチャする方式(以下、ストリームキャプチャ方式)である。ただしこの方法は高性能なCPU・I/O・ディスクを備えたキャプチャ端末が必要であり、収録システムの大型化・コスト増につながる。

これとは別の方法として、AVCHDカメラ内にファイルとして蓄積された撮影データを撮影後に取り出す方式(ファイルキャプチャ方式)が考えられる。この方法ではカメラ内のファイルをコピーするだけでよい。そのためキャプチャ端末は高性能である必要はなく、組み込み用PCやマイクロサーバを用いて低コストで収録システムを構築することが可能となる。

以下、両方式の概要とメリット・デメリットについて述べる。

### 2.1 ストリームキャプチャ方式

HDV規格はDVテープにハイビジョン映像を収録するための規格として2003年にキヤノン、シャープ、ソニー、日本ビクターによって策定された規格である[11]。最高で1440×1080解像度の1080i画像の収録が可能であり、IEEE1394インタフェースを通じてMPEG-2 TSストリームとして映像・音声の外部出力を行うことができる。1080i解像度のHDVストリームの帯域は25Mbpsであり、近年の一般的なデスクトップ/ノートPCであれば支障なくキャプチャを行える。ただし、組み込み用CPUを用いた小型PCではディスクへのデー

タの書き込みが追いつかないためにキャプチャは実際上不可能である。

IEEE1394ケーブルの長さは規格上1本あたり5mまでであるが、リピータケーブルを多段接続することで最大で50mまで延長することが可能である。このためカメラとキャプチャ端末の設置位置は自由度が高い。IEEE1394インタフェースを備えた機器であればWindows, Mac, Linuxと幅広いOSで利用可能であり自動収録システムの構築に適しているほか、高性能PCとGigabit Ethernetなどの広帯域ネットワークを組み合わせることで遠隔地にHDVストリームを中継することも可能である。

HDMI規格は非圧縮デジタル音声・映像を伝送するためのインタフェースとして作成された規格であり、10Gbpsという高い帯域を持つために1440×1080解像度だけでなく1920×1080解像度の非圧縮映像を伝送することができる。近年ではハイビジョンカメラだけでなくゲーム機・PCの映像出力にもHDMIが利用されるようになっており、HDMI信号の収録システムを構築することで幅広い映像素材の収録が可能になると考えられる。

映像圧縮後のデータが出力されるHDV規格とは異なり、HDMIでは撮像素子の信号が圧縮されずに伝送されるため画質が高いという特徴がある。その代わり、広帯域の映像信号をキャプチャするために専用のキャプチャボード、高性能なCPU・I/O・ディスクを備えたキャプチャ端末が必要である。例えばトムソン・カノーブス株式会社製のHD対応ノンリニアビデオ編集システムHD STORMでは、専用のハードウェアエンコードボードを用いてHDMI信号を独自のCanopus HQ形式にリアルタイムにエンコードする。フルハイビジョン映像ではエンコード後のビットレートが100Mbpsを超えることもあるため、Pentium 4 2.8GHz以上のCPU、回転数7200rpm・転送速度300MbpsのSerial ATAディスクの利用が推奨されている。

撮影カメラを天井に設置することを考えた場合、カメラからキャプチャ端末までのHDMI信号の引き回しに留意する必要がある。市販のHDMIケーブルで手軽に入手できるものは10m程度の長さまでであるが、カメラからキャプチャ端末までの引き回しを考えると十分な長さとは言えない。市販のHDMIリピータを使って中継することで合計40m程度まで延長できるものの、長尺ケーブルによる接続では機器の組み合わせによって信号が認識されないことがあるだけでなく、長尺ケーブルとリピータだけで10万円程度のコストが必要である。

## 2.2 ファイルキャプチャ方式

AVCHD 規格はハイビジョン映像収録のためにソニーとパナソニックが 2006 年に策定した映像フォーマットであり [12]，最大解像度 1920×1080 のフルハイビジョン映像 (帯域 24Mbps) までサポートされている。撮影データがカメラ内のファイルシステム上に保存される点の特徴であり，記録メディアとして DVD, HDD, メモリーカードに対応している。この特徴を生かし，AVCHD 対応ビデオカメラでは HDD を使った長時間録画モデル，SD カードを使った小型モデルなど撮影用途別に様々な機種が存在している。また 2009 年には，AVCHD Lite と呼ばれる解像度を 1280×720 に限定したサブセット規格が策定され，ハイビジョン動画の撮影が可能なデジタルカメラも登場している。

