

## 2. COM コーナー

### 誰でもできるマルチメディア授業、その2 —SMILによる動画・プレゼンテーションの同期— A Simple Way to Construct Multimedia Lectures, Part 2. Synchronization between Movie and Presentation by SMIL

文学部 非常勤講師 奥山徹  
(朝日大学経営学部)

#### 概要

前回[1]は、マルチメディア授業を展開するための手軽なツールを2つ紹介した。それらのツールを使うことで、PowerPointのプレゼンテーションやWWWのホームページによる資料公開とは違った側面でのマルチメディア授業を展開できる可能性を示唆した。しかしながら、これらは単独で使ってもそれほど意味はなく、たとえば授業を撮影した動画と同期をとりながら利用することで、「授業をよりよく再現できる」、「自習用のオンデマンド教材が作成できる」といったより有機的な利用が可能となる。そこで、今回は、前回利用したツールの中から画面キャプチャリングソフトウェアを使ったPowerPointのスライドデータを動画と同期させて組み合わせる仕組みについて考える。

#### 1. はじめに

現在、多くの大学で社会人や一般家庭の人々を受け入れるプロジェクトが社会人の再教育あるいは生涯教育の一環として実現されつつある。しかしながら、特に勤めを持つ人々にとって、時間と場所が限定される授業形態では、なかなか受講の機会を得られないのが現実である。

そこで、多くの大学が様々な工夫を凝らし、たとえば遠隔教育を取り入れるなどして、時間と場所の制約を幾分でも緩和することを試みている。しかし、そのような努力にもかかわらず、現在のシステムでは受講することが難しいと考えている人は少なからず存在する。

そのような人々の需要を満たす一つの方法として、非同期型の授業形態を模索する動きがある。これは、必ずしも同期的な対

面あるいは遠隔教育にこだわることなく、授業内容をオンデマンドで取り寄せ、それについてたとえば電子メールのような非同期媒体をなどの種々の方法で議論することを可能とした学習方法である。

残念ながら、このような学習方法は現在では補助的な手段として扱われており、たとえば同期的な教育がある程度の比率で行われない限り大学の単位として認定するのは現在の制度上は難しい。しかしながら、今後は増大する社会人や一般人の学習需要を満たすためには、少なからず非同期的な授業を採用していく必要がある。

そこで、問題となるのが、このような非同期的なオンデマンド教材をどのようにして作成するかということである。著者は1996年度から始まった豊橋技術科学大学のMUPS (Multimedia University Pilot

Study) [2]において、授業をスタジオ収録する場合の費用と労力が膨大なものであることを経験した。

そこで、通常行われている授業を簡単にオンデマンド教材とすること考え、これまで種々の試みを行ってきた。前回と今回述べている内容もその一部である。実際にビデオに落とした授業内容と資料として使用した種々の材料を人海戦術で組み合わせることで、多くの場合ビデオ教材を作成することは可能である。しかしながら、人が介在するプロセスは費用対効果はあまりよくなく、さらにアルバイトなどを使っている場合は、成果物の品質を管理することが重要であり、そのために余分な労力をかけなければならない。

ビデオ教材自動生成の目的はそのような現実的な問題点を検討することで、いかに人が介在するプロセスを少なくし、しかも間違わないように規格化することである。それにより、短時間で多数のオンデマンドビデオ教材を作成することが可能であると考えている。

そこで、本報告はそのためのひとつのアイデアを示し、現実に小さな試作システムを稼働させた結果を示すものである。

## 2. 基本的概念

具体的な話に入る前に、本報告で用いる基本的概念について確認する。

### 【対象】

本報告において対象とするものは、あくまでも通常行われている授業の内容であり、特別にアレンジされたものではない。ただし、たとえば PowerPoint のプレゼンテー

ション資料を使うなど、通常の授業よりマルチメディア化されていることを前提としている。

### 【授業内容】

授業内容も通常授業を想定しており、特にオンデマンド教材用のアレンジされたものではない。また、ビデオはあくまでも音声による講述内容を中心としたものであり、映像はオンデマンド教材としての授業内容を補助するものであると位置付ける。

