

文字認識 API を用いた講義アーカイブ閲覧システムの設計

大西 淑雅^{*1}, 下井倉 慶紀^{*2}, 山口 真之介^{*1}, 近藤 秀樹^{*1}, 西野 和典^{*3}
九州工業大学 ^{*1} 学習教育センター ^{*2} 情報工学部 ^{*3} 教養教育院

Design of Browsing System for Lecture Archived Video Using Character Recognition API

Yoshimasa OHNISHI^{*1}, Yoshiki Shimoikura^{*2}, Shin'nosuke Yamaguchi^{*1},
Hideki Kondo^{*1}, Kazunori NISHINO^{*3}

Kyushu Institute of Technology ^{*1} Learning and Teaching Center,
^{*2} Graduate School of Computer Science and Systems Engineering, ^{*3} Institute of Liberal Arts

あらまし:我々は、市販の Web カメラとフリーウェアを用いた講義アーカイブシステムに関する研究開発を行っている。本稿では、LMS との連携を考慮した講義アーカイブ閲覧システムの検討について報告する。具体的には、Moodle やインターネットサービスで提供される API を用いることで、LMS 上における講義アーカイブ動画の閲覧環境の改善について報告を行う。

キーワード: Moodle, 文字認識, 講義アーカイブ, システム連携

1. はじめに

動画を用いた教材を提供するシステムとして、講義(授業)アーカイブシステムがあり、多くの研究開発や実践が行われている^(1~10)。講義アーカイブを受講生に提供する目的の多くは復習であると考えられるが、授業に参加できなかった学生への救済、教職員の FD の一環としての授業公開、翌年度以降の教育素材としての記録、などの利用も一般的である。

講義アーカイブは講義室の設備として、手軽に導入・実践できる半面、管理コストを削減するために、動画編集を行わないことが多い。しかし、講義の中でアーカイブに相応しくない部分も録画されることも多く、講義アーカイブを活用する上での課題の一つである。市販の講義アーカイブシステムの中には、スライドデータの変化を検知し、自動的にインデックスを付与するものもある。

我々は、録画側で対応する方法として、教員自身に録画制御行ってもらう方法⁽¹¹⁾といった試みを行ってきた。今回は、再生側において講義アーカイブの閲覧支援として、テキスト検出といったインターネットサービスを用いる方法について報告を行う。

2. 講義アーカイブシステムの概要

本学に導入されている講義アーカイブシステム(第2期:2015.3~)の構成を図1に示す。本システムは、ネットワークカメラとフリーウェア openRTSP⁽¹²⁾を用いたオリジナルシステム⁽¹³⁾であり、ストリーミングサーバに Adobe Media Server⁽¹⁴⁾または Wowza Streaming Engine⁽¹⁵⁾を用いる。

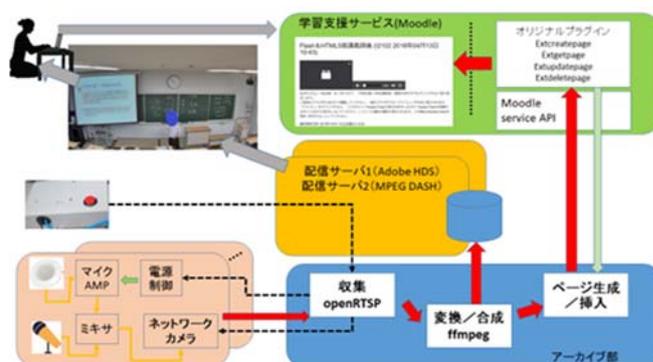


図 1 講義アーカイブシステムの概要

また、Learning Management System (LMS)との連携を重視し、該当コース内にあるスライドファイル(情報)との LMS 上での連携を検討してきた。具体的には、特定コース上に講義アーカイブの閲覧ページを自

