

Title	マルチビデオストリームを配信するオンデマンド学習システムに関する研究
Author(s)	澤田, 憲志
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1537
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修 士 論 文

**マルチビデオストリームを配信する
オンデマンド学習システムに関する研究**

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

澤田 憲志

2002 年 3 月

修 士 論 文

マルチビデオストリームを配信する オンデマンド学習システムに関する研究

指導教官 丹 康雄 助教授

審査委員主査 丹 康雄 助教授

審査委員 篠田 陽一 教授

審査委員 敷田 幹文 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

010050 澤田 憲志

2002 年 2 月 15 日

要 旨

本研究では、提案する複数の動画像を配信するオンデマンド学習システムの有効性を検証し、そのシステムの実装上の課題を検討する。

まず、提案するシステムの有効性を明らかにするために、講師映像とスライドによる一般的な表示形態との比較評価実験をおこなった。その実験結果を分析し考察する。つぎに、コンテンツ制作の作業工程数および作業時間による制作コストをもとに、システムの有効性を検証する。

複数の動画像の配信を考慮に入れ、クライアントの要求に対応するユーザインタフェースとして、視点別動画像選択方式とレイアウトのリアルタイム表示切替方式を提案する。

広域ネットワークにおいて、高品質かつ大容量コンテンツである複数の動画像の同期再生を実現するためには、エンドユーザへの安定した配信が求められる。オンデマンド配信においては、広域に分散配置されたキャッシュサーバの利用を検討する。ライブ配信においては、複数のサーバからのストリームデータを単一送信元としてクライアントへ転送する中間同期化ノードを提案する。

目次

1	はじめに	1
2	遠隔教育用オンデマンド配信の現状と課題	3
2.1	ストリーミング伝送技術	3
2.1.1	ストリーミング	3
2.1.2	ストリーム配信方式とシステム製品	3
2.1.3	ストリーミングで利用されるプロトコル	4
2.1.4	同期再生表示のための記述言語 SMIL	9
2.2	”e-Learning ”の実例	10
2.3	コンテンツ表示形態の課題	11
3	コンテンツ表示形態の比較評価実験	12
3.1	評価実験システム概要	12
3.1.1	コンテンツ表示形態	12
3.1.2	コンテンツの講義内容	15
3.1.3	実験システム環境	15
3.1.4	評価方法	16
3.1.5	アンケート項目	16
3.2	評価結果	17
3.2.1	平均値の評価	17
3.2.2	総合的评价	19
3.2.3	因子分析	19
3.2.4	因子の内部相関	20
3.2.5	単回帰分析	22
3.2.6	重回帰分析	23

3.3	評価結果の考察	23
3.3.1	「従来型」の考察	23
3.3.2	「提案型」の考察	24
3.3.3	「従来型」「提案型」共通の考察	24
3.3.4	「提案型」実装上の課題	25
4	配信コンテンツの制作コストによる表示形態の比較	26
4.1	企画・制作アプローチによる分類と制作工程	26
4.1.1	スタジオ収録蓄積型	27
4.1.2	ライブ収録加工蓄積型	28
4.1.3	ライブ多視点収録蓄積型	30
4.2	コンテンツ制作標準作業時間の算出	32
4.2.1	スタジオ収録蓄積型の標準作業時間	32
4.2.2	ライブ収録加工蓄積型の標準作業時間	34
4.2.3	ライブ多視点収録蓄積型の標準作業時間	36
4.3	制作コストによる比較検討	37
4.3.1	コンテンツ制作最短作業時間の比較	37
4.3.2	コンテンツ制作作業時間の典型値による比較	40
4.3.3	制作コストの比較	43
5	複数の動画像を視聴するためのユーザインタフェース	45
5.1	視点別動画像選択方式	45
5.2	レイアウトのリアルタイム表示切替方式	46
6	サーバ間の負荷分散とトラフィック分散	52
6.1	負荷分散配置と制御	52
6.1.1	複数のサーバへのコンテンツの分散配置	52
6.1.2	負荷分散制御	54
6.2	広域ネットワークにおける分散配置とオンデマンド配信	55
6.2.1	コンテンツ配信ネットワーク	55
6.2.2	プロキシによるストリーミングのサポート	57
6.2.3	複数のストリームのオンデマンド配信	60

7 複数のストリームのライブ配信における同期再生の課題と中間同期化ノード	61
7.1 複数のストリームのライブ配信における課題	61
7.2 複数のストリームの同期再生	63
7.3 中間同期化ノード	63
8 今後の課題	67
8.1 マルチビデオストリームを用いた表示形態による学習効果の検証	67
8.2 ユーザインタフェースの検討	67
8.3 ライブ配信における中間同期化ノードの実装と動作検証	68
9 おわりに	69

目 次

2.1	ストリーミング・プロトコルの構成	4
2.2	RTP Header Fields (RFC1889)	5
2.3	SR : Sender report RTCP packet (RFC1889)	6
2.4	RTSP サーバ／クライアントの状態遷移図	8
2.5	SMIL によるマルチメディアデータの制御	9
3.1	講師映像とスライドを表示する「従来型」のレイアウト	13
3.2	複数の映像を表示する「提案型」のレイアウト	14
3.3	実験システムの構成	15
3.4	各質問項目の平均値	18
3.5	総合的評価	20
3.6	因子成績	22
4.1	スタジオ収録蓄積型コンテンツ制作工程	28
4.2	動画像とスライドとテキストとの同期再生表示のための記述ファイルの構成例	29
4.3	ライブ収録加工蓄積型コンテンツ制作工程	30
4.4	ライブ多視点収録蓄積型コンテンツ制作工程	31
4.5	複数の動画像を同期再生表示するための記述ファイルの構成例	31
4.6	スタジオ収録蓄積型の作業時間の定義	33
4.7	ライブ収録加工蓄積型の作業時間の定義	35
4.8	ライブ多視点収録蓄積型の作業時間の定義	36
5.1	視点別動画像選択フォーム	46
5.2	プロトタイプのレイアウト	48
5.3	レイアウト選択フォーム	50
5.4	レイアウトのリアルタイム表示切替リストボックス	51

6.1	配信サイト内のオリジンサーバのミラーリング	53
6.2	配信サイト内のキャッシュサーバの配置	53
6.3	コンテンツ配信ネットワーク	56
6.4	CDN ピアリング	56
6.5	キャッシングの接続	57
6.6	スプリッティングの接続	58
6.7	マルチキャストの接続	58
6.8	パススルーの接続	59
7.1	異なる伝送経路を通る複数のライブストリーム	62
7.2	異なる伝送手段を使用する複数のライブストリーム	62
7.3	中間同期化ノードの配置イメージ	64
7.4	中間同期化ノードを配置した場合の配信動作フロー	65
7.5	中間同期化ノードを配置した場合のチャンネルのセットアップ	66

表 目 次

2.1	RTP ヘッダフィールドの定義	5
2.2	Sender report RTCP packet ヘッダフィールドの定義	7
2.3	RTSP の制御機能リスト	8
3.1	因子抽出結果	21
3.2	因子成績間の内部相関	21
3.3	各因子成績から総合的評価を単回帰予測したときの回帰係数	22
4.1	企画・制作アプローチの違いによる分類	27
4.2	コンテンツ制作作業時間の典型値による比較	43
5.1	ICML と既存のスクリプト言語との比較	47

第 1 章

はじめに

近年、ネットワークの大容量・高速化と常時接続料金の低価格化により、動画像などを中心とする大容量コンテンツの配信サービスが提供されるようになってきた。いわゆる“ e-Learning ”と呼ばれているインターネット上でのオンライン学習をはじめとする教育コンテンツのオンデマンド配信は、高等教育機関における単位取得や企業内研修等で利用されており、今後も生涯学習と結び付いて、あらゆる人々の知識向上のためのひとつの手段として活用されていくであろう。

現在、遠隔教育用オンデマンド配信のコンテンツとして、講師映像とプレゼンテーションソフトウェアのスライドとの同期再生表示をおこなう形態が一般的に用いられている。しかし、この表示形態では、教室でおこなわれている通常の講義を収録して配信する場合、講師がスライドを指し示しながら説明する場面において、視聴者は指示場所を知ることができないという問題が生じる。そして、スライド表示以外の手段を用いる講義、例えば板書や OHP などを使用している講義の場合には、その講義に対応したコンテンツ制作が必要になるという問題がある。

これらの問題を解決するためには、ライブ講義を収録するのではなくオンデマンド配信コンテンツとして講義を企画し収録することがひとつの方法であるが、制作コストが大きくなる。

そこで、本研究ではライブ講義を複数のビデオカメラで収録し、オンデマンド配信によって複数の動画像コンテンツの同期再生表示を提供するシステムを提案し検討する。

これまでも複数のビデオストリームを用いた遠隔教育システムの提案 [15][16] はなされている。これらの提案では、実際の教室に複数のビデオカメラを設置し、受講生の視点を複数のビデオストリームとして提供することで、教室の対面講義に近い効果が得られるシステムを検証している。ただし、これらはライブ配信、すなわち生中継で配信データを

残さない配信方式を前提としていた。

本論文では、まず遠隔教育用オンデマンド配信において複数の動画像を用いることの有効性を検証するために、ローカルエリアネットワークに実験システムを構築し、一般的に用いられている講師映像とスライドによる同期再生表示との比較評価実験をおこなった。

つぎに、配信コンテンツの企画・制作アプローチの違いによる表示形態の分類をおこない、コンテンツ制作の作業工程数と作業時間に基づく制作コストによって表示形態を比較する。

そして、複数のストリームをオンデマンドで配信し同期再生表示させるための実装上の課題を述べる。配信される動画像は高品質かつ大容量コンテンツであり、広域ネットワークにおける配信にあたっては、エンドユーザに安定してストリームデータを提供するための手法が必要である。一方、受信側においては、複数の動画像の表示数がディスプレイの画面領域サイズに制約を受けることから、それに対応するためのユーザインタフェースが必要である。これら実装上の課題の解決手法を検討する。

複数のストリームのライブ配信では、複数の送信元サーバからクライアントまでを常に接続している必要がある。広域ネットワークにおいては各ストリームの伝送手段や伝送経路が異なることも考えられ、安定したストリーミングの実現とともに複数のストリーム間の同期再生の実現が課題となる。この課題の解決として、中間同期化ノードを提案する。

第 2 章

遠隔教育用オンデマンド配信の現状と課題

この章では、まず、ストリーミングの伝送技術に関して述べる。つぎに、いわゆる ”e-Learning ” といったオンデマンド学習コンテンツ配信サービスの実例を挙げ、そこで用いられているコンテンツ表示形態の課題を述べる。

2.1 ストリーミング伝送技術

ストリーミングの伝送技術として、ストリーム配信方式、ストリーミングで利用されているプロトコル、そしてコンテンツの同期再生表示を制御する SMIL について述べる。

2.1.1 ストリーミング

ストリーミングとは、サーバとクライアント間で伝送制御をおこない、データを受信しながらリアルタイムに順次再生していく技術である。コンテンツファイルをすべてダウンロードしてから再生するというダウンロード再生とは異なり、待ち時間が少なく、ユーザのローカルディスクにはデータが残らないため、動画像や音楽データの配信に広く利用されている。

一般的に受信側で再生用のバッファを用いて、伝送遅延やジッタを吸収している。

2.1.2 ストリーム配信方式とシステム製品

ストリーム配信方式には、ライブ配信とオンデマンド配信がある。

- ライブ配信

ライブ配信はある定められた時間帯にある場所でおこなわれているイベントの音声、映像を生放送としてストリーミングする配信方式である。

- オンデマンド配信

オンデマンド配信は記録したコンテンツを配信可能な状態にエンコードし、あらかじめサーバに蓄積しておいて、ユーザからの要求に応じてストリーミングを開始する配信方式である。

現在普及しているストリーム配信システムには、次の代表的な3社の製品がある。

- RealSystem (RealNetworks)
- QuickTime (AppleComputer)
- Microsoft Windows Media Technology (Microsoft)

2.1.3 ストリーミングで利用されるプロトコル

Application		
RTP/RTCP	RTSP	HTTP
UDP		TCP
IP		

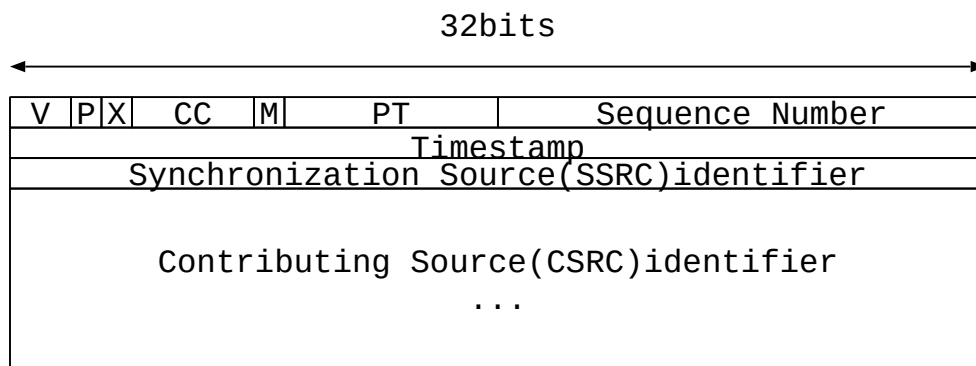
図 2.1: ストリーミング・プロトコルの構成

ストリーミングでは、UDP/IP とともに RTP (Real-time Transport Protocol) [1]、RTCP (Real-time Control Protocol) [1]、RTSP (Real-Time Streaming Protocol) [2] といったプロトコル (図 2.1) が利用されている。

- RTP

RTP は、リアルタイムデータを伝送するアプリケーションのために、エンドツーエンドの配信サービスを提供するプロトコルである。もともとはテレビ会議用のプロトコルとして設計されたものであるが、ストリーミングにも応用されている。

RTP は次のような機能を提供する。



V:Version (2bits)
 P:Padding (1bit)
 X:Extension (1bit)
 CC:CSRC Count (4bits)
 M:Marker (1bit)
 PT:Payload Type (7bits)

図 2.2: RTP Header Fields (RFC1889)

フィールド	長さ	説明
Version	2bit	RTP のバージョンを示す.
Padding	1bit	1 の場合, パケットの最後にパディングが付加
Extension	1bit	1 の場合, パケットヘッダの後ろに拡張ヘッダが付加
CSRC Count	4bit	ヘッダ中の CSRC の数を示す. 最大 15 個.
Marker	1bit	ペイロードタイプごとに定義されている
Payload Type	7bit	伝送ストリームの種別を示す
Sequence Number	16bit	順序番号. 初期値は乱数によって定める.
Timestamp	32bit	RTP データ部の最初のバイトのクロック値を示す
SSRC	32bit	同期送信元識別子
CSRC	32bit	寄与送信元識別子. ストリームの源を示す.

表 2.1: RTP ヘッダフィールドの定義

- 送信元識別
各 RTP パケットの送信元は、SSRC(Synchronization Source Identifier) と呼ばれる RTP ヘッダフィールドによって受信側で識別される
- ペイロードタイプ識別
RTP パケットに含まれるペイロードタイプは、ペイロードタイプ識別子と呼ばれる RTP ヘッダフィールドによって示され、受信側はペイロードタイプ識別子に基づいてパケットの内容を解釈する
- シーケンス番号付加
パケットの順序化とパケット損失検出のためにシーケンス番号を付加する
- タイムスタンプ付加
異なるメディアストリームとの間で同期をとるためにタイムスタンプを付加する

図 2.2 に RTP ヘッダフィールドを示す。これらの RTP ヘッダフィールドの定義をまとめたものが表 2.1 である。

● RTCP

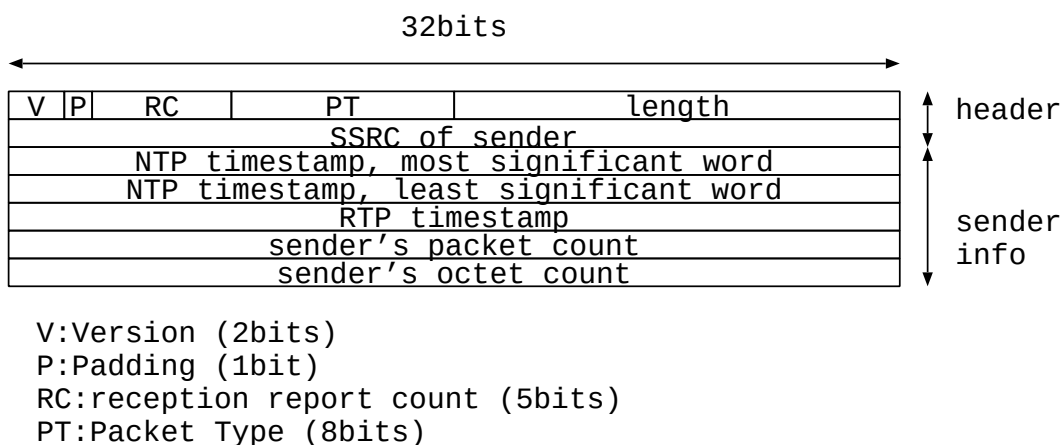


図 2.3: SR : Sender report RTCP packet (RFC1889)

RTCP は、RTP を補助するために、ストリームに関するフロー制御、メディア同期、発信元情報といった制御情報の通知を送受信間でおこなう。

図 2.3 のように、RTCP パケットには送信装置からストリームに関する情報の通知をおこなう SR(Sender Report) タイプがあり、このパケットは RTP データを送出

フィールド	長さ	説明
Version	2bit	RTP のバージョンを示す.
Padding	1bit	1 の場合, パケットの最後にパディングが付加
Packet Type	8bit	RTCP パケットの種別
Length	2byte	4byte 単位で表したパケットの長さから 1 を引いた値
SSRC	4byte	RTCP パケット送信者の SSRC 識別子を示す
RC	5bit	SR,RR パケットのみ:reception report block 数
SC	5bit	SDES,BYE パケットのみ
Timestamp	32bit	RTP データ部の最初のバイトのクロック値を示す
NTP	64bit	パケットが送出された絶対時刻を示す
RTP Timestamp	32bit	パケット送出時刻を示し, 基準クロックを復元
sender's packet count	32bit	これまでの RTP パケット総数を示す
sender's octet count	32bit	これまでの RTP パケットのペイロード部累積バイト数を示す

表 2.2: Sender report RTCP packet ヘッダフィールドの定義

した時点から周期的に送出される。SR タイプの Sender info には、このパケットの送出時刻における NTP タイムスタンプと RTP タイムスタンプの値が記入されている。SDES(Source Description) タイプには、送信装置に関する具体的な情報が含まれており、RTP ヘッダの SSRC と対応させることで送信元の識別が可能となっている。SR タイプの RTCP パケットヘッダフィールドの定義をまとめたものが表 2.2 である。

その他のタイプには、受信端末から受信したストリームに関する情報を通知する RR(Receiver Report) タイプ、通信の離脱を通知する BYE タイプ、アプリケーション固有の情報を通知する APP(Application) タイプがある。

- RTSP

RTSP は、サーバとクライアント間におけるストリームの送信などに関する制御コマンドである。停止、早送り、巻戻し、指定した位置からの再生など、VCR 形式の制御機能を提供する。

規定されている制御機能には、必須メソッドとして、OPTIONS、PLAY、SETUP、TEARDOWN の 4 種類と、サーバあるいはクライアントの状態を変化させるメソッドとして、PAUSE、PLAY、RECORD、SETUP、TEARDOWN の 5 種類がある。RTSP の制御機能リストを表 2.3 に示す。

メソッド	動作内容とステート変化の有無
DESCRIBE	コンテンツの仕様要求. 変化無
ANNOUNCE	コンテンツの仕様打診. 変化無
GET_PARAMETER	メディアのパラメータ再送要求. 変化無
OPTIONS	利用可能なオプション仕様のチェック. 変化無
PAUSE	リソースを保持したまま伝送中断. 変化有
PLAY	SETUP で準備したメディアの送信開始. 変化有
RECORD	メディアの記録開始. 変化有
REDIRECT	サーバからクライアントへの接続変更要求. 変化無
SETUP	メディアのためのリソース要求とセッション開始. 変化有
SET_PARAMETER	メディアのパラメータ設定. 変化無
TEARDOWN	リソースを解放してセッション終了. 変化有

表 2.3: RTSP の制御機能リスト

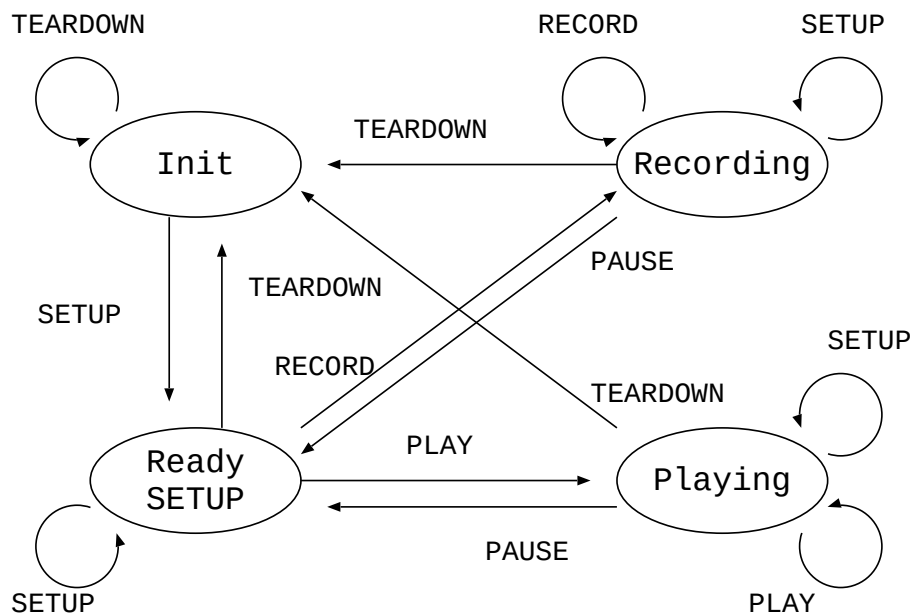


図 2.4: RTSP サーバ／クライアントの状態遷移図

RTSP クライアントのステートには、初期状態である Init(Initial)、コマンドの発行可能な状態である Ready、ストリームの受信状態である Playing、ストリームの送信状態である Recording の4状態がある。