AVCHD カメラでは撮影データはカメラ内の記録媒体にファイルとして蓄積されるため，通常はビデオクリップを撮影後に取り出す収録方式 (ファイルキャプチャ方式) を用いる。多くの機種では撮影データの取り出し用に USB 2.0 High Speed インタフェース (最大 480Mbps) が設けられており，キャプチャ端末からは USB ストレージデバイスとしてアクセスすることができる。キャプチャ操作は単なるファイルのコピーであるため，キャプチャ端末の性能が低くてもキャプチャに要する時間が延びるだけでデータの取りこぼしが生じるわけではない。このため，キャプチャに要する時間と収録システムのコストのトレードオフを考慮しながら幅広い機種・OS をキャプチャ端末に利用することができる。ただし，カメラを USB デバイスとして認識させるためには通常，カメラ側の USB 接続ボタンを押すなどの操作が必要であり，自動収録システムを構築するにはこの点に工夫が必要となる。

ファイルキャプチャ方式では撮影カメラとキャプチャ端末の間は USB ケーブルで接続することになる。USB ケーブルの長さは規格上 1 本あたり 5m までであるが，リピータケーブルを多段接続することで最大で 20m まで延長することが可能である。しかしながら，撮影カメラを天井に設置するケースではキャプチャ端末までのケーブル長に留意が必要である。

## 3 講義ビデオ自動撮影加工システム

この章では，今回開発した講義ビデオ自動撮影加工システムの構成について述べる。自動撮影システム設計の基本方針として，機材の設置・導入を容易にするために市販の安価な小型機材を用いることとした。また，仮想カメラワーク [14] を用いて撮影後に収録ビデオを加工することを前提とし，撮影カメラの向きを制御するための機構は設けていない。

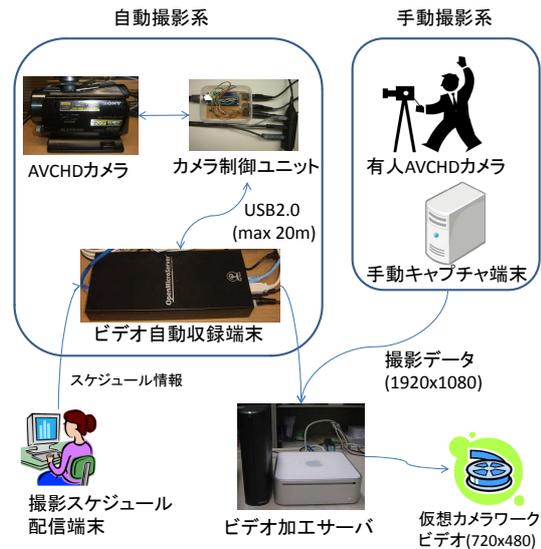


図 1: 講義ビデオ自動撮影加工システムの構成

3.1 節でシステムの構成について述べ，3.2 節で撮影スケジュールの管理方法について説明する。

### 3.1 システム全体構成

開発したシステムはスケジュール配信端末，ビデオ収録端末，ビデオ加工サーバから構成される。スケジュール配信端末から配信される撮影スケジュールに従ってビデオ収録端末が自動的に講義のフルハイビジョン撮影を行い，撮影データをビデオ加工サーバに自動アップロードする。ビデオ加工サーバは新規データの到着を検出し，仮想カメラワークにもとづいたビデオの自動生成を行う。撮影後のカメラをビデオ収録端末に接続して手動でキャプチャ開始の操作を行うことも可能であり，講義撮影のみ有人で行うという使い方も想定している。

大規模に講義収録を行うことを考えた場合，各講義室に設置する機材は極力小型・安価・頑健であることが望ましい。そこで，今回は Linux マイクロサーバ [15] を用いてファイルキャプチャ方式 (2.2 節参照) による自動収録を実現することとした。マイクロサーバを採用することでビデオ収録端末そのものを天井・床下等の狭い空間に設置することが可能となる。なお本システムでは収録端末とカメラ間の制御信号・撮影データのやりとりは全て USB 経由で行うため，USB2.0 High Speed インタフェースを備えた機種であればマイクロサーバに限らず，設置形態・利用形態に応じてデスクトップ PC・ノート PC など様々な機種を収録端末として利用可能である。