動挿入^(16~18)することで録画終了後、数時間で自動公開できる機能を持つ。

表 1 2018 年度のアーカイブ例

科目名	日時 (時間)
科目 1 : アルゴリズム設計 S	毎週月曜日 4 時限, 金曜日 2 限 (4 月 6 日~6 月 1 日)
科目 2 : 離散アルゴリズム特論	毎週水曜日 2 時限, 金曜日 2 限 (6 月 8 日~7 月 27 日)
科目 3 : プログラミング	毎週月曜 3 限, 4 限, 5 限 (4 月 9 日~5 月 28 日) 毎週月曜 3 限, 4 限 (6 月 11 日~7 月 30 日)

2.1 アーカイブの事例

表 1 に 2018 年度前期のアーカイブ例を示す。科目 1 では、講義冒頭に「何人かの学生を指名し、黒板上で回答させる」といった演習時間が設定されているため、自作の制御装置⁽¹²⁾を講義室に設置した。教員が制御装置のボタンを操作することによって、講義アーカイブシステムを直接制御できるようになっている。科目 2 は大学院の科目であり、講義録画のみを行い、受講生への公開は行わなかった。そのため、制御装置は使用せず、時刻にあわせて録画の開始・終了をスケジュールした。科目 3 は PC を用いた演習付き講義であるが、前半は週 3 コマの進むため、学生の復習(救済)するために利用された。

2.2 再生方法

図 2 に示すように、「Flash&HTML5 講義録画」のようなアーカイブ名称のページが自動的に挿入される。利用者はこのページを開くことで、図 3 に示すような、再生ページが表示される。再生デバイスの自動判定は行っていないため、HTTP Dynamic Streaming⁽¹⁹⁾と HTTP Live Streaming⁽²⁰⁾の 2 種類のストリーミング再生コードを並べている。

再生プレーヤは標準的な、再生・停止・タイムライン・ボリューム・再生時間表示などが使用できる。しかし、先行研究や市販の講義アーカイブシステムのように、スライドの切り替わりを判定してインデックスを自動生成する機能はない。

2. 計算モデルと漸近記法 4/9

Flash & HTML5 版講義録画 (12102 2018年04月09日14:50)

計算モデルと漸近記法

計算モデルと漸近記法 (印刷用)



3. ソーティング I: アルゴリズムの正当性, 分割統治法 4/13

Flash & HTML5 版講義録画 (12102 2018年04月13日10:43)

ソーティング I: アルゴリズムの正当性, 分割統治法



図 2 講義アーカイブ閲覧ページの配置例 (コース上にリソースとして自動挿入)



図 3 講義アーカイブ閲覧ページの表示例

3. 閲覧システムの改善方針

講義アーカイブの閲覧に関する課題を解決するためには、講義ビデオの再生環境の改善が必要である。また、Flash Player を用いた閲覧ページは、インターネット環境の変化に合わせ MPEG Dynamic Adaptive Streaming over HTTP⁽²¹⁾を用いた方法に変更することとした。以下、主な改善方針を示す。

- LMS(Moodle)との親和性を維持
- 閲覧者(学生)に対する利便性の向上
- 閲覧履歴の記録と再利用

本学の講義アーカイブシステム⁽¹³⁾は、フリーウェアを活用としたシステムであるため、アーカイブされた講義動画ファイルを、サーバ上で解析する仕組みの組み込みが比較的容易である。そこで、閲覧者に対する利便性の向上を目標に、再生環境において講義アーカイブ動画に対して、文字検索可能なインデックスを提供することにした。インデックス情報を動画データから生成するために、まず動画データから静止面データを作成し、インターネットサービス Google Cloud Vision API⁽²²⁾を用いることにした。また、Google Cloud Speech-to-Text API⁽²³⁾の活用も検討することにした。

3.1 講義ビデオの再生環境

Moodle との親和性を考慮すると、講義ビデオの再生環境は、コース内のリソース「ページ」として構築することが望ましい。一方、ビデオの閲覧と講義資料の参照は同時に行われることも多いため、コース内のリソース「ファイル」にも簡単な操作でアクセスすることが望ましい。

例えば、図 4 に示すように Moodle のウィンドウを並べて、左側に講義ビデオとインデックスボタン（図では 4 つの例）を配置し、右側にコース内の資料を連動して表示するなど考えられる。この場合のインデックスの静的生成は、他の研究と同様に、スライドの切り替わりを検知する必要がある。