RTSP サーバのステートには、初期状態である Init、コマンドの受信可能な状態である Ready、ストリームの送信状態である Playing、ストリームの受信状態である Recording の4状態がある。

RTSP サーバ／クライアントの状態遷移図を図 2.4 に示す。

2.1.4 同期再生表示のための記述言語 SMIL

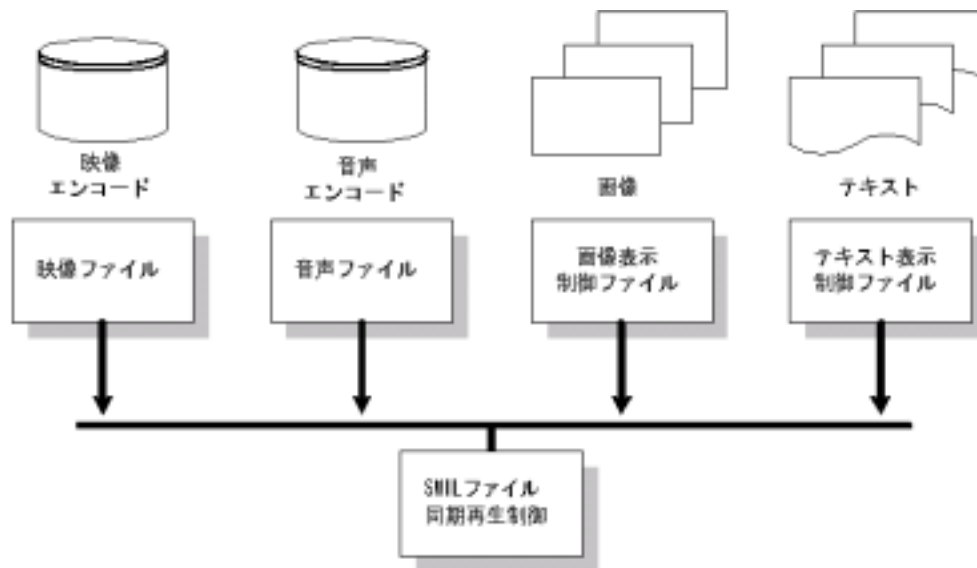


図 2.5: SMIL によるマルチメディアデータの制御

ストリーミングで用いられるマルチメディアデータを同期再生表示させる技術として、SMIL(Synchronized Multimedia Integrated Language)[5] が広く利用されている。SMIL は、W3C(World Wide Web Consortium) によって推奨され、音声、動画像、静止画像、テキスト等、マルチメディアデータの構成および制御をおこなう記述言語であり、複数のメディアストリームデータのレイアウトや同期のタイムライン制御を実現するメカニズムが定義されている。(図 2.5)

2.2 ”e-Learning ”の実例

いわゆる ”e-Learning ”といったオンデマンド学習コンテンツを配信しているサービスについて、講師映像とスライドとの表示形態を提供している高等教育や社会人教育、および企業内研修に関する実例を以下に挙げる。

- Stanford Online (Stanford Center for Professional Development (SCPD)) [7][8]

スタンフォード大学では、企業と連携して企業内学生の専門的能力を開発するため、主として機械工学、コンピュータサイエンスなどの遠隔教育をおこなっている。1970年代からテレビ放送を、1995年にはテレビ授業をインターネット上で見ることができるオンデマンド方式を開始した。

- SOI (WIDE University School of Internet) [9][10]

1997年9月より WIDE プロジェクトでは WIDE University School of Internet (SOI) を開始し、インターネットを基盤として、インターネットやコンピュータについて学ぶための学習環境を人々に提供してきた。講義および講演は、WIDE プロジェクトに所属する大学教員によって提供され、オンデマンド型で配信されている。これを利用する学生は、現役の学生のみならず社会人も含まれている。

SOI のオンデマンド授業では、講師の音声・映像、スライドなどの講義資料、板書などをデジタル化し同期再生表示させる手法を用いて、講義をインターネット上に配信している。

- LearningSite21 [11]

NTT ラーニングシステムズが運営する教育ポータルサイトである LearningSite21 は、社会人や企業を対象にオンライン学習システムによる人材育成サービスを提供している。オンデマンド配信は、NTT ラーニングシステムズのサーバへのアクセスによる利用、または顧客企業のイントラネット上のサーバにオンデマンドコンテンツを構築して利用、という2つの利用形態がある。コンテンツは講師映像とそれに同期して資料スライドの表示が自動的に切り替わる表示形態を用いている。

- 大原 WAK² Web [12]

大原 WAK² Web は大原グループと株式会社 NTT-ME/X が共同で開発した資格取得者向けサイトであり、専任講師による講義映像とポイント説明のためのスライド表示とともに、講師が板書した記述内容をリアルタイムに同期表示させる手法を用

いている。コンテンツはオンデマンド学習の視聴者用に収録された講義を加工して配信している。

- インターネット講座「現代版早稲田講義録」 [13]

インターネット講座「現代版早稲田講義録」は、早稲田大学で実施されている生涯学習講座を早稲田大学ラーニングスクエア株式会社がインターネット用に加工したもので、講師映像を中心に構成されている。

ケーブルテレビインターネットを活用したコンテンツ配信サービスプロバイダである AII 株式会社によって、2001 年 11 月から配信サービスが開始されている。

2.3 コンテンツ表示形態の課題

現在提供されているオンデマンド配信による一般的な表示形態は、講師映像とプレゼンテーションソフトウェアのスライドとの同期再生表示である。ところで、講義や講演によってはスライド表示以外の手段を使用することが考えられ、例えば板書や OHP を用いて講義がおこなわれる場合にはそれに対応した特別な措置が必要となる。そして、講師映像とスライドとの表示だけでは、場合によって講師がスライドを指し示す動作を視聴者が確認できないという問題も生じる。

これまでに、複数のビデオストリームを用いた遠隔教育システムの提案 [15][16] がなされており、登 [15] は実際の教室に複数のビデオカメラを設置し、受講生の視点を複数のビデオストリームとして配信して受信者に選択させることで、教室の対面講義に近い効果が得られるシステムを提案している。この提案では、ライブ配信、すなわち生中継で配信データを残さない配信方式を想定していた。

本研究では、通常の教室における講義を複数のビデオカメラで収録し、複数の動画像をオンデマンド配信するシステムを検討する。

第 3 章

コンテンツ表示形態の比較評価実験

この章では、提案する複数の動画像の同期再生表示の有効性を検証するためにおこなった比較評価実験について、その実験内容と評価結果を示し、分析および考察をおこなう。

3.1 評価実験システム概要

提案するオンデマンド配信による複数の動画像の同期再生表示について、視聴者に与える影響を調査し、その有効性を検証するために、現在、一般的に配信されている講師映像とプレゼンテーションソフトウェアのスライドによる同期再生表示との間で比較評価実験をおこなった。

以下、一般的な表示形態である講師映像とスライドとの同期再生表示を「従来型」、提案する複数の動画像による同期再生表示を「提案型」と呼ぶことにする。

3.1.1 コンテンツ表示形態

「従来型」は、講師映像である動画像とスライド切替記述ファイルによるスライド表示と、それらの同期再生を制御する SMIL 記述ファイルから構成されている。実験で用いた「従来型」コンテンツのレイアウトを図 3.1 に示す。

「提案型」は、今回の実験では講師映像とスライドスクリーン映像と板書映像の 3 つの動画像を用い、それらの同期再生表示を制御する SMIL 記述ファイルから構成されている。実験で用いた「提案型」コンテンツのレイアウトを図 3.2 に示す。

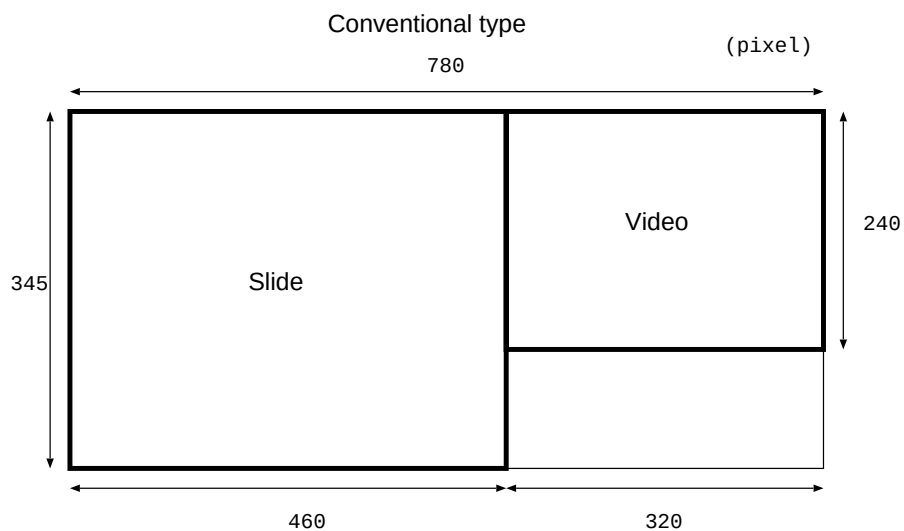
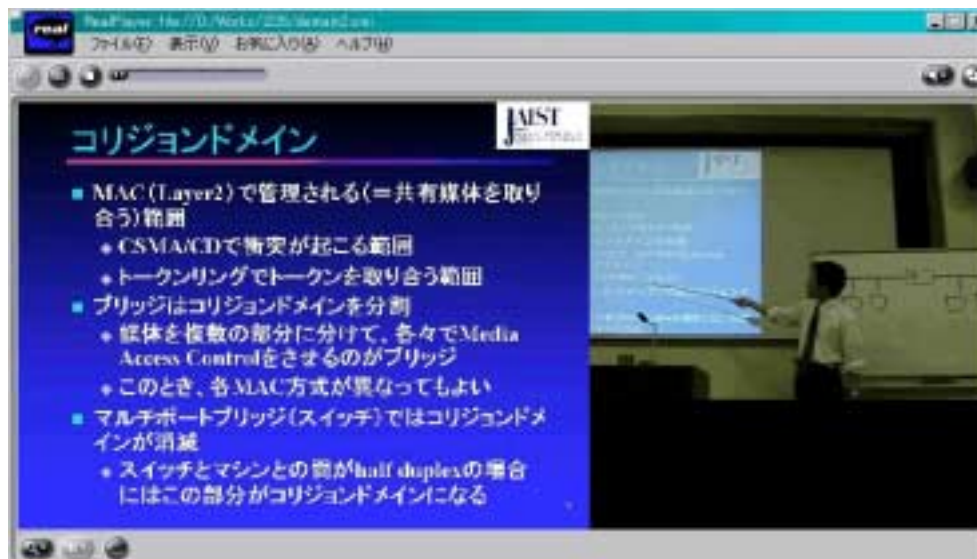


図 3.1: 講師映像とスライドを表示する「従来型」のレイアウト

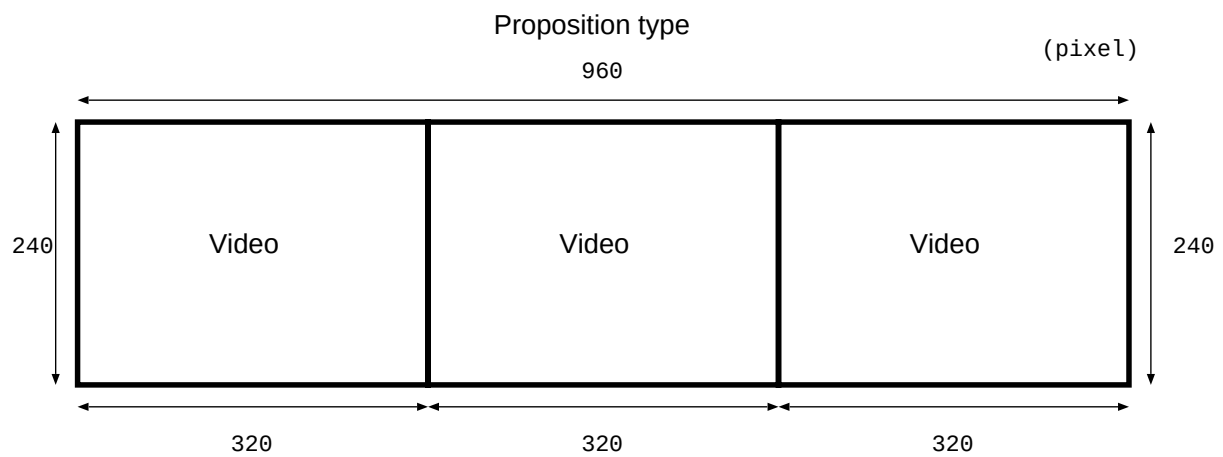
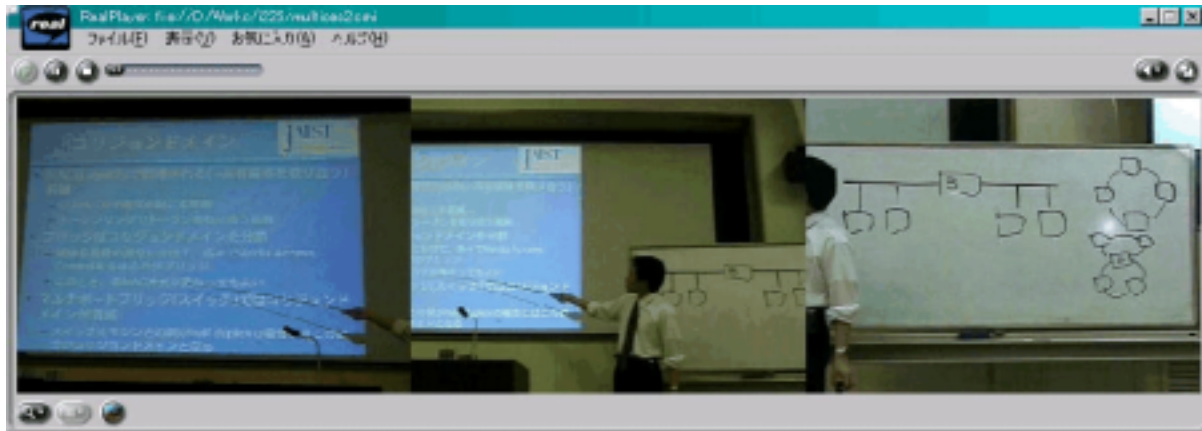


図 3.2: 複数の映像を表示する「提案型」のレイアウト

3.1.2 コンテンツの講義内容

2つの表示形態とも同一の講義内容とし、本学基幹講義である I226「コンピュータネットワーク特論」第6回（2001年6月28日）収録映像のうち500秒間を評価実験用コンテンツとして使用した。

3.1.3 実験システム環境

実験システムはローカルエリアネットワークに構築した。システムの全体構成は図3.3に示すように、配信サーバにある各表示形態の同期再生表示ファイル（SMIL ファイル）の場所を Web ページによって指定し、配信サーバからのストリームデータをクライアントシステムの StreamingPlayer（RealPlayer）が再生する仕組みになっている。

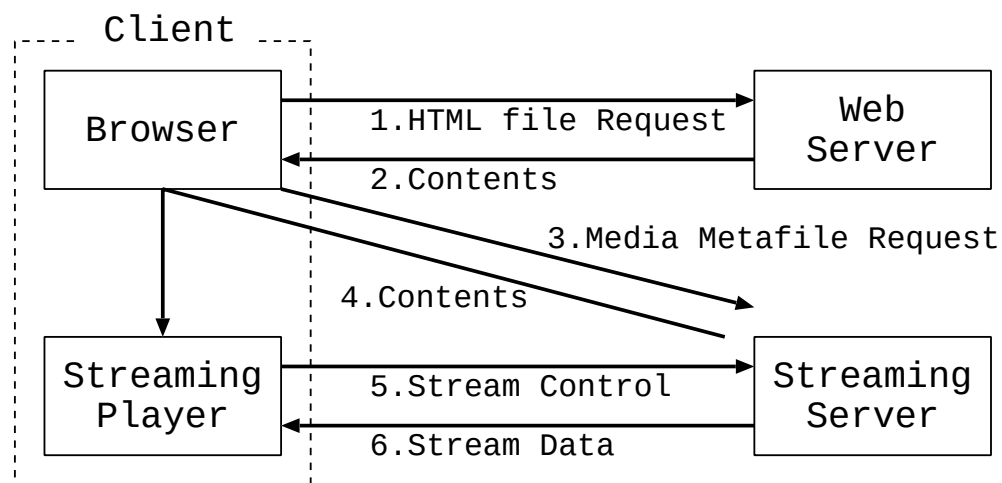


図 3.3: 実験システムの構成

- 配信サーバシステム環境
 - 10Mbps Ethernet LAN
 - 配信サーバ：CPU:400MHz ; RAM:288MB ; OS:WindowsNT4.0, RealServer8.0
 - 1つの動画像のビットレートは150Kbpsの帯域幅に対応した品質でRealVideo/Audioに符号化されている
- 評定者およびクライアントシステム環境

評定者として本学情報科学研究科博士前期課程の学生 11 名に依頼した。評定者は各自のクライアントシステムを用いて配信サーバからストリーミングによって視聴し比較評価をおこなった。

評定者が用いたクライアントシステムの性能は以下のとおりである。

- CPU:400MHz ; RAM:256MB ; OS:Solaris8
- CPU:500MHz ; RAM:256MB ; OS:Solaris8
- CPU:600MHz ; RAM:128MB ; OS:Windows2000
- CPU:733MHz ; RAM:192MB ; OS:WindowsMe
- CPU:350MHz ; RAM:128MB ; OS:Windows95

3.1.4 評価方法

CCIR (ITU-R) で推奨された 5 段階品質尺度をもとにした評価用語「非常に良い」「良い」「普通」「悪い」「非常に悪い」の 5 つのカテゴリー [25] を用いた。「従来型」「提案型」それぞれについて同一のアンケートによる主観評価をおこなった。

評定用紙には、アンケート項目と評定記入用罫線（長さ 100mm の横線、20mm 間隔にカテゴリー配置）を用意した。評定者がアンケートの各項目に最もふさわしいと判断した罫線上に印を記し、罫線の右端から評定者が記入した印までの距離をその質問項目の評点とした。

3.1.5 アンケート項目

アンケート項目は以下の 20 項目とした。

- (1) 講義の内容は興味深かった
- (2) 講義内容のポイントはつかめた
- (3) 講師の動作がわかった
- (4) 提示画面（スライドやボードなど）はよくわかった
- (5) 講師が指し示している場所がわかった
- (6) 全体的に画面構成はよかった
- (7) 画像の乱れやちらつきは気にならなかった

- (8) 目の疲れはなかった
- (9) 画像と音声の時間的ずれは気にならなかった
- (10) 音声が遅れることは気にならなかった
- (11) 講師の音声は聞き取ることができた
- (12) 視覚的にわかりやすく学習できた
- (13) 講義の雰囲気伝わった
- (14) 臨場感が得られた
- (15) 興味深く視聴することができた
- (16) 集中して講義を受けることができた
- (17) 講師の場所へ行くよりも遠隔教育がよい
- (18) オンデマンド学習で講義を受けることができるのはよい
- (19) このような講義を今後も受けたい
- (20) 総合的にこの表示形態はよい

3.2 評価結果

3.2.1 平均値の評価

「従来型」「提案型」それぞれの表示形態に対する評価の違いを明らかにするために、アンケートの各質問項目ごとに平均値を求め、2つの表示形態の間の平均値に対してt検定をおこなって差異を検証した。その有意差の大きい順に質問項目を並べたものが図3.4である。有意差が得られた項目の上に、5%有意水準には「*」、1%有意水準には「**」の記号を付けた。

- 「従来型」が高い評価を得ている項目
 - (7) 画像の乱れやちらつきは気にならなかった（有意水準1%）
 - (9) 画像と音声の時間的ずれは気にならなかった（有意水準1%）
 - (11) 音声が遅れることは気にならなかった（有意水準5%）