### 【オンデマンド教材】

非同期での遠隔授業の中心をなす教材であり、現実に講述された授業の内容を忠実に再現できるものであり、次の各要素を含む。

- (1) 講師により講述された内容に関するデジタル化された動画。この中で重要な要素は音声は明瞭に聞こえることであり、画像は補助的なものとする。
- (2) 授業に使用された PowerPoint のスライドなどのマルチメディア化された教材。
- (3) (1) の動画と (2) の教材間の時間的な同期情報。
- (4) その他、授業の各種の静的な情報を集めたデータベース。

## 3. 動画とプレゼンテーション教材の時間同期

さて、前節の基本概念の中で、オンデマンド教材が非同期の遠隔授業を構成する中心的な存在と考えている。この中で動画とプレゼンテーションのための教材との時間



## synchronization

図1. 動画とプレゼンテーション教材の時間同期

的な同期情報を要求している。ここでは、はじめに時間的な同期の概念を解説する。

図1は時間同期の概念を示している。すなわち、時間同期とは動画上で講述されている内容に応じたプレゼンテーション教材が、明確にしめされることである。ここでの教材は、同画面上で表示されるだけではだめで、再利用可能な状態で学生に提示されなければならない。なぜなら、このようなデジタル化されたマルチメディア教材は、そのまま電子的な学生のノートブックシステムに挿入できるべきであると考えている。なお、ここで、電子的なノートブックシステムとは難しく考える必要はなく、たとえば MS-WORD のようなワードプロセッサでも十分な機能を果たすことが考えられる。

問題は、このような動画とプレゼンテーション教材の時間同期データをどのように収集するかということである。人手による収集はもっとも効率よいと思われがちであるが、作業者の錯誤や怠慢による人為的な

ミスを完全に排除することは難しい。そのため、何らかの手段で同期データを自動的に収集する必要がある。

#### 4. 自動化への基本的なアイデア

同期データの自動化への基本的なアイデアは前回報告した画面キャプチャリングソフトウェアを活用することである。すなわち、画面キャプチャリングソフトウェアはプレゼンテーションを表示しているパソコンの画面をすべて、ある一定時間間隔でディスク上に保存してくれる。そこで、図2に示したように、ネットワーク上で NTP (Network Time Protocol) [3]により時間同期が取れたコンピュータを使い、それらの上で動画とプレゼンテーション教材の画面キャプチャリングを行う。このようにして動画の進行とキャプチャされた画面の時間同期を取ることができる。

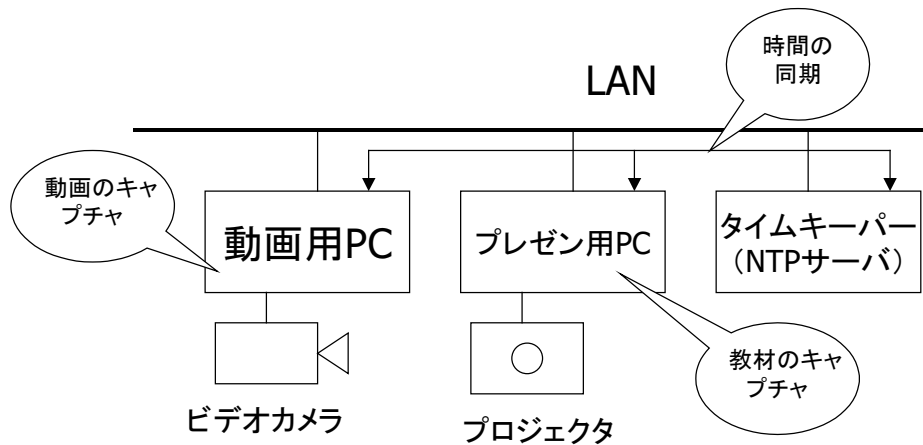


図2. 全体のシステム構成

ところで、この段階では、ただ単にキャプチャされたデータをそのまま組み合わせることで時間同期をとることができるだけである。すなわち、ビデオカメラからのキャプチャされた動画と同時に教材として提示されている画面表示をすべてキャプチャしたデータの両者をそのままつなぎ合わせているだけに過ぎない。特に、キャプチャされた画面データは膨大な量となるために、このままでは保存するために大量のディスク容量を必要とする。たとえば単純に試算すると次のようになる。

