

## 内容概要

現在、急速にブロードバンドといった高速なインターネットを安価に利用できる時代になってきた。これまでのテキスト・画像に加え、音声や動画が扱われるようになってきた。それと同時にリアルタイム性が要求され、ストリーミング技術の必要性は更に高まっている。しかし、文字や静止画といった比較的軽いものならともかく、動画や音声をスムーズにストリーミング配信するには、その劣化が避けられない。また、配信できる容量にも限界がある。そこで同期マルチメディアの技術を用い、その欠落した情報を補足し、さらに分かりやすいコンテンツにすること、ストリーミング配信の実験考察を行うことが本研究のテーマである。

本研究では同期マルチメディアを用いて並列処理という難しい概念を説明するコンテンツを作成した。並列処理を分かりやすく説明するモデルとして、相撲モデルを CG で作成した。さらに、その相撲モデルを使用し、キーワードアニメーションを作成した。しかし、アニメーションではすべて伝えきれないため、そのアニメーションにテキストや静止画を同期させ、補助説明をする。よって視聴者に伝えることのできる情報が増え、視聴者の理解度を深めることが可能になることが分かった。次にさまざまな環境への対応を行い、ストリーミング実験として、完成したコンテンツをストリーミングサーバからの配信と Web サーバから先行ダウンロード方式での配信を同時に 20 人見た場合の比較を行い、考察した。

目次：

1	はじめに.....	1
2	同期マルチメディアとストリーミング.....	2
2.1	ストリーミングとは.....	2
2.2	同期マルチメディアとは.....	3
2.3	同期マルチメディア言語：SMIL.....	3
3	同期マルチメディアを用いた並列処理教材コンテンツ.....	5
3.1	コンテンツの構成.....	5
3.2	並列処理のキーワード.....	6
3.3	各メディアの同期方法.....	6
4	相撲モデルによる並列処理教材の作成.....	8
4.1	作成の流れ.....	8
4.2	3D Studio Max における CG とアニメーションの作成.....	9
4.2.1	3D Studio Max とは.....	9
4.2.2	相撲 CG の作成.....	10
4.2.3	相撲アニメーションの作成.....	11
4.3	静止画の作成.....	11
4.4	音声の作成.....	12
4.5	Premiere におけるアニメーションの作成.....	12
5	キーワードアニメーションの詳細と評価.....	14
5.1	並列処理とは.....	14
5.2	負荷均衡.....	17
5.3	スケーラビリティ.....	20
5.4	評価.....	23
6	ストリーミング配信実験と考察.....	24
6.1	ストリーミング配信実験.....	24
6.2	結果と考察.....	25
7	おわりに.....	27
	謝辞.....	28
	参考文献.....	29
	付録 SMIL ソースファイル.....	30

図目次：

図 1：ストリーミングの流れ.....	2
図 2：SMIL の概要.....	4
図 3：コンテンツの構成図.....	5
図 4：コンテンツの完成図の例（並列処理とは）.....	7
図 5：RealText による時間指定.....	7
図 6：相撲モデルによる並列処理教材の作成の流れ.....	8
図 7：3D Studio Max のインターフェース.....	9
図 8：力士.....	10
図 9：土俵.....	10
図 10：マテリアルエディタ.....	10
図 11：Adobe Photoshop の画面.....	11
図 12：午後のこ～だ for windows.....	12
図 13：Premiere の画面.....	13
図 14：ストリーミング形式のファイルを作成中の画面.....	13
図 15：逐次アニメ初期画面.....	14
図 16：逐次アニメ 5 秒後.....	14
図 17：並列処理の会話（1）.....	15
図 18：並列処理の会話（2）.....	15
図 19：並列処理の会話（3）.....	15
図 20：並列処理の会話（4）.....	15
図 21：並列アニメ初期画面.....	16
図 22：並列アニメ 5 秒後.....	16
図 23：負荷が等しくないときの初期画面.....	17
図 24：負荷が等しくないときの 5 秒後.....	17
図 25：負荷均衡の会話（1）.....	18
図 26：負荷均衡の会話（2）.....	18
図 27：負荷均衡の会話（3）.....	18
図 28：負荷均衡の会話（4）.....	18
図 29：負荷が等しいときの初期画面.....	19
図 30：負荷が等しいときの 5 秒後.....	19
図 31：1 台、2 台、4 台、8 台を比較したアニメの初期画面.....	20
図 32：1 台、2 台、4 台、8 台を比較したアニメの 7 秒後.....	21
図 33：1 台、2 台、4 台、8 台を比較したアニメの 14 秒後.....	21
図 34：1 台、2 台、4 台、8 台を比較したアニメの 28 秒後.....	21

図 3 5 : 1 台、2 台、4 台、8 台を比較したアニメの 5 6 秒後 ( 最終画面 ) .....	22
図 3 6 : スケーラビリティの会話 ( 1 ) .....	22
図 3 7 : スケーラビリティの会話 ( 2 ) .....	22
図 3 8 : スケーラビリティの会話 ( 3 ) .....	23
図 3 9 : スケーラビリティの会話 ( 4 ) .....	23
図 4 0 : <switch>...</switch>タグ .....	24

表目次 :

表 1 : 相撲モデルの対応表 .....	5
表 2 : 並列処理教材の評価 .....	23
表 3 : 初期バッファタイムの比較.....	25
表 4 : スムーズに再生できるかの比較.....	26

## 1 はじめに

ブロードバンドといった高速なインターネットを安価に利用できるようになってきた。「インターネット接続サービスの利用者等の推移」によれば、2001年8月に約51万人だったDSLサービス加入者数は1年後の2002年8月には約390万人に増えたそうだ。それだけブロードバンド環境の普及が進んだといえる。これに伴い、今まで静止画やテキストを中心としたWebコンテンツは、アニメーションや映像、音声などの非常にデータ量の多いストリーミングデータを中心としたマルチメディアを駆使したものになると考えられる。現在、ストリーミング技術を用いた映像表現やコミュニケーション手段として、インターネットの機能を効果的に利用した教育システムであるEラーニング、映画や音楽などを配信するエンターテインメントコンテンツ、IRや商品プロモーションといったビジネスプレゼンテーションなどがあり、確実に発展してきている。[6]

しかし、高速なインターネットでもデータ量の多いストリーミングコンテンツの配信において、複数の利用者が同時に同じサーバにアクセスすることは、サーバ側にとって多大な負担となる。そのような負担を軽減するための1つの手法として、同期マルチメディアの技術が挙げられる。[3]

同期マルチメディアとはストリーミング用に作成された映像や音声のファイルにテキストや静止画などのファイルを同期して1つのコンテンツとして表示させるための技術である。映像や音声の圧縮時における情報の劣化を文字や静止画によって補うことができ、また、複数のメディアを統合することでより魅力的なコンテンツを表示することができる。

本研究では、Eラーニングに注目し、本研究室の研究分野である並列処理という難しい概念を楽しく理解できるように同期マルチメディアを用いて並列処理教材を作成した。また、そのコンテンツに環境の対応を行い、ストリーミングの配信実験を行った。

まず、並列処理の基本的な概念で特に重要と思われる概念を3つのコンテンツに分け、概念の分かりやすい例として相撲モデルを使用し、作成した。そして、ユーザの環境がモデム・ISDNならモデム・ISDN用のコンテンツ、それ以上の環境ならそれ以上の環境用のコンテンツを自動的に配信できるようにし、ストリーミング配信し検討した。