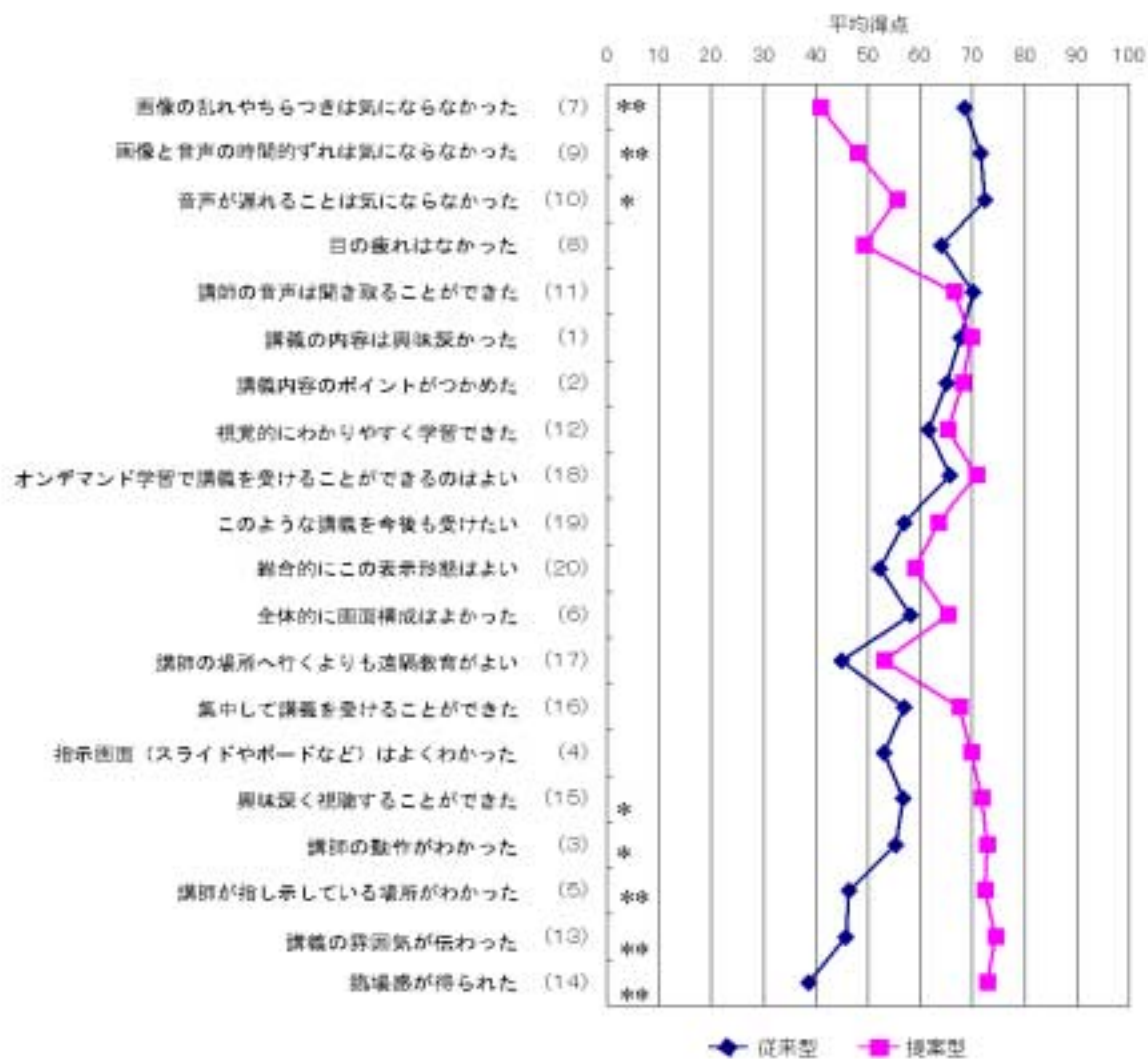


図 3.4: 各質問項目の平均値

- 「提案型」が高い評価を得ている項目
 - (14) 臨場感が得られた（有意水準 1 %）
 - (13) 講義の雰囲気伝わった（有意水準 1 %）
 - (5) 講師が指し示している場所がわかった（有意水準 1 %）
 - (15) 興味深く視聴することができた（有意水準 5 %）
 - (3) 講師の動作がわかった（有意水準 5 %）

3.2.2 総合的評価

アンケート質問項目 20 項目のうち以下の 6 項目については、総合的な評価を求める質問項目であり、これらの質問項目に対する平均値を表示形態の総合的評価とした。

- (15) 興味深く視聴することができた
- (16) 集中して講義を受けることができた
- (17) 講師の場所へ行くよりも遠隔教育がよい
- (18) オンデマンド学習で講義を受けることができるのはよい
- (19) このような講義を今後も受けたい
- (20) 総合的にこの表示形態はよい

2つの表示形態に対する総合的評価の結果が図 3.5 である。「従来型」の総合的評価は 51 点（カテゴリー「普通」中）よりも 5 %水準で有意に高い評価を得ている。一方、「提案型」の総合的評価は 60 点（カテゴリー「良い」）よりも 5 %水準で有意に高い評価を得ている。

両者の総合的評価の間で t-検定をおこなったところ、「提案型」が 1 %水準で有意に高く評価されていることがわかった。

3.2.3 因子分析

アンケートの質問項目の回答を分類し、これらの規定している因子を抽出して、主観評価の評価観点を明確にするために因子分析をおこなった。

因子分析には、質問項目 20 項目から総合的評価に関する質問項目 6 項目を除き、残る 14 項目に対して、SMC 法、VARIMAX 回転を用いて因子を抽出した。その結果、4 因子

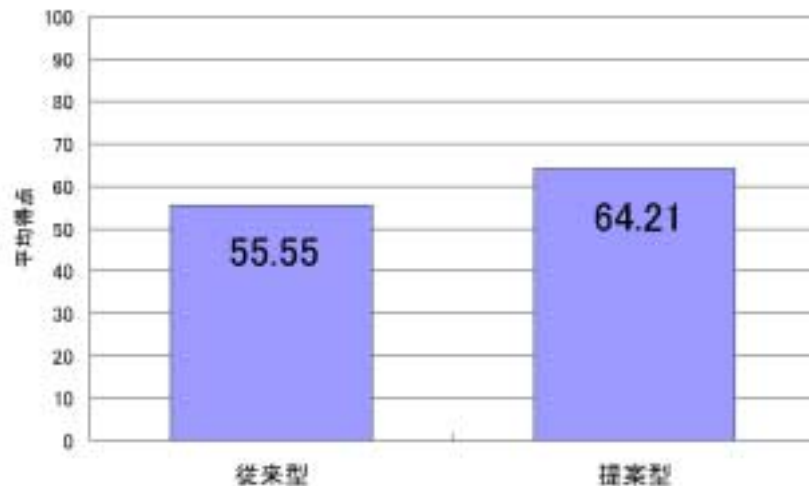


図 3.5: 総合的評価

を抽出することができた。このときの因子負荷量、寄与率を表 3.1 に示す。それぞれの因子は、抽出された質問項目から判断して、「画質・音声」「臨場感」「視覚効果」「内容」とした。なお、これら 4 因子の合計寄与率は 78.80 %であった。

各因子を構成する質問項目の回答平均を因子成績と定義し、2つの表示形態の評価の違いを検討するために、各因子成績ごとに t-検定をおこなった。それぞれの表示形態の各因子の因子成績を図 3.6 に示す。

その結果、第 1 因子「画質・音声」では 1 %水準で「従来型」が有意に高い評価を得ている。一方、第 2 因子「臨場感」と第 3 因子「視覚効果」では 1 %水準で「提案型」が有意に高い評価を得ている。なお、第 4 因子「内容」についての両者の間の有意差は認められなかった。

3.2.4 因子の内部相関

各因子成績は、各因子間相互にも関連していることが予測されるため、各因子成績間の内部相関係数を求めた。それが表 3.2 である。

その結果、第 2 因子「臨場感」と第 3 因子「視覚効果」と第 4 因子「内容」との間で高い相関係数が得られており、それぞれの評価が相互に影響していると推測される。

第1因子 画質・音声	寄与率 25.57 %
(8) 目の疲れはなかった	0.8573
(9) 画像と音声の時間的ずれは気にならなかった	0.8423
(7) 画像の乱れやちらつきは気にならなかった	0.8187
(10) 音声が遅れることは気にならなかった	0.7974
(11) 講師の音声は聞き取ることができた	0.5807
第2因子 臨場感	寄与率 20.93 %
(14) 臨場感が得られた	0.8935
(13) 講義の雰囲気伝わった	0.8792
(5) 講師が指し示している場所がわかった	0.5427
(6) 全体的に画面構成はよかった	0.4838
第3因子 視覚効果	寄与率 18.48 %
(4) 指示画面（スライドやボードなど）はよくわかった	0.8636
(3) 講師の動作がわかった	0.8610
(12) 視覚的にわかりやすく学習できた	0.6498
第4因子 内容	寄与率 13.83 %
(1) 講義の内容は興味深かった	0.9015
(2) 講義内容のポイントはつかめた	0.8315

表 3.1: 因子抽出結果

	画質・音声	臨場感	視覚効果	内容
画質・音声	1.0000	-0.0787	0.1877	0.2787
臨場感		1.0000	0.6367	0.5639
視覚効果			1.0000	0.5602
内容				1.0000

表 3.2: 因子成績間の内部相関

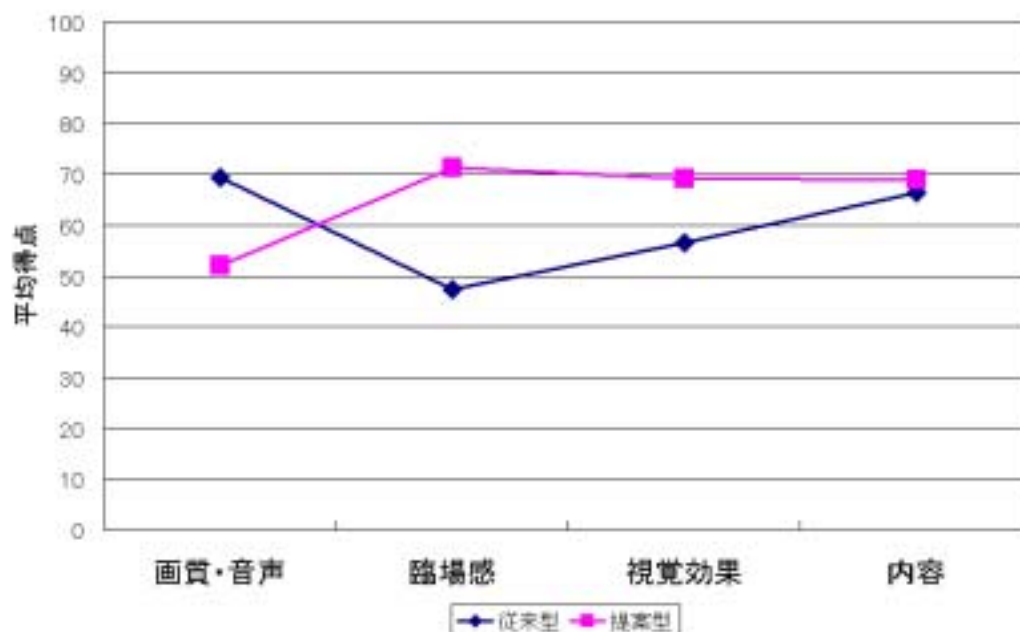


図 3.6: 因子成績

3.2.5 単回帰分析

各因子がどのように総合的評価に影響を与えているかを検討するために単回帰分析をおこなった。それぞれの因子成績から総合的評価を単回帰予測したときの回帰係数を表 3.3 に示す。表の有意「**」は 1 % 有意水準、「*」は 5 % 有意水準を表す。なお、「従来型」と「提案型」の評定をあわせて「全体」と定義した。

	全体		従来型		提案型	
	回帰係数	有意	回帰係数	有意	回帰係数	有意
画質・音声	0.2605		0.4830		0.7709	*
臨場感	0.6155	**	0.7555	**	0.9961	**
視覚効果	0.5095	**	0.5743	**	0.4181	
内容	0.4830	**	0.4860	*	0.4352	

表 3.3: 各因子成績から総合的評価を単回帰予測したときの回帰係数

「全体」の結果を見ると、第 2 因子「臨場感」、第 3 因子「視覚効果」、第 4 因子「内

容」の相関係数が比較的に高く、密接に関係している。

「従来型」の結果においては、第2因子「臨場感」、第3因子「視覚効果」、第4因子「内容」で有意な係数が得られ、総合的評価に強く影響していることを示している。

「提案型」の結果においては、第2因子「臨場感」、第1因子「画質・音声」で有意な係数が得られ、総合的評価に強く影響していることを示している。

3.2.6 重回帰分析

総合的評価に各因子の与える影響の度合いを見るために重回帰分析をおこない、以下のような回帰式を得た。ただし、 x_1 :画質・音声、 x_2 :臨場感、 x_3 :視覚効果、 x_4 :内容、とする。

$$y = 3.7265 + 0.3138x_1 + 0.6109x_2 + 0.1070x_3 - 0.0857x_4$$

この結果から、第2因子「臨場感」の係数が有意で大きく、次いで第1因子「画質・音声」の係数が有意であることから、これらの因子が総合的評価に影響を与えていることがわかる。

3.3 評価結果の考察

評価結果の分析から、オンデマンド配信における表示形態について、ユーザは総合的に「臨場感」「画質・音声」を求めていることがわかった。

以下では評価結果をもとに、「従来型」「提案型」それぞれの表示形態に関する考察と、2つの表示形態に共通に見られた点の考察を述べ、「提案型」実装上の課題を挙げる。

3.3.1 「従来型」の考察

「従来型」は、スライドと1つの動画像との表示により、画像の乱れがなく、「画質・音声」の高い評価に影響を与えている。その一方で、スライドとその説明のための動画像であることから、「臨場感」「視覚効果」の低い評価に影響を与えている。なかでも「臨場感」の低い評価が総合的評価に影響を与えている。

とくに評価が高かった項目は、第1因子「画質・音声」に属する以下の項目である。

- (10) 音声が遅れることは気にならなかった
- (9) 画像と音声の時間的ずれは気にならなかった
- (7) 画像の乱れやちらつきは気にならなかった

とくに評価が低かった項目は、第2因子「臨場感」に属する以下の項目である。

- (14) 臨場感が得られた
- (13) 講義の雰囲気伝わった
- (5) 講師が指し示している場所がわかった

3.3.2 「提案型」の考察

「提案型」は、複数の動画像の表示が「臨場感」「視覚効果」の高い評価に影響を与えている。これらの評価の要因として、講師映像の他にスライドスクリーン映像や板書映像を表示させていることが挙げられる。その一方で、評定者によっては画像のちらつきからストレスを感じ「画質・音声」の低い評価に影響を与えている。総合的評価には「臨場感」の高い評価、「画質・音声」のやや低い評価が影響を与えている。

とくに評価が高かった項目は、第2因子「臨場感」、第3因子「視覚効果」に属する以下の項目である。

- (13) 講義の雰囲気伝わった
- (14) 臨場感が得られた
- (5) 講師が指し示している場所がわかった
- (3) 講師の動作がわかった

とくに評価が低かった項目は、第1因子「画質・音声」に属する以下の項目である。

- (7) 画像の乱れやちらつきは気にならなかった

3.3.3 「従来型」「提案型」共通の考察

総合的評価については、両者の表示形態とも平均得点 50 点以上（カテゴリー「普通」中）の評価を得ていた。

第4因子「内容」については、平均得点 60 点以上（カテゴリー「良い」）の高い評価を得ており、両者の間に有意差は認められなかった。

「音声」に関する質問項目である、“(11) 講師の音声は聞き取ることができた”については、主観的には両者の表示形態とも音声の遅れは認められず、高い評価を得ていた。この理由として、音声の情報量が動画像に比べて小さいことが挙げられる。

3.3.4 「提案型」実装上の課題

評価実験の結果から、オンデマンド配信による複数の動画像を用いた表示形態において、臨場感や視覚的な表示効果の面での有効性が明らかになった。

そこで、複数のストリームをオンデマンドで配信し表示させるシステムを実装するにあたり、考慮しなければならない問題として次の2つを挙げる。

一つは、配信される動画像は高品質かつ大容量コンテンツであり、配信にあたっては、エンドユーザに安定してストリームデータを提供するための手法が必要である。もう一つは、受信側において、複数の動画像の表示数がディスプレイの画面領域サイズに制約を受けることから、それに対応するためのユーザインタフェースが必要である。

これら実装上の課題の解決手法については後章で検討していく。

第 4 章

配信コンテンツの制作コストによる表示形態の比較

この章では、配信コンテンツに対する企画・制作アプローチの違いによって表示形態を分類し、その制作工程における作業工程数と作業時間をもとに制作コストの比較をおこない、複数の動画像を配信する表示形態の有効性を検証する。

4.1 企画・制作アプローチによる分類と制作工程

遠隔教育用オンデマンド配信で用いられているコンテンツ表示形態は、例えば、通常の講義室でおこなわれているライブ講義を収録しコンテンツを制作する場合や、あるいは遠隔講義用にその視聴者向けに新たに講義をおこなって収録しコンテンツを制作する場合など、企画・制作アプローチの違いによって制作工程が異なっている。

講師映像とプレゼンテーションソフトウェアのスライド表示とを中心とする、実験コンテンツで用いた「従来型」のような表示形態の制作工程は、企画・制作アプローチの違いから大きく2つに分類することができる。

一つはオンデマンド配信用として講義を企画し収録する「スタジオ収録蓄積型」であり、もう一つは、ライブ講義を収録し加工してオンデマンド配信する「ライブ収録加工蓄積型」である。実験コンテンツの「従来型」は後者にあたる。

そして、複数の動画像に用いる、すなわち実験コンテンツの「提案型」のような表示形態の制作工程は、ライブ講義を収録してオンデマンド配信する「ライブ多視点収録蓄積型」である。

これらの企画・制作アプローチの違いによる分類をまとめたものが表 4.1 である。

	収録場所	映像	スライド表示	テキスト表示
スタジオ収録蓄積型	スタジオ	講師のみ	使用する	有
ライブ収録加工蓄積型	ライブ			無
ライブ多視点収録蓄積型		複数	映像収録	

表 4.1: 企画・制作アプローチの違いによる分類

制作されるコンテンツの講義の基本要素として、講師の音声と映像、プレゼンテーションソフトウェアのスライドを前提とし、制作工程の内容を型別に列挙する。ただし、同期再生表示のための技術は SMIL を用いるものとして考察する。

4.1.1 スタジオ収録蓄積型

スタジオ収録蓄積型では、コンテンツ制作においてオンデマンド配信による講義を企画し、収録スタジオ等で講師が視聴者に向けておこなう講義を収録する。

表示形態には、講師映像とプレゼンテーションソフトウェアのスライドとの同期再生表示が一般的であり、それに加えて視聴者の理解を手助けするために講師の口述内容をテキスト表示させているものが多い。

制作工程には、

- 講師の音声と映像をビデオカメラで収録し、エンコードする
- プレゼンテーションソフトウェアのスライドを表示させるため、各スライドの画像変換をおこなう
- 動画像と各スライドとの同期再生表示のために、講義進行にあわせてスライドのページを変化させるタイミング（スライド切替タイミング）を記録する
- スライド切替タイミングを基にして、スライド切替表示のための記述ファイルを作成する
- 講師の口述内容をテキストに変換するとともに、動画像とテキストとの同期再生表示のために、講義進行にあわせてテキスト表示を変化させていくタイミング（テキスト表示タイミング）を記録する
- テキスト表示タイミングを基にして、テキスト切替表示のための記述ファイルを作成する

- 動画像とスライドとテキストとの同期再生表示のための記述ファイルを作成する
- といった内容がある。

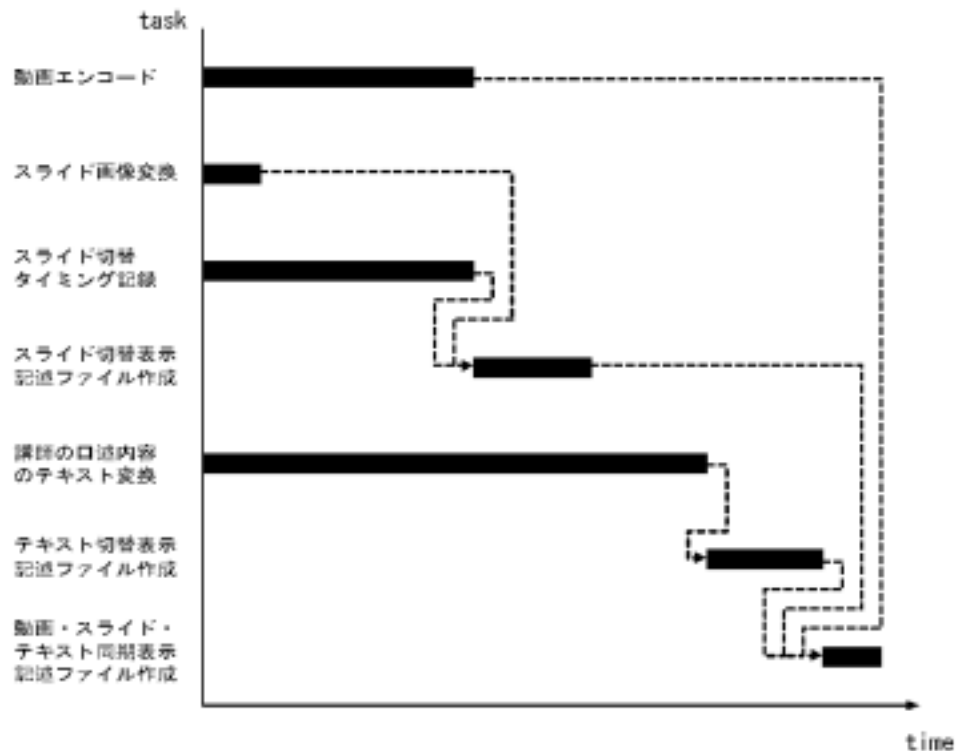


図 4.1: スタジオ収録蓄積型コンテンツ制作工程

スタジオ収録蓄積型コンテンツ制作工程を表したガントチャートが図 4.1 である。

4.1.2 ライブ収録加工蓄積型

ライブ収録加工蓄積型では、通常の教室等でおこなわれている講義や講演会場における講演などをライブ収録し、オンデマンド配信用に加工する。

表示形態には、講師映像とプレゼンテーションソフトウェアのスライドを表示させるものが一般的であり、講義の状況によっては、板書内容をテキストや画像に変換し HTML で表示させている場合もある。