AVCHD 形式のビデオデータを加工するには高性能な CPU と大容量 HDD を備えた PC が必要であるが，導入コストや設置・メンテナンスの手間を考えると各

講義室にそのような PC を設置することは現実的ではない．そこで，別途ビデオ加工サーバを設け，ビデオ収録端末とビデオ加工サーバを LAN でつなく構成としている．撮影データの伝送は LAN 上のファイル転送として行われるためデータの長距離伝送が可能であり，キャプチャ端末とビデオ加工サーバをキャンパスをまたいで設置することも可能である．また，ビデオ加工に必要な高性能 PC をサーバ室などに集約して設置できるという利点もある．

なお，本システムはネットワーク上に分散した構成となるため，各構成要素はキャンパスネットワークに障害が発生した場合にも自律的に動作する必要がある．この点に関する工夫は 3.2 節，5.1 節で述べる．

### 3.2 撮影スケジュールの管理方法

科目の自動収録を大規模に行うには，撮影スケジュールの管理を手軽に行える仕組みが不可欠である．講義の撮影は基本的には学期当初に決めた科目スケジュールの通りに実施すればよいので，学務システムのデータを元に機械的にスケジュールデータを生成すればよいように思われがちである．しかし，実際には休講や補講のために臨時に科目開講日の変更されることがあるため，撮影スケジュールを一部修正できる仕掛けが必要になる．

今回開発したシステムでは自動収録端末を Linux で構築しているため，cron を用いれば撮影コマンドを自動的に実行することは容易である．しかし，crontab の書式は必ずしも見やすいものではないため，込み入ったスケジュールを設定するには不向きである．また，crontab 編集のために自動収録端末にログインして作業することは操作ミスが発生するおそれがあるため極力避けることが望ましい．

スケジュール管理のための標準データ形式として近年 iCalendar 形式 [16] が普及しており，Google Calendar, iCal, Sunbird などの主要なスケジュール管理ソフトがこの形式に対応している．撮影スケジュールの一括登録や確認作業，共同編集のしやすさなどを考え，今回は iCalendar 形式対応のスケジュール管理ソフトを用いて撮影スケジュールの管理を行うこととした．

撮影スケジュールを登録するには，スケジュールソフト上で撮影制御コマンドを記入した予定を登録する．現在利用可能なコマンドは (1)power on ( 予定開始時刻に電源 ON )，(2)power off ( 予定開始時刻に電源 OFF )，(3)record(予定開始時刻から終了時刻まで録画実施)，(4)capture ( 予定開始時刻にキャプチャ開始 ) の 4 つである．スケジュール管理ソフトを用いることで毎週指定曜日の繰り返し録画といった指定を手軽に行うことができ，さらに撮影スケジュールを一目で確認することができる ( 図 2 ) ．



図 2: iCal を用いた自動撮影スケジュール管理の例

```

lecvideo@nagi193:~$ crontab -l
00 07 01 04 * /home/lecvideo/toolbox/bin/util/startRecording.sh
15 07 01 04 * /home/lecvideo/toolbox/bin/util/stopRecording.sh
20 07 01 04 * /home/lecvideo/toolbox/bin/util/capture.sh
40 08 01 04 * /home/lecvideo/toolbox/bin/util/startRecording.sh
10 10 01 04 * /home/lecvideo/toolbox/bin/util/stopRecording.sh
30 10 01 04 * /home/lecvideo/toolbox/bin/util/capture.sh
30 11 01 04 * /home/lecvideo/toolbox/bin/util/powerOff.sh

```

図 3: 撮影スケジュールから生成された crontab の例

作成されたスケジュールデータをスケジュール配信端末に保存しておくことで，当日分の撮影スケジュールがスケジュール配信端末から一定時間おきに各自動収録端末に配信される．自動収録端末では受信したデータをもとにしてカメラ制御用のコマンドを指定時刻に実行するための crontab を生成する ( 図 3 ) ．当日のスケジュールデータを繰り返し配信するのは一見無駄に見えるが，これはネットワーク障害や各端末のメンテナンス停止があっても安定して動作するようにするためである．例えば，日中にキャンパスネットワークの障害が発生したとしても，当日の明け方までに自動収録端末にスケジュールデータが届いていれば自動収録自体は問題なく実行される．