本研究ではSMILという同期マルチメディア言語を利用して、アニメーション、音声、文字、静止画を同期させてコンテンツを作成した。アニメーション部分は3D Studio MaxとPremiere使用し、音声はマイクで録音しPremiereで編集した。文字はRealText、静止画は主にAdobe Photoshopで作成した。

本論文では、2章で同期マルチメディアとストリーミング技術について述べる。3章では、作成したコンテンツの構成、並列処理キーワードについて、そして各メディアの同期方法について述べる。4章では相撲モデルによる並列処理キーワードの作成について、5章ではキーワードアニメーションの詳細について説明し、評価する。6章ではストリーミング配信実験と考察について述べる。7章では本論文のまとめと今後の課題について述べる。

## 2 同期マルチメディアとストリーミング

### 2.1 ストリーミングとは

従来、動画や音声の通信技術は、インターネット上にある動画データや音声データ全体をいったん自分のコンピュータに取り込んだ後に再生するダウンロードが一般的であった。ダウンロード後は何度でもローカル上で見ることができるという利点がある反面、ファイル全体のダウンロードが完全に終了するまで再生できない欠点がある。このような悩みを一気に解消できるのが、ストリーミング技術である。[4]

ストリーミングとは、インターネットやイントラネット上で、一方が音声や映像などのマルチメディアデータを送信し、同時にもう一方が受信しながら再生する技術である。つまり、リクエストした動画や音声データを再生する前に、数秒間データを溜め込み、その後、順次データを受信しながら再生することである。ダウンロードのように長い時間待たされることなく再生できるのが最大の特徴だ。ローカル上のコンピュータにはデータが残らないため繰り返し見るといことはできないが、コンテンツを配信する側にとっては、無断で複製や再配布されることを防ぎ、コンテンツの著作権を保護するという利点がある。

ストリーミングを実現するためには、音声やサウンドまたはビデオ映像をストリーミング再生できるようなファイルにエンコーディングする必要がある。収録されたアナログあるいはデジタル素材をコンピュータに取り込み、配信したいフォーマットのデジタルデータにエンコーディングして、インターネット上のサーバを経由して配信する。また、受け取ったファイルは RealPlayer、Windows Media Player、QuickTime などの再生プレーヤーでデコーディングして見ることができる(図1)。[4] [5] [6]

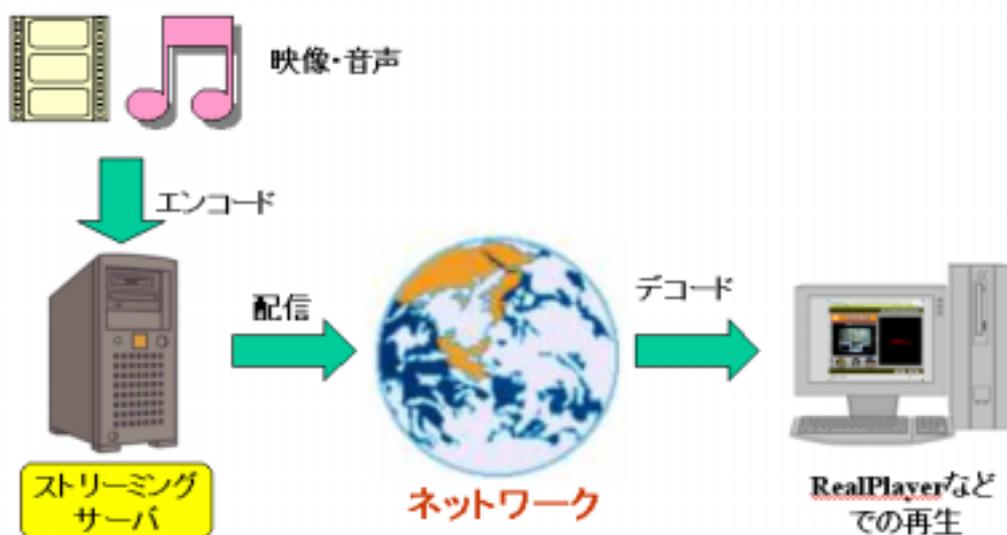


図1：ストリーミングの流れ

図1がストリーミングを実現するための流れだが、配信方法は2つの形態がある。

1つはビデオやCDのように見たい、聴きたいときにいつでも視聴することができるもので、「オンデマンド」と呼ばれる。これは、あらかじめ作成しておいたコンテンツを視聴者のリクエストに応じて配信するもので、ストリーミングコンテンツの大半がこのオンデマンドにあたる。

もう1つは、テレビやラジオのように音声や映像をリアルタイムに中継、配信する方法で、「ライブ」と呼ばれる。しかし、ライブを実現するためには専用サーバを利用しなければならない。

ストリーミング配信は利点ばかり目立つが問題もある。ストリーミングを楽しむにも普通のインターネット接続を行うときと同じようにインターネット回線やサーバの混み具合に依存すること。また、ネットワークの帯域幅に準じた映像を配信されるため、低速なネットワークの場合、映像や音声の品質の低下が避けることができないことだ。しかし、それらはADSL、光ファイバーネットワークなどの高速通信網の普及と整備、ストリーミング自体の普及によりまもなく解消される。また、後述する同期マルチメディアの技術を用いることで、情報の劣化を補うことが可能である。

## 2.2 同期マルチメディアとは

同期マルチメディアとは、ストリーミングの音声や映像に同期して、たとえばプレゼンテーションの資料をWebブラウザで表示するための技術である。これは、動画で配信する際のコンテンツ情報の劣化を容量の少ない文字や画像ファイルを用いて付加することで失われた部分の情報を補う。同期情報を含んだ同期マルチメディアデータは、サーバのハードディスクなどに保存され、ユーザが必要なときにネットワークを通じて受信し観覧することになる。同期マルチメディアのコンテンツを作成する場合、SMIL(synchronized Multimedia Integration Language)という言語を使用する方法が一般的である。[12]

## 2.3 同期マルチメディア言語：SMIL

SMILは、1998年6月にW3C(World Wide Web Consortium)のSynchronized MultimediaワーキンググループによってSMIL1.0として、2001年8月にSMIL2.0として策定された言語で、マルチメディアを統合する言語という意味である「Synchronized Multimedia Integration Language」の略である。SMILを使うことにより、ストリーミングコンテンツとして使われているビデオやオーディオ、Flashアニメーション、静止画、テキストなどのメディアファイルを1つのコンテンツとして統合できる。(図2)また、ユー

ザの環境に合わせた自動配信はでき、画面のどの部分にこういったタイミングで表示するか正確に、複数のメディアデータを同期させ再生させることができる。[3] [5]



図 2 : SMIL の概要

### 3 同期マルチメディアを用いた並列処理教材コンテンツ

#### 3.1 コンテンツの構成

本研究では、並列処理の基本概念を説明するコンテンツを作成した。コンテンツの構成は並列処理教材コンテンツのメニューページから、今回参考として使用した山崎勝弘教授の「コンピュータは進化する」の1部分を引用したHTML、並列処理と逐次処理の違いと粒度の説明をした「並列処理とは」というSMILコンテンツ、負荷均衡を説明したSMILコンテンツ、スケーラビリティを説明したSMILコンテンツの4つのコンテンツに移動することができる。HTMLのテキスト部分のリンクからSMILコンテンツに移動することができ、またその逆も可能である。

