



Kumamoto University

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

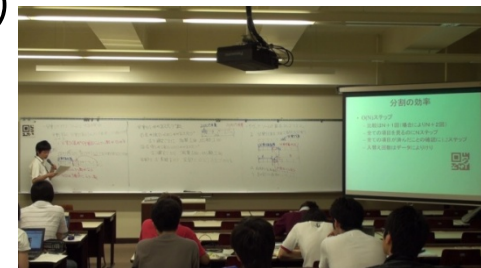
CMS研究会(2009/5/15 三重大学)

熊本大学総合情報基盤センター

准教授 永井孝幸

講義収録を大規模に行うには？

- 有人収録(2003～@鳥環大)
 - 不可能ではないが大変
- 自動収録システム
 - 高価なシステムは設置場所が限定される
 - 自動追尾カメラ、複数カメラ切替など
 - 固定フルハイビジョンカメラはどうか？
 - カメラ自体は普及価格帯(10万円程度)
 - カメラワークは後処理で生成可能
 - **安価な自動収録システムがあればよい**



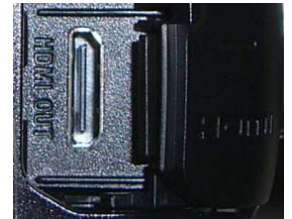
コスト・労力削減のアイデア

- 撮影カメラマン/自動追尾カメラ
 - 固定ハイビジョンカメラ＋仮想カメラワークで代替
- 撮影データの加工
 - ffmpegその他によりUnix上で自動化
- 撮影スケジュール管理
 - iCalendar形式対応ソフトで単純化



ハイビジョン映像のキャプチャ手段

- 映像ストリーム(HDV,HDMI)のキャプチャ
 - 映像信号のリアルタイム保存
 - ドロップフレームあり
 - キャプチャボードや高性能PC・HDDが必要
- AVCHDファイルのキャプチャ
 - USBストレージ内ファイルのコピー
 - ドロップフレームなし
 - USB 2.0(High-Speed)付きPCであればよい
 - 汎用マイクロサーバの採用が可能



今回開発したシステム

- 市販機材による小型・安価な構成
- 自動収録システム(約20万円)
 - AVCHDフルハイビジョンカメラ(HDR-SR12)
 - マイクロサーバ(OMS400)
 - 素材蓄積用HDD(HD-CE1.0TU2)
- 自動加工システム(約10万円)
 - 小型PC(Mac Mini)
- Linux/Unix上で実装



システム全体構成



撮影スケジュール
配信端末

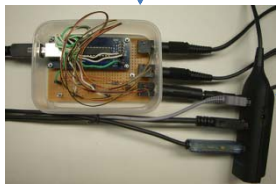
スケジュール情報

自動撮影系



ビデオ自動収録端末

USB2.0
(max 20m)



カメラ制御ユニット



AVCHDカメラ

手動撮影系



有人AVCHDカメラ



手動キャプチャ端末

撮影データ
(1920x1080)



仮想カメラワーク
ビデオ(720x480)

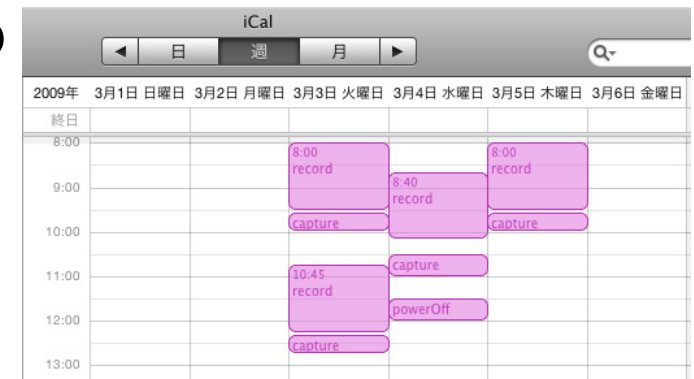


ビデオ加工サーバ



撮影スケジュールの管理

- 学期当初の科目スケジュール通りではない
 - 休講・補講、休校などによる予定変更
 - 学期途中でスケジュール更新が必要
- iCalendar準拠のスケジュール管理ソフトを利用
 - 主要ソフト(iCal,Sunbird,Google Calendar)が対応
 - スケジュールが一目で分かる
 - ネットワーク上での共同編集