## 4 撮影カメラ制御方法について

2.2 節で述べたファイルキャプチャ方式を用いて講義の自動収録を行うには，録画の開始・終了と撮影データのキャプチャを自動化する必要がある．また，実運用においては撮影範囲調整のためのズーム制御と電源 ON/OFF の制御も必要である．この章では市販機材を用いてこれらの制御を実現する方法について述べる．

これらの制御を行える民生カメラは多くないが，要求を満たす機種として今回は SONY 製 AVCHD カメラ HDR-SR12 を用いた．この機種は LANC コントロール端子を備えており，LANC プロトコル [17] に従ってコマンドを送信することで録画の開始・終了，ズーム制御，電源 OFF を行うことができる．一部，USB 接続の制御など LANC で対応できない部分についてはリレー回路を用いたカメラ制御ユニットを自作して対応している ( 表 1 ) ．

表 1: 撮影カメラ (HDR-SR12) の制御手段

操作	制御手段
ズーム制御	LANC
録画開始・終了	
電源OFF	
電源ON	リレー回路
USB接続開始	

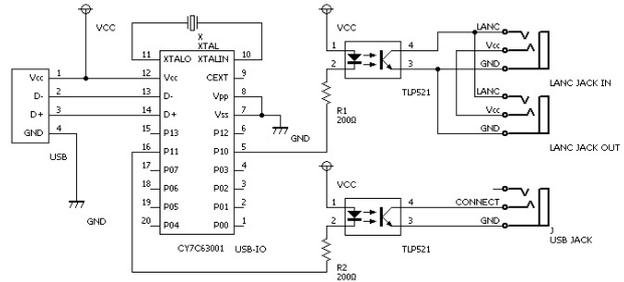


図 4: カメラ制御ユニット回路図

以下, 4.1 節で LANC プロトコルによるカメラ制御, 4.2 節で自作のカメラ制御ユニットによる制御について説明する。

#### 4.1 LANC プロトコルによる制御について

LANC 規格自体は民生用ビデオカメラ・ビデオデッキの制御に古くから利用されており, 現在でも高級機種では LANC 端子を備えているものが多い。しかし, ノンリア編集の普及に伴って業務用機材 [18] も含めてパソコンに接続可能な LANC インターフェースは一般に販売されなくなっている。国内で現在入手可能な製品には RS-232C 経由で LANC 通信を行うための LANC アダプター [19][20] があるものの, 特定の LANC コマンドしか送信できないなどの制約がある。LANC による通信は基本的には 9600bps のシリアル通信であり通信用インタフェースを自作することも可能であるが, 今回のシステムではイギリスの AVITResearch 社から販売されている USB-LANC アダプタ [21] を用いることにした。

アダプタ付属の開発キットでは Windows 用のライブラリしか提供されていないが, このアダプタは USB の HID 機器として取り扱えるため, 汎用の USB ドライバを用いて任意の LANC コマンドを送信することができる。今回 UNIX からこのアダプタを制御するためのライブラリを libusb[22] を用いて自作し, Linux, FreeBSD からカメラ制御が行えることを確認している<sup>1</sup>。

電源 ON 操作も LANC 規格で規定されているが, LANC 信号線を 140ms 以上 GND レベルに保つことで電源 ON が実行されるという規格のため, 4.2 節で述べるように簡単な外付けリレー回路を作成して対処している。

#### 4.2 カメラ制御ユニットについて

LANC コマンドの送信だけでは扱えない電源 ON と USB 接続の制御については, 信号線の制御をするためのリレー回路を備えたカメラ制御ユニットを作成し, 制

御ユニットを USB 経由で制御することで対処した。このユニットは USB マイコン [23] を使ってフォトカプラを用いたリレー回路を制御し, 該当する信号線を GND レベルに下げる操作をするだけの簡易な回路である (図 4)。OS 側からは汎用の HID デバイスとして認識され, libusb[22] を用いて Linux, FreeBSD から制御が行えることを確認している。