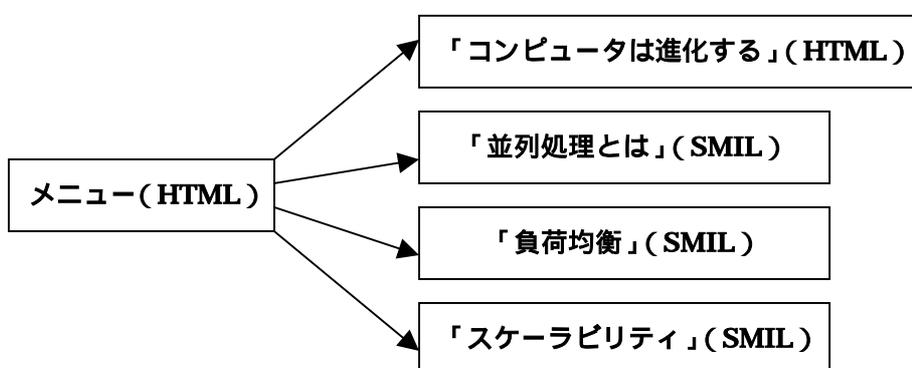


図3：コンテンツの構成図

また、説明するモデルとして、日本の国技である相撲を使用した。並列処理の仕事を相撲の取り組み、プロセッサを土俵と置き換えている。相撲の取り組み時間は全て等しくし、理想的な状態にしている（表1）。

表1：相撲モデルの対応表

並列処理	相撲モデル
仕事	取り組み
プロセッサ	土俵

さらに並列処理に詳しい力士が並列処理を全く知らない力士に並列処理を説明するアニメーションを加えている。面白い動きのあるモデルを使用し、さらに分かりやすい説明を加えることにより、並列処理を全く知らないユーザでも並列処理の基本概念を楽しく簡単に理解できるような仕組みとなっている。

### 3.2 並列処理のキーワード

ここでは各キーワードの説明をする。

(a) 並列処理とは

ここでは並列処理と逐次処理の違い、粒度の説明をした。並列処理とはひとつの仕事を均等に分散化し、それぞれで処理を行うことである。仮に4台のプロセッサで並列処理を行うならば、単純に考えて処理時間が4分の1となる。対して逐次処理とはひとつの仕事をひとつのプロセッサで行うことである。大規模な問題を実行する場合は何時間もかかってしまい、並列処理にしたほうが速い。また、粒度とは並列処理の単位となる処理の大きさのことである。ここでは相撲の4取り組みを1台の時、4台の時で処理することにより並列処理と逐次処理の違いを説明した。

(b) 負荷均衡

並列処理では全体の処理時間は、各プロセッサでの演算時間とプロセッサ間の通信時間の合計となる。また、最も演算時間の長いプロセッサでの時間が全体の処理時間に影響する。各プロセッサでの負荷をできるだけ等しくすることを負荷均衡という。ここでは土俵が3台の時、6取り組みをバランスよく処理する時とバランスが悪く処理する時のアニメーションを表示することにより負荷均衡を説明した。

(c) スケーラビリティ

プロセッサの台数を増やしただけで単純に速度向上が得られるとは限らない。台数を増やした時にそれに見合う性能向上が得られる可能性をスケーラビリティという。ここでは相撲の8取り組みを1台、2台、4台、8台の時で処理を表現し、同時に表示することにより説明した。

### 3.3 各メディアの同期方法

各キーワードのコンテンツはSMILを利用して同期をとった。SMILによってかかれたファイルは、SMILに対応しているRealPlayerに表示される。SMILを用いて、複数のメディアを利用することで、非常にわかりやすいコンテンツが作成できた。図4がコンテンツ完成の例である。例として「並列処理とは」を利用した。



図 4 : コンテンツの完成図の例 ( 並列処理とは )

の部分に映像が流れる。映像はCGと静止画と音声を組み合わせたものである。の部分にはRealTextが表示される。RealTextは文字表示を時間指定できる。

```

heiretu02 - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<window width="300" height="20" duration="2:10" bgcolor="black">
<center>
<font charset="x-sjis" face="osaka" color="white" size="2"/>
<time begin="0:34">
1台 (2.8秒)
<time begin="1:55">
4台 (7秒)
</font>
</center>
</window>

```

図 5 : RealTextによる時間指定

図 5 は「 並列処理とは 」のRealText部分で、一番上のduration="2:10"が全体の時間を表す。つまり、このファイルは2分10秒のコンテンツである。最初の文字が全体の時間の34秒後に表示され、次の文字が全体の時間の1分55秒に表示される。

さらに静止画をうまくレイアウトして同期を図り、リンクを貼る事でコンテンツが完成する。

## 4 相撲モデルによる並列処理教材の作成

### 4.1 作成の流れ

作成の流れについて説明する。図6は相撲モデルによる並列処理教材の作成の流れ図である。まず、3D Studio Maxで並列処理の仕事に置き換えた力士、プロセッサに置き換えた土俵などを作成する。さらに作成した力士、土俵を利用し、相撲アニメーションと静止画を作成する。静止画はAdobe Photoshopで作成する。次に作成した相撲アニメーションと静止画、午後のこ～だ for windowsで作成した音声を使用し、編集してキーワードアニメーションを作成する。編集するソフトにPremiereを使用した。さらに作成したキーワードアニメーションをPremiereでストリーミング形式に出力する。最後にストリーミング形式のファイルとテキスト、静止画をSMILで統合することによりコンテンツが完成する。

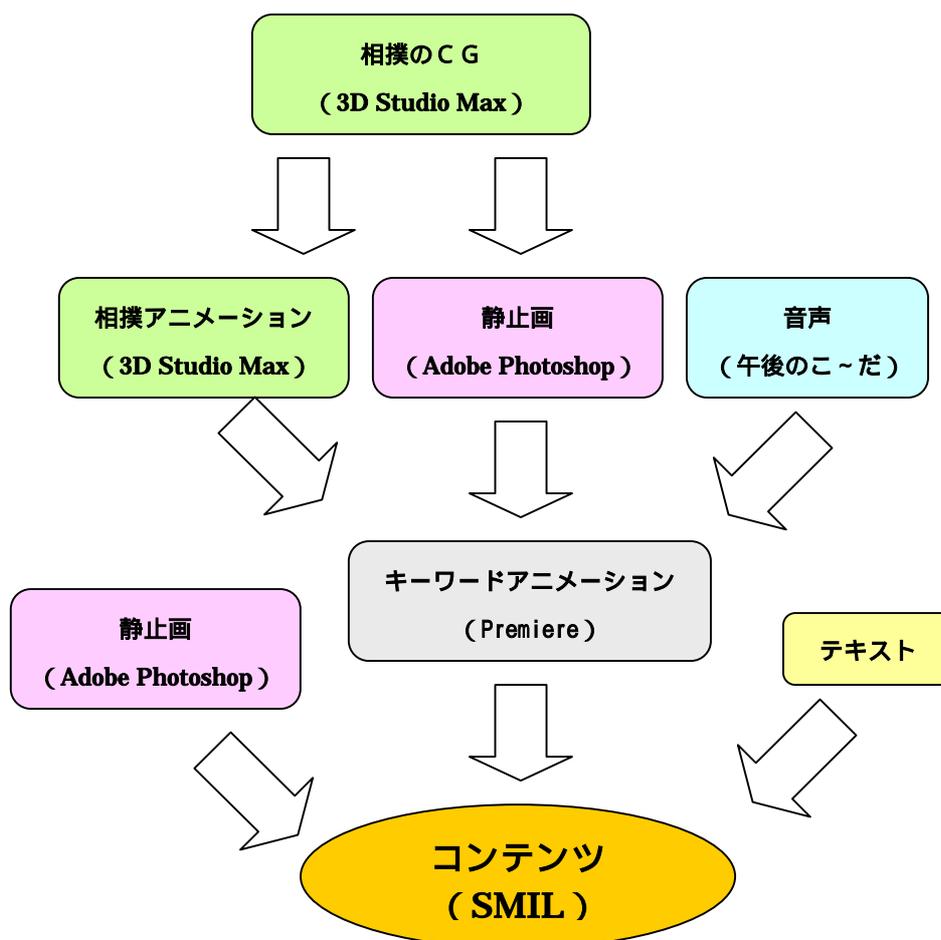


図6：相撲モデルによる並列処理教材の作成の流れ

## 4.2 3D Studio MaxにおけるCGとアニメーションの作成

### 4.2.1 3D Studio Maxとは

相撲のCG、アニメーションを作成するソフトとして3D Studio Maxを使用した。3D Studio Maxは、ビジュアルエフェクト、キャラクターアニメーション制作といった目的で使用される3Dモデリング、レンダリング、アニメーションソフトウェアである。

