

## 簡易利用可能な授業の自動アーカイビング/配信システムの開発

小林 裕之<sup>†</sup>      関 秀行<sup>†</sup>      千代倉 弘明<sup>†</sup>

ネットワーク環境の充実に伴い、ビデオ教材のストリーミング配信に対する期待と需要が高まっている。筆者らは、教員がビデオ教材のための特別な準備をせずに日常の授業をそのまま収録し、ストリーミング配信を行いたいという需要に注目した。この場合、授業の収録から配信の準備までをいかに手軽に行えるかが重要になる。本稿ではそのために開発した授業の自動アーカイブ/配信システムについて報告する。本システムは、mpeg-1 などのソースビデオファイルを web ページからアップロードするだけで、ストリーミング配信の準備をすべて自動で行うことを可能にする。

### Development of Easily Operatable Automatic Lecture Archiving and Streaming System

KOBAYASHI, HIROYUKI,<sup>†</sup> SEKI, HIDEYUKI<sup>†</sup>  
and CHIYOKURA, HIROAKI<sup>†</sup>

The more computer network grows, the larger demand for movie streamings becomes. There are demands for authoring and streaming video contents of ordinary lectures in classroom, on which the authors focused. In this report, a new movie archiving and streaming system which is easily operatable by everybody is introduced. One who wants to create a new streaming content only needs to upload a source movie file. Then, automatically streaming is go.

#### 1. はじめに

教育の場での IT における動画の利用は、情報が豊富であること、容易に臨場感を得られること、さまざまな教材に応用でき汎用性が高いことなど、他の方法にはない利点が多く、今日ますます期待が高まっている。

その反面、ビデオ教材はコンテンツとしての完成度を高めようとするとき非常に多くの手間がかかるため、教員一人一人が手軽に作成するのは容易でない場合が多い。現在のビデオ教材の多くは、事前に綿密に計画され、専門スタッフが十分な時間と労力をかけて作成する『作り込み型』のコンテンツであると言える。

しかしながら、現実には常に作り込み型レベルのクオリティが必須であるとは限らない。ビデオコンテンツの質的充実でなく、量的/時間的な充実が優先される場合も多く、その場合、例えば通常の授業をビデオで記録し、それをアーカイブするだけでも十分役に立つ。

このような方向の試みとして、現在、既に複数の製

品が市場に出回っている。しかしながら、その多くが特定のプレゼンテーションソフトウェアに特化した製品である。この種の製品は、特定のソフトウェア (e.g. MS PowerPoint) で完結したプレゼンテーションをアーカイブし配信するのは簡単に行えるが、学校で行われる通常の授業をアーカイブするという目的には向かない。操作がどれほど簡単であっても、教員はプレゼンテーションソフトで完結する授業を事前に準備し、アーカイブ専用にと時間をとって撮影を行う『作り込み型』と言える。

このような背景から、我々は誰もが簡単に授業をアーカイブ/配信できるようなシステムの開発を行った。基本目標は

- できるだけ単純なインターフェイスであること
- 講義室で行う通常の授業をそのまま手軽にアーカイブできること
- システムとしてオープンであり、拡張や他のシステムとの連係が容易であること

である。

#### 2. ストリーミングとは

ストリーミングとは、ネットワークを介してムービーを送り、実時間で表示する際に起こるものである [3]。

<sup>†</sup> 慶應義塾大学  
Keio University

ストリーミングが通常の動画ファイルダウンロードと大きく異なるのは以下の2点である。

- ファイルを転送しつつ、実時間再生が可能  
多くの場合、ストリーミング技術を利用する最大の理由はこれであろう。転送と再生が同時に行われるので、視聴者は時間をかけて大きなデータの転送が完全に終わるのを待つことなく、すぐに再生を始めることができる。
- 時間軸上の移動が可能  
視聴者は再生中の任意の時刻に、コンテンツの任意の位置へスキップができる。