撮影スケジュールの配信

- iCalendarデータ

```
atlantis.cc.kumamoto-u.ac.jp:22 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
END:VTIMEZONE
BEGIN:VEVENT
SEQUENCE:6
TRANSP:OPAQUE
UID:C15D0183-C38C-4BC0-B2E1-13AC1994B52B
DTSTART;TZID=Japan:20090225T130000
DTSTAMP:20090225T025336Z
SUMMARY:record
CREATED:20090225T020051Z
DTEND;TZID=Japan:20090225T143000
END:VEVENT
BEGIN:VEVENT
SEQUENCE:8
:
```

- スケジュール配信サーバ

1. 当日分のイベントを抽出
2. VEVENTデータのSUMMARYを抽出
3. 撮影コマンド+実行時刻を配信
powerOn,powerOff,record,capture

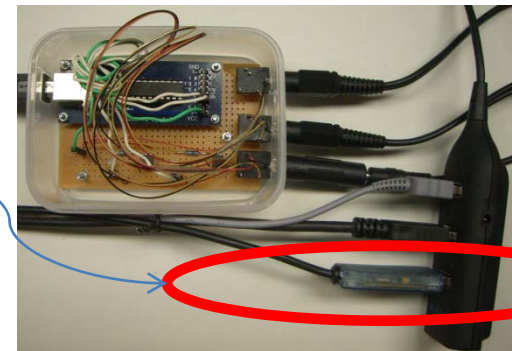
```
lecvideo@nagil93:~$ crontab -l
00 07 01 04 * /home/lecvideo/toolbox/bin/util/startRecording.sh
15 07 01 04 * /home/lecvideo/toolbox/bin/util/stopRecording.sh
20 07 01 04 * /home/lecvideo/toolbox/bin/util/capture.sh
40 08 01 04 * /home/lecvideo/toolbox/bin/util/startRecording.sh
10 10 01 04 * /home/lecvideo/toolbox/bin/util/stopRecording.sh
30 10 01 04 * /home/lecvideo/toolbox/bin/util/capture.sh
30 11 01 04 * /home/lecvideo/toolbox/bin/util/powerOff.sh
```