AVCHD カメラ内に蓄積された撮影データをキャプチャするには, カメラを制御端末に USB ストレージとして認識させる必要がある。通常, この操作はカメラ側のスイッチを操作して行うようになっており, 完全に自動的に素材を取り込むことは想定されていない。カメラの機種に依存した解決法ではあるが, この部分の操作を自動化するための簡単な外付けリレー回路を作成することで撮影データの自動キャプチャを実現した。

HDR-SR12 では, 付属のハンディカムステーション (USB クレイドル) にカメラをセットし, ステーション上のワンタッチディスクボタン (機械式プッシュボタン) を押すことでパソコンに USB 接続を行うことができる。そこで, このスイッチ操作を電気的に行えるように信号線を引き延ばし, 制御ユニット内のリレーに接続した。制御ユニットの操作により信号線を GND レベルに変化させることで USB 接続が開始される。なお, USB 接続解除の正式な操作はカメラ本体の液晶タッチパネルからしか行うことができないため, LANC コマンド送信によるカメラの電源 OFF によって間接的に USB 接続の解除を実現している。

### 5 撮影素材の自動加工手順

この章では, 自動収録端末 (あるいは手動キャプチャ端末) とビデオ加工サーバの連携による講義ビデオ自動加工手順について説明する。AVCHD カメラが USB ストレージとして端末に接続されると, 端末内のキャプチャ用スクリプトが自動実行され, 撮影データが端末内の HDD にコピーされる。その後, 端末からビデオ加工サーバに素材が自動アップロードされ, 仮想カメラワークを用いて素材の自動加工が行われる。以下, 上記動作の詳細について述べる。

<sup>1</sup>Mac OS X では OS 標準の HID デバイスドライバと干渉するため動作しない

## 5.1 ビデオ素材自動キャプチャについて

AVCHD カメラが USB ストレージデバイスとしてアクセスできることを利用し、撮影データを自動的にキャプチャするための仕組みを Linux 上に構築した。Linux ではプラグアンドプレイ機器の接続・切り離しに応じて指定されたプログラムを実行するための udev[24] と呼ばれる機構が実装されており、この機構を通じてキャプチャ用プログラムを呼び出すことで自動キャプチャを実現している<sup>2</sup>。

自動収録端末を用いた自動キャプチャの手順は次のようになる。まず (1) カメラ制御ユニット内の USB 接続用リレーに通電し、(2) USB ストレージの接続を収録端末が検知する。これにより (3) udev による自動キャプチャスクリプトの実行開始、(4) カメラを USB ストレージとしてマウント、(5) 撮影データを収録端末にコピー、(6) カメラの USB 接続を解除、(7) 撮影データをビデオ加工サーバにアップロード、という一連の動作が自動的に行われる。なお、手動キャプチャ端末の場合は上記の (1) の動作が「(1') キャプチャ端末にカメラを USB 接続する」に置き換わるだけでその後の動作は同じである。

上記 (5) の処理を飛ばしてカメラ内のデータを直接アップロードすることも考えられるが、ネットワークの不具合によりアップロード先と正常に通信できなくなった場合に問題が生じる。アップロードが完了しないためにカメラの USB 接続の解除が次の撮影に間に合わず、自動収録に支障をきたすケースである。そこで、撮影データを収録端末に接続された HDD にいったんコピーしておき、バックグラウンドでアップロード処理を行う実装としている。特に建物・キャンパスをまたいで転送するようなケースでは、知らないうちにアップロード先でネットワーク障害やサーバメンテナンス・計画停電などが発生している可能性があるため、長時間のネットワーク途絶を前提としたストア・アンド・フォワード型の通信方式の実装が必要になる。

この実装ではネットワークの不具合と関係なく、カメラ・端末間でのみキャプチャ処理が完了するため撮影に支障は生じない。またこの方式では、手動キャプチャ端末の場合はカメラからのデータコピーが完了した時点ですぐにカメラを持ち出すことができ、カメラの稼働率を上げられるというメリットもある。

なお今回利用したカメラには 12 時間分の撮影データしか保存できないため、キャプチャの済んだデータはカメラから削除するようにしている。ただし、長期間の運用においてはキャプチャ用プログラムの不具合やハードウェア故障、瞬電・停電のためにコピーが正常終了しないことが考えられる。そのためトランザクション管理を行い、全てのビデオクリップが正常にコピー