これらのうち、第1項はFast Start [3] などと呼ばれる非ストリーミング技術でも実現可能である。この方式は、視聴者側が適当な大きさのバッファを持つことで通常の動画ファイルを転送しつつ再生する方式である。ただし、原則としてオリジナルデータのビットレートを下回る回線においてはリアルタイム性が実現できないことや、あくまでも視聴者側の技術であるため、配信における精密な管理が難しいことなどの問題点がある。そのような制約なしに上記の特徴を実現するのがストリーミングであるが、そのためには

- ストリーミング配信に対応した動画ファイル
- ストリーミングサーバ
- ストリーミング再生クライアント

の3つが揃う必要がある。特に第1点の、コンテンツ自体がストリーミングに対応したもの (mpeg-4[4] や ヒントトラックを持つ QuickTime ムービーなど) でなければならない点が重要である。つまり、現在手軽に広く用いられている mpeg-1 などのストリーミング非対応フォーマットの場合は、ふさわしい形式に変換する必要がある。本報告で紹介するシステムでは QuickTime のヒントムービーを採用した。ヒントムービーとはオリジナルのムービーに『ヒントトラック』というストリーミングのための情報を付加したムービーである (図1)。

### 3. 自動マルチメディアアーカイブ/配信システム

教員誰もが手軽にストリーミング教材を作成できるようにとの要求から、筆者らは自動マルチメディアアーカイブ/配信システム“ViShNu” (VIdео SHaring NUcleus) を開発している。本節では、そのシステムを具体的に紹介する。

#### 3.1 特徴

本システムは慶應義塾大学湘南藤沢キャンパス (SFC) のエンドユーザ (授業を収録する教員や視聴

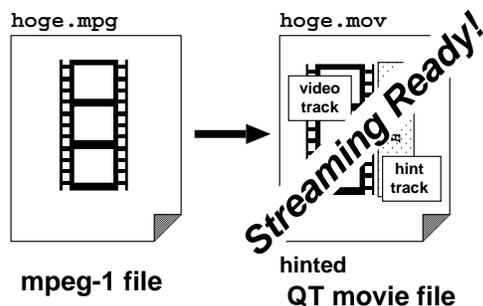


図1 Adding hint track to original file

者である学生) とシステム管理者の双方が楽に利用できることを目指して設計した。主な特徴は以下のとおり。

- 動画ファイルのみをサービス対象とする  
本システムは動画 (およびそれに付随するタイトルや作者などの最小限のメタデータ) のみを扱うため、使い方の自由度が高い。言い替えると、本システムはプレゼンテーションソフトのファイルなどの付録資料は敢えて扱わず、より上位のシステムに任せることで汎用性を高めるようにした。
- web によるインターフェイスで操作可能である  
OS に依存しないので、利用者の日常の計算機環境から利用できる。
- 全自動システムである  
利用者は動画ファイルをサーバにアップロードするだけで、ストリーミング形式へのファイル変換、サムネイル生成、閲覧ページ生成などの一連の流れが完全に自動的に行える。
- ネットワーク上で登録/管理が可能である  
mpeg などの動画ファイルを用意すれば、それを HTTP でアップロードすることでそのままストリーミング配信の準備ができる。つまり、サーバルームなどへ直接足を運ぶことなく、収録した講義室、教員の研究個室、自宅など好きな場所からネットワーク越しにストリーミング教材の登録/管理が行える。
- UNIX 系 OS 上で構築されたシステムである  
SFC のキャンパスネットワークは UNIX 系 OS を基本としているので、導入や保守が楽に行える。加えてセキュリティ面での安心感もある。

---

慶應義塾大学 SFC ではこの目的のために、Web Learning System (WLS) というアプリケーションを開発し、現在運用している。ViShNu は WLS と連携運用が可能ないように開発している。

- 大部分がフリーソフトウェアで作成されている後述する動画フォーマット変換のライブラリを除いて、全てオープンソースのフリーあるいはそれに近いライセンスのソフトウェアを利用しているので導入およびランニングコストが極めて安価である。ストリーミング配信のユーザ数にもライセンス的な制限はない。
- 機能がモジュール化されている  
個別の機能毎に独立したプログラムが存在し、それらが TCP/IP などによって通信し、全体として機能することで、システムの拡張性が高くなっている。