<ディスク容量の試算>

- ・ 90分授業を考慮
- ・ キャプチャ間隔は1秒とする
  - 1) 必要画像数  $90 \times 60 = 5,400$  枚
- ・ 1枚の画像の平均的容量を 50k バイトとする
  - 2)  $5,400 \times 50,000 = 270,000,000$

すなわち、1つの授業の教材データを保

存する容量として約 270M バイトが必要であり、たとえば平均的な PowerPoint のファイルの平均容量が 270k バイトと仮定すると、1,000 倍の容量を必要とする。そこで、これらの画像から、変化した画像とその変化した時刻を抽出することでデータを圧縮する。つまり、PowerPoint のようなスライド型のデータは図3に示すような時系列データとして記録される。ここから、以下に示すアルゴリズムにより、自動的に変化した画像と時刻を抽出する。

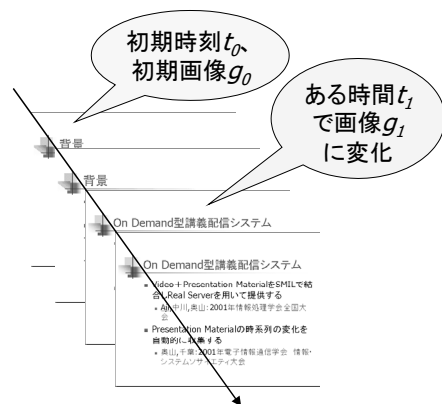


図3. キャプチャ画像の時系列変化

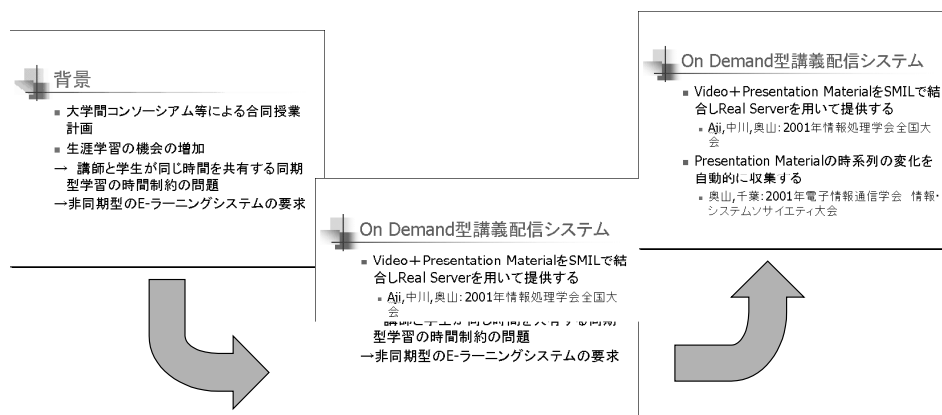


図4. 変化中のスライドのキャプチャ

#### <同期データ取得アルゴリズム>

- (1) 表示画面を一定時間（最短1秒）ごとに収集
- (2) 画面データを適当な領域に分割し、変化量を測定
- (3) 変化のあった画像の集合を  $G$  とし、そのときの計測時間の集合を  $T$  とする
- (4)  $G$  とそれと同期した時間集合  $T$  への写像は全単射となるので、 $G$  の  $i$  番目の要素  $g_i$  の像である  $t_j$  の時刻を SMIL にて組み込む
- (5) これにより、動画とプレゼンテーション資料の同期が取れる