制作工程には、

- 講師の音声と映像をビデオカメラで収録し、エンコードする

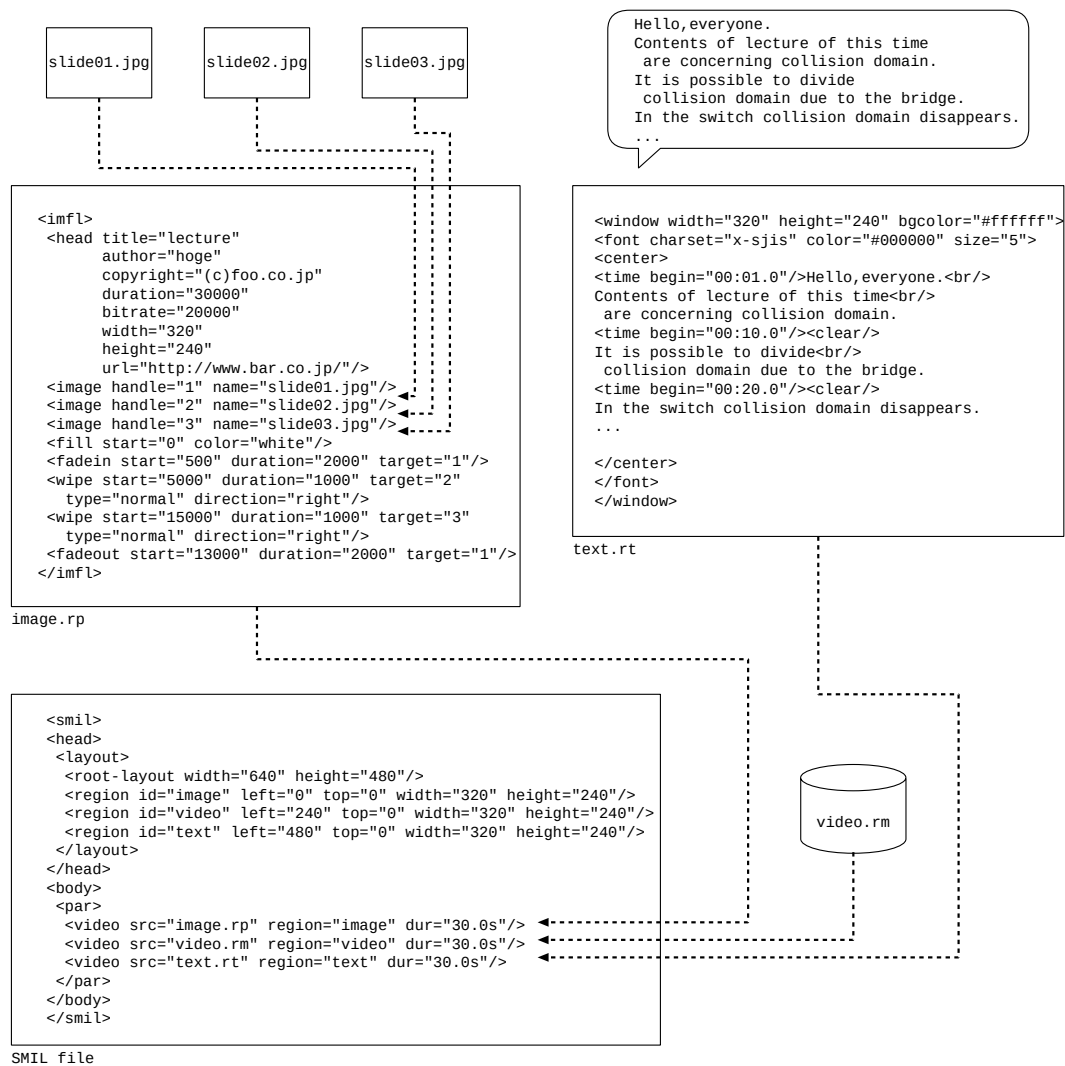


図 4.2: 動画像とスライドとテキストとの同期再生表示のための記述ファイルの構成例

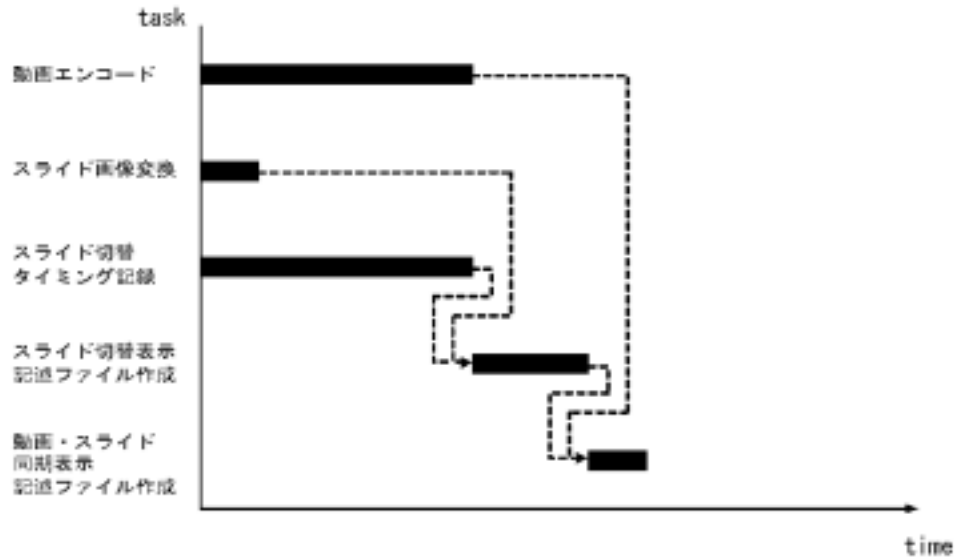


図 4.3: ライブ収録加工蓄積型コンテンツ制作工程

- プレゼンテーションソフトウェアのスライドを表示させるため、各スライドの画像変換をおこなう
- 動画像と各スライドとの同期再生表示のために、講師がスライドのページを変化させたタイミング（スライド切替タイミング）を記録する
- スライド切替タイミングを基にして、スライド切替表示のための記述ファイルを作成する
- 動画像とスライドとの同期再生表示のための記述ファイルを作成する

といった内容がある。

ライブ収録加工蓄積型コンテンツ制作工程を表したガントチャートが図 4.3 である。

4.1.3 ライブ多視点収録蓄積型

ライブ多視点収録蓄積型では、通常の教室等でおこなわれている講義や講演会場での講演において、講師、スライドスクリーン、黒板、OHP スクリーンなど、撮影対象の数だけ複数台のビデオカメラを用意し収録する。

表示形態は、収録した複数の動画像の同期再生表示である。

制作工程には、

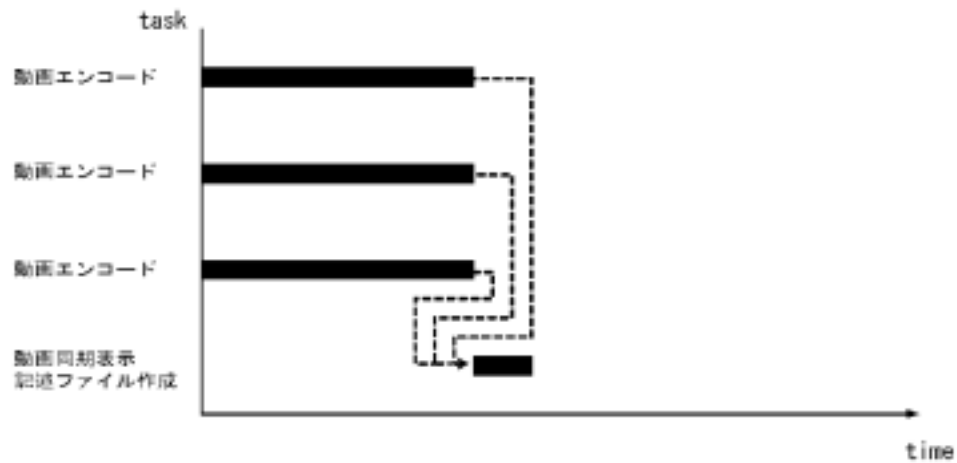


図 4.4: ライブ多視点収録蓄積型コンテンツ制作工程



図 4.5: 複数の動画像を同期再生表示するための記述ファイルの構成例

- 複数のビデオカメラで収録し、エンコードする
- 複数の動画像を同期再生表示するための記述ファイルを作成する

といった内容がある。

ライブ多視点収録蓄積型コンテンツ制作工程を表したガントチャートが図 4.4 である。

4.2 コンテンツ制作標準作業時間の算出

分類した各タイプのコンテンツ制作工程をもとに、各タイプの標準作業時間を算出する。

4.2.1 スタジオ収録蓄積型の標準作業時間

スタジオ収録蓄積型コンテンツ制作工程における各作業の作業時間を以下のとおり定義する。

- $t(e)$: 動画像エンコード処理時間
- $t(sx)$: スライド画像変換処理時間
- $t(str)$: スライド切替表示タイミング記録処理時間
- $t(pf)$: スライド切替表示記述ファイル作成処理時間
- $t(txr)$: 講師の口述内容のテキスト変換および切替表示タイミング記録処理時間
- $t(tf)$: テキスト切替表示記述ファイル作成処理時間
- $t(sf_studio)$: 動画像・スライド・テキスト同期再生表示記述ファイル作成処理時間

以上を図 4.6 に示す。

制作工程における各作業間の依存関係により、スライド収録蓄積型の作業時間 $T(Studio)$ は、

$$T(Studio) = t(sf_studio) + \max\{t(tf) + t(txr), t(pf) + \max\{t(str), t(sx)\}, t(e)\} \quad (4.1)$$

$t(sx)$ は、講師が用意したプレゼンテーションソフトウェアのスライドをクライアントシステムにおいて表示可能な画像ファイルに変換する作業時間であり、この作業はソフトウェアにより一括して瞬時に変換可能であることから、 $t(e)$ の時間内に作業が終了する。

$$t(sx) < t(e) \quad (4.2)$$

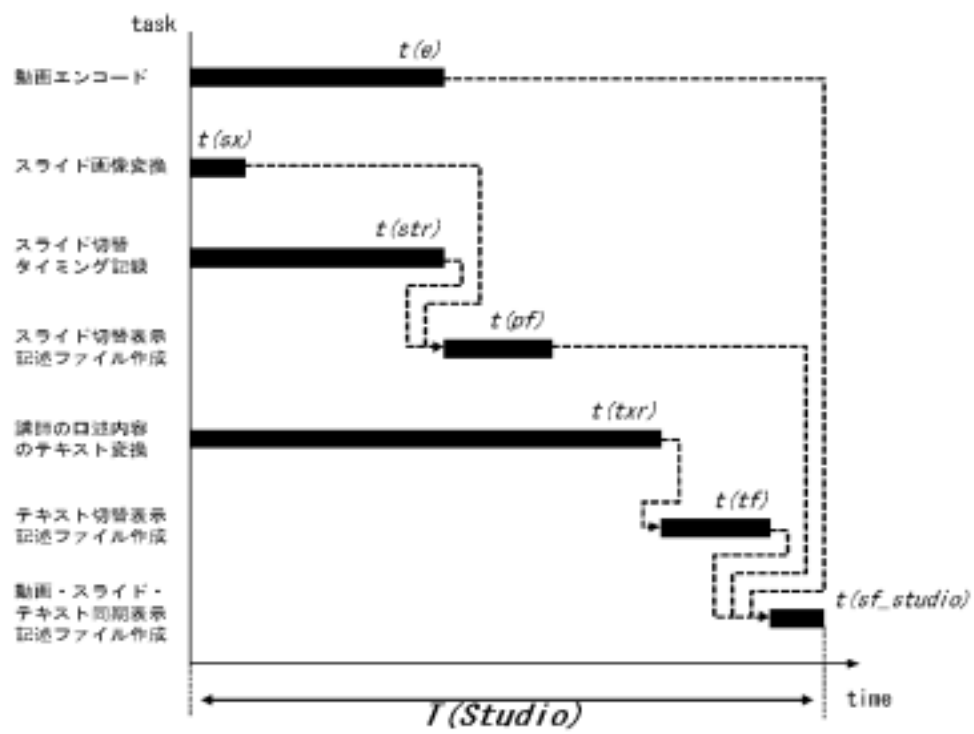


図 4.6: スタジオ収録蓄積型の作業時間の定義

$t(str)$ は、エンコードされている動画像の時間情報をもとにスライド切替のタイミングを記録する作業時間であり、 $t(e)$ とほぼ等しくなる。

$$t(str) \cong t(e) \quad (4.3)$$

$t(pf)$ の作業開始時刻は、 $t(sx)$ と $t(str)$ の作業終了時刻に依存する。ただし、 $t(sx)$ と $t(str)$ の工程は並行作業が可能である。(4.2) (4.3) 式より、

$$t(str) > t(sx) \quad (4.4)$$

よって、

$$t(pf) + t(str) > t(e) \quad (4.5)$$

$t(txr)$ は、収録した動画像やスライドをもとに講師の口述内容のテキスト変換およびテキスト表示タイミングの記録作業時間であり、 $t(e)$ よりも大きくなる。

$$t(txr) > t(e) \quad (4.6)$$

$t(tf)$ の作業内容は、テキスト表示レイアウトの指定と表示形式およびテキストの表示時刻の設定からなり、 $t(pf)$ の作業内容は、表示するスライド画像ファイルの指定と表示形式および表示時刻の設定からなる。よって、

$$t(tf) + t(txr) > t(pf) + t(str) \quad (4.7)$$

(4.5) (4.7) 式より、

$$t(tf) + t(txr) > t(pf) + t(str) > t(e) \quad (4.8)$$

(4.1) (4.4) (4.8) 式より、スタジオ収録蓄積型の標準作業時間 $T(Studio)$ は、

$$T(Studio) = t(sf_studio) + t(tf) + t(txr) \quad (4.9)$$

となる。

4.2.2 ライブ収録加工蓄積型の標準作業時間

ライブ収録加工蓄積型コンテンツ制作工程における各作業の作業時間を以下のとおり定義する。

- $t(e)$: 動画像エンコード処理時間

- $t(sx)$: スライド画像変換処理時間
- $t(str)$: スライド切替表示タイミング記録処理時間
- $t(pf)$: スライド切替表示記述ファイル作成処理時間
- $t(sf_ls)$: 動画像・スライド同期再生表示記述ファイル作成処理時間

以上を図 4.7 に示す。

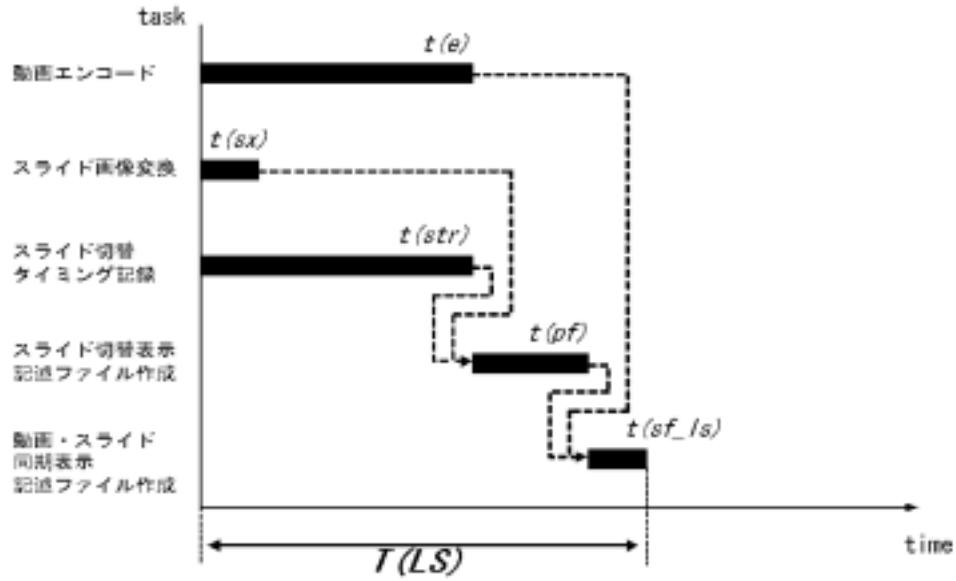


図 4.7: ライブ収録加工蓄積型の作業時間の定義

制作工程における各作業間の依存関係により、ライブ収録加工蓄積型の作業時間 $T(LS)$ は、

$$T(LS) = t(sf_ls) + \max\{t(pf) + \max\{t(str), t(sx)\}, t(e)\} \quad (4.10)$$

$t(sx)$ は、講師が用意したプレゼンテーションソフトウェアのスライドをクライアントシステムにおいて表示可能な画像ファイルに変換する作業時間であり、この作業はソフトウェアにより一括して瞬時に変換可能であることから、 $t(e)$ の時間内に作業が終了する。

$$t(sx) < t(e) \quad (4.11)$$

$t(str)$ は、エンコードされている動画像の時間情報をもとにスライド切替のタイミングを記録する作業時間であり、 $t(e)$ とほぼ等しくなる。

$$t(str) \cong t(e) \quad (4.12)$$

$t(pf)$ の作業開始時刻は、 $t(sx)$ と $t(str)$ の作業終了時刻に依存する。ただし、 $t(sx)$ と $t(str)$ の工程は並行作業が可能である。(4.11) (4.12) 式より、

$$t(str) > t(sx) \quad (4.13)$$

よって、

$$t(pf) + t(str) > t(e) \quad (4.14)$$

(4.10) (4.13) (4.14) 式より、ライブ収録加工蓄積型の標準作業時間 $T(LS)$ は、

$$T(LS) = t(sf_ls) + t(pf) + t(str) \quad (4.15)$$

となる。

4.2.3 ライブ多視点収録蓄積型の標準作業時間

ライブ多視点収録蓄積型コンテンツ制作工程における各作業の作業時間を以下のとおり定義する。

- $t(e)$: 動画像エンコード処理時間
- $t(sf_multi)$: 複数の動画像の同期再生表示記述ファイル作成処理時間

以上を図 4.8 に示す。

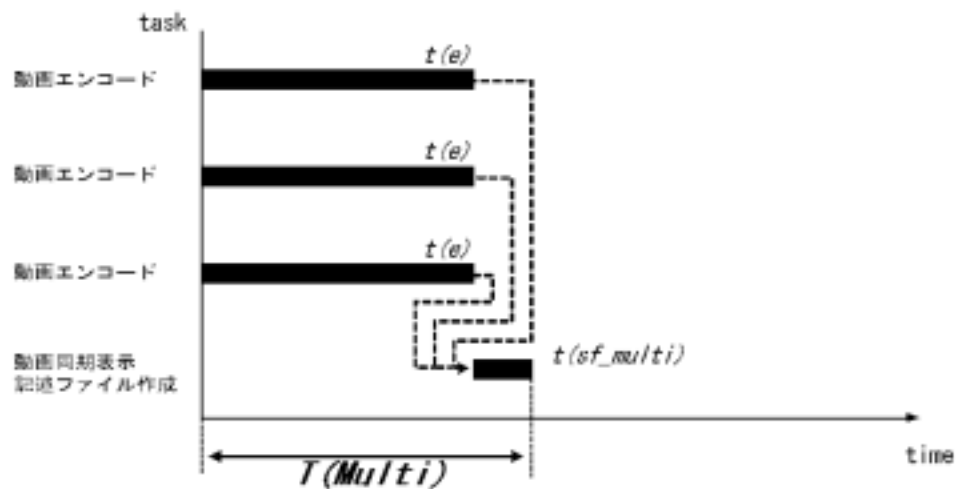


図 4.8: ライブ多視点収録蓄積型の作業時間の定義

$t(e)$ は1つの動画画像のエンコード処理時間であるから、複数の動画画像のエンコード処理時間は配信する動画画像の数に依存する。よって、収録するビデオカメラの台数を c 、エンコードの数を n とすると、

$$\left\lceil \frac{c}{n} \right\rceil * t(e) \quad (4.16)$$

ただし、 $\lceil \cdot \rceil$ は小数点以下切り上げを意味する。

$t(sf_multi)$ の作業開始時刻は、すべての動画画像のエンコード処理の終了時刻すなわち (4.16) 式に依存する。よって、ライブ多視点収録蓄積型の標準作業時間 $T(Multi)$ は、

$$T(Multi) = t(sf_multi) + \left\lceil \frac{c}{n} \right\rceil * t(e) \quad (4.17)$$

となる。

4.3 制作コストによる比較検討

コンテンツ制作の作業工程数と最短作業時間をもとに、各タイプの制作コストを比較し検証する。

4.3.1 コンテンツ制作最短作業時間の比較

コンテンツ制作標準作業時間の算出結果をもとに、各タイプの作業時間の短縮化をはかった最短作業時間を算出し比較する。

分類した3つのタイプにおいて、同一の講義内容を収録してコンテンツを制作するものとし、動画画像はビデオカメラで収録しながらエンコードをおこない、各種作業においては既製の自動化ソフトウェアの使用による作業時間の短縮化をはかるものと仮定する。

スライド切替タイミング記録の自動化ソフトウェアを使用すると収録時にその作業がおこなわれ、スライド切替表示タイミング記録処理時間 $t(str)$ は動画画像エンコード処理時間 $t(e)$ 内に作業が終了する。

$$t(str) < t(e) \quad (4.18)$$

スライドタイミング記録をもとにスライド切替表示記述ファイルを作成したうえで、動画画像とスライドとの同期再生表示記述ファイルを作成する自動化ソフトウェアの使用により、スライド切替表示記述ファイル作成処理時間 $t(pf)$ を省くことができる。

$$t(pf) = 0 \quad (4.19)$$

音声データのテキスト化をおこなうソフトウェアを使用すると、収録時に講師の口述内容のテキスト化の作業がおこなわれ、テキスト変換および切替表示タイミング記録処理時間 $t(txr)$ を短縮することができる。しかし、収録後、テキスト内容における間投詞などの不要な話し言葉の削除や、不適切な部分の修正をおこなう必要がある。よって、

$$t(txr) > t(e) \quad (4.20)$$

各タイプの同期再生表示記述ファイルの作成処理時間はほぼ等しいので、

$$t(sf_studio) \cong t(sf_ls) \cong t(sf_multi) \quad (4.21)$$

とする。

(スタジオ収録蓄積型最短作業時間)

スタジオ収録蓄積型作業時間 $T(Studio)$ は (4.1) 式より、

$$T(Studio) = t(sf_studio) + \max\{t(tf) + t(txr), t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

(4.18) (4.19) (4.20) 式より、

$$t(tf) + t(txr) > t(e) > t(pf) + t(str)$$

よって、スタジオ収録蓄積型最短作業時間 $Max_T(Studio)$ は、

$$Max_T(Studio) = t(sf_studio) + t(tf) + t(txr) \quad (4.22)$$

となる。

(ライブ収録加工蓄積型最短作業時間)

ライブ収録加工蓄積型作業時間 $T(LS)$ は (4.10) 式より、

$$T(LS) = t(sf_ls) + \max\{t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

(4.18) (4.19) 式より、

$$t(e) > t(pf) + t(str)$$

よって、ライブ収録加工蓄積型最短作業時間 $Max_T(LS)$ は、

$$Max_T(LS) = t(sf_ls) + t(e) \quad (4.23)$$

となる。

(ライブ多視点収録蓄積型最短作業時間)