また、閲覧者が講義ビデオを検索する方法として、検索ワードを入力できる仕組みを構築し、検索結果からインデックスを動的に生成する方法（図 5）も考えられる。この場合は講義ビデオの画像および音声から文字情報を生成し、文字情報のマッチング率を使ってインデックスを生成する。なお、Moodle コース上での、検索ワードの入力をどのように実現するかを検討する必要がある。その他にも、複数の閲覧者が講義ビデオを閲覧した箇所（タイムライン）をインデックスとして生成することも可能であると思われる。



図 4 リンクベースのインデックス表示案

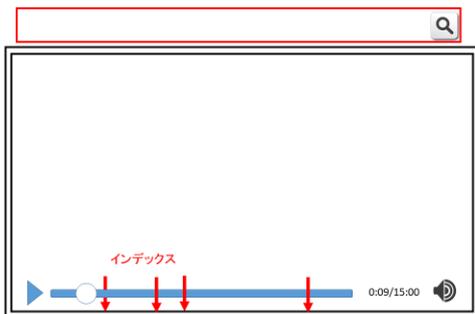


図 5 検索ボックスとインデックス表示案

3.2 静止画からのテキスト検出

Google Cloud Vision API は Google が提供している有償の画像分析サービスである。2018 年 8 月現在で、画像内のテキスト検出や画像内の顔検知など、合計 10 個の機能を利用できる。以下、PHP を用いた使い方を簡単に紹介する。

3.2.1 Google Cloud Vision API の利用

API を利用するために、以下の手順でプロジェクト作成と API の選択を行う。なお、必要に応じて API のアクセス制限を設定する。

1. 「Google Cloud Platform」でプロジェクト作成
2. 「API とサービス」を選択。API ライブラリから「Cloud Vision API」を選択し、API を有効化
3. 「API とサービス」を選択し、認証情報を作成

3.2.2 TEXT DETECTION 機能の呼び出し

Google Cloud Vision API では、リクエストとレスポンスの両方で JSON 形式を採用している。テキスト検出のドキュメント⁽²⁴⁾によると、プロトコルは図 6 に示す通りである。“image”には画像を指定し、“features”のタイプには “TEXT_DETECTION” を指定する。画像がローカルに格納されている場合、BASE64 エンコード文字列に変換した上で指定する。画像の指定方法には複数の方法を使用できる。なお、付録.1 に予備実験に使用した実験用 PHP コードを示す。



図 6 TEXT DETECTION プロトコル
<https://cloud.google.com/vision/docs/detecting-text?hl=ja#vision-text-detection-protocol> より引用

4. インデックスデータの生成

講義アーカイブの閲覧に必要なインデックスは、講義動画に記録された画像化された文字情報により生成されるものと、音声データにより生成されるものを組

み合わせる必要がある。本研究で対象としている講義動画には、スクリーンに投影されたスライドによる説明や板書による解説などを想定している。

4.1 スライドからのテキスト抽出

LMS との連携を前提としているため、講義動画内に登場するスライドデータは、LMS に登録されていると思われる。そこで、対象コース上のリソース情報を Moodle API⁽²⁵⁾ を用いて取得し、該当コース上のすべての PDF・PPT ファイルからキーワードとなり得るテキストデータの抽出をあらかじめ行う。

その上で、講義動画から静止画を生成し、Google Cloud Vision API を用いて、テキスト検出を行う。これら二つのデータを用いてインデックスの対象データを生成する。なお、講義動画からの静止画の生成間隔やその方法については、現在検討中である。