図7は3D Studio Maxの起動画面である。インターフェースは上部のメニューバー、右端部分のコマンドパネル、左にあるビューポット、下部のタイムスライダの4つに大別できる。コマンドパネルによりさまざまなオブジェクトが選択可能であり、ここで元となるオブジェクトを選択し、左のビューポットでCGグラフィックを作成する。また、このオブジェクトにはさまざまな変化が可能である。ビューポットも様々に変化することができ、基本は図のように4分割され、左上のウィンドウがオブジェクトの真上から、右上が右側から、左下が左側から、右下がユーザの任意の視点からオブジェクトを見ることができる。また、この他の視点からもオブジェクトを見ることができる。そして、タイムスライダを動かしてそのシーンごとに作成することによりアニメーションを作成することができる。[7]

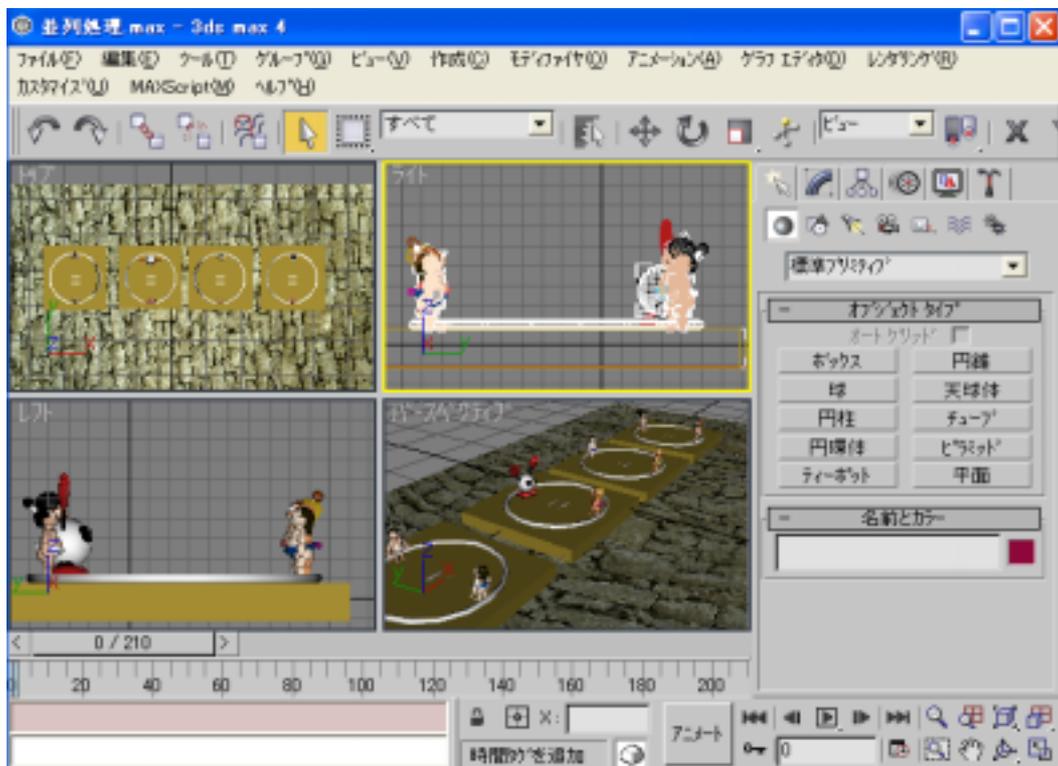


図7：3D Studio Maxのインターフェース

## 4.2.2 相撲CGの作成

並列処理の仕事とプロセッサに置き換えた力士と土俵の作成について説明する。

### (a) 力士の作成

力士は16人作成した。面白みを出すために髪の毛やまわし、体格を変えることで各力士に特徴をつけた。主な作成手順として、まず FFD という技法と面を押し出したりして、リアルな顔を作成した。FFD とは物体を直方体で囲んで、直方体を変形させることで物体も変形させることができる技法である。次にオブジェクトで体を作成し、完成である。後でアニメーションしやすいように体は関節ごとにパーツを作成している。図8が力士の完成図である。



図8：力士

### (b) 土俵の作成

土俵はオブジェクトを組み合わせて作成する。単にカラーだけで土を表現するのは面白くないので、マテリアルエディタを使うことでリアルな土を表現した(図9)。マテリアルエディタではオブジェクトがどのように見えるか定義することができる(図10)。

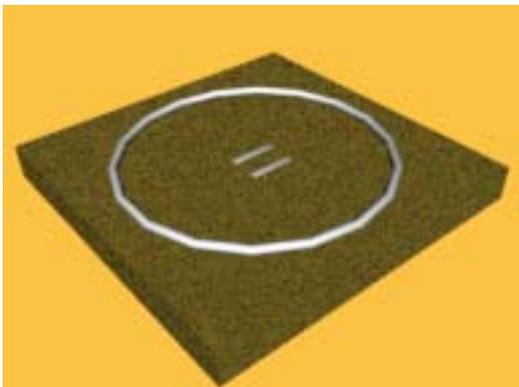


図9：土俵



図10：マテリアルエディタ

### 4.2.3 相撲アニメーションの作成

作成した力士と土俵を使って相撲アニメーションを作成した。本研究では、8 取り組みのアニメーションを作成した。アニメーションは コマ1コマ作成していたら莫大な時間がかかってしまう。そこで、アニメーション中のキーとなる部分のみ作成し、その間は補完するというキーフレームアニメーションを利用した。すべての取り組みはフレーム数を210とし時間を等しくした。そして、1つのアニメーションにキーを20個作成した。また、力士の動きがスムーズな動きになるようにフォワードキネマティクスを利用した。フォワードキネマティクスとは、パーツに親、子、孫といった関係を作って、親が動いたら、子及び孫も親とともに動くという技法だ。これによりパーツが制御しやすくなり、スムーズな動きが実現した。最後に完成したキーフレームアニメーションをレンダリングして、出力する。

### 4.3 静止画の作成

静止画の作成は主に Adobe Photoshop で作成した。Adobe Photoshop はさまざまなデジタル画像処理ができる画像編集ツールである。図 1 1 が作成画面である。作成した画像は JPEG 形式で保存した。

