卒業論文

# 同期マルチメディアを用いた<br>並列処理教材の作成（Ⅰ）


氏名　　：　小永章子
学生番号　：　2210990051－0
指導教員　：　山崎勝弘教授

提出日　　：　2003 年 2 月 21 日


立命館大学　理工学部　情報学科

## 内容梗概

現在インターネット上には多くのコンテンツが存在する。技術の進歩、ブロードバンド化にともない、メディアも多様化してきている。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。その中で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高まっている。しかし、静止画やテキスト等の比較的軽いものなら問題ないが、動画や音声など大容量ものは圧縮がかけられ、情報の劣化が見られる。それを同期マルチメディア技術によって補い、さらに情報を付加することで分かりやすいコンテンツにする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技術を用いて、視聴者に分かりやすい並列処理教材を作成する。並列処理の基本的な概念を、アニメーションを通して楽しみながら理解できるようにする。並列処理アニメーションのモデルとして、ボーリングモデルを CG で作成した。アニメーションだけでは分かりにくいので、そのアニメーションにテキストや音声を同期させた。これによって視聴者に伝えることのできる情報が増える。また、より理解しやすくなるように、同期アニメを見る前に人物やレーンの説明画面を出し、アニメの最後にまとめとして結果を加えた。

また作成した並列処理教材を Web 上におき、同期アニメーションのひとつについて、ダウンロード配信とストリーミング配信を行った。

**目次：**



**図目次：**

**表目次：**

# 1　はじめに

パーソナルコンピュータの低価格、高性能化によって、一般家庭でも CG のモデリングやレンダリングといった複雑で時間のかかる処理や計算を行えるようになった。また、ブロードバンドといった高速ネットワークを安価に利用できる時代になっている。これによって、3D や合成された映像を、テレビや映画、ゲームだけでなく、インターネット上でも多く見られるようになった。これまでのテキスト･画像に加え、動画や音声を用いて情報を伝える時代になっている。その中で、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高まっている。

しかし、高速ネットワークを利用しても静止画やテキスト等の比較的軽いデータ配信は問題ないが、動画や音声など大容量のコンテンツ配信はサーバに大きな負担をかける。こ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が挙げられる。

同期マルチメディアとは、動画や音声など動的なストリーミングコンテンツに、文字や静止画など、静的なコンテンツを同期的に表示させる技術のことである。ストリーミングでは、配信時の回線状況にあわせて圧縮をかけるため、画像の劣化が激しく、肝心な情報が伝わらない場合がある。同期マルチメディアの技術を用いることで、失われた情報を補うことができ、さらに情報を付加することも可能になる。[6] [8]

本研究では、同期マルチメディアを用いて、高性能計算研究室の研究テーマである並列処理の教材を作成する。今日、情報処理の世界では並列処理は大分浸透してきている。ここ数年間で様々な高性能並列計算機が製品化され、価格性能比も飛躍的に向上している。実際並列計算機は気象予測、物理・化学（流体計算、遺伝子情報相合検索など)、人工知能システムなど先端的な分野で活用され、実際に大きな成果を上げている。また PC クラスタによって高性能計算はより身近になり、並列処理の技術はますます発展している。しかし、それでも並列計算機は一部の専門家や学生にしか使われておらず、まだ一般的になっているとは言い難い。

そこで、まず並列処理に興味を持ってもらうために、複数のメディアを同期させコンテンツを作成する。教材として、視聴者に分かりやすく楽しみながら理解できるものを目指した。並列処理の基本的な概念をボーリングモデルを用いて、動画や音声、静止画、文字といった多種のメディアで表現する。

作成したアニメーションは 4 パターンで、人物の能力が同じ場合と異なる場合がある。能力が同じ場合は全員同じ人物で、4 人の場合と 10 人の場合を作成した。4 人の場合でまず、並列処理とはどういうものかを説明している。ここで、逐次処理と並列処理の違いを理解してもらう。10 人の場合では、負荷の違いによって、どのような結果になるのかを表した。負荷が等しい時と等しくない時を表現した。能力が異なる場合のモデルで、負荷均衡と同期のアニメーションを作成した。