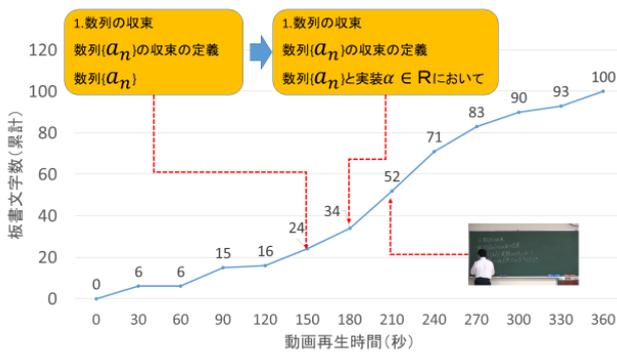


図 7 板書速度の調査結果

4.2 板書からのテキスト抽出

板書を中心とする講義動画の場合には、(a)板書をある程度終えて説明、(b)板書しながら説明、といった授業スタイルによって、静止画像の生成間隔が異なると思われる。そこで板書タイプの講義動画を実際に閲覧し板書速度を簡単に測定した。図 7 に結果を示す。

概ね 60 秒程度で、数行の板書文字に相当し、20～40 文字程度の情報が増えることがわかる。3～4 行の板書毎にテキスト検出をできればよいと思われる。そこで、静止画の動画分割の間隔は 90～120 秒程度でまずは検討することにした。なお、板書タイプの講義動画には、教授者自身が板書文字を隠すこともあるため、テキスト検出を複数回実施することで、文字情報の補間を行う必要があることがわかった。

5. 既存動画を用いた予備実験

予備実験では、ビデオカメラで板書タイプの講義を撮影した既存動画（表 2）を用いた。FFmpeg を使って、動画から静止画を生成し、幾つかの静止画を Google Cloud Vision API に渡し処理をした。以下の例では、フレームレートを小さく設定 (0.1) することで、10 秒間隔で静止画ファイルを抽出できる。

- `ffmpeg -i 動画ファイル -ss 0 -t 7200 -r 0.1 静止画ファイル`

図 8 および図 9 に実験結果の一部を示す。図 8 は実験した中で最も認識率がよかったものである。オリジナルの板書とはかなりの差があるが、2～3 文字程度の単語レベルで、なんとか認識できそうである。

テキスト検出の認識率の向上には、解像度をあげる必要があるが、一方で Google Cloud Vision API に送信可能なサイズ上限やレスポンス時間を考慮する必要もある。なお、図 9 は認識率が悪い結果である。

表 2 予備実験に用いた既存動画

項目	サイズ
長さ	86 分
ファイルサイズ	約 2GB
解像度	1,920 x 1,080
フレームレート	29.97 フレーム/秒
ビットレート	3,328kbps

6. まとめ

本稿では、Moodle やストリーミングサーバに登録されたファイル (PDF 資料, スライド資料, 動画など) より文字情報を抽出し、講義アーカイブシステムの再生機能の改善を検討した。講義動画ファイルからもある程度の文字情報を抽出し、スライド資料との連携に向けたキーワード検索による、インデックス生成が可能であることを確認した。なお、講義動画ファイルから音声データを抽出し Google Cloud Speech-to-Text API⁽²³⁾にてテキスト検出も行ってみた。こちらもインデックスの元になるデータとして使えそうである。

なお、キーワード検索文字で静止画の場所を検索できても、その説明は静止画の前後付近の音声データに合わせて、インデックスの位置を再調整する必要がある。

る。今後は、本検討結果を踏まえ、講義アーカイブ閲覧システムの実装を進めていく予定である。

参考文献

man-D $\forall n$ 数列(an)の収束の定義 $\forall n$ 敬列)(An)と実数(XER)h:n て表す. $\forall n$ 自然数 neNfBAR大きくした 25 (定理 11)数歹ソの,, $\forall n$ daX に収束す $\forall n$ 陳荷と, $\forall n$

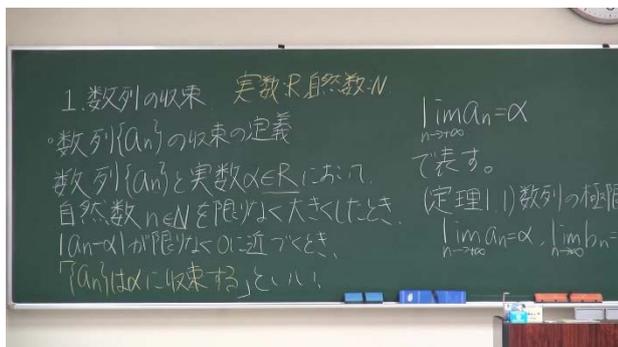


図 8 検出結果 1 (上) と静止画例 (下)
(1,920 x 1,080, 96dpi, 約 200KB)