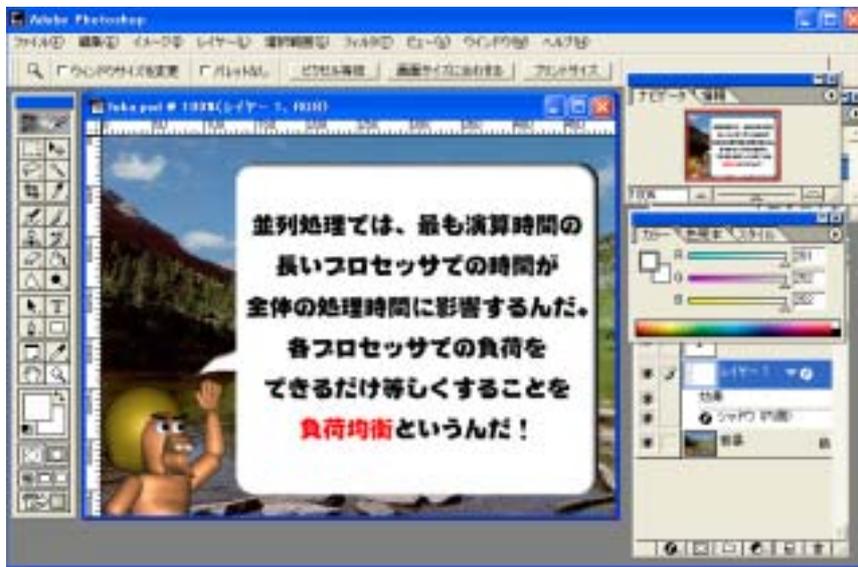


図 1 1 : Adobe Photoshop の画面

## 4.4 音声の作成

音声を作成するために午後のこ～だ for windows を使用した。午後のこ～だ for Windows は、wav や aiff などさまざまな形式の音声データを MPEG 規格にて定められた音声ストリームデータの"MP3"形式に変換するためのフリーのソフトウェアである。また、音声の録音もできる。これを使用して音声を録音し MP3 形式で保存した(図12)。

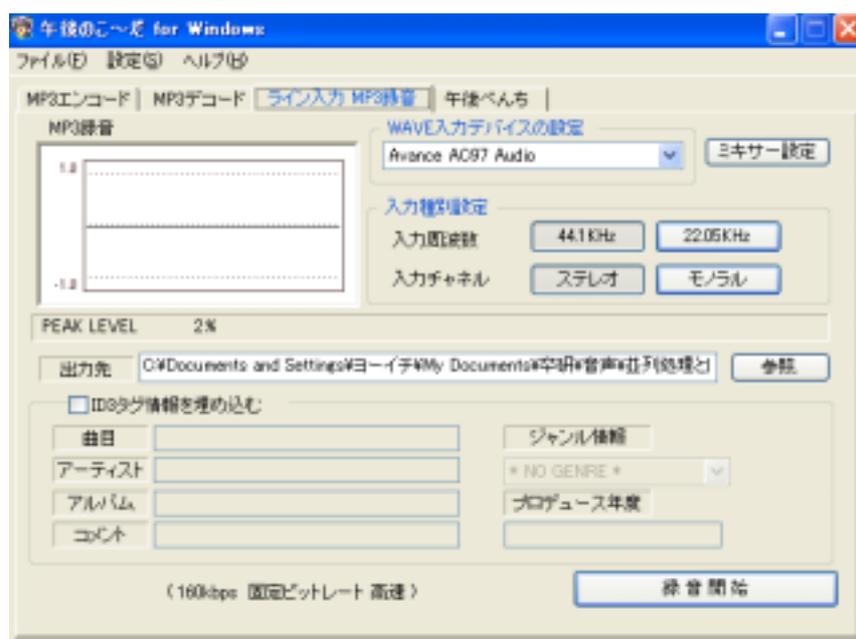


図12 : 午後のおこ～だ for windows

## 4.5 Premiere におけるアニメーションの作成

3D Studio Max で作成したアニメーションだけでは、単なる素材にすぎない。特殊効果を加えることにより、より分かりやすく面白いアニメーションになる。3d Studio Max で作成したアニメーションと Adobe Photoshop で作成した静止画、午後のおこ～だで作成した音声を同期して、より分かりやすく面白いアニメーションにした。そのためのツールとして Premiere を使用した。Premiere はDVD・VHSなど多くのメディアに対応し、操作性、特殊効果が豊富なツールである。

図13はPremiereの作成画面である。赤い線で囲まれた部分がタイムラインである。タイムラインで、映像と音声の編集ができる。タイムラインのビデオトラックで映像、オーディオトラックで音声の編集作業を行う。例えば、アニメーションに音声を同期させたいというときにはタイムラインのビデオトラックにアニメーションを配置し、オーディオ

ラックに音声を同じ時間に配置すればよい。

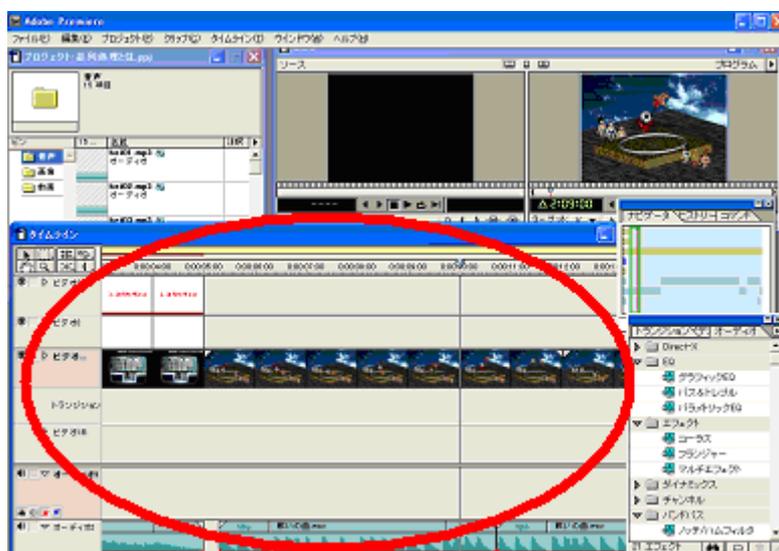


図 1 3 : Premiere の画面

Premiere では、完成したアニメーションをストリーミング形式のファイルに出力することができる。しかも、モデム用、ISDN 用、Broadband/LAN 用のファイルといった公開した回線速度にあわせたファイルを作成することができる。図 1 4 は Broadband/LAN 用のファイルを作成中の画面である。

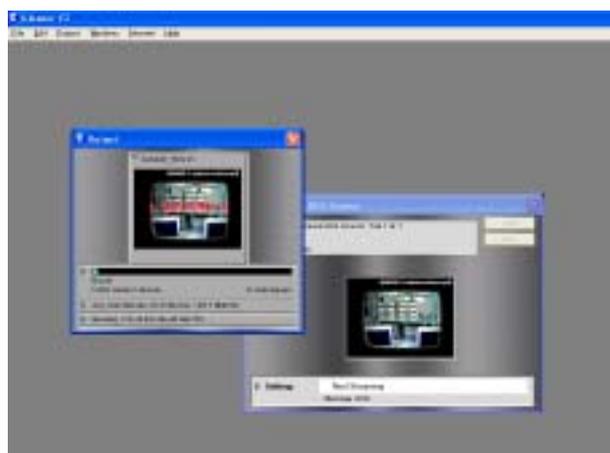


図 1 4 : ストリーミング形式のファイルを作成中の画面

## 5 キーワードアニメーションの詳細と評価

### 5.1 並列処理とは

ここでは、並列処理と逐次処理の違いと粒度の説明を相撲の取り組みを例にとって説明した。アニメーションは逐次処理アニメーション、説明を含む会話アニメーション、並列処理アニメーションの3つを組み合わせた。