### 3.2 アーキテクチャ

ViSHNu は図 2 に示すように、5 つのサーバから構成される。

**管理インターフェイスサーバ** ファイルの登録と管理のユーザインターフェイスを提供する。インターフェイスは web で http サーバには Apache を利用する。動的な処理 (html 生成や DB サーバとの通信) には Perl プログラムを CGI として用いる。

**配信インターフェイスサーバ** 視聴者 (学生など) がブラウザなどを使ってアクセスするサーバで、ストリーミング配信を行うまでのユーザインターフェイス (検索、サムネイル表示など) を提供する。Apache および Perl を使用。

**配信ストリーミングサーバ** RTSP によって動画を視聴者にストリーミング配信するためのサーバ。Apple 社の QuickTime Streaming Server (もしくはそのオープンソース版の Darwin Streaming Server) を使用。

**データベース (DB) サーバ** 動画に付随するメタデータやファイルの保存パスなどを管理する。DBMS には PostgreSQL を使用。

**ファイルサーバ** 動画ファイルとインデクス静止画ファイルなどを保管するファイルサーバ。ファイル共有は NFS で行う。

**フォーマット変換サーバ** 図 1 で示したように、通常の形式の動画ファイルをストリーミング形式に変換するサーバ。c++ で記述し、デーモンプロセスとして動作する。QuickTime のライブラリで実現している。従ってこのサーバは QuickTime ライブラリが提供されていて NFS マウントが可能な OS である必要があり、現状では MacOS X

が唯一の現実的な選択肢である。

これらのサーバは、全て 1 台の計算機に実装しても良いし、別々の計算機に割り当てても良く、規模に合わせた柔軟なシステム構成が可能である。(我々が現在実験中のシステムは Linux(x86) 機 と Xserve の 2 台で構成している。) プラットフォームについては、現在の UNIX 系 OS であれば事実上制約はないが、フォーマット変換サーバだけは MacOS X である必要がある。

### 3.3 原理

本システムによって、ソースとなる動画ファイルがストリーミング可能な状態に準備されるまでを流れて追って説明する。図 2 中の番号が以下の説明の項目番号に対応する。

- (1) はじめに利用者は動画ファイルを用意する。対応する動画フォーマットは QuickTime によって読み込み、編集が可能なものである。具体的には mpeg-1, AVI などである。配信はこのファイル (にヒントトラックを付加したもの) がそのまま用いられるので、編集やビットレートの調整はこの時点で完了しているものとする。
- (2) 次に、そのファイルにアクセス可能な計算機で管理インターフェイスサーバの web ページを開き、必要に応じてメタデータ (講義番号、タイトル、著者名、編集キー、タイトル静止画、etc.) を入力し、データとファイルを HTTP の POST メソッドでアップロードする。(図 3)

図 3 Registration page

- (3) 送信データは管理インターフェイスサーバの Perl プログラムが処理する。Perl プログラム



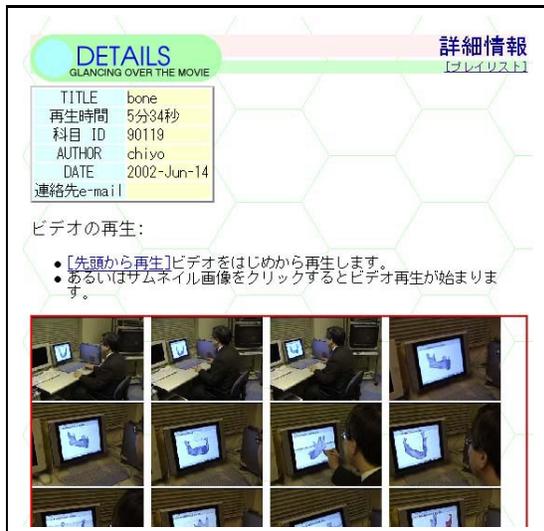


図 6 Details of the movie page

オトラック、オーディオトラックなどに加えヒントトラックが必要である(図1)。この種の変換を実装するにあたり、商用のライブラリを用いる方法や、フリーソフトを用いる方法を検討したが、前者はコストが非常に高くなってしまいう問題があり、後者は適当なものが見つからなかった。そこで、MacOS や Windows で利用可能な QuickTime のライブラリを利用することにした。QuickTime のライブラリは c (あるいは c++) 言語や Java 言語から利用可能である [1, 2]。ViShNu では c++ を使用した。