1 2475%の有界(LAFR $\forall n$ イ 4 意の neN に対し, an 式が成 1 $\forall n$ (a ふ頂界でおと.. $\forall n$ (注意よると $\forall n$ は 1 み 35 $\forall n$ Cin $\forall n$ 田) $\forall n$ つ. $\forall n$ 田 $\forall n$ 11 齊チヨ-7 $\forall n$

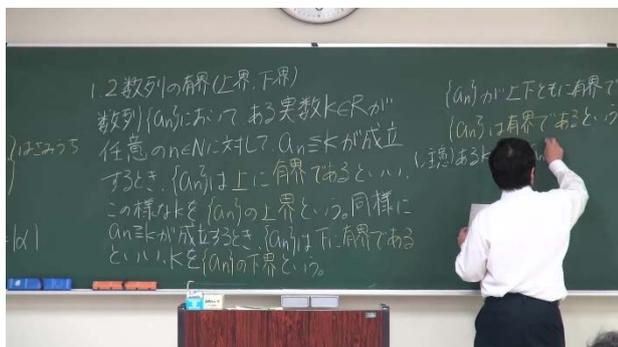


図 9 検出結果 2 (上) と静止画例 (下)
(1,920 x 1,080, 96dpi, 約 200KB)

謝辞

本稿では、豊坂裕樹先生の講義ビデオ⁽⁶⁾の一部を利用して頂いた。本研究の一部は、科学研究費補助金（基盤研究(C) JP16K01069, 基盤研究(C) JP16K01116 及び基盤研究(C) JP17K01136)の助成を受けた。

- (1) 市村 哲, 福井登志也, 井上亮文, 松下 温: “Web 学習用講義コンテンツを自動作成する板書講義収録システム”, 情報処理学会論文誌, Vol.47, No.10, pp.2938-2946 (2006)
- (2) 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”, 情報処理学会研究グループ報告, 第 9 回 CMS 研究発表会, pp.56-63 (2008)
- (3) Le Hieu Hanh, Thitiporn Lertrusdachakul, 渡部徹太郎, 横田治夫: “講義映像ビデオからダイジェスト自動作成のための重要シーン抽出手法の評価”, DEWS2008 E4-1
- (4) 山口真之介, 大西淑雅, 西野和典, 小林史典: “情報工学部における自動講義アーカイブシステムの試み”, 九州工業大学情報科学センター広報第 23 号(2011)
- (5) 川口雄一, 目時光紀, 小島洋一郎: “HTML5 によるストリーミング教材配信基盤の構築”, 大学 ICT 推進協議会 2012 年度年次大会予稿集, H4-2(2012)
- (6) Yamaguchi, S., Taenaka, Y., Toyosaka, Y., Yoshimasa, O., Nishino, K., & Fukumura, Y.: “The Distribution Method of the High Resolution Video for a Blackboard Based Lecture”, International Journal of Technology and Educational Marketing (IJTEM), 4(2), pp.28-42, doi:10.4018/ijtem.2014070103 (2014)
- (7) 吉良元, 長谷川忍: “大学院生の補完的学習環境としての講義アーカイブシステムの運用と分析”, 教育システム情報学会誌 Vol.32 No.1, pp.98-110 (2015)
- (8) 早稲田大学, Waseda Course Channel, <http://course-channel.waseda.jp/> (2018.8 参照)
- (9) 奈良先端科学技術大学院大学総合情報基盤センター学術情報サービスグループ: “講義アーカイブシステム 電子図書館システムの一つの可能性として, 大学図書館における先進的な取り組みの実践例”, No.16, 2 ページ, http://www.mext.go.jp/component/a_menu/science/detail/icsFiles/afieldfile/2015/08/31/1361413_16.pdf
- (10) 長谷川 忍, 小林弘彬: “講義アーカイブシステムにおけるホワイトボード領域の鮮明化”, 先進的学習科学と工学研究会, 人工知能学会, Vol.78, pp.30-33(2016)
- (11) 大西淑雅, 荒川等, 山口真之介, 近藤秀樹, 西野和典: “小型ハードウェア機器を用いた講義活動の記録システムの検討”, 第 41 回 JSiSE 全国大会, B2-4(2 ページ) (2016)