#### (a) 逐次処理アニメーション

土俵が1台あり、力士が8人いる(図15)。そして、相撲の4取り組みが順に処理される(図16)。プロセッサを土俵、仕事を相撲の取り組みとしている。また、分かりやすくするために相撲の各取り組みの時間は全て等しくした。



図15：逐次アニメ初期画面



図16：逐次アニメ5秒後

(b) 説明を含む会話アニメーション

並列処理に詳しいキャラクターが並列処理に詳しくないキャラクターに対して、並列処理と粒度の説明をするアニメーションである。単に説明だけのアニメーションするだけでなく、会話を通じて説明することにより分かりやすいアニメーションとなっている。

図 1 7 , 1 8、1 9 , 2 0 は並列処理の会話の一場面である。



図 1 7 : 並列処理の会話 ( 1 )



図 1 8 : 並列処理の会話 ( 2 )

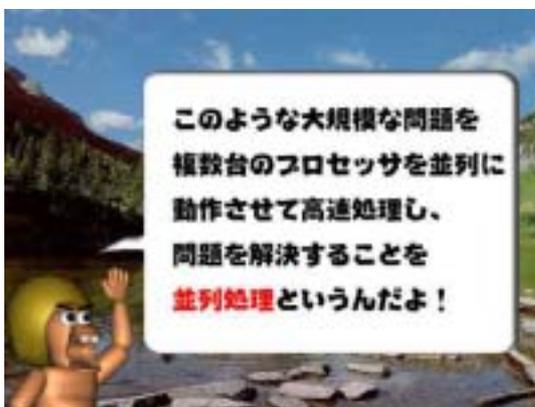


図 1 9 : 並列処理の会話 ( 3 )

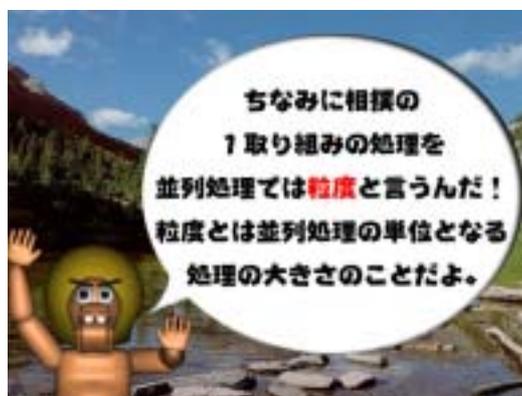


図 2 0 : 並列処理の会話 ( 4 )

(c) 並列処理アニメーション

土俵が 4 台あり、力士が 8 人いる ( 図 2 1 )。そして、相撲の 4 取り組みが同時に処理される ( 図 2 2 )。プロセッサを土俵、仕事を相撲の取り組みとしている。また、分かりやすくするために相撲の各取り組みの時間は全て等しくした。



図 2 1 : 並列アニメ初期画面



図 2 2 : 並列アニメ 5 秒後

## 5.2 負荷均衡

ここでは、負荷均衡を相撲の取り組みを例にとって説明した。アニメーションは負荷が等しくないときのアニメーション、説明を含む会話アニメーション、負荷が等しいときのアニメーションの3つを組み合わせた。

### (a) 負荷が等しくないときのアニメーション

土俵が3台あり、力士が12人いる(図23)。そして、1台は4つの取り組み、残り2台は1つの取り組みが行われる。1台に取り組みが集中して効率が悪い様子を表している(図24)。また、分かりやすくするために相撲の各取り組みの時間は全て等しくした。



図23：負荷が等しくないときの初期画面



図24：負荷が等しくないときの5秒後

(b) 説明を含む会話アニメーション

ここでは、並列処理に詳しいキャラクターが並列処理に詳しくないキャラクターに対して、負荷均衡の説明をするアニメーションである。

図 2 5、2 6、2 7、2 8 は負荷均衡の会話の一場面である。



図 2 5 : 負荷均衡の会話 ( 1 )



図 2 6 : 負荷均衡の会話 ( 2 )

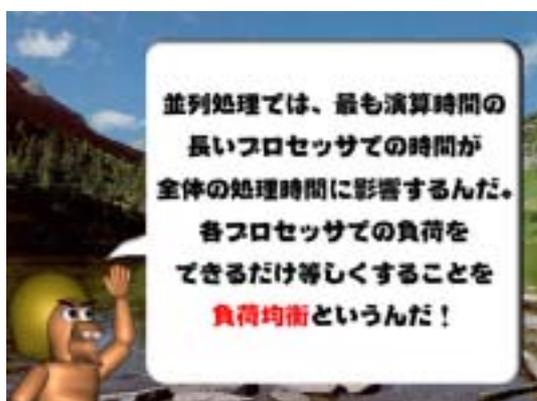


図 2 7 : 負荷均衡の会話 ( 3 )



図 2 8 : 負荷均衡の会話 ( 4 )

(c) 負荷が等しいときのアニメーション

土俵が 3 台あり、力士が 1 2 人いる ( 図 2 9 )。各土俵に 2 取り組みが行われる。各土俵の取り組み数が同じなので、効率よく取り組みが終わる様子を表している ( 図 3 0 )。また、分かりやすくするために相撲の各取り組みの時間は全て等しくした。



図 2 9 : 負荷が等しいときの初期画面



図 3 0 : 負荷が等しいときの 5 秒後

### 5.3 スケーラビリティ

ここではスケーラビリティを相撲の取り組みを例にとって説明した。ここでは、1台、2台、4台、8台を比較したアニメーションと説明を含む会話アニメーションの2つを組み合わせた。

#### (a) 1台、2台、4台、8台を比較したアニメーション

まず、相撲の8取り組みを1台、2台、4台、8台で行うアニメーションを作成する。全ての取り組みの時間を7秒に固定し、説明が分かりやすくなるようにした。さらに各台ごとにタイマーのアニメーションを作成し、そのアニメーション各台のアニメーションに埋め込む。最後に4つのアニメーションを統合し1つのアニメーションとして表示する。左上に1台のとき、右上に2台のとき、左下に4台のとき、右下に8台のときのアニメーションが表示される(図3 1)。7秒後に8台のときのアニメーションが終了する。終了したことを分かりやすく表現した(図3 2)。14秒後には4台のときのアニメーションが終了する(図3 3)。28秒後には2台のときのアニメーションが終了する(図3 4)。最後に56秒後に1台のときのアニメーションが終了して全体のアニメーションが終わる(図3 5)。