以上のアルゴリズムにより、時系列で変化する画像集合  $G$  とそれと同期する時刻集合  $T$  がえられる。 $G$  の要素と  $T$  の要素は1対1に対応する（全単射な）ので、 $T$  の各要素の時刻に  $G$  の対応する要素の画像をプレゼンテーション資料として、図1に示したように動画と同期して表示すればよい。SMIL(Synchronized Multimedia Integration Language)[4]はそのための言語系として設

計されたものであり、独立した言語というよりはXML(Extensible Markup Language)[5]の一部として定義されている。

#### 5. 画像集合 $G$ の決定方法

ところで、画像集合  $G$  をどのように決定するかは重大な問題を含む。一つは、画面キャプチャリングにおける問題であり、他のものはどこまで詳細に画面の変化を捉えるかの問題である。これらはいずれも、<同期データ取得アルゴリズム>における(2)のステップに変化量の測定に間する問題点である。すなわち、変化量の測定は、次のようにしておこなっている。

#### <変化量計測アルゴリズム>

- (1) 初期画像（24ビットBMP等）を8ビットのBMP画像に変換する
- (2) 画像を決められた  $n$  個の区分に分割する
- (3) 分割された画像のビットマップ上のRGBで示される特性ベクトルの線形結合を取り、それをその区分の特性値とする

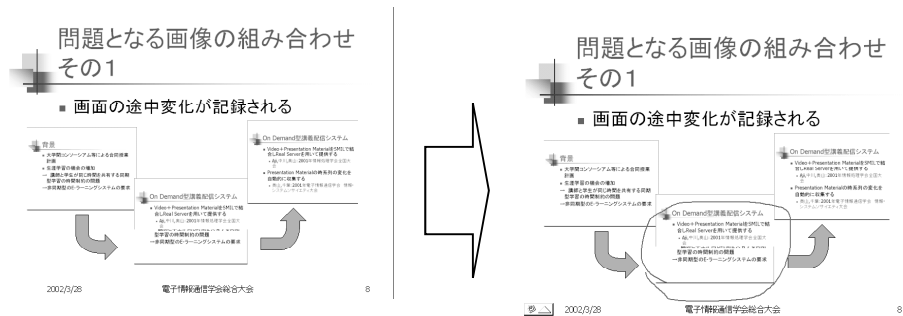


図5. 書き込み状態のスライドのキャプチャ

- (4)  $i$  番目の画像  $g_i$  とその次の画像  $g_{i+1}$  特性値に変化のあった区分を計数する
- (5) 計数した値がある一定比率  $m$  以上であれば変化したと認識する

前者については、1秒間隔のキャプチャリングにおいて、しばしば図4に示すような事態が起きる。すなわち、2つの画像の変化している状態をキャプチャしてしまう。VRAMへの画像データの書き込みがDMA(Direct Memory Access)によって瞬時に切り替わるなら1秒という間隔ではこのような事態は起きない。しかし、PowerPointの画像書き換えモードではメモリー上に蓄えられた画像情報をVRAM上に瞬時に送るわけではないので、図に示したような問題がおきる。つまり、2つのスライドが切り替わりつつある中間状態のスライドというものが現れる。これを上のアルゴリズムでは画像として認識してしまうのである。このような中間状態のスライドを、以下では「不要スライド」と呼ぶ。

一方後者の問題は、たとえばPowerPointのスライドのプレーン上にペンでハイライトする操作を行う、あるいは、あらたな情報を書き加えるような操作をどこまで細か

く記録するかを決定することである。たとえば、図5はスライドの変化中の状態をペンでハイライトした画像を示している。この程度の書き込みであれば、書き込み前のスライドと書き込み後のスライドの2枚がキャプチャされることが多いが、場合によっては書き込み中のスライドが数枚にわたりキャプチャされることがある。このような中間状態の画像をどのように取り扱うのか決定する必要がある。また、これらの画像を自動的に排除する方法を同時に考える必要がある。