また、フォーマット変換サーバは付属機能として、インデクス生成機能を持つ。これは、動画像全体から任意枚数 (DB に登録された枚数) のサムネイル静止画 (jpeg) を生成する機能である。サムネイルの抽出アルゴリズムは最も単純な、時間を等分割するものを採用した。この点に関しては、より複雑なアルゴリズムも数多く研究/提案されている [5-8]。例えば XEROX 社 FX Palo Alto 研究所の MBase system では、フレーム間の変化を統計的に処理する方法を基本とし、そこに音声やメタデータなども加えて総合的にキーフレームを抽出する手法を提案して、更にその結果に基づいたユーザインターフェイスの提案 [7, 8] もあり、興味深い。しかし我々は、日常の授業風景からそのままストリーミングコンテンツを作る、という本システムの目的に鑑みて、単純な時間等分割方式が最適と判断した。なぜなら典型的な日常の授業の画像というものは、単調な画面であり、変化点や特徴点は視聴者にとってそれほど意味がなく、むしろ一定間隔のシーンの方が検索に便利であると考えられるためである。

以上が ViShNu の全容である。次に、このシステムをより高度に運用するための、講義室環境の例を示す。

#### 4. 講義室環境の支援による、より高度な教材への対応

教員が ViShNu を用いて授業をアーカイブするために必要な機材は、mpeg-1 や AVI など授業を撮影できる装置と、web ページにアクセスする環境のみであるため、極端な例を挙げると CCD カメラと無線 LAN を装備したラップトップ PC 1 台で済む。このような軽快な運用とは別に、より凝った教材も手軽にその場で作ることができるように、我々は講義室側の設備の開発も行っている。ここで、講義室側の目標は

- PC 画面 (プレゼンテーション) を録画できること
- 授業終了と同時に教材が完成すること (撮影後の編集を行わずに済むこと)
- さまざまな授業形態に柔軟に対応できること

である。我々は、この要求を満たすため図7に示すような講義室環境を開発している。この環境の特徴は、複数のソース (教員の姿、PC 画面、etc.) を全て NTSC にし、セレクトで切り替えるという割り切った単純さにある。

PC 画面も NTSC にしてしまうため、PC 本来の解像度は得られないが、通常の授業で多人数を相手に用いるような資料であれば、多くのものは実用的な品質を得られるであろう。また、スキャンコンバータに、クロマキー合成機能のあるものを採用したため、教員が事前に準備さえしておけば、教員の姿と教材を同一画面に納める図8のようなコンテンツもリアルタイムで作成可能である。

#### 5. まとめと今後の課題

我々は、日常の授業をそのまま記録/蓄積し、配信するシステムの開発を行っている。大学の教員が特に準備せずに気軽に利用可能なものを作るという最大の目標は、現在の ViShNu によってほぼ達成できたと思われる。さらに、このような単純な動画像配信システムでも、教室環境を適切に作ることで、幅広い教材に対応できる可能性も示した。

現在の開発の中心課題は、フォーマット変換の高機能化である。現在の ViShNu のフォーマット変換は図1のように『ヒントトラックを付加するだけ』である。そのため、教員がアップロードしたソースムービーのビットレートがそのまま配信のビットレートになって