- (12) <http://live555.com/openRTSP/> (2018.8 参照)
- (13) 大西淑雅,山口真之介,西野和典:“フリーウェアを用いた講義アーカイブシステムの実現”, 大学 ICT 推進協議会 2015 年次大会, CDRROM[3D3-3], 8 ページ(2015)
- (14) <https://www.adobe.com/jp/products/adobe-media-server-family.html>
- (15) <https://www.wowza.com/products/streaming-engine>
- (16) 大西淑雅, 福丸浩史, 林朗弘, 本田宏:“Moodle service API を用いたページ内容の自動挿入”, UeLA&TIES 合同フォーラム 2016 ポスタ発表, pp. 32-35 (2017)
- (17) 大西淑雅, 山口真之介, 西野和典:“Moodle におけるリソース情報の自動追加の試み”, 第 42 回 JSiSE 全国大会インタラクティブ発表, pp. 315-316 (2017)
- (18) “ローカルプラグインの設計・開発”, <http://www.ltc.kyutech.ac.jp/center/research/ohnishi/> (2018.8 参照)
- (19) <https://www.adobe.com/jp/products/hds-dynamic-streaming/faq.html>(2018.8 参照)
- (20) <https://developer.apple.com/streaming/> (2018.8 参照)
- (21) <https://www.ite.or.jp/contents/keywords/1701keyword.pdf> (2018.8 参照)
- (22) <https://cloud.google.com/vision/> (2018.8 参照)
- (23) <https://cloud.google.com/speech-to-text/?hl=ja> (2018.8 参照)
- (24) <https://cloud.google.com/vision/docs/detecting-text?hl=ja> (2018.8 参照).
- (25) The Moodle Project : “Web service API functions”, https://docs.moodle.org/dev/Web_service_API_functions (2018.8 参照)
- (26) Yuta Arai : “Cloud Vision API の使い方まとめ” <https://syncer.jp/cloud-vision-api> (2018.8 参照)

付録.1 Google Cloud Vision API の TEXT DETECTION 機能の呼び出しテストに用いた PHP コードを以下に示す. 参考文献⁽²⁶⁾ からコードを一部引用し作成した.

```
<?php
$api_key = "xxxxxxxxx";
$image_path = "ファイル名";

$json = json_encode(
array(
    "requests" => array(// [
```

```
array(
    "image" => array(
        "content" =>
base64_encode( file_get_contents( $image_path ))
    ),
    "features" => array(// [
        array("type" => "TEXT_DETECTION"),
        array("maxResults" => 1)
    )// features[]
)
) // request[]
);
// リクエストを実行
$curl = curl_init();
curl_setopt( $curl, CURLOPT_URL,
"https://vision.googleapis.com/v1/images:annotate?key=" .
$api_key );
curl_setopt( $curl, CURLOPT_HEADER, true );
curl_setopt( $curl, CURLOPT_CUSTOMREQUEST,
"POST" );
curl_setopt( $curl, CURLOPT_HTTPHEADER,
array("Content-Type: application/json" ) );
curl_setopt( $curl, CURLOPT_SSL_VERIFYPEER,
false );
curl_setopt( $curl, CURLOPT_RETURNTRANSFER,
true );
if( isset($referer) && !empty($referer) ) curl_setopt( $curl,
CURLOPT_REFERER, $referer );
curl_setopt( $curl, CURLOPT_TIMEOUT, 15 );
curl_setopt( $curl, CURLOPT_POSTFIELDS, $json );
$res1 = curl_exec( $curl );
$res2 = curl_getinfo( $curl );
curl_close( $curl );
// 取得したデータ
$json = substr( $res1, $res2["header_size"] );
// 出力
print_r($json);
```