<sup>2</sup>FreeBSD, NetBSD でも usbd による同様の実装が可能である

```
ffmpeg -i input -vfilters camera=XOFFSET:  
YOFFSET: 720: 480: camerawork.txt -y output
```

図 5: 開発したカメラワークフィルタの使い方

されたことを確認して初めてカメラ内のビデオクリップを削除するようにしている。

## 5.2 ビデオ自動加工について

ビデオ加工サーバでは監視対象フォルダに新しいビデオ素材がアップロードされたことを検知し、仮想カメラワークにもとづいたビデオの自動加工を行う。仮想カメラワークには以前著者が考案した手法 [14] をそのまま用いているが、フルハイビジョン解像度のフレームから NTSC 解像度のフレームにトリミングする処理を ffmpeg 自体に組み込むことで計算時間を短縮することに成功した。また、以前はハイビジョン映像の加工に一部 Windows 上のソフトを用いた手作業が必要であったが、今回、オープンソースのツール類を組み合わせることにより Linux 上で処理を自動化した。AVCHD データの加工方法は文献 [25] を参考にしている。

基本的な処理は次のとおりである。まず、受け取った AVCHD ファイルのヘッダ情報を読み取り、どのファイルがどの撮影に対応するかを判定する。長時間の撮影データは 2GB 単位で分割された複数の AVCHD ファイルとしてカメラ内に記録されるため、撮影の区切りに合わせて AVCHD ファイルを連結する必要があるためである。次に、xport[26] を用いて AVCHD ファイルから H.264 ビデオストリームを取り出す。その後、ffmpeg を用いてビデオストリームをデコードし<sup>3</sup>、仮想カメラワークを計算する。

最後に、計算したカメラワークに従ってフルハイビジョン映像の各フレームから NTSC 解像度に対応する範囲のトリミングを行い、NTSC 解像度のビデオを出力する。ffmpeg で直接この処理を行うために、libavfilter[27] を用いて仮想カメラワーク用ビデオフィルタ *camera* を作成した。

作成したビデオフィルタ *camera* は引数で指定されたカメラワーク情報ファイルの内容を読み取り、各フレーム毎に指定された範囲のトリミングを行う (図 5)。カメラワーク情報ファイルには先頭行から順にフレーム毎のトリミング範囲の左上座標が「x 座標 y 座標」という書式で記載されている (図 6)。90 分の講義映像 (約 162000 フレームに対応) に対して、カメラワーク情報ファイルのサイズは 1.2MB 程度である。

<sup>3</sup>PAFF インターレース対応バージョンの ffmpeg が必要

```
#書式: (トリミング範囲左上の) x 座標 y 座標
825 350
831 350
838 350
844 350
```

図 6: カメラワーク情報 (camera.txt) の例

表 2: 自動収録システム構築費用

名称	品名	費用(円)
AVCHDカメラ	HDR-SR12	110,000
マイクロサーバ	OpenMicroServer OMS400	56,000
USB-LANCアダプタ	Control-L USB Cable	12,000
外付けUSBディスク	HD-CE1.0TU2	17,000
カメラ制御ユニット	自作	2,000
合計		197,000

## 6 システム稼働状況

この章では開発した講義ビデオ自動撮影加工システムの稼働状況について述べる。

今回は撮影カメラにソニー製ハイビジョンカメラ HDR-SR12, 自動収録端末にプラットフォーム社製 Open Micro Server[15] と BUFFALO 社製 USB 外付けディスク HD-CE1.0TU2 を用いて自動収録システムを構築した。費用は約 20 万円である (表 2)。カメラ制御・自動キャプチャ・自動アップロードの処理は自作の C プログラムとシェルスクリプト (約 3,000 行) で記述し、コマンドのバイナリと合わせて約 150KB とコンパクトに実装している。

この実装では、90 分の撮影完了後、ビデオ加工が始まるまでのファイル転送に要する時間は合計で 80 分程度である。90 分の撮影 1 回につき 10.3GB の AVCHD データがカメラ内に蓄積され、このデータを収録端末を介してカメラから外付け USB HDD へコピーするのに約 30 分要している。収録データをビデオ加工サーバにアップロードする処理は、サーバのフォルダを CIFS フォルダとしてリモートマウントし、リモートフォルダに対してファイルコピーを行う方式で実装している。この方式で 100Mbps の LAN を経由して 10GB のデータをアップロードするのに約 50 分要している。コピーに要する時間が長いのはマイクロサーバの CPU がボトルネックであることが分かっており、他機種での実装も検討中である。