ライブ多視点収録蓄積型標準作業時間 $T(Multi)$ は (4.17) 式より、

$$T(Multi) = t(sf_multi) + \left\lceil \frac{c}{n} \right\rceil * t(e)$$

収録するビデオカメラの台数 c にエンコーダの数 n が対応しているので、

$$c = n$$

より、

$$\left\lceil \frac{c}{n} \right\rceil * t(e) = t(e)$$

となる。

よって、ライブ多視点収録蓄積型最短作業時間 $Max_T(Multi)$ は、

$$Max_T(Multi) = t(sf_multi) + t(e) \quad (4.24)$$

となる。

・ 3つのタイプの最短作業時間の比較

(4.21)(4.22)(4.23)(4.24) 式より、

$$t(sf_studio) \cong t(sf_ls) \cong t(sf_multi)$$

$$Max_T(Studio) = t(sf_studio) + t(tf) + t(txr)$$

$$Max_T(LS) = t(sf_ls) + t(e)$$

$$Max_T(Multi) = t(sf_multi) + t(e)$$

よって、3つのタイプの最短作業時間の関係は、

$$t(sf_studio) + t(tf) + t(txr) > t(sf_ls) + t(e) \cong t(sf_multi) + t(e) \quad (4.25)$$

であるから、

$$Max_T(Studio) > Max_T(LS) \cong Max_T(Multi) \quad (4.26)$$

となる。

したがって、スタジオ収録蓄積型は、作業工程数および作業時間が他のタイプよりも大きくなり、一方、ライブ収録加工蓄積型とライブ多視点収録蓄積型は、作業工程数および作業時間が小さく抑えられ、ほぼ等しくなるといえる。

4.3.2 コンテンツ制作作業時間の典型値による比較

・典型値による標準作業時間

各制作工程の標準作業時間を以下のように設定し、各タイプの標準作業時間の典型値による比較をおこなう。

- $t(e)$: 動画像エンコード処理時間 = 講義収録時間 + 操作時間 = 5700 秒
- $t(sx)$: スライド画像変換処理時間 (20 枚) = 30 秒
- $t(str)$: スライド切替表示タイミング記録処理時間 = 5400 秒
- $t(pf)$: スライド切替表示記述ファイル作成処理時間 = 450 秒
- $t(txr)$: テキスト変換および切替表示タイミング記録処理時間 = 10800 秒
- $t(tf)$: テキスト切替表示記述ファイル作成処理時間 = 600 秒
- $t(sf_studio)$: 動画像・スライド・テキスト同期再生表示記述ファイル作成処理時間 = 300 秒
- $t(sf_ls)$: 動画像・スライド同期再生表示記述ファイル作成処理時間 = 300 秒
- $t(sf_multi)$: 複数の動画像の同期再生表示記述ファイル作成処理時間 = 300 秒

講義時間は 5400 秒 (90 分) とする。講義で使用するスライドは 20 枚用意される。スライドの表示可能な画像ファイルの変換作業は、ソフトウェアにより一括しておこなわれる。テキスト変換に関しては、講師が 1 秒間に 4 文字分話すと仮定した場合、5400 秒間では 21600 字となり、テキスト入力に要する時間が 1 秒間に 2 文字であるとき、テキスト変換には 10800 秒を必要とする。

(スタジオ収録蓄積型標準作業時間)

スタジオ収録蓄積型の作業時間 $T(Studio)$ は、(4.1) 式より、

$$T(Studio) = t(sf_studio) + \max\{t(tf) + t(txr), t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

であるから、スタジオ収録蓄積型の標準作業時間 $T(Studio)$ は、

$$T(Studio) = 300 + \max\{600 + 10800, 450 + \max\{5400, 30\}, 5700\} = 300 + 600 + 10800 = 11700$$

となる。

(ライブ収録加工蓄積型標準作業時間)

ライブ収録加工蓄積型の作業時間 $T(LS)$ は、(4.10) 式より、

$$T(LS) = t(sf_ls) + \max\{t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

であるから、ライブ収録加工蓄積型の標準作業時間 $T(LS)$ は、

$$T(LS) = 300 + \max\{450 + \max\{5400, 30\}, 5700\} = 300 + 450 + 5400 = 6150$$

となる。

(ライブ多視点収録蓄積型標準作業時間)

ライブ多視点収録蓄積型の標準作業時間 $T(Multi)$ は、(4.17) 式より、

$$T(Multi) = t(sf_multi) + \left\lceil \frac{c}{n} \right\rceil * t(e)$$

であるから、

ビデオカメラ 3 台、エンコーダ 1 台 ($c = 3, n = 1$) の場合、

$$T(Multi) = 300 + \left\lceil \frac{3}{1} \right\rceil * 5700 = 17400$$

ビデオカメラ 3 台、エンコーダ 2 台 ($c = 3, n = 2$) の場合、

$$T(Multi) = 300 + \left\lceil \frac{3}{2} \right\rceil * 5700 = 11700$$

ビデオカメラ 3 台、エンコーダ 3 台 ($c = 3, n = 3$) の場合、

$$T(Multi) = 300 + \left\lceil \frac{3}{3} \right\rceil * 5700 = 6000$$

となる。

・典型値による最短作業時間

各制作工程において自動化ソフトウェアなどの使用による短縮化をはかった作業時間、すなわち最短作業時間を以下のように設定し、各タイプの最短作業時間の典型値による比較をおこなう。

- $t(e)$ ：動画像エンコード処理時間（講義を収録しながらエンコードする）＝ 5400 秒
- $t(sx)$ ：スライド画像変換処理時間（20 枚）＝ 30 秒

- $t(str)$: スライド切替表示タイミング記録処理時間 = 5000 秒
- $t(pf)$: スライド切替表示記述ファイル作成処理時間 = 0 秒
- $t(txr)$: テキスト変換 (音声データのテキスト化) および切替表示タイミング記録処理時間 = 8000 秒
- $t(tf)$: テキスト切替表示記述ファイル作成処理時間 = 600 秒
- $t(sf_studio)$: 動画像・スライド・テキスト同期再生表示記述ファイル作成処理時間 = 300 秒
- $t(sf_ls)$: 動画像・スライド同期再生表示記述ファイル作成処理時間 = 300 秒
- $t(sf_multi)$: 複数の動画像の同期再生表示記述ファイル作成処理時間 = 300 秒

(スタジオ収録蓄積型最短作業時間)

スタジオ収録蓄積型の作業時間 $T(Studio)$ は、(4.1) 式より、

$$T(Studio) = t(sf_studio) + \max\{t(tf) + t(txr), t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

であるから、スタジオ収録蓄積型の最短作業時間 $Max_T(Studio)$ は、

$$Max_T(Studio) = 300 + \max\{600 + 8000, 0 + \max\{5000, 30\}, 5400\} = 300 + 600 + 8000 = 8900 \quad (4.27)$$

となる。

(ライブ収録加工蓄積型最短作業時間)

ライブ収録加工蓄積型の作業時間 $T(LS)$ は、(4.10) 式より、

$$T(LS) = t(sf_ls) + \max\{t(pf) + \max\{t(str), t(sx)\}, t(e)\}$$

であるから、ライブ収録加工蓄積型の最短作業時間 $Max_T(LS)$ は、

$$Max_T(LS) = 300 + \max\{0 + \max\{5000, 30\}, 5400\} = 300 + 5400 = 5700 \quad (4.28)$$

となる。

(ライブ多視点収録蓄積型最短作業時間)

ライブ多視点収録蓄積型の標準作業時間 $T(Multi)$ は、(4.17) 式より、

$$T(Multi) = t(sf_multi) + \left\lceil \frac{c}{n} \right\rceil * t(e)$$

であるから、ビデオカメラ 3 台、エンコーダ 3 台 ($c = 3, n = 3$) の場合、ライブ多視点収録蓄積型の最短作業時間 $Max_T(Multi)$ は、

$$Max_T(Multi) = 300 + \left\lceil \frac{3}{3} \right\rceil * 5400 = 5700 \quad (4.29)$$

となる。

	標準作業時間 (秒)	最短作業時間 (秒)
スタジオ収録蓄積型	11700	8900
ライブ収録加工蓄積型	6150	5700
ライブ多視点収録蓄積型 (カメラ：3, エンコーダ：3)	6000	5700
(カメラ：3, エンコーダ：2)	11700	
(カメラ：3, エンコーダ：1)	17400	

表 4.2: コンテンツ制作作業時間の典型値による比較

・ 3つのタイプの最短作業時間の比較

(4.27) (4.28) (4.29) 式より、3つのタイプの最短作業時間の関係は、

$$Max_T(Studio) > Max_T(LS) = Max_T(Multi) \quad (4.30)$$

となる。

4.3.3 制作コストの比較

コンテンツ制作の作業工程数および最短作業時間を用いて、各タイプの制作コストを検討する。

スタジオ収録蓄積型はオンデマンド配信用に講義を企画・制作するため、ライブ収録よりも作業工程数、作業時間が多く、制作コストは最も大きくなる。仮に講師の口述内容を

テキスト変換するソフトウェアを用いたとしても、変換されたテキストの人間による検査作業がなお必要であると考えられる。

コンテンツ制作の自動化ソフトウェアを使用することによって、ライブ収録加工蓄積型とライブ多視点収録蓄積型の最短作業時間はほぼ等しくなった。ただし、ライブ収録加工蓄積型の最短作業時間は、講師映像とスライドとの同期再生表示の場合のみを前提にしている。

ライブ収録加工蓄積型は通常おこなわれている講義や講演を収録する分、スタジオ収録蓄積型よりも制作コストが小さくなる。しかし、講義の状況によっては、プレゼンテーションソフトウェアのスライド以外の要素、例えば板書や OHP などを使用する場合もあり、それに対応した加工作業をおこなう必要が生じる。したがって、講義の要素が増えれば、それに応じて加工するための作業工程数、作業時間が追加されていくことになり、講義内容あるいは講師に依存して制作コストが大きくなっていく。

ライブ多視点収録蓄積型は講義の要素をすべて撮影対象として収録するため、あらゆる講義内容に対応することができ、しかも作業工程数が最も少ない。また、複数台のビデオカメラとそれに対応する数のエンコーダがあれば並行作業により、最短作業時間を講義時間に近づけることが可能となる。

したがって、ライブ多視点収録蓄積型すなわち実験コンテンツの「提案型」は、複数のビデオカメラを用いて講義をライブ収録することによって、あらゆる講義内容に柔軟に対応し、制作コストを小さく抑え、ライブ講義収録後最も速く蓄積したコンテンツを配信することができると考えられる。

第 5 章

複数の動画像を視聴するためのユーザインタフェース

この章では、エンドユーザの要求に対応した複数の動画像を視聴するためのユーザインタフェースとして、エンドユーザが選択した動画像のみを配信するためのフォームと、クライアントシステム環境の画面サイズの制約に対応したレイアウトのリアルタイム表示切替を提案する。

5.1 視点別動画像選択方式

オンデマンド配信を利用するエンドユーザは、特定の動画像を繰り返し視聴することが考えられる。例えば、板書の内容をもう一度視聴したい場合には、板書の映像のみを要求するであろう。

複数のビデオカメラを用いて収録したさまざまな視点からの講義映像の静止画像を表示しておき、クライアントが視聴したい視点からの動画像を選択させるユーザインタフェースのプロトタイプを図 5.1 に示す。

この視点別動画像選択方式を利用することによって、エンドユーザは視聴したい映像のみの配信を受けることが可能となる。

その他、クライアントが複数の動画像を受信するにあたって、使用する通信回線の帯域幅による受信可能な動画像配信数の制約を受ける場合がある。限られた帯域幅しか利用できず、複数の動画像を受信することが制限される環境では、クライアントのシステム環境に応じてエンドユーザが動画像を選択できるとよい。

この場合にも視点別動画像選択方式を利用することによって、配信を受ける前にあらか

じめエンドユーザが使用する帯域幅を考慮して、視聴したい動画像を選択することが可能となる。



図 5.1: 視点別動画像選択フォーム

5.2 レイアウトのリアルタイム表示切替方式

複数の動画像をクライアントが視聴する際、動画像のレイアウト表示がクライアントシステムのディスプレイ画面サイズに制約を受ける場合がある。表示される動画像の数は画面領域による制約を受けるため、クライアントが複数の動画像をディスプレイ画面上に効果的に配置し、講義の状況に応じてリアルタイムにレイアウトを変更できることが求められる。

例えば、何も書かれていない板書の映像やシートの置かれていない OHP スクリーン映像など、講義の状況によってはクライアントにとって不要な動画像が発生する。そこで、講義の進行に従ってクライアントがレイアウトを適宜切り替えて、視聴したい映像を変更

していくことができるようなユーザインタフェースの提供によって、限られた画面スペースの有効活用をはかることができる。

SMIL のような同期やレイアウトの概念を含むコンテンツ配信記述言語に、ICML (Intermedia Casting Markup Language)[29] がある。

ICML は、動画およびそれに同期して表示される要素を規定するための規則の集合である。その規則は XML の考え方に準じており、マークアップ言語によって表現されている。この記述言語には、動画およびその動画との同期によるデータの表示、レイアウトの変更やシナリオによる管理などの機能がある。

SMIL と異なる特徴として ICML では、コンテンツ配信にあたって選択可能な複数のレイアウトをあらかじめ用意しておき、受信側でリアルタイムに視聴したい動画像を自由に選択することができる。また、HTML をエレメントとして表示できるため、JavaScript や CGI などによって機能を加えることができる。

この ICML と既存のスクリプト言語との比較を表 5.1[29] に示す。なお、BML (Broadcast Markup Language) は社団法人電波産業会 (ARIB: Association of Radio Industries and Businesses) によって策定され、BS デジタル放送 (衛星放送) で使用されている XML ベースのスクリプト言語であり、従来の動画と音声による放送に双方向性やインターネットとのリンクを提供するための技術である。

	ICML	SMIL	BML
マルチシンクロ	○	○	○
マルチレイアウト	○	△	△
マルチシナリオ	○	×	×
オーサリングの容易性	○	△	△
メンテナンスの容易性	○	×	×
物理的記録メディアによる配布	○	○	×
オンデマンド配信	○	○	×
ライブ配信	○	×	×
HTML ファイルの制御	○	×	×
各種エンジンとの親和性	○	△	×
配信の容易性	○	○	×

表 5.1: ICML と既存のスクリプト言語との比較

ICML の機能を利用して、例えば「講師映像とスライドスクリーン映像」の表示から「講師映像と板書映像」の表示へ、というように講義の進行状況に応じたリアルタイム表示切替が可能となる。これによって、クライアントシステムの限られたディスプレイ画面領域サイズにおける複数の動画像の表示の問題が解決できる。

ICML における必要な構成要素として、同期のもととなる動画を”MainMovie タグ”で指定し、その動画とそれ以外の表示するタイプのエレメントを”Elements タグ”、それらの表示レイアウトを”Layouts タグ”、シナリオを”Scenarios タグ”で設定する。

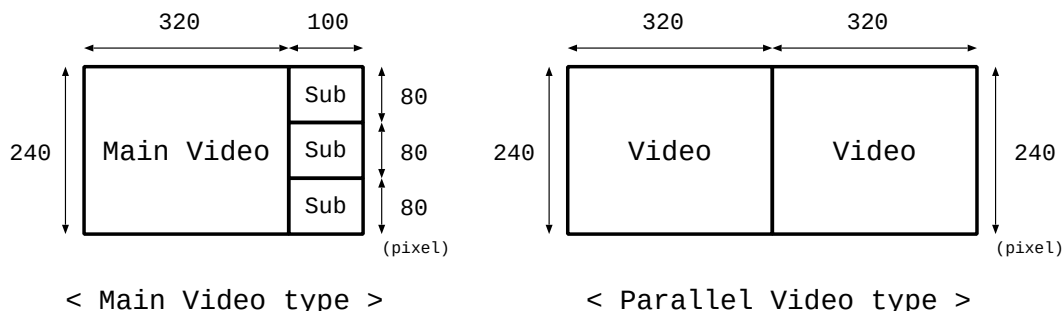


図 5.2: プロトタイプのレイアウト

ICML によるレイアウトのリアルタイム表示切替のプロトタイプを図 5.3 と図 5.4 に示す。このプロトタイプでは、図 5.2 のように 1 つの動画像をメインに表示するタイプと 2 つの動画像を並列に表示するタイプの 2 種類のレイアウトを用意している。そのレイアウトに、講師、スライドスクリーン、板書、OHP スクリーンの 4 種類の動画像を配置するパターンのシナリオを設定している。以下にそのプロトタイプのコードの一部を示す。

```
<!-- 同期のもととなる動画の指定：MainMovie タグ -->
<MainMovie>lecturer.avi</MainMovie>

<!-- 表示する要素の指定：Elements タグ -->
<Elements>
  <Element Name="MainMovie" Type="Movie" Main="True" Border="0" />
  <Element Name="Movie02" Type="Image" Border="0" />
  <Element Name="Movie03" Type="Image" Border="0" />
  <Element Name="Movie04" Type="Image" Border="0" />
</Elements>

<!-- 表示する要素の配置の指定：Layouts タグ -->
```

```

<Layouts>
  <Layout Name="講師メイン" StretchType="Center" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >
    <LayoutElement Target="MainMovie" Left="0" Top="0" Width="320"
      Height="240" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
      Height="20" />
    <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
      Height="20" />
    <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
      Height="20" />
    <LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
      Height="20" />
    <LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
      Height="20" />
    <LayoutElement Target="Movie02" Left="320" Top="0" Width="100"
      Height="80" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="Movie03" Left="320" Top="80" Width="100"
      Height="80" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="Movie04" Left="320" Top="160" Width="100"
      Height="80" StretchType="Position" BGColor="00FF00" />
  </Layout>
</Layouts>

```

<!-- 複数のシナリオを定義し、ユーザに複数の表示を提供：Scenarios タグ -->

```

<Scenarios>
  <Scenario Name="講師中心" />
  <Scenario Name="スライド中心" />
  <Scenario Name="板書中心" />
  <Scenario Name="OHP中心" />
  <Scenario Name="講師・スライド" />
  <Scenario Name="講師・板書" />
  <Scenario Name="講師・OHP" />
</Scenarios>

```

<!-- シナリオにおいて同期する要素の指定：Sequences タグ -->

```

<Sequences>
  <Sequence Time="0">
    <CommandSet Scenario="講師中心">

```

```
<Action Target="Movie02" Command="slide.avi" />
<Action Target="Movie03" Command="board.avi" />
<Action Target="Movie04" Command="ohp.avi" />
<Action Target="講師メイン" Command="Change" />
</CommandSet>
</Sequence>
</Sequences>
```

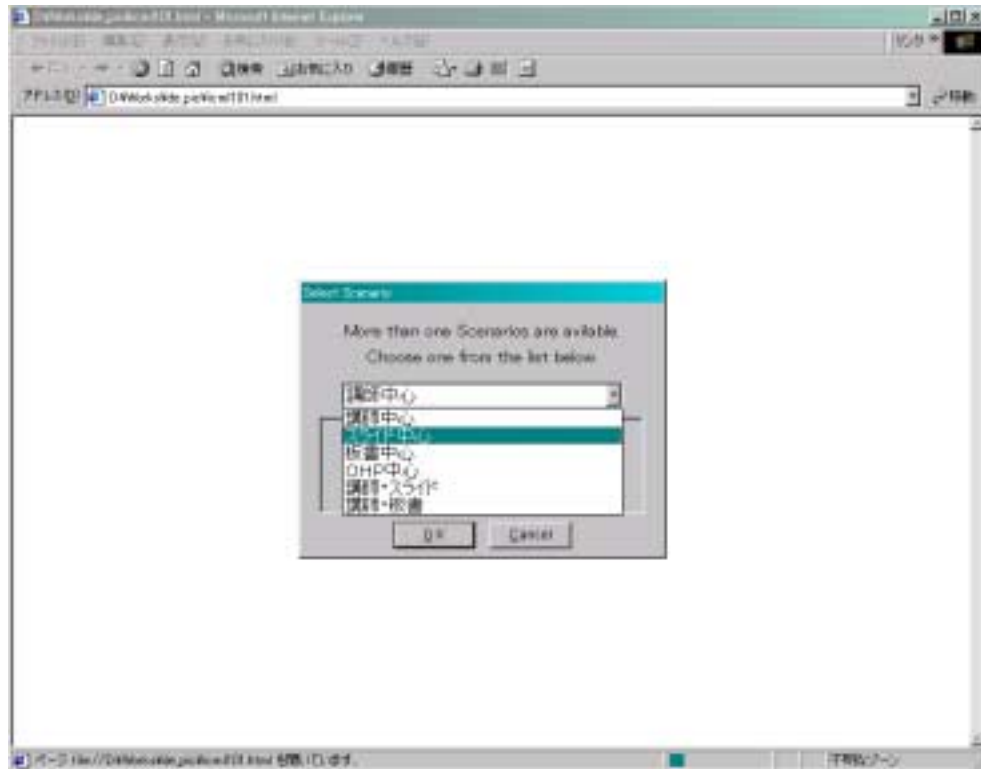


図 5.3: レイアウト選択フォーム



図 5.4: レイアウトのリアルタイム表示切替リストボックス

第 6 章

サーバ間の負荷分散とトラフィック分散

この章では、複数の高品質かつ大容量コンテンツをクライアントに安定して配信するための技術として、サーバ間の負荷分散とトラフィック分散、コンテンツ配信ネットワークに関して述べる。

6.1 負荷分散配置と制御

高品質動画像のような大容量コンテンツの配信においては、サーバへのトラフィック集中による輻輳を回避するために、複数のサーバにコンテンツの分散配置をおこない、クライアントからのアクセスを分散する構成をとっている場合が多い。