- 自動収録端末

1. 撮影コマンドを受信
2. 制御コマンド系列に変換
3. crontabを生成・登録

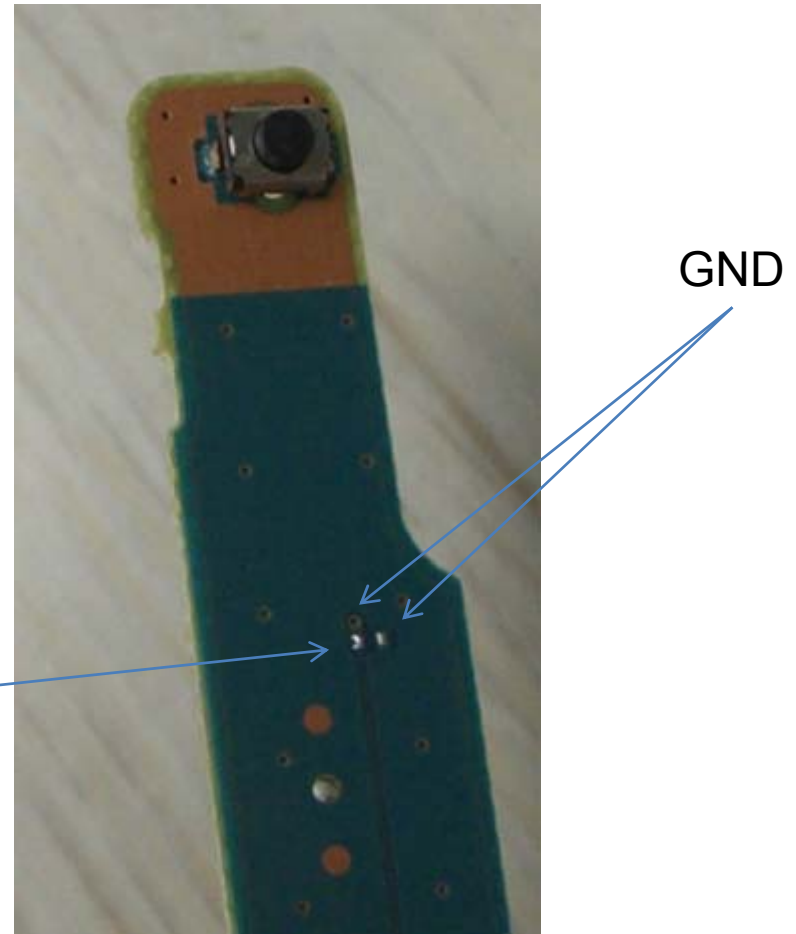
撮影カメラの制御

- 基本はLANC端子を介した制御
 - 市販のUSB-LANCアダプタを利用
- 電源ON・キャプチャ制御はリレー回路を作成
 - USB-IO + フォトカプラ
- カメラ制御・キャプチャを全てUSB経由で実現
 - Linuxマイクロサーバで実装(仮想PCでも動作)



自動キャプチャ制御

- カメラをUSBストレージとして認識させる必要あり
 - 通常はカメラ/クレイドルのボタンを操作
 - HDR-SR12での実装
 - クレイドル側の信号線をリレーで制御
 - ピン間0.25mm
接続制御信号
 - LANCコマンドから直接制御できればよいのですが...



収録素材の自動加工

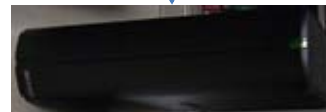
- カメラのUSB接続をトリガーとしてキャプチャ
 - udevによるスクリプトの自動実行
 - トランザクション処理付きアップロード
 - ネットワーク障害に備えて一旦蓄積
- 監視フォルダの新着データを自動加工
 - 1920x1080 AVCHD → 720x480 MPEG-2(仮想カメラワーク)



AVCHDカメラ



ビデオ自動収録端末



一時蓄積用ストレージ



ビデオ加工サーバ



ビデオ自動加工処理の実装

- AVCHD→MPEG-2変換
 - xport, ffmpegの組み合わせ
 - AVCHDファイル連結、PAFFインタレース処理
- 仮想カメラワークの処理
 - 自作プログラム(領域検出、カメラワーク生成)
 - 算出カメラワークにもとづいた動的トリミング
 - ffmpegに自作ビデオフィルタを実装
- ffmpegが動作するUnix/LinuxならOK