ビデオ自動加工サーバには、小型デスクトップ PC の MacMini(Core 2 Duo 2GHz) と作業用外付けディスク (BUFFALO 社製 HD-CE1.0TU2) を用いた .xport,ffmpeg と自作プログラムをシェルスクリプト (約 500 行) で組

み合わせるによりビデオの自動加工を実現している。90 分の AVCHD データ (約 10GB) から仮想カメラワークを施した NTSC 解像度の MPEG-2 ビデオを生成するのに、この構成では約 5 時間 40 分かかる。このうち約 3 時間が AVCHD データの変換に費やされており、仮想カメラワークに基づいたビデオのトリミング処理そのものは約 40 分で完了する。

2009 年 3 月から 4 月までの一ヶ月間、研究室内にシステムを設置して常時稼働させ運用試験を行った。カメラとサーバ間を市販の USB リピーターケーブルを用いて 15m 延長した状態で、定時スケジュールにもとづいた自動撮影・自動加工が問題なく動作することを確認している。HDD 録画型カメラでは長期間の運用における内蔵 HDD の故障が問題となる。そこで撮影終了後はカメラの電源を落とし、HDD の稼働時間を減らすようにしている。

なお、自動収録端末については VMWare 上の仮想 Linux での実装も終わっており、既存のノート PC・デスクトップ PC をすぐに自動収録端末・手動キャプチャ端末として利用可能である。

## 7 まとめ

本報告では市販のフルハイビジョン AVCHD カメラとマイクロサーバを組み合わせた講義ビデオ自動撮影加工システムの実装について述べた。このシステムではネットワークから配信される撮影スケジュールにもとづいて講義のフルハイビジョン撮影を行い、仮想カメラワークによってビデオの自動加工を行う。ハイビジョン映像のキャプチャにファイルキャプチャ方式を採用し、専用のキャプチャボード・ソフトやキャプチャ用高性能端末が不要ことが特徴である。カメラ制御と撮影データの転送を全て USB2.0 インタフェースを介して行うため、市販の小型 Linux サーバによる低コストなシステム構築が可能である。特にマイクロサーバを用いることで自動収録端末を天井・床下に設置することができる。

AVCHD ビデオカメラで撮影したデータを自動的にキャプチャするには、制御用 PC に対してカメラを USB ストレージとして認識させる必要がある。しかし、LANC プロトコルではこのような機能は提供していないため、USB 接続制御用の回路をキャプチャ用 USB クレイドルに追加して対応した。カメラ制御のために簡単な外付け回路を自作する必要があるが、カメラ本体の改造は一切不要である。

仮想カメラワークにもとづいてビデオを自動加工するシステムをオープンソースのツール類を組み合わせで実現し、撮影から加工までを自動化することに成功した。また撮影スケジュールの管理には iCalendar 形式

のデータを用い, iCal, Sunbird など iCalendar 対応スケジュール管理ソフトを用いた編集を可能としている。

現在のシステム実装では 90 分ビデオの加工に約 6 時間かかっているが, そのうち 3 時間ほどが AVCHD ビデオのデコード処理である。最近では AVCHD ビデオのハードウェアデコーダを搭載したビデオカードが安価に入手可能になっており, ハードウェアによるデコード処理の高速化を検討している。

研究室内で一ヶ月間の連続稼働試験を行い, スケジュール撮影, 自動キャプチャ, 自動レンダリングが問題なく稼働していることを確認した。今回は OpenMicroServer を用いて自動収録端末を実現したが, 100Base-T ネットワークと USB 2.0 High Speed インタフェースを備えた Linux 対応機種であれば同様に利用可能である。

現在は実際の講義室での運用に向け, 機材設置方法の検討を行っているところである。撮影用カメラを天井や壁に固定設置する場合, カメラの取り付け位置・角度に応じてレンズのズーム調整や仮想カメラワーク計算のためのパラメータ調整を行う必要がある。この作業を簡素化することが今後の課題として挙げられる。また, 長期間の運用における各機材の耐久性の確認, 撮影素材管理システムの開発も今後の課題である。