サーバの負荷分散をおこなうことによって、クライアントへのレスポンスタイムを高速化すると同時に、クライアントへの安定した配信が期待できる。

6.1.1 複数のサーバへのコンテンツの分散配置

コンテンツのサーバへの配置方法として、オリジンサーバのミラーリングやキャッシュサーバの設置によるものがおこなわれている。

- オリジンサーバのミラーリング（図 6.1）

オリジンサーバのミラーリングによるコンテンツの配置では、オリジンサーバのすべてのコンテンツと同一の複製としてミラーサーバを構築する。

ミラーリングにおいて、オリジナルのコンテンツが新たに生成されたときや更新されたときには、オリジンサーバとミラーサーバとの一貫性を保つための更新作業が

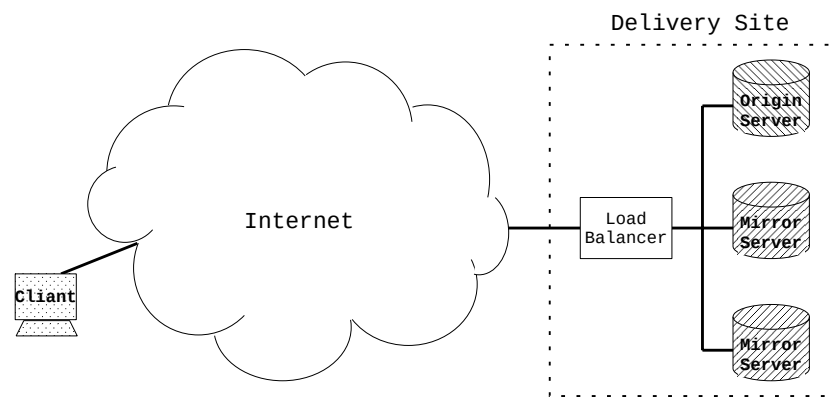


図 6.1: 配信サイト内のオリジンサーバのミラーリング

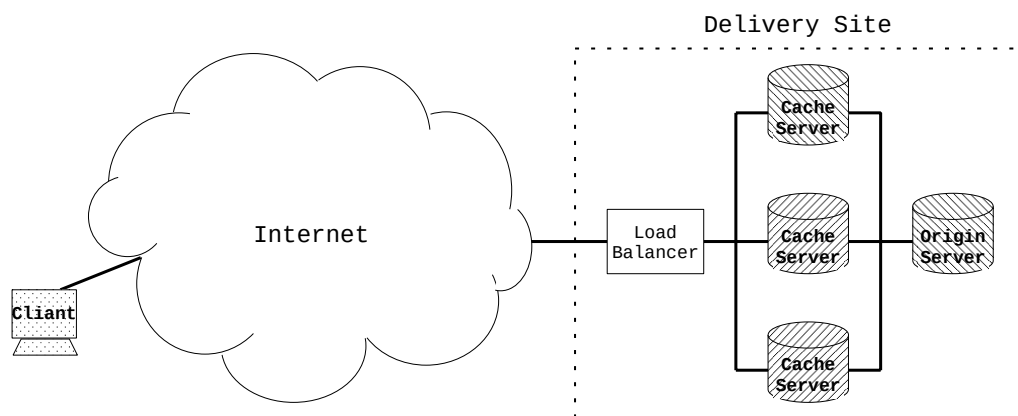


図 6.2: 配信サイト内のキャッシュサーバの配置

必要となる。そのうえ、場合によってはクライアントから要求のないコンテンツも保持しなければならないことになる。

- キャッシュサーバの設置（図 6.2）

キャッシュサーバの設置によるコンテンツの配置では、クライアントからの要求はまずキャッシュサーバが受け、キャッシュに要求されたコンテンツが存在すれば、それをクライアントへ返す。

キャッシュに要求コンテンツが存在しなければ、オリジンサーバへ要求し、コンテンツを受け、クライアントへ送り、それと同時にキャッシュサーバはそのコンテンツをキャッシュとして保持する。

サーバのファイル容量が一定数に達した場合には、クライアントからの要求の少ないコンテンツから削除されていくことになる。

図 6.2 のキャッシュサーバの配置例では、キャッシュサーバがボトルネックにならないように、オリジンサーバの前段に複数のキャッシュサーバを配置している。

6.1.2 負荷分散制御

複数のサーバにコンテンツの分散配置がおこなわれたならば、サーバの負荷分散とトラフィック分散の効果を充分なものとするために、クライアントからのアクセスに対するサーバへの適切な振り分け処理が必要となる。

- ローカル制御

ローカル制御は、配信サイト内においてオリジンサーバのミラーサーバへのアクセスを振り分けて負荷分散する。ローカル制御には、一般的にロード・バランサ（負荷分散制御装置）が用いられている。

クライアントからのアクセスに対するサーバへの振り分け方法には、順番に処理を振り分けていくラウンドロビン方式をはじめ、クライアントごとに振り分け先サーバを固定する方式、サーバ間に振り分け順位や重み付けをおこなう方式、各サーバの負荷状況に応じて動的に振り分ける方式、[30] などがある。

- グローバル制御

広域ネットワークでは、地理的に分散配置されたサーバ間へのアクセスを振り分け、トラフィックを分散させるグローバル制御が要求される。その代表的な制御手法とし

ては、インターネットでの名前解決に用いられている DNS(Domain Name System) への参照をリダイレクトする方式がある。この方式では、DNS サーバが備えるラウンドロビン選択機能を利用して、ユーザの IP アドレスやホップ数をもとに最寄りのサーバの IP アドレスを返信することで振り分けをおこなう。

6.2 広域ネットワークにおける分散配置とオンデマンド配信

ストリーミングでは帯域幅を使うだけでなく、クライアントとサーバとが常に接続しているため、ダウンロードとは異なり、多くのセグメント、ルータ、スイッチを介した複雑なネットワーク経路における安定したパケット配信が要求される。つまり、ストリーミングでは、HTML コンテンツよりも大きな帯域を使用することと、データが中断しないように継続して安定供給することを考慮しなければならない。

それには、すべてのエンドユーザに対して安定した品質を提供できるような高速ネットワークとエンドユーザまでスムーズにデータを配信する仕組みが必要となる。

6.2.1 コンテンツ配信ネットワーク

コンテンツ配信ネットワーク (CDN: Contents Delivery Network) は、複数のキャッシュ・サーバをインターネットとは別の独立した網内に分散させて設置し、分散配置したシステムにコンテンツの複製を配置し、エンドユーザに対して最寄りのサイトからコンテンツを提供する。これによって利用者からのアクセスを振り分けることができ、1つのサイトに集中するトラフィックを分散させると同時に、コンテンツ配信サーバからエンドユーザまでの伝送距離を短くすることで配信の高速化が期待できる。(図 6.3)

CDN が提供する機能には、以下のようなものがある。

- リダイレクト・サービス
エンドユーザからのリクエストを最寄りのキャッシュサーバに送る
- ディストリビューション・サービス
オリジンサーバの代わりにコンテンツをキャッシュサーバへ複製あるいは分散させる
- アカウンティング・サービス
コンテンツの利用を処理、測定、記録する

コンテンツ配信サービスプロバイダ (CDSP: Contents Delivery Service Provider) の大規模な事業者は、一般にキャッシュサーバやミラーサーバを世界中に配置し、エンド

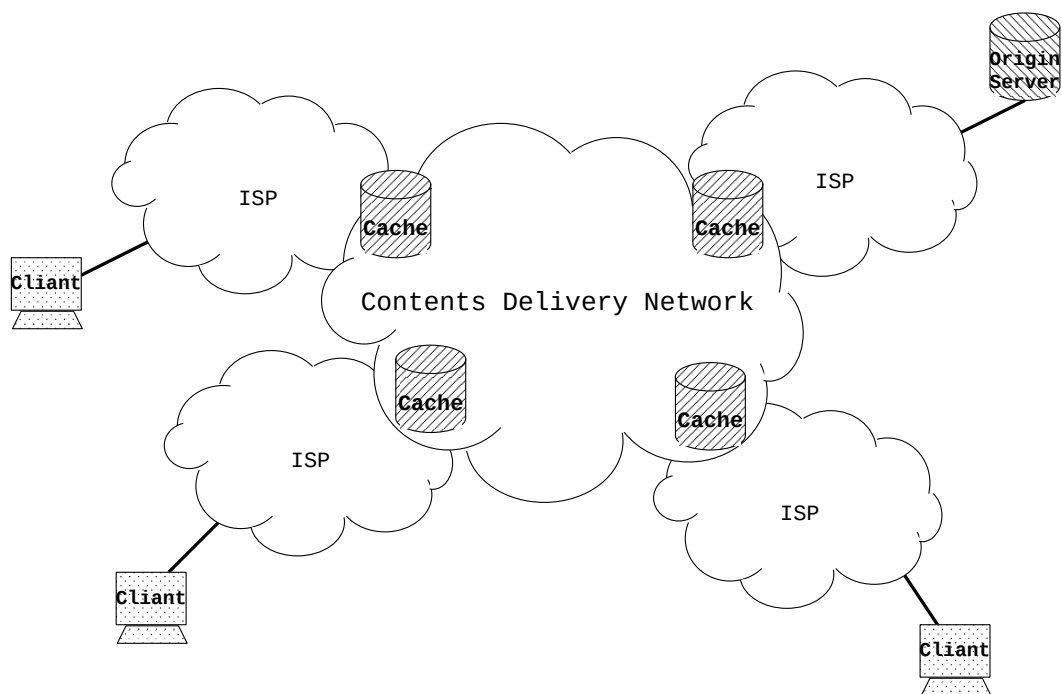


図 6.3: コンテンツ配信ネットワーク

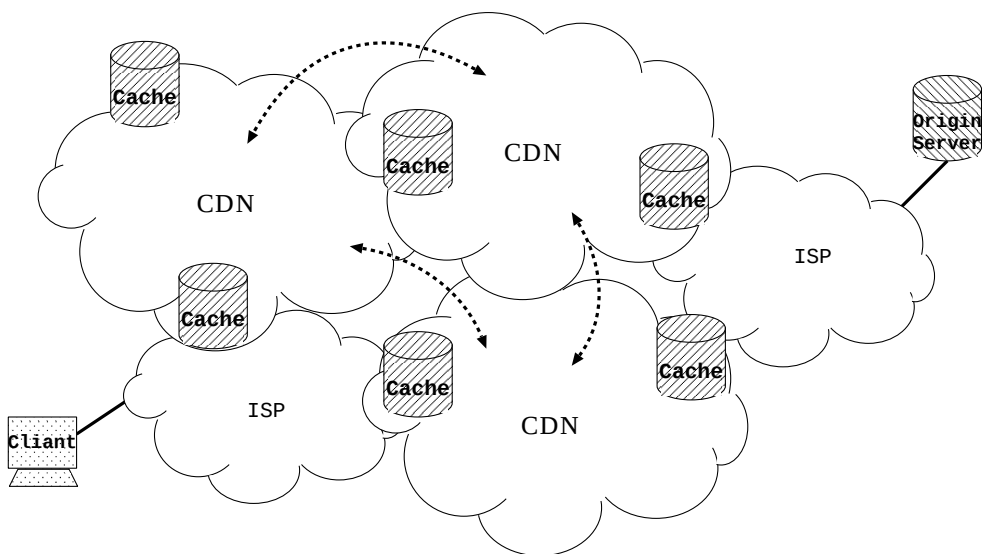


図 6.4: CDN ピアリング

ユーザの近くからコンテンツ配信サービスを提供している。このような大規模な事業者には”Akamai”、”Digital Island”、”mirror image”、”adero”、”SPEEDERA” などがある。国内における事業者には、不特定多数のユーザを対象にコンテンツを配信する”J-Stream”、”Accelia” などや、CATV、xDSL、FTTH 加入者などの特定ユーザを対象にコンテンツを配信する”AII”、”有線ブロードバンド” などがある。

これら複数の CDN 間を相互接続することによって、エンドユーザに幅広いサービスの提供をはかることを CDN ピアリングという。(図 6.4)

CDN ピアリングには、エンドユーザへ配信する CDN の選択をおこなうリクエスト・ルーティングや、ピアリングしている CDN 間のコンテンツ・ディストリビューション、そして、ピアリングしている CDN 間の課金決済のアカウンティング制御がある。

6.2.2 プロキシによるストリーミングのサポート

ストリーミングは、大きな帯域の使用とサーバ／クライアント間の安定した接続という課題をもっている。以下では、サーバとクライアントシステムの間でプロキシを用いて、クライアントへのストリーミングデータの接続をサポートする基本的な配信方式を挙げる。

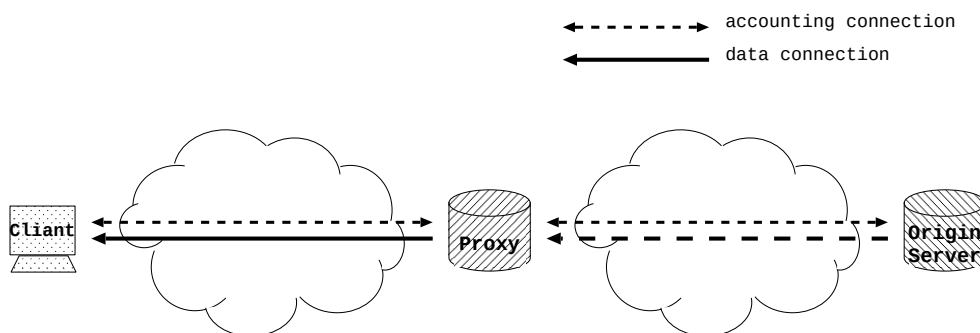


図 6.5: キャッシングのコネクション

- オンデマンド配信におけるプロキシの機能

- － キャッシング

- プロキシにキャッシュが装備されており、最初のリクエストでコンテンツのキャッシングをおこなう。

- その後、キャッシュに格納されたコンテンツのリクエストであるかどうかを確認

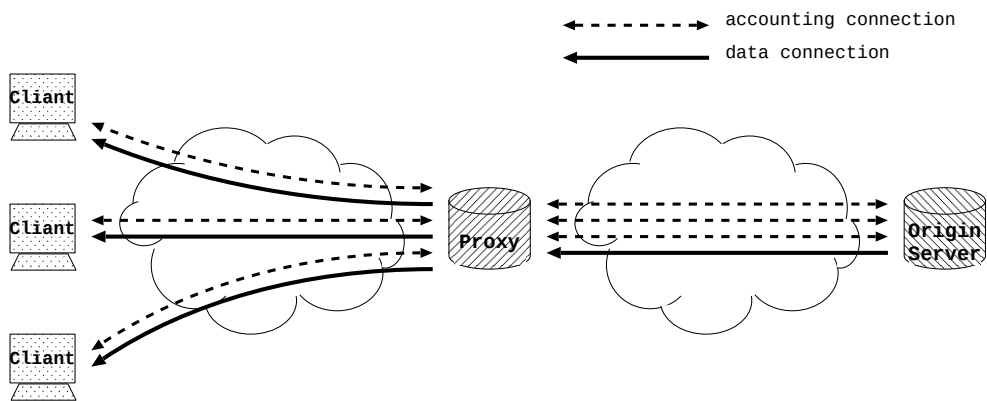


図 6.6: スプリッティングのコネクション

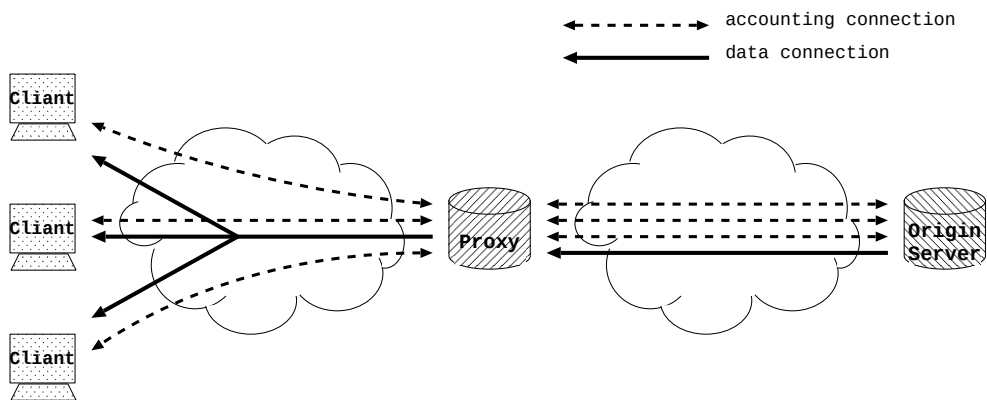


図 6.7: マルチキャストのコネクション

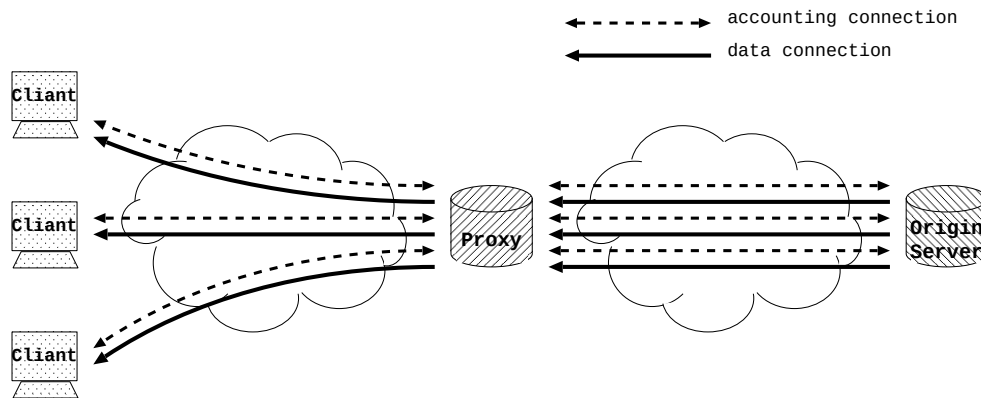


図 6.8: パススルーのコネクション

認し、キャッシュにない場合はオリジンサーバからコンテンツを入手すると同時にクライアントに配信する。

キャッシングのコネクションを図 6.5 に示す。

- ライブ配信におけるプロキシの機能

- － スプリッティング

プロキシがオリジンサーバに単一のデータ・チャンネルを接続した後、プロキシに接続されるクライアントへライブ・ブロードキャストを分流する。

他のクライアントから同じライブストリームの要求を受けた場合、プロキシから配信されることになる。

スプリッティングのコネクションを図 6.6 に示す。

- － マルチキャスト

プロキシとクライアント間、およびクライアントシステムが IP マルチキャストに対応している場合、プロキシはクライアントへライブストリームを再放送することができる。

マルチキャストのコネクションを図 6.7 に示す。

- － パススルー

ライブコンテンツを分流できない場合、プロキシはクライアントとの接続を確立した状態でクライアントへストリームを転送する。

パススルーのコネクションを図 6.8 に示す。

6.2.3 複数のストリームのオンデマンド配信

複数の高品質かつ大容量の動画コンテンツをオンデマンド配信する場合、送信元がオリジンサーバである必要性はない。したがって、オリジンサーバから配信するのではなく、エンドユーザに最寄りのキャッシュサーバからコンテンツを配信した方が、レスポンスは速くなり、品質は良くなり、帯域使用は削減する。

エンドユーザに安定してデータを提供するために、広域ネットワークに分散配置されたキャッシュサーバを利用してサーバのデータ転送負荷を軽減し、できるだけクライアントと配信サーバとの伝送距離が短くなるように、エンドユーザに最寄りのキャッシュサーバから複数のストリームを配信する手法が効果的である。

オンデマンド配信の特徴を利用して、エンドユーザに最寄りのキャッシュサーバやミラーサーバにあらかじめコンテンツの事前配置をおこなうことも可能である。

第 7 章

複数のストリームのライブ配信における同期再生の課題と中間同期化ノード

この章では、広域ネットワークのライブ配信におけるマルチビデオストリームの同期再生の課題を述べ、その課題解決として中間同期化ノードを提案する。

7.1 複数のストリームのライブ配信における課題

オンデマンド配信では、送信元はオリジンサーバである必要性がないため、エンドユーザに最寄りのサーバからの配信を利用することができた。

一方、ライブ配信では、送信元はオリジンサーバでなければならないという制約がある。このため、広域ネットワークにおいては、サーバとクライアントとの間で多くのセグメント、ルータ、スイッチを介する複雑なネットワーク経路上をストリームデータが配信されることになる。

複数のストリームを伝送するライブ配信では、複数の送信元サーバからクライアントまでを常に接続していなければならない。このとき、図 7.1 のように広域ネットワークにおいては、複数の送信元サーバからの各ストリームデータが異なる伝送経路を通過してクライアントに到着する場合がある。あるいは図 7.2 のように映像は衛星回線、音声は地上回線というように、配信するコンテンツの特性によって異なる伝送手段を使用する場合がある。

複数のストリームが異なる送信元サーバから配信され、なおかつ各ストリームデータが異なる伝送経路を通過してクライアントに到着するため、クライアントシステムにおいて複数のストリーム間の同期再生が保証されないという問題が生じる。