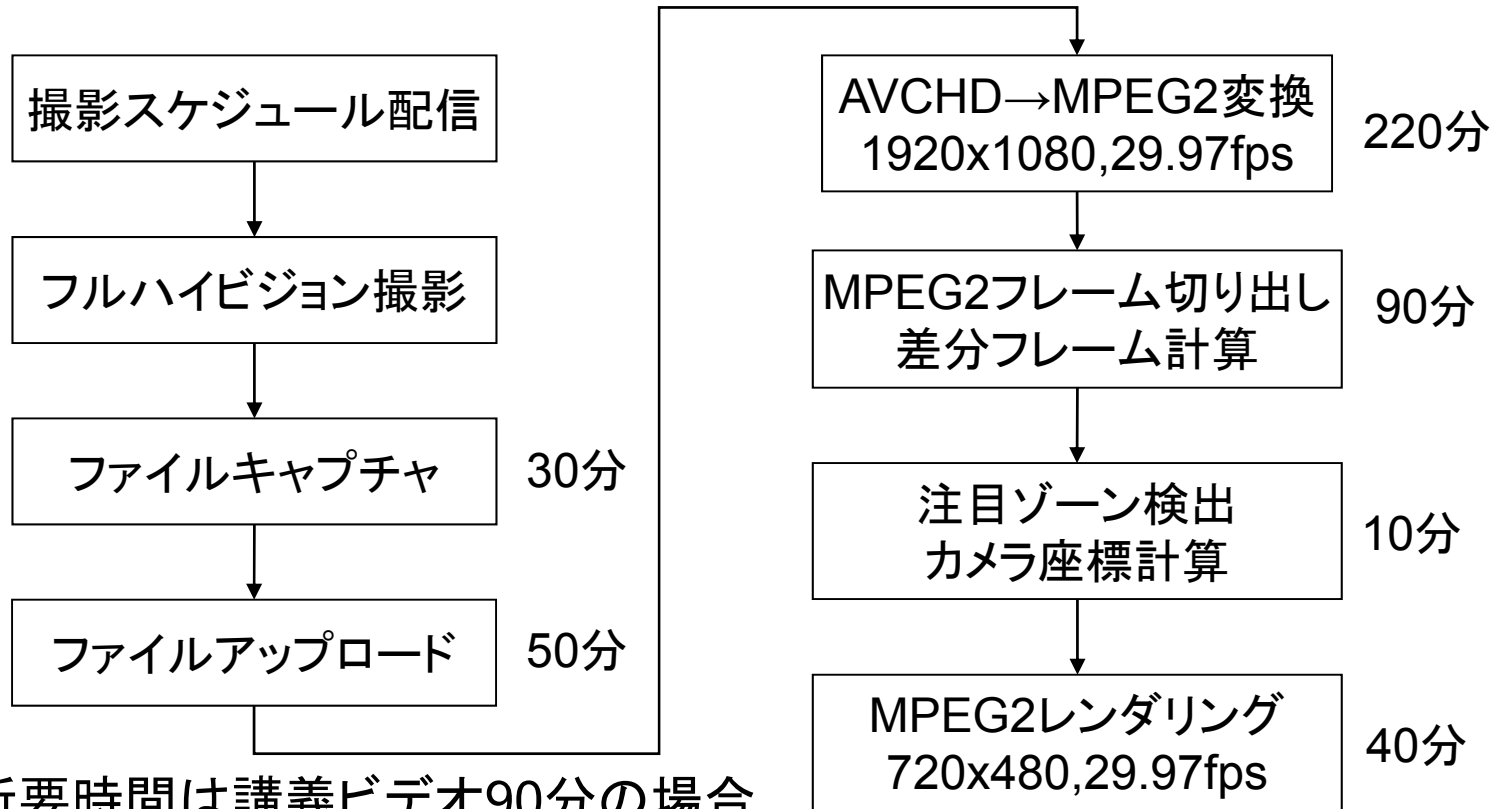
libavfilterによるフィルタ実装

- ffmpegのaudio/videoフィルタAPIを利用
 - 例: `ffmpeg -i input -vfilters camera=XOFFSET:YOFFSET: 720: 480: camerawork.txt -y output`
- 動的フィルタの実装自体は簡単
 - `start_frame`関数でトリミング範囲を登録
 - `draw_slice`関数で指定範囲を描画
- 講義ビデオ(90分)のトリミング処理
 - MPEG-2(1920x1080)→MPEG-2(720x480)トリミング処理
 - Mac Mini (Core2 Duo 2GHz)で約40分



ワークフロー

- 撮影完了→加工開始: 80分
- 加工開始→処理済みビデオ出力: 360分



システム稼働状況

- 研究室内で試験運用(2009年3月～)
 - 定時スケジュール撮影・自動加工
- 講義室内仮設置・動作テスト(2009年5月～)



手動キャプチャ



自動キャプチャ



床パネル: 50cm四方
床下空間: 高さ12cm



自動収録端末(床下設置)



まとめ

- 小型・安価な講義ビデオ自動収録システム
 - ハイビジョン＋ファイルキャプチャ＋小型サーバ
 - USB機器・ストレージの制御システム
 - オープンソースによる仮想カメラワーク実装
- 今後の課題
 - 講義室への常設・定常運用
 - 各機材の耐久性の確認
 - HDR-SR12以外の機種への対応
 - 素材管理システム(OpenCast?/Sakai?/自作?)