## 6. スライドのロスト問題

次に、現在のアルゴリズムではスライドがロストするという他の問題を抱えている。たとえば、多くのスライドを用意しすぎたために、あるスライドの表示時間がキャプチャ間隔以下となり、そのスライドがキャプチャされない事態が生じる。多くの場合、このようなスライドは重要性が低いと考えられるから、ロストしても問題ないと考えられることもできる。

しかしながら、たとえば一連の時系列変化を示しているスライドを多数見せる場合など、中間の画像が失われることにより、後にオンデマンドでその授業を聴講した場

表1. 実際の授業における使用スライドの枚数と不要スライドおよびロスト枚数

講義	PPT枚数	1秒間隔		2秒間隔		5秒間隔		10秒間隔	
		不要枚数	ロスト枚数	不要枚数	ロスト枚数	不要枚数	ロスト枚数	不要枚数	ロスト枚数
朝日大①	65	6	0	2	1	0	2	0	4
朝日大②	81	11	0	3	2	1	3	0	7
朝日大③	43	6	0	1	0	0	1	0	3
朝日大④	52	5	0	2	0	1	2	0	2
朝日大⑤	58	3	0	0	0	0	1	0	1
朝日大⑥	63	8	0	3	1	1	2	0	4
朝日大⑦	35	4	0	1	0	0	0	0	1
朝日大⑧	121	18	1	8	6	2	8	0	15
朝日大⑨	73	9	0	4	2	3	3	1	4
朝日大⑩	82	7	0	6	1	2	2	0	3
愛知大①	52	5	0	2	0	0	1	0	3
愛知大②	55	4	0	1	1	0	2	0	2
愛知大③	46	4	0	2	0	0	0	0	3
愛知大④	62	6	0	3	1	0	1	0	5
名城大①	38	2	0	0	0	0	0	0	2
名城大②	56	8	1	3	2	0	2	0	4
豊田工大	36	1	0	0	1	0	1	0	3
大阪工大	70	7	0	4	1	0	3	0	6
平均	60.44444	6.333333	0.111111	2.5	1.055556	0.555556	1.888889	0.055556	4
標準偏差	20.71153	3.834825	0.323381	2.093407	1.433721	0.921777	1.811366	0.235702	3.162278
比率	100	10.47794	0.183824	4.136029	1.746324	0.919118	3.125	0.091912	6.617647

キャプチャ間隔と不要比率、ロスト率の関係

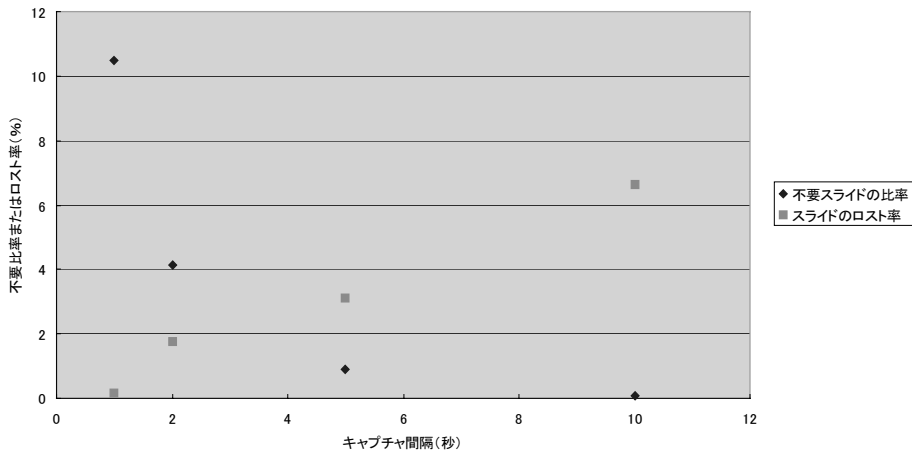


図6. 不要スライドの比率とスライドのロスト率

合、理解しにくくなる場合も考えられる。したがって、できる限るスライドのロストは避ける必要がある。

以上の2つの問題点、特に不要スライドの問題とスライドのロスト問題についていくつかの大学で講義をした際の実測データがあるので紹介する。表1はいくつかの大学で著者が行った講義について、実際に使