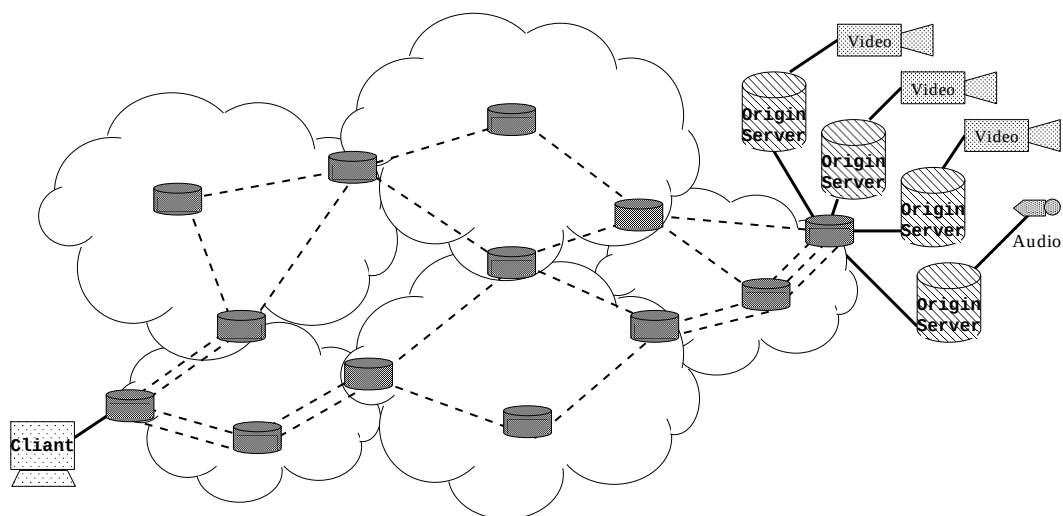


図 7.1: 異なる伝送経路を通る複数のライブストリーム

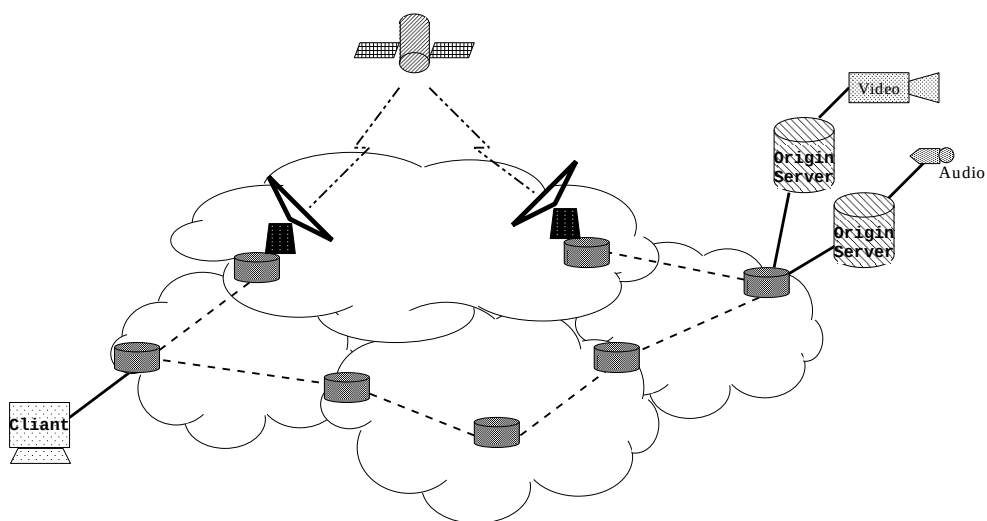


図 7.2: 異なる伝送手段を使用する複数のライブストリーム

すなわち、マルチビデオストリームを伝送するライブ配信では、安定したストリーミングの実現と同時に複数のストリーム間の同期再生の実現が課題となる。

7.2 複数のストリームの同期再生

複数のストリームの同期再生のメカニズムは、クライアントシステムに実装されている。配信される各ストリームデータの packets ヘッダには、送信元の識別子と配信サーバのクロックに基づいてタイムスタンプがつけられている。受信側であるクライアントシステムがバッファリングによって遅延やジッタを吸収しながら、データパケットのヘッダ情報から送信元を識別し、送信元が同一であるデータとの間の同期再生を実現している。

したがって、複数のサーバから配信される異なる送信元からのストリームデータの同期再生をクライアントシステムが容易に実現できるようにするために、複数のサーバとクライアントとの間に中継ノードを配置する必要がある。この中継ノードは、各ストリームデータの送信元識別子を単一送信元サーバからのストリームデータとして付け替えて配信することによって、クライアントシステムにおける同期再生を実現する。

7.3 中間同期化ノード

中継ノードは複数のサーバからのストリームを収束化し、この中継ノードが単一送信元として複数のストリームをクライアントへ配信することから、「中間同期化ノード (i-sync node : intermediate synchronization node)」としてクライアントに最寄りの網内に配置する。中間同期化ノードの配置イメージを図 7.3 に示す。

中間同期化ノードは、ストリームデータの RTP、RTCP パケットヘッダの送信元識別情報をもとに複数のサーバとの間でセッション処理をおこない、クライアントは、中間同期化ノードを単一送信元として複数のストリームデータ間の同期再生を実現する。

中間同期化ノードを配置した場合の動作フローとして図 7.4 に示す。クライアントから配信制御サーバへリクエストが送信されると、配信制御サーバは中間同期化ノードの ID をクライアントへ通知する。クライアントは中間同期化ノードへリクエストを送信し、それを受けて中間同期化ノードは各配信サーバへ配信リクエストを送信し、各配信サーバからのストリームデータをクライアントへ中継する。クライアントから切断要求を受けた場合、中間同期化ノードはクライアントとのセッションを閉じ、各配信サーバに切断要求を送信する。

実際に UDP の上位に RTP を使ってストリームデータを配信する場合、セッション制

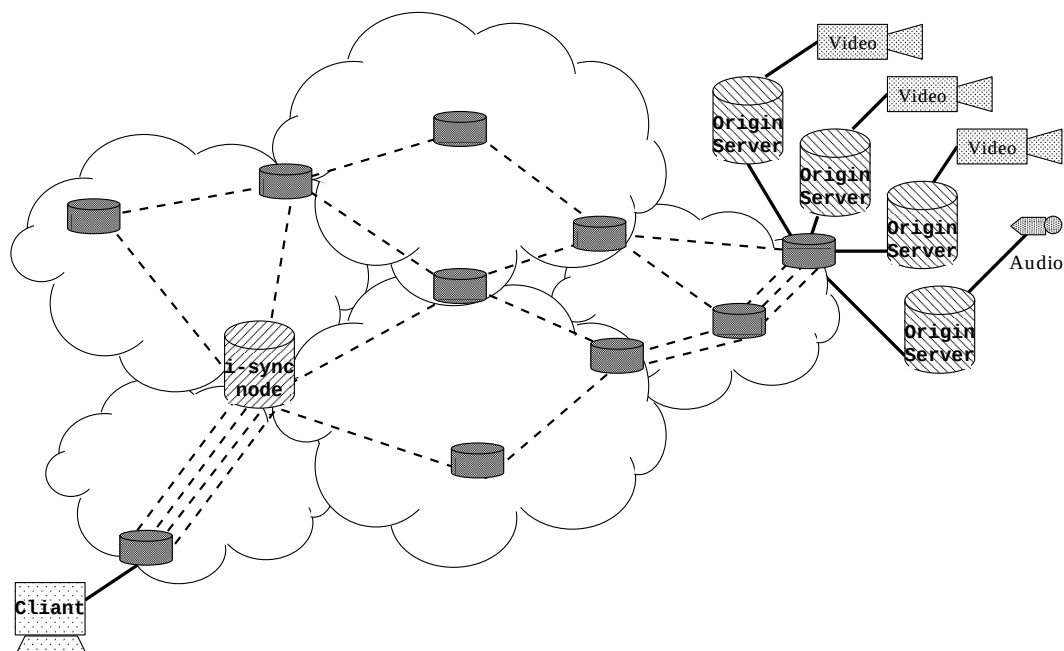


図 7.3: 中間同期化ノードの配置イメージ

御に関する TCP チャンネル、実際のデータを配信する RTP/UDP チャンネル、データ伝送に関する情報のやりとりをおこなう RTCP/UDP チャンネル、これら 3 つのチャンネルが各サーバと中間同期化ノードとの間で図 7.5 のようにセットアップされる。

中間同期化ノードは、複数のサーバからストリームを受けてクライアントに転送するプロキシサーバ上にアプリケーションとして構築する。

中間同期化ノードは、データの RTP、RTCP パケットヘッダの送信元識別情報をもとに同期処理をおこなう。実装にあたっては、複数のサーバと中間同期化ノードとの間で伝送される各サーバにおける RTP ヘッダのタイムスタンプ、シーケンス番号、RTCP ヘッダの Sender/Receiver Reports、これらの情報の各サーバに対するマッピングが必要となる。

中間同期化ノードと複数の各サーバとの RTSP コネクションにおいても、セッション ID はタイムスタンプとシーケンス番号に基づいているため、各サーバに対するセッション ID のマッピングが必要となる。

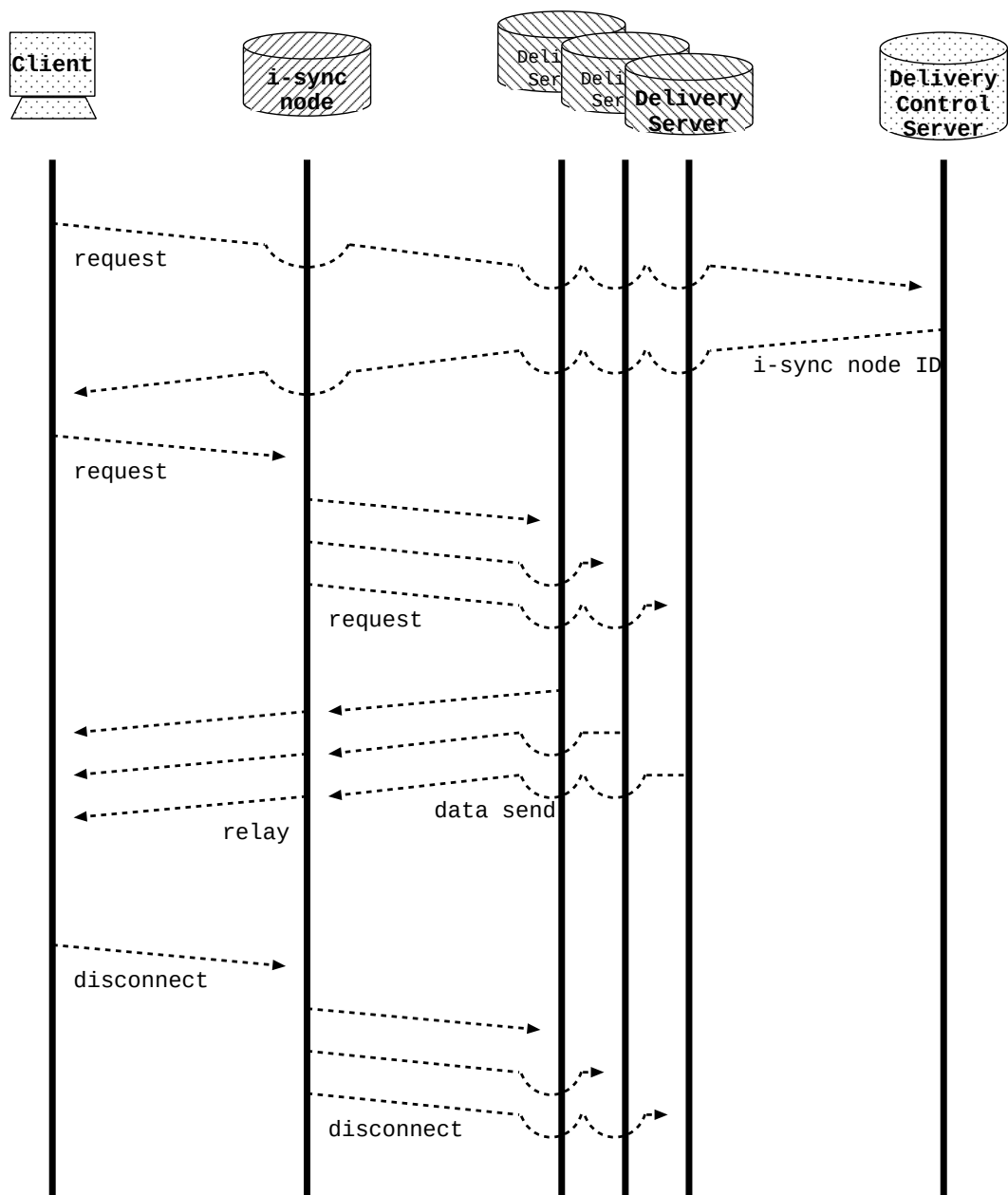


図 7.4: 中間同期化ノードを配置した場合の配信動作フロー

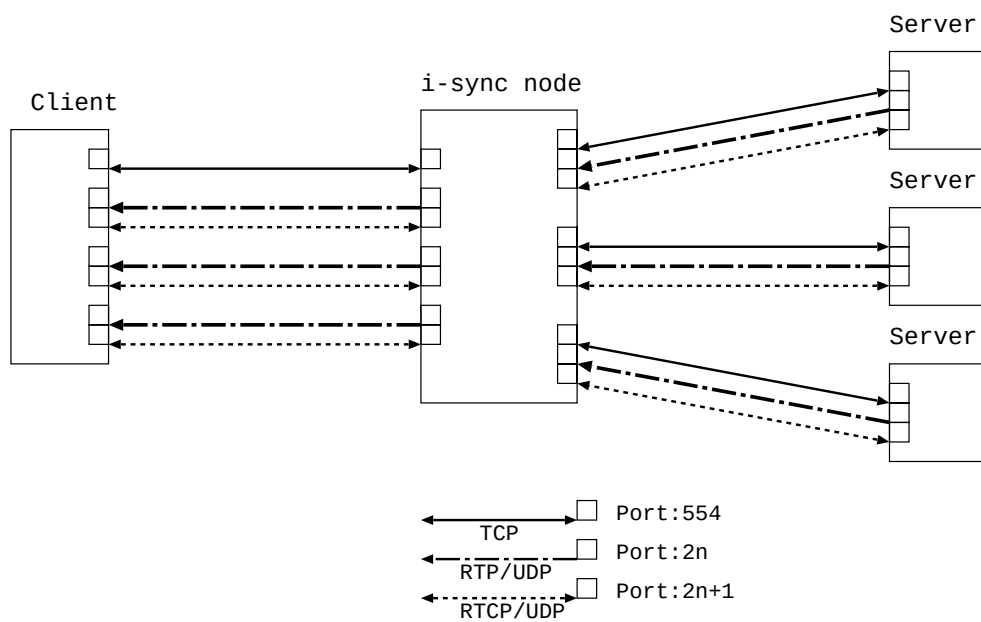


図 7.5: 中間同期化ノードを配置した場合のチャネルのセットアップ

第 8 章

今後の課題

8.1 マルチビデオストリームを用いた表示形態による学習効果の検証

今回の比較評価実験では、表示形態による視聴者への影響について評価したが、学習システムとして、マルチビデオストリームを用いた表示形態がどのような学習効果をもたらすのかを検証することも重要である。

学習効果を評価する例として、受講生を2つのグループに分け、講師映像とプレゼンテーションソフトウェアのスライドとの同期再生表示である「従来型」と、複数の動画画像の同期再生表示である「提案型」をそれぞれグループ別に視聴した後、講義内容に関する理解度をチェックするペーパーテストをおこなう。つぎに、視聴していないもう一方の表示形態を視聴した後で、理解度をチェックするペーパーテストをおこない、それぞれのテストの結果から学習効果の比較を評価する方法が考えられる。

このペーパーテストを用いた比較評価によって、コンテンツ表示形態が学習効果に及ぼす影響を検討する。

8.2 ユーザインタフェースの検討

複数の動画画像を視聴するためのユーザインタフェースの課題として、提案した視点別動画画像選択方式とレイアウトのリアルタイム表示切替方式の有効性を明らかにするために評価実験をおこない、問題点を検討することが挙げられる。

視点別動画画像選択方式は、ユーザが特定の映像を繰り返し視聴したい場合に利用される

ビデオライブラリの配信サービスとして提供できると考えられる。

レイアウトのリアルタイム表示切替方式は、クライアントシステムの画面領域サイズを考慮に入れ、講義の進行に応じてリアルタイムにユーザが視聴したい映像を切り替えていくことを可能にした。

ただし、レイアウトのリアルタイム表示切替のプロトタイプは、ICML を用いて記述したが、ユーザが選択可能なレイアウトをあらかじめ用意しておく必要があり、配信されるストリームの数に応じて表示される映像レイアウトの組合せが増加し、制作コストが大きくなってしまう。

配信サイトに動画像のデータベースを構築し、ユーザからリクエストされた特定のストリームを配信することによって、視点別動画像選択方式とレイアウトのリアルタイム表示切替方式の両方の機能を実現することが可能となる。しかし、広域ネットワークにおいてキャッシュサーバを用いたオンデマンド配信の場合には、データベースの利用ができなくなるという課題が残る。

これらの問題点をふまえ、複数の動画像を視聴するためのユーザインタフェースの検討が必要である。

8.3 ライブ配信における中間同期化ノードの実装と動作検証

ライブ配信における複数のストリーム間の同期再生の実現を容易にするために提案した中間同期化ノードの実装と動作検証をおこなう必要がある。

複数のサーバからのストリームデータの識別をおこない、クライアントへ配信するアプリケーションをプロキシサーバ上に実装し、複数のサーバとクライアントとの間で中間同期化ノードとして動作検証する。

第 9 章

おわりに

本研究では、提案する複数の動画像を配信するオンデマンド学習システムについて、表示形態の比較評価実験の結果とコンテンツ制作コストの比較によって有効性を検証し、実装上の課題を検討した。

まず、提案する複数の動画像を用いた同期再生表示（「提案型」）の有効性を明らかにするために、一般的に用いられている講師映像とスライドによる同期再生表示（「従来型」）との比較評価実験をおこなった。

この表示形態の比較評価の結果、総合的評価にあたる平均値では「提案型」が「従来型」よりも有意に高い評価が得られた。評価項目を因子分析したところ、「画質・音声」「臨場感」「視覚効果」「内容」の4因子が抽出された。因子成績を比較すると、「画質・音声」については「従来型」が「提案型」よりも有意に高く、「臨場感」「視覚効果」については「提案型」が「従来型」よりも有意に高い評価が得られ、「内容」については両者の間に有意差は認められなかった。よって、オンデマンド配信による複数の動画像を用いた表示形態は、臨場感や視覚的な表示効果の面での有効性をもつことが明らかになった。

コンテンツ制作コストによる表示形態の比較をおこない、ライブ講義を複数のビデオカメラで収録してオンデマンド配信する形態が有利であることを検証した。

以上のように評価実験と制作コストの検証から、複数のビデオストリームのオンデマンド配信の有効性を明らかにしたうえで、つぎに実装上の課題を挙げた。一つは、配信される動画像は高品質かつ大容量コンテンツであるため、エンドユーザへのストリームデータの安定した提供方法が必要である。もう一つは、複数の動画像の表示可能な数はクライアントシステムのディスプレイ画面領域サイズに制約を受けるため、それに対応するユーザインタフェースが必要である。

複数の動画像の表示については、エンドユーザの要求に対応した複数の動画像を視聴す

るためのユーザインタフェースとして、エンドユーザが選択した動画像のみを配信するためのフォームと、クライアントシステムのディスプレイ画面領域サイズの制約に対応したレイアウトのリアルタイム表示切替を提案した。

高品質かつ大容量コンテンツ配信については、オンデマンド配信における送信元サーバはオリジンサーバである必要性がないことから、コンテンツ配信ネットワークのサービスのようなエンドユーザに最寄りのキャッシュサーバを利用する手法を検討した。

ライブ配信では、送信元サーバからクライアントまでを常に接続していなければならないため、広域ネットワークにおいて複数のストリーム間の同期再生の実現が課題となる。この課題の解決として、中間同期化ノードを提案した。中間同期化ノードは、複数のサーバとクライアントとの間の網内に位置し、複数のサーバからのストリームを受取り、単一送信元としてクライアントへ転送する。クライアントにとって、最寄りの中間同期化ノードから複数のストリームが配信されることになり、複数のストリーム間の同期再生を容易にする。

今後の課題として、マルチビデオストリームを用いた表示形態による学習効果の検証、複数の動画像を視聴するためのユーザインタフェースの検討、ライブ配信における中間同期化ノードの実装と動作検証を挙げた。

謝辞

本研究において、指導教官である丹康雄先生には終始御指導して頂き、心から深く感謝申し上げます。

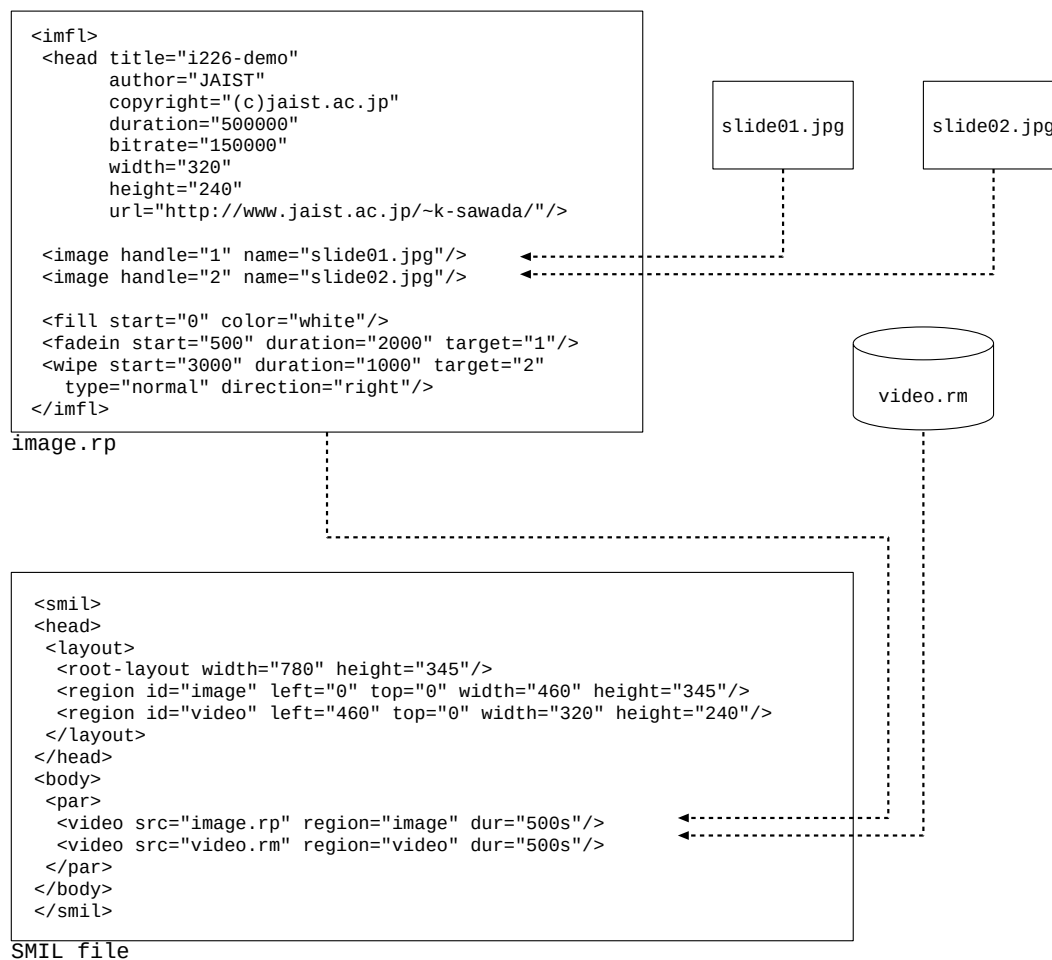
また、丹研究室のOB諸先輩方をはじめとし、皆様方には貴重な意見や助言をして頂き、厚く御礼申し上げます。