## 謝辞

本研究は科研費 ( 課題番号:21700818 ) の助成を受けたものである。

## 参考文献

- [1] Y.Rui, L.He, A. Gupta and Q.Liu: “Building an Intelligent Camera Management System”, Proceedings of the ninth ACM international conference on Multimedia, pp. 2 – 11, 2001.
- [2] Q.Liu, Y.Rui, A.Gupta and J.Cadiz: “Automating Camera Management for Lecture Room Environments”, Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 442 – 449, 2001.
- [3] 西口敏司, 亀田能成, 角所考, 美濃導彦: 大学における実運用のための講義自動アーカイブシステムの開発, 電子情報通信学会論文誌 D-II, vol.J88-D-II, no.8, pp.530–540, 2005
- [4] 大西正輝, 泉正夫, 福永邦雄: “デジタルカメラワークを用いた自動映像生成”, 画像の認識・理解シンポジウム (MIUR2000), vol.I, pp.331-336, 2000.
- [5] 横井隆雄, 藤吉弘巨: “高解像度映像からの自動講義ビデオ生成-仮想カメラワークの実現-” 第 11 回画像センシングシンポジウム, pp.73-76, 2005.
- [6] 横井隆雄, 桐井孝嘉, 藤吉弘巨: “講義イベント検出に基づく短縮講義ビデオの自動生成”, 第 12 回画像センシングシンポジウム予稿集, pp.535-530, 2006.
- [7] 永井孝幸: “ハイビジョンカメラと仮想カメラワークを用いた簡易な講義ビデオ撮影方式について”, 情報処理学会研究報告 2008-CMS-09, pp.56-63 (2008-09)
- [8] NEC 製 追尾型講義コンテンツ自動作成ソフト i-Collabo.AutoRec <http://www.i-collabo.jp/autorec/>
- [9] Takayuki Nagai, “Laboursaving Video Archive System for Daily Manual Recording”, 8th International Conference on Information Technology Based Higher Education and Training
- [10] 長瀧寛之, 永井孝幸, 都倉信樹, 「学生が作業スタッフとして参加する学科内全科目の講義撮影・配信への取り組み」, 社団法人日本工学教育協会論文誌工学教育 2007 年 9 月号, vol.55 no.5 pp.8-14
- [11] HDV 規格  
<http://www.hdmi.org/>
- [12] AVCHD 規格  
<http://www.hdv-info.org/>
- [13] HDMI 規格  
<http://www.avchd-info.org/index.html>
- [14] Takayuki NAGAI, “Simple lecture recording with HDV camera and virtual camerawork”, Proceedings of the 8th International Conference on Applications and Principles of Information Science, pp.321-324, 2009.
- [15] ぷらっとホーム株式会社製 OpenMicroServer OMS400  
<http://www.plathome.co.jp/products/microserver/oms400/>
- [16] RFC 2445, Internet Calendaring and Scheduling Core Object Specification,  
<http://tools.ietf.org/html/rfc2445>
- [17] LANC プロトコル  
<http://www.boehmel.de/lanc.htm>
- [18] LANC インターフェイスボックス IF-FXE2  
[http://www.ecat.sony.co.jp/business/video\\_production/products/index.cfm?PD=12327](http://www.ecat.sony.co.jp/business/video_production/products/index.cfm?PD=12327)
- [19] RS-232/Control-L Interface Adapters  
<http://www.addenda-elect.com/control.htm>
- [20] LANC 232C コントローラ  
[http://www.cypress.ne.jp/t-komeda/products/lanc\\_rs/index.html](http://www.cypress.ne.jp/t-komeda/products/lanc_rs/index.html)
- [21] Control-L USB Cable  
<http://www.avitresearch.co.uk/control-l.htm>
- [22] libusb  
<http://libusb.wiki.sourceforge.net/>
- [23] USB-IO  
<http://km2net.com/usb-io/index.shtml>
- [24] udev  
<http://en.wikipedia.org/wiki/Udev>
- [25] “特集 3 Linux でフルハイビジョン, ビデオ編集から再生まで”, 日経 Linux 2008 年 10 月号, pp.57–73, 2008.
- [26] xport Transport Stream Demuxer  
<http://www.w6rz.net/>
- [27] libavfilter  
<http://wiki.multimedia.cx/index.php?title=Libavfilter>