図3 1：1台、2台、4台、8台を比較したアニメの初期画面



図 3 2 : 1 台、2 台、4 台、8 台を比較したアニメの 7 秒後



図 3 3 : 1 台、2 台、4 台、8 台を比較したアニメの 1 4 秒後



図 3 4 : 1 台、2 台、4 台、8 台を比較したアニメの 2 8 秒後



図35：1台、2台、4台、8台を比較したアニメの56秒後（最終画面）

(b) 説明を含む会話アニメーション

ここでは、並列処理に詳しいキャラクターが並列処理に詳しくないキャラクターに対して、スケーラビリティの説明をするアニメーションである。

図36、37、38、39はスケーラビリティの会話の一場面である。



図36：スケーラビリティの会話（1）



図37：スケーラビリティの会話（2）



図38：スケラビリティの会話（3）

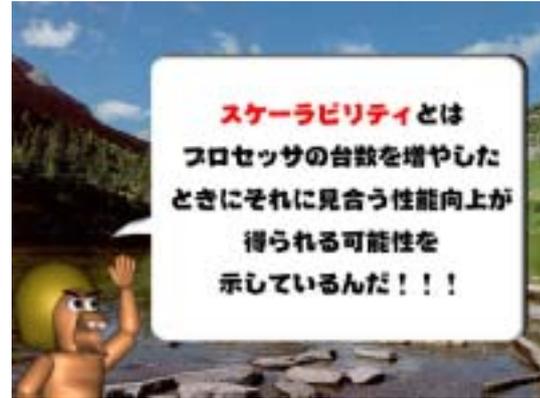


図39：スケラビリティの会話（4）

## 5.4 評価

完成した並列処理教材を20人の方に視聴していただき、評価をいただいた。表2にその結果を示す。

表2：並列処理教材の評価

	コンテンツの内容	コンテンツの配置
分かりやすい	14	12
普通	5	8
分かりにくい	1	0
計	20	20

コンテンツの内容は、並列処理の基本的な概念が簡単に理解できるという意見が多かった。しかし、コンピュータの知識が全くない人に見せたら内容が分からないという意見があった。これは並列処理以前にコンピュータの知識がないため、コンテンツにでてくるコンピュータの用語が分からなかったためだと考えられる。そのような人も考慮したコンテンツを作成する必要があることが分かった。また、コンテンツの配置は見やすいという意見が多かった。これは、ユーザの立場にたって、各メディアを配置したのがよかったのだと思う。

## 6 ストリーミング配信実験と考察

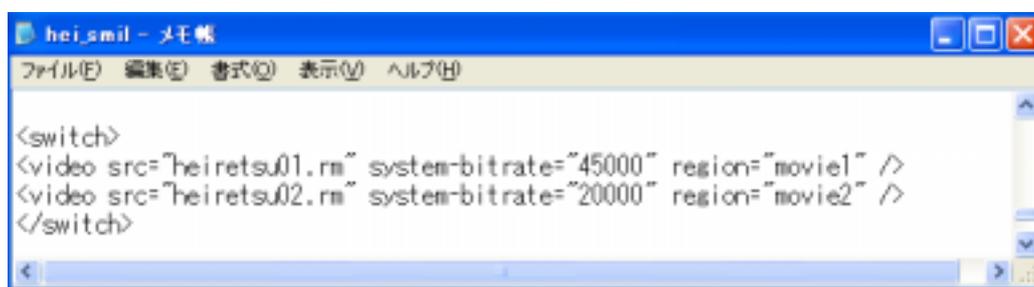
### 6.1 ストリーミング配信実験

作成した並列処理教材でストリーミングサーバによる配信と Web サーバによる配信との比較を行った。実験内容は1つのコンテンツを同時に多くの人に視聴してもらい、コンテンツにアクセスしてから再生するまでの時間である初期バッファタイム、スムーズに再生できるかを比較した。

ストリーミング配信にはストリーミングサーバとして「RealSystem Server」を使用した。「RealSystem Server」はネットワークを介してマルチメディアをストリーミングすることを目的として開発されたものであり、複数のクリップの同期を取り、ネットワークの状態が悪くてもスムーズにクリップのストリーミングができるようになっている。

Web サーバでは、先行ダウンロード方式で配信を行う。先行ダウンロード方式は、端末側でデータを少しずつダウンロードしながら、その完了を待たず、届いたものから順に再生する方式である。ダウンロードのスピードが再生スピードより速ければ、すばやく再生を始め、途切れなく再生することができる。ダウンロードのスピードが再生スピードより遅い場合には、基本的にはダウンロード終了を待って再生することになるため時間がかかるが、映像はとてもきれいに再生できる。また、ハードディスクにそのファイルは残るが、通常は一時ファイルとして格納され、自動的に排除される。

また、実験をする前に環境への対応を考えた。高速 LAN を使える人がいるかと思えば、低速なアナログモデムしか使えない人がいるなどユーザの環境はさまざまだからである。環境の対応として SMIL の <switch>...</switch> タグを使用した。SMIL の <switch>...</switch> タグを使用することにより閲覧側の環境にふさわしいメディアファイルを選択できるように設定することができる。[3]



```
hei_smil - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

<switch>
<video src="heiretsu01.rm" system-bitrate="45000" region="movie1" />
<video src="heiretsu02.rm" system-bitrate="20000" region="movie2" />
</switch>
```

図 4 0 : <switch>...</switch>タグ

図 4 0 は「並列処理とは」の SMIL の <switch>...</switch> タグ部分である。モデム、ISDN 以上(64kbps 以上)の環境のユーザには「heiretsu01.rm」を自動的に配信する。何故 45kbps になっているかということと実際の転送レートはネット上の混雑等を考慮すると低下するので、

RealSystem では 64kbps 用は実質 45kbps として扱っているからだ。それ以下で 28.8kbps 以上の環境のユーザには「heiretsu02.rm」が自動的に配信される。同様に、RealSystem では 28.8kbps 用は 20kbps として扱っている。

## 6.2 結果と考察

表 3、4 に実験結果を示す。今回は、20 人の方に同時にコンテンツを視聴してもらった。コンテンツは「並列処理とは」を使用した。また、モデム、ISDN は 405KB の容量、モデム、ISDN 以上には 1.48MB の容量のコンテンツが自動的に配信される。

表 3：初期バッファタイムの比較

	ストリーミング		先行ダウンロード	
	ブロードバンド	モデム・ISDN	ブロードバンド	モデム・ISDN
10 秒以内	16	0	7	0
20 秒以内	0	2	1	0
30 秒以内	0	1	0	3
視聴できない	0	1	8	1
計	20		20	

ブロードバンド環境で、ストリーミング配信を行った場合は、ストレスなく視聴できることが分かった。しかし、Web サーバでは、エラーで視聴できない人が多かった。原因を調べてみると、Web サーバで SMIL ファイルを配信した際、SMIL ファイルに直接リンクを行ったことが問題だった。これは、Web サーバが拡張子.smil のついたファイルの中の要素を効率よく同期してくれないためである。この問題を解決する方法として、拡張子.ram のついたメタファイルを作り、SMIL ファイルに直接ではなくメタファイルにリンクするようにすればよい。これは一般に擬似ストリーミングという。

一方、モデム・ISDN 環境で、ストリーミング配信を行った場合は、多少時間がかかった。また、モデムの方は、時間切れで切断され全く視聴できなかった人もいた。Web サーバでは、ストリーミング配信より表示されるまで時間がかかることが分かった。モデム・ISDN 環境では、ダウンロードのスピードが再生スピードより遅いためである。

次に、表 4 にコンテンツがスムーズに再生できたかの比較を表示する。

表 4：スムーズに再生できるかの比較

	ストリーミング		先行ダウンロード	
	ブロードバンド	モデム・ISDN	ブロードバンド	モデム・ISDN
再生できた	16	1	8	0
再バッファリングがある	0	2	0	3
再生できない	0	1	8	1
計	20		20	

ブロードバンド環境で、ストリーミング配信を行った場合は、スムーズに再生できたことが分かる。Web サーバの方でも、エラーで視聴できない人以外、スムーズに再生できたことが分かる。一方、モデム・ISDN 環境で、ストリーミング配信を行った場合は、途中で止まり、バッファリングを行う症状が現れた。同様に Web サーバの方でもある。

今回、ストリーミングサーバによる配信と Web サーバによる配信との比較を行ったが、大差はなかった。これはファイル容量が小さかったためだ。ファイル容量を増やし、もっと大勢の人が同時に視聴する実験が必要である。また、さまざまな環境に考慮してメディアファイルを自動配信できるようにしたが、実験では環境にあったメディアファイルを配信することができた。ブロードバンド環境では、メディアファイルの劣化はなかった。しかし、モデム、ISDN 用のコンテンツは速度を意識しすぎてしまい、コンテンツの劣化が激しく逆に分かり難くなってしまった。