7. 実際の講義における状況

用した PowerPoint のスライドの枚数と、ある一定間隔でスライドをキャプチャした場合の不要スライド数とスライドのロスト数を示したものである。ただし、実測したのは 1 秒間隔のデータだけであり、2 秒間隔よりも広い間隔のデータは 1 秒間隔のデータをもとに当該秒数の中間データを削除して作成した。なお、いずれも 90 分の講義時間を使って行われた。

以上のデータから、不要スライドの比率とスライドのロスト率を計算し、プロットすると図 6 のようになる。ここで、不要スライドの比率 = 平均不要スライド枚数 / 平均スライド枚数、ロスト率 = 平均ロスト枚数 / 平均スライド枚数である。結果は、予想されるとおり、キャプチャ間隔を長く取れば不要スライドの比率は減るが、スライドのロスト率は増える。

なお、原理的に両者の割合を 0 にすることは不可能である。前者は **chance correlation** の問題であり、変化中のスライドをたまたまキャプチャする確率は 0 ではない。したがって、どのような場合でも、不要スライドが現れる場合がある。当初キャプチャソフトウェアとしてロータスの Screen CAM を使用していたが、現在は自作のプログラムを使用している。どちらも Windows の割り込み制御を利用して画面をキャプチャしているため、キャプチャ間隔を 1 秒以下にすることは難しい (Screen CAM では 1 秒以下の設定は不可能である。自作プログラムでは、設定することはできるが、正しい時間間隔で動作しているかの検証は行っていない。) そのため、どうしても速い変化を捉えることはできず、スライドのロスト率を 0 とすることはできない。

ただし、この問題は講師が十分な余裕をもってスライドを表示することで対処可能である。

## 8. 問題解決のための試行

スライドのロスト問題は現状では自動的な解決法はない。したがって、最後は人によるチェックを省くことはできない。しかしながら、不要スライドの問題とスライドへの書き込みなどの問題はある程度対処できる。ここでは、試行した対処法について述べるが、こちらも完全な解をえることはできない。

ここでは、まず変化量を計測する画面分割数  $n$  と変化した部分の比率  $m$  について考察した。すなわち、一般的に次のことが言える。

- (1) 画面分割数が 1 の場合：どのような微細な変化も記録できる。
- (2) 画面分割数を  $n$  とし、閾値を  $m$  とした場合：
  - ・  $n$  と  $m$  の値を適当に取ることで、微小変化を無視できる
  - ・ 例えば、 $n=25$  ( $5 \times 5$ ) とし、 $m=2$  とすれば 4% 以下の変化は無視できる

ところで、全変化記録では微小変化による影響が大きすぎる。一方、 $n$  と  $m$  の最適値を求めることは難しい。そこで、たとえばその都度適用できるいくつかの変化量の計測パターンを用意し、発見的なアルゴリズムにより、できるだけ不要スライドを除去することを考えた。いずれにしても、不要スライドの除去も最後は人手にたよるしかない。したがって、ここでの目的はいかに人手による作業量を減らすかを考えること



になる。

そこで、スライドが変化している状態をキャプチャした場合、図7に示したように上下に変化することから、分割を上下方向に限定したパターンを使うことで、ほとんどの場合は除去できた。この場合、 $m$ は変化した部分の比率ではなく、個数として、1～2個の変化部分がある場合だけ除去する。これにより、不要スライドの大部分は