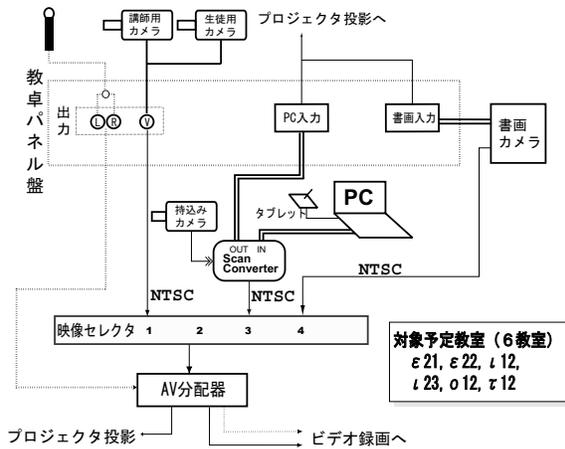


図 7 Proposed lecture room schematic

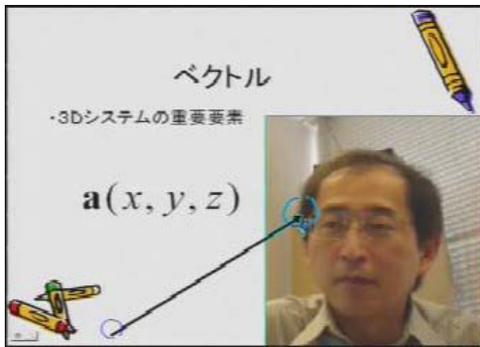


図 8 Sample content employing video overlaying

しまい、いわゆるナローバンドへの対応が不十分である。そこで、高ビットレートのソースムービーから、様々な環境の視聴者に対応できるように、複数のビットレートのストリーミングムービーを自動生成する機能を計画している (図 9)。

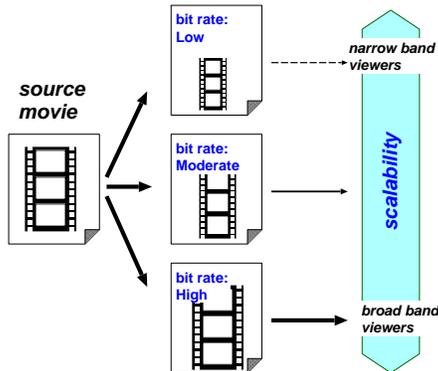


図 9 Scalable streaming feature

また、冒頭で触れたとおり、慶應義塾大学 SFC では教員と学生に対し WLS というサービスを運用中である。この WLS との連携も次の課題である。WLS との連携が適切に行えれば、学生が WLS で授業を検索し、必要な教材をダウンロードし、該当するストリーミングコンテンツを視聴し、レポートを提出するという一連の流れがシームレスに行えるようになる。

本システムは、大学における現場の要求を元に開発している。今後はこのシステムで実際の授業をできるだけ多く蓄積/配信する運用実験を繰り返し、教員や学生からのフィードバックを得て、さらに改善を図りたい。

### 参考文献

- 1) George Towner, "Discovering QuickTime: An Introduction for Windows and Macintosh Programmers," Academic Press/Morgan Kaufmann, ISBN 0-12-059640-7 (1999)
- 2) Tom Maremaa and William Stewart, "QuickTime for Java: A Developer Reference," Academic Press/Morgan Kaufmann, ISBN 0-12-305440-0 (1999)
- 3) Apple Computer, Inc., "QuickTime for the Web," Academic Press/Morgan Kaufmann, ISBN 1-55860-780-3 (2002)
- 4) 大久保 肇, 川島 正久 監修, MCR 編, 「H.323/MPEG-4 教科書」, IE インスティテュート, ISBN 4-901280-03-1 (2001)
- 5) M. Bertini, A. Del Bimbo, P. Pala, "Content-based indexing and retrieval of TV news," *Pattern Recognition Letters*, vol. 22, pp. 503-516 (2001)
- 6) Anastasios D. Doulamis, Nikolaos D. Doulamis and Stefanos D. Kollias, "A fuzzy video content representation for video summarization and content-based retrieval," *Signal Processing*, vol.80, pp. 1049-1067 (2000)
- 7) Andreas Girgensohn, John Boreczky, Lynn Wilcox, and Jonathan Foote, "Facilitating Video Access by Visualizing Automatic Analysis," *Human-Computer Interaction INTERACT '99*, IOS Press, pp. 205-212 (1999)
- 8) J. Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi "An Interactive Comic Book Presentation for Exploring Video," *CHI 2000 Conference Proceedings*, ACM Press, pp. 185-192 (2000)