## 7 おわりに

本研究では、同期マルチメディアを用いた並列処理教材において、並列処理の基本的な概念を如何に分かりやすく、面白くユーザに説明するかに重点をおいた。まず、概念を説明するモデルとして相撲モデルを考えた。これは、ただ単に説明するより、動きが多い相撲モデルを使うことで、ユーザが楽しく理解できると考えたからだ。その相撲モデルを 3D Studio Max と Premiere を使用し、さらに音声を付け加えてアニメーション化した。それだけでは、伝えきれない部分や、配信による情報の欠落が考えられるため、SMIL という同期マルチメディア言語を用いて作成したアニメーションにアニメーションを補足説明する文字や静止画を同期させた。最後にそのコンテンツをストリーミング配信した。

同期マルチメディアを用いることで非常に分かりやすいコンテンツを作成することができた。また、相撲モデルはユーザがコンテンツを見ていて飽きないなど効果的であることが分かった。ストリーミングはユーザの環境を考慮した配信を行ったが、モデム、ISDN 用のコンテンツは速度を意識しすぎてしまい、コンテンツの劣化が激しく逆に分かり難くなってしまった。また、Web サーバとの比較で、同時に 20 人見た場合でも、ストリーミングサーバでの配信はスムーズに視聴することができることが分かった。Web サーバでの先行ダウンロードは SMIL ファイルを配信することは困難だということが分かった。

今後の課題として、ユーザの学習意欲を促進するために各コンテンツに関連したゲームや問題を作成することが挙げられる。見るだけのコンテンツでは面白くない。ゲームや問題を実際に解いてみることで、さらに並列処理の理解度が深まると考えたからだ。ストリーミング配信では実験を行ったが、ファイル容量を増やし、もっと大勢の人で実験をする必要がある。また、ユーザの環境を考慮した配信を行ったが、さらに考えていく必要がある。どの環境に対してもコンテンツの劣化が少なく、スムーズに配信されるようなコンテンツを作成することが今後の課題である。

## 謝辞

本研究の機会を与えてくださり、数々の助言を頂きました山崎勝弘教授と小柳滋教授に心より感謝致します。また、本研究にあたり、いろいろな面で貴重なご意見、ご指導を頂きました本研究室の院生である末富俊樹氏、並びに本研究室の皆様に、心より感謝申し上げます。

## 参考文献

- [1] 湯浅太一・安村道晃・中田登志之 編：はじめての並列プログラミング、bit 別冊、共立出版、1999.
- [2] 山崎勝弘：コンピュータは進化する、1999.
- [3] SMIL でいこう：<http://www.smi.co.jp/smil/index.html/>.
- [4] 平明弘：誰でもできるストリーミングかんたん入門、メディア・テック出版、2002.
- [5] 月間コンピュータグラフィックスワールド7月号別冊：ストリーミングワールド第2号、IDG Japan、2002.
- [6] 月間コンピュータグラフィックスワールド12月号別冊：ストリーミングワールド第3号、IDG Japan、2002.
- [7] 3D STUDIO MAX R3 チュートリアル、KINEX、1999.
- [8] C&R 研究所：あっと驚く達人の技 Premiere6.0、ナツメ社、2002.
- [9] KUMIKO：Photodhop5.0J パーフェクトマスター、秀和システム、1998.
- [10] 午後のこ～だ3：[http://member.nifty.ne.jp/~pen/free/gogo3/mct\\_gogo.htm/](http://member.nifty.ne.jp/~pen/free/gogo3/mct_gogo.htm/).
- [11] リアルネットワークス社：<http://www.jp.real.com/>.
- [12] 大塚良知：ストリーミングと同期マルチメディアを用いた研究室紹介システム、立命館大学理工学部情報学科卒業論文、2002.
- [13] 末富俊樹：同期マルチメディア言語を用いた研究キーワード紹介システム、立命館大学理工学部情報学科卒業論文、2002.
- [14] 田村智樹：同期マルチメディアを用いた研究紹介コンテンツの作成と物理環境への適応、立命館大学理工学部情報学科卒業論文、2002.
- [15] 小永章子：同期マルチメディアを用いた並列処理教材（ ）立命館大学理工学部情報学科卒業論文、2003.

## 付録 SMIL ソースファイル

### ( 1 ) 「並列処理とは」の SMIL ファイル ( hei\_smil.smil )

```
<smil>

<head>
<layout>

<meta name="title" content="卒業研究"/>
<meta name="author" content="Y.Kono"/>
<meta name="copyright" content="(C)2002 Y.Kono"/>

<root-layout width="680" height="480" background-color="#333300"/>

<region id="back" top="40" left="0" width="680" height="400" z-index="1"/>
<region id="photo1" top="47" left="10" width="340" height="56" z-index="2"/>
<region id="movie1" top="110" left="22" width="320" height="240" z-index="3"/>
<region id="movie2" top="110" left="22" width="320" height="240" z-index="4"/>
<region id="text1" top="65" left="360" width="310" height="350" z-index="5"/>
<region id="photo2" top="90" left="370" width="140" height="105" z-index="6"/>
<region id="photo3" top="90" left="520" width="140" height="105" z-index="7"/>
<region id="text2" top="210" left="370" width="300" height="20" z-index="8"/>
<region id="photo4" top="10" left="375" width="280" height="20" z-index="9"/>
<region id="photo5" top="203" left="82" width="200" height="48" z-index="10"/>
<region id="photo6" top="450" left="575" width="80" height="20" z-index="11"/>
<region id="photo7" top="450" left="475" width="80" height="20" z-index="12"/>

</layout>
</head>

<body>
<par>


<textstream src="heiretu01.rt" region="text1" fill="freeze"/>
<textstream src="heiretu02.rt" region="text2" fill="freeze"/>
```

```




<a href="http://www.hpc.cs.ritsumei.ac.jp/" show="new">

</a>
<a href="hei_smil.smil" target="_player">

</a>
<a href="text.html" show="new">

</a>
<a href="index.html" show="new">

</a>
<switch>
<video src="heiretsu01.rm" system-bitrate="45000" region="movie1" />
<video src="heiretsu02.rm" system-bitrate="20000" region="movie2" />
</switch>

</par>
</body>

</smil>

```

( 2 ) 「 並列処理とは 」 のテキストファイル 1 ( heiretu01.rt )

```

<window width="310" height="350" duration="2:10" bgcolor="black">

<center>
<font charset="x-sjis" face="osaka" color="red" size="5"/>

<br/><br/><br/><br/><br/><br/><br/><br/><br/><br/>
<b>並列処理とは</b><br/>
<font color="white" size="3"/>

```

```
<br/>
<time begin="1:03"/>
大規模な問題を複数台のプロセッサを<br/>
<time begin="1:06"/>
並列に動作させて高速処理し、<br/>
<time begin="1:09"/>
問題を解決すること<br/><br/><br/>
<time begin="1:40"/>
<font color="yellow" size="2"/>粒度</font>
<time begin="1:42"/>
<font color="white" size="2"/>...並列処理の単位となる処理の大きさ</font>

</font>
</center>

</window>
```

### ( 3 ) 「並列処理とは」のテキストファイル 2 ( heiretu02.rt )

```
<window width="300" height="20" duration="2:10" bgcolor="black">

<center>
<font charset="x-sjis" face="osaka" color="white" size="2"/>

<time begin="0:34"/>
1台 ( 28秒 )
<time begin="1:55"/>
4台 ( 7秒 )

</font>
</center>

</window>
```