最後に、私の研究を影で支えてくれた家族、友人に感謝します。

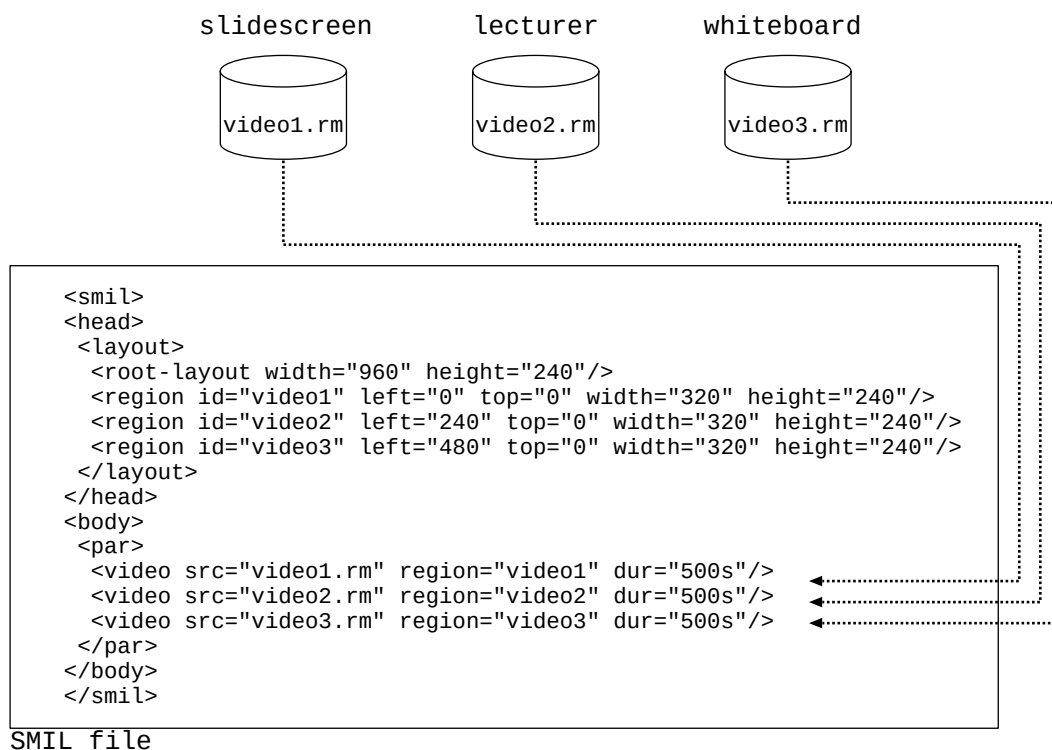
付録 A

評価実験で用いた表示形態の記述ファイルの構成

1. 講師映像とスライドとの同期再生表示（「従来型」）



2. 複数の動画の同期再生表示（「提案型」）



付録B

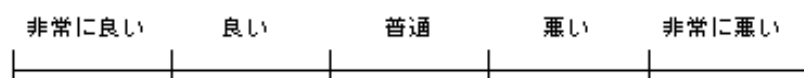
実験における同一質問紙による主観評価の手順

1. 質問項目の選考

実験の目的がオンデマンド配信における表示形態の比較評価であることから、講義の内容に関すること、画像・音声に関すること、臨場感に関すること、オンデマンド学習に関すること、遠隔教育に関すること、総合評価に関すること、これらを中心に質問項目を選考した。

2. 評価尺度

CCIR(ITU-R)で推奨された5段階品質尺度をもとにした評価用語「非常に良い」「良い」「普通」「悪い」「非常に悪い」の5つのカテゴリーを用いた。図のように、評価用紙の評価記入用罫線（横線）は長さ100mmとし、20mm間隔にカテゴリーを配置した。



3. 回答方法

評価者がアンケートの各項目に最もふさわしいと判断した罫線上に黒丸印を記入させた。例えば、カテゴリー「普通」と判断した場合でも、比較的「良い」と感じたならば罫線左方に、比較的「悪い」と感じたならば右方に記すように指示した。

なお、実際に評価者に与えた指示は次のとおりである。

「アンケートの各項目に最もふさわしいと判断した線上の位置に黒丸 ”●” で示して下さい。記入する位置は線上であればどこでも構いません。例えば、「普通」と判断した場合でも、比較的良いと感じた場合は左の方に、比較的悪いと感じた場合は右の方に記入して下さい。評価にあたっては、既に採点した評点に左右されることなく、それぞれ感じたまに採点して下さい。」

4. 集計

各質問項目の評定罫線上において、横線右端から評定者が記入した印までの距離をその質問項目の評点とした。

5. 統計分析

5.1. 平均値の差の検定 (t-検定)

2つのグループに対する評価の違いを明らかにするために、アンケート項目ごとの平均値を求め、両者の平均値に対して t-検定をおこなった。

t-検定は、両者の平均値の差が母集団においても差として認められるか推測する検定であり、間隔あるいは比率尺度から得られた量的データにおいて、2つの標本平均間 ($\bar{x}$) の相違が母平均間 (μ) においても相違として認められるか推測する方法である。その手順は以下のとおりである。

1. 帰無仮説の設定

A 群の母平均と B 群の母平均には差がない、または A 群の母平均と B 群の母平均は等しいと仮定する

2. 有意水準 α の設定

3. 共通分散 V の算出

2つの母分散は等しいと仮定した場合、 $\sigma_A^2 = \sigma_B^2 = \sigma^2$ と表現でき、 σ^2 は未知であるから、V を σ^2 の推定値として用いる

$$V = \frac{S_A + S_B}{\phi_A + \phi_B}$$

S_A = A 群の平方和、 S_B = B 群の平方和

n_A = A 群のデータ数、 n_B = B 群のデータ数

$\phi_A = n_A - 1$ $\phi_B = n_B - 1$

4. 検定統計量 t 値の算出

$$t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{V(\frac{1}{n_A} + \frac{1}{n_B})}}$$

5. p 値の算出

有意水準と比較する確率 p 値は、自由度 $\phi_A + \phi_B$ の t 分布において、 $|t|$ 以上の値が発生する確率である

6. 判定

p 値 $\leq$ 有意水準 $\alpha \rightarrow$ 帰無仮説を棄却する

p 値 $>$ 有意水準 $\alpha \rightarrow$ 帰無仮説を棄却しない

5.2. 因子分析

アンケートの質問項目の回答を分類し、変数間の関係を成り立たせている因子を抽出して、主観評価の評価観点を明確にするために、変数と因子との関係から変数間の関係を説明する因子分析をおこなった。その手順は以下のとおりである。

1. 共通性の推定値

変数の分散のうち因子によって説明される部分の比率を共通性という。因子分析にさきだち共通性の推定値を与える必要があり、共通性の推定値を決める代表的な手法には、すべての変数について 1 を推定値にする主成分分析と、重相関係数の 2 乗を推定値にする SMC 法がある。

2. 因子の数

抽出する因子の数が多ければ多いほど、変数間の相関関係を適切に説明できるはずであるが、あまり多くの因子を抽出したのでは、情報を圧縮したことにはならないため、因子の数は分析結果に重大な影響を及ぼすことに留意しなければならない。因子の数の決定には、固有値と累積寄与率が目安として使われる。

3. 因子軸の回転

変数との関係から因子の意味を解釈していくための作業が因子軸の回転である。軸の回転する手法には、抽出した因子相互に相関関係がないようにして軸の回転をおこなう直交回転と、因子間に相関関係が存在することを認める斜交回転がある。これらの手法で最も一般的なものは、直交回転の VARIMAX 回転である。

4. 因子分析の結果

因子分析の結果、抽出された因子を独立変数、各変数を従属変数とする重回帰分析の形として表される。変数と因子との関係の強さは、因子負荷量として表される。

5. 因子の解釈と命名

ある因子に大きな因子負荷をもつ変数はその因子の特徴を反映し、その変数はその因子に所属していると考えられる。したがって、所属する変数から因子の解釈と命名をおこなう。

5.3. 相関分析

各因子成績の因子間相互の関連をみるために、各因子成績間の内部相関係数を求めた。相関係数は一方の増加に伴って他方が増加または減少していく関係の強さを表しており、その値は $-1 \sim +1$ をとる。

5.4. 回帰分析

回帰分析は、ある1つの変数の値を別の変数の値を使って、予測、制御したい場合に利用される。予測したい変数のことを目的変数と呼び、予測に使う変数のことを説明変数と呼ぶ。制御の場合には目的変数が結果として制御される変数で、説明変数が直接制御する変数である。

回帰分析には、説明変数が1つである単回帰分析と説明変数が2つ以上である重回帰分析がある。

評価実験では、各因子がどのように総合的評価に影響を与えているかを検討するために単回帰分析をおこない、総合的評価に各因子の与える影響の度合いをみるために重回帰分析をおこなった。

付録C

ICMLによるレイアウトのリアルタイム表示切替のコード例

```
<html>
<head>
<SCRIPT LANGUAGE="ICML">
<!--

<title>ICML in HTML</title>
<MainMovie>lecturer.avi</MainMovie>
<Authentication Name="SAWADA Kenji" Address="Tan-Lab,JAIST"
    EMail="k-sawada@jaist.ac.jp" />

<Elements>
    <Element Name="MainMovie" Type="Movie" Main="True" Border="0" />
    <Element Name="Movie02" Type="Image" Border="0" />
    <Element Name="Movie03" Type="Image" Border="0" />
    <Element Name="Movie04" Type="Image" Border="0" />
</Elements>

<Layouts>
    <Layout Name="講師メイン" StretchType="Center" Left="0" Top="0"
        Width="640" Height="260" BGColor="ffffff" >
        <LayoutElement Target="MainMovie" Left="0" Top="0" Width="320"
            Height="240" StretchType="Position" BGColor="00FF00" />
        <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
            Height="20" />
        <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
            Height="20" />
        <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
```



```

        Height="20" />
<LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
    Height="20" />
<LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
    Height="20" />
<LayoutElement Target="Movie02" Left="320" Top="0" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
<LayoutElement Target="Movie03" Left="320" Top="80" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
<LayoutElement Target="Movie04" Left="320" Top="160" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
</Layout>

<Layout Name="スライドメイン" StretchType="Center" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >
    <LayoutElement Target="Movie02" Left="0" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
        Height="20" />
    <LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
        Height="20" />
    <LayoutElement Target="MainMovie" Left="320" Top="0" Width="100"
        Height="80" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="Movie03" Left="320" Top="80" Width="100"
        Height="80" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="Movie04" Left="320" Top="160" Width="100"
        Height="80" StretchType="Position" BGColor="00FF00" />
</Layout>

<Layout Name="板書メイン" StretchType="Center" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >
    <LayoutElement Target="Movie03" Left="0" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />

```

```

<LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
  Height="20" />
<LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
  Height="20" />
<LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
  Height="20" />
<LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
  Height="20" />
<LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
  Height="20" />
<LayoutElement Target="MainMovie" Left="320" Top="0" Width="100"
  Height="80" StretchType="Position" BGColor="00FF00" />
<LayoutElement Target="Movie02" Left="320" Top="80" Width="100"
  Height="80" StretchType="Position" BGColor="00FF00" />
<LayoutElement Target="Movie04" Left="320" Top="160" Width="100"
  Height="80" StretchType="Position" BGColor="00FF00" />
</Layout>

<Layout Name="OHPメイン" StretchType="Position" Left="0" Top="0"
  Width="640" Height="260" BGColor="ffffff" >
  <LayoutElement Target="Movie04" Left="0" Top="0" Width="320"
    Height="240" StretchType="Position" BGColor="00FF00" />
  <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
    Height="20" />
  <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
    Height="20" />
  <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
    Height="20" />
  <LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
    Height="20" />
  <LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
    Height="20" />
  <LayoutElement Target="MainMovie" Left="320" Top="0" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
  <LayoutElement Target="Movie02" Left="320" Top="80" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
  <LayoutElement Target="Movie03" Left="320" Top="160" Width="100"
    Height="80" StretchType="Position" BGColor="00FF00" />
</Layout>

```

```

<Layout Name="講師スライド並" StretchType="Position" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >
    <LayoutElement Target="MainMovie" Left="0" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
        Height="20" />
    <LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
        Height="20" />
    <LayoutElement Target="Movie02" Left="320" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />
</Layout>

```

```

<Layout Name="講師板書並" StretchType="Position" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >
    <LayoutElement Target="MainMovie" Left="0" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />
    <LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
        Height="20" />
    <LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
        Height="20" />
    <LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
        Height="20" />
    <LayoutElement Target="Movie03" Left="320" Top="0" Width="320"
        Height="240" StretchType="Position" BGColor="00FF00" />
</Layout>

```

```

<Layout Name="講師OHP並" StretchType="Position" Left="0" Top="0"
    Width="640" Height="260" BGColor="ffffff" >

```

```

<LayoutElement Target="MainMovie" Left="0" Top="0" Width="320"
    Height="240" StretchType="Position" BGColor="00FF00" />
<LayoutElement Target="PlayButton" Left="0" Top="240" Width="20"
    Height="20" />
<LayoutElement Target="PauseButton" Left="20" Top="240" Width="20"
    Height="20" />
<LayoutElement Target="StopButton" Left="40" Top="240" Width="20"
    Height="20" />
<LayoutElement Target="PositionGuage" Left="60" Top="240" Width="260"
    Height="20" />
<LayoutElement Target="ScenarioBox" Left="320" Top="240" Width="100"
    Height="20" />
<LayoutElement Target="Movie04" Left="320" Top="0" Width="320"
    Height="240" StretchType="Position" BGColor="00FF00" />
</Layout>

</Layouts>

<Scenarios>
    <Scenario Name="講師中心" />
    <Scenario Name="スライド中心" />
    <Scenario Name="板書中心" />
    <Scenario Name="OHP中心" />
    <Scenario Name="講師・スライド" />
    <Scenario Name="講師・板書" />
    <Scenario Name="講師・OHP" />
</Scenarios>

<Sequences>
    <Sequence Time="0">
        <CommandSet Scenario="講師中心">
            <Action Target="Movie02" Command="slide.avi" />
            <Action Target="Movie03" Command="board.avi" />
            <Action Target="Movie04" Command="ohp.avi" />
            <Action Target="講師メイン" Command="Change" />
        </CommandSet>
        <CommandSet Scenario="スライド中心">
            <Action Target="Movie02" Command="slide.avi" />
            <Action Target="Movie03" Command="board.avi" />

```

```

        <Action Target="Movie04" Command="ohp.avi" />
        <Action Target="スライドメイン" Command="Change" />
    </CommandSet>
    <CommandSet Scenario="板書中心">
        <Action Target="Movie02" Command="slide.avi" />
        <Action Target="Movie03" Command="board.avi" />
        <Action Target="Movie04" Command="ohp.avi" />
        <Action Target="板書メイン" Command="Change" />
    </CommandSet>
    <CommandSet Scenario="OHP中心">
        <Action Target="Movie02" Command="slide.avi" />
        <Action Target="Movie03" Command="board.avi" />
        <Action Target="Movie04" Command="ohp.avi" />
        <Action Target="OHPメイン" Command="Change" />
    </CommandSet>
    <CommandSet Scenario="講師・スライド">
        <Action Target="Movie02" Command="slide.avi" />
        <Action Target="講師スライド並" Command="Change" />
    </CommandSet>
    <CommandSet Scenario="講師・板書">
        <Action Target="Movie03" Command="board.avi" />
        <Action Target="講師板書並" Command="Change" />
    </CommandSet>
    <CommandSet Scenario="講師・OHP">
        <Action Target="Movie04" Command="ohp.avi" />
        <Action Target="講師OHP並" Command="Change" />
    </CommandSet>
</Sequence>
</Sequences>

//-->
</SCRIPT>
</head>

<body bgcolor="ffffff">
<center>
    <OBJECT width = 640 height = 260
        classid="clsid:E439637A-CB77-11D4-B05C-00B0D03168C1"
        codebase="http://icml.lope.co.jp/public/icml/

```

```
cab/ICMLXControl.cab#version=1,1,0,0">  
  <PARAM NAME="FileName" value="Self">  
    <PARAM NAME="AutoStart" value="True">  
  </OBJECT>  
</center>  
</body>  
</html>
```

参考文献

- [1] RTP: A Transport Protocol for Real-Time Applications, RFC1889, January, 1996
- [2] Real Time Streaming Protocol (RTSP), RFC2326, April, 1998
- [3] D.Wu, Y.T.Hou, W.Zhu, Y-Q.Zhang, J.M.Peha, “Streaming Video over the Internet: Approaches and Directions”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Vol.11, No.1, pp.1-20, February, 2001
- [4] M.Cheshire, A.Wolman, G.Voelker, and H.Levy, “Measurement and Analysis of a Streaming-Media Workload”, In Proc. The Usenix Symposium on Internet Technologies and Systems (USITS), San Francisco, CA, USA, March 2001.
- [5] Synchronized Multimedia Integration Language (SMIL),
<http://www.w3.org/>
- [6] 大久保榮, 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, 2001
- [7] STANFORD UNIVERSITY Stanford Center for Professional Development (SCPD),
<http://scpd.stanford.edu/>
- [8] 高津直己, ”スタンフォード大学におけるメディアの活用”, NIME Newsletter, No.20, pp.8, 2000
- [9] 大川恵子, 伊集院百合, 村井純, ”School of Internet – インターネット上での「インターネット学科」の構築”, 情報処理学会論文誌, Vol.40, No.10, pp.3801-3810, 1999.10
- [10] WIDE University, School of Internet,
<http://www.soi.wide.ad.jp/>

- [11] LearningSite21, NTT LEARNING SYSTEMS CORPORATION,
<http://www.learningsite21.com/>
- [12] 大原 WAK² Web, O-HARA, NTT-ME CORPORATION, NTT-X,
<http://www10.wakwak.com/o-hara/>
- [13] Waseda University Learning Square Corporation,
<http://www.wls.co.jp/>
- [14] 大川恵子, 泉山英孝, 加藤朗, 村井純, “次世代インターネットを利用した高等教育環境の構築実験 GIOS プロジェクト”, 情報処理学会誌, Vol.42, No.1, pp.51-57, 2001.1
- [15] 登 弘聡, “複数のビデオストリームを用いた遠隔教育システムに関する研究”, 北陸先端科学技術大学院大学修士論文, 2001
- [16] 結城皖曠, “遠隔教育への高速ネットワークの活用”, NIME Newsletter, No.20, pp.6-7, 2000
- [17] RealNetworks,Inc. , “REALSERVER 管理ガイド RealServer8.0”, 2000,
<http://www.jp.realnetworks.com/>
- [18] RealNetworks,Inc. , “REALSYSTEM SERVER 8 での RTSP 相互運用性”, 2001.3,
<http://www.jp.realnetworks.com/>
- [19] RealNetworks,Inc. , “REALSYSTEM プロダクションガイド”, 2000,
<http://www.jp.realnetworks.com/>
- [20] RealNetworks,Inc. , “REALPIX オーサリングガイド”, 2001.3,
<http://www.jp.realnetworks.com/>
- [21] RealNetworks,Inc. , “REALTEXT オーサリングガイド”, 2001.2,
<http://www.jp.realnetworks.com/>
- [22] 宇井修, 中山実, 清水康敬, “衛星通信講座における講義形態と学習者評価の関係”, 電子情報通信学会論文誌, D-II, Vol.J80-D-II, No.4, pp.892-899, 1997.4
- [23] 村上正行, 八木啓介, 角所考, 美濃導彦, “受講経験・日米受講習慣の影響に注目した遠隔講義システムの評価要因分析”, 電子情報通信学会論文誌, D-I, Vol.J84-D-I, No.9, pp.1421-1430, 2001.9

- [24] 近藤喜美夫, "大学間教育交流ネットワーク SCS の開発 ", メディア教育研究, No.7, pp.1-27, 2001
- [25] 成田長人, "主観評価で用いる日本語評価用語の知覚間隔の検証 ", 電子情報通信学会論文誌, D-II, Vol.J75-D-II, No.11, pp.1968-1974, 1992.11
- [26] 内田治, "すぐわかる EXCEL による統計解析 ", 東京図書, 1996
- [27] 木下栄蔵, "わかりやすい数学モデルによる多変量解析入門 ", 近代科学社, 1995
- [28] 古谷野亘, "数学が苦手な人のための多変量解析ガイド 調査データのまとめかた ", 川島書店, 1988
- [29] Intermedia Casting Markup Language (ICML), LOPE CORPORATION,
<http://icml.lope.co.jp/>
- [30] 巳波弘佳, 熊谷和則, 能上慎也, 阿部威郎, "負荷分散制御アルゴリズムの性能評価と適用領域 ", 電子情報通信学会技術研究報告, NS2001-3, Vol.101, No.8, pp.15-20, 2001.4
- [31] 下川俊彦, 吉田紀彦, 牛島和夫, "多様な選択ポリシーを利用可能なサーバ選択機構 ", 電子情報通信学会論文誌, D-I, Vol.J84-D-I, No.9, pp.1396-1403, 2001.9
- [32] S.Gruber, J.Rexford, A.Basso, "Protocol Considerations for a Prefix-Caching Proxy for Multimedia Streams", In Proc. 9th International World Wide Web Conference, Amsterdam (2000).
- [33] R.Brussee, H.Eertink, W.Huijsen, B.Hulsebosch, M.Rougoor, W.Teeuw, M.Wibbels, H.Zandbelt, "Content Distribution Networks ", Telematica Instituut, 2001
- [34] B.Reeves, C.Nass, 細馬宏通 訳, "人はなぜコンピュータを人間として扱うか 「メディアの等式」の心理学 ", 翔泳社, 2001
- [35] 澤田憲志, 丹康雄, "マルチビデオストリームを用いたオンライン学習システムの構築 ", 平成 13 年度 電気関係学会北陸支部連合大会 講演論文集, C-55, pp.231, 2001