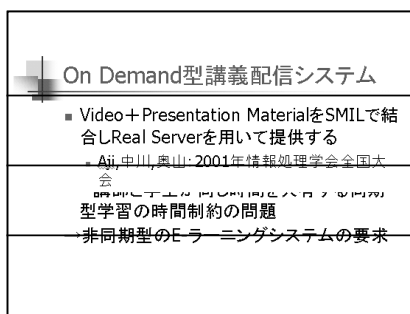


図7. 分割パターンの変更

除去できる。

次に、描画途中のスライドは、 $n$ と $m$ の値を適切に決めることにより多くの部分は除去できると考えた。実際には最適値を決めることは難しい。そこで、経験的に $n=64 \times 48$ とし、 $m$ を5%程度と定めた。これにより、全体の7割程度 of 描画途中でキャプチャされたスライドを除去することができる。 $m$ の値をこれ以下にすると最終的に必要とされる描画を終えたスライドも除去する危険性が増大し、これ以上にするとほとんどの描画途中のスライドを残してしまう。

このようにして、ほとんどの不要スライドは除去できるが、残る部分は手作業で除去する必要がある。しかしながら、これらの操作により、手作業の割合は飛躍的に減少する。

## 9. 今後の課題

以上のような種々のアルゴリズムを組み合わせても、最終的には完全な自動化は達成できない。全自動化を達成するためにはあらたなアルゴリズムの工夫が必要である。

このような工夫は、著者が考えている次のステップを実現するためにも重要である。すなわち、前回紹介した **mimio** による板書記録 (図8参照) の適切な中間状態を記録して提示する、あるいは、WEB ページの適切な表示状態を保存することができれば、PowerPoint のようなスライドデータだけでなく、より多くの教材データの同期を自動化できる。

今後は、より変化を適切に捕らえて分析

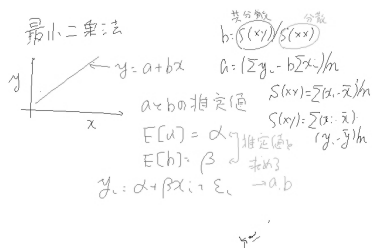


図8. mimio による板書データの例

できる新しいアルゴリズムの考案が必要である。しかしながら、そのようなアルゴリズムは難しい。また、たとえ実現できたとしてもそれが著しく時間を浪費するものであってはならない。いずれにしても、より詳細な解析が可能なアルゴリズムについて今後考えたい。

## 10. まとめ

以上、非同期型の授業を実現する一つの方法としてのオンデマンド提供する教材を自動的に作成するための一つの方法を示し

た。ところで、そのような自動化は必要なのかという疑問が残る。なぜなら、これらのことははじめから人手でやることで十分な結果が得られることであり、無理に不完全な自動化を目指す必要はないという議論である。しかしながら、今後、このような教材を作成する機会は増えつつけることが予想され、それらすべてを人手で行うことが必ずしも妥当であるとは思えない。さらに、人手によって作成された教材の信頼性を確保するためには、結局、クロスチェックなどが必要となる。

そこで、不完全な自動化でも、初期バージョンを機械的に作成し、そのときの詳細なログを残すことで、人手による作業量を減らすことができる。また、ログと照合することで作業者の作為を排除することができ、しかもクロスチェックがやりやすくなる。

以上のように、非同期型の遠隔授業のためのビデオ教材を効率よく作成するための一つのアイデアとそのための実装系について提案した。このシステムを導入することにより、手作業の 1/10 以下の時間で教材を作成することが可能となった。今回のバージョンは主にコマンドラインの命令群を組み合わせて作業を行う。そのため、必ずしも効率の良いものではない。そこで、今後は、より使い易いインターフェイスの開発が必要である。

#### 【文献など】

- [1] 奥山徹、「誰でもできるマルチメディア授業」、愛知大学情報処理センター紀要 COM、Vol.12、No.2、2001.
- [2] 豊橋技術科学大学マルチメディアセン

ター、「MUPS 事業プロジェクト研究」、1997～2001.

<http://www.mc.tut.ac.jp/mups/project.html>

[3] D.L. Mills, “Network Time Protocol (Version 3), Specification, Implementation and Analysis”, RFC1305, IETF, 1992.

[4] <http://www.w3.org/AudioVideo/>.

[5] <http://www.w3.org/XML/>.