動画作成には、3D Studio MAX を用いた。3D Studio MAX とは、映画やテレビ、ゲー

ムなど、幅広い分野で使用されている統合型動画作成アプリケーションである。また SMIL という同期マルチメディア言語を利用して、各メディアを同期させた。

作成した並列処理教材を Web 上に置いた。同期アニメーションのひとつをストリーミング配信し、ナローバンドとブロードバンドでの再生状態を比較した。

本論文では、2 章でストリーミングと同期マルチメディアについて述べる。3 章では、作成した並列処理教材の構成、アニメーションの作成、各メディアの同期方法について説明する。4 章では並列処理キーワードのアニメーション表現、5 章でストリーミング配信と並列処理教材としての考察を述べ、6 章で本論文のまとめとする。

## 2　ストリーミング技術と同期マルチメディア

### 2.1　ストリーミングとは

ストリーミングとはインターネットやイントラネットの上で、音声や画像のマルチメディアファイルをユーザが受け取りながらリアルタイムで再生する技術のことである。従来の方法では、ダウンロードが完了してからでないと再生することができなかった。文字データなどのファイルに比べ、音声や動画のファイルは非常に容量が大きいので、ダウンロードが終わるまでにかなりの時間がかかり、ハードディスクの領域を圧迫するという問題があった。しかし、ストリーミングではデータを受け取りながら再生するので、待ち時間も少なく、ハードディスクなどの領域を圧迫しないし、データを保存することがないのでコピーされる危険が少ない。また、テレビやラジオのように生放送も可能になった。ストリーミングに必要なのは、インターネットに接続できる環境とストリーミング再生に対応したソフトウエアだけで、誰でも気軽に楽しむことができる。しかし、ネットワークの帯域幅に準じた映像を配信することが求められているので、低速ネットワークの場合、映像や音声の品質低下を避けることができない。しかし、それらはケーブル TV ネットワークや、ADSL、光ファイバーネットワークなどの高速通信網の普及に伴い、解消される。また、同期マルチメディアを用いることで、情報の劣化を補うことが可能である。[4] [6]

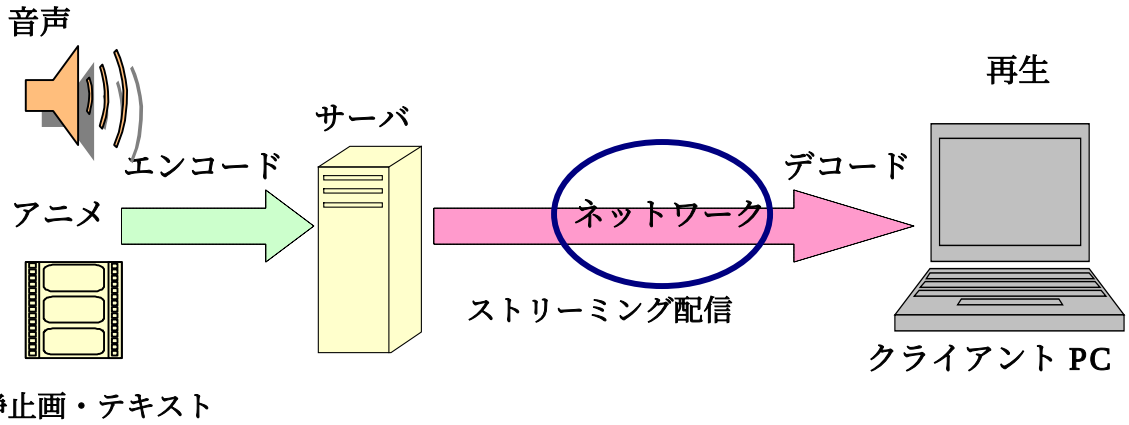


図 1：ストリーミングの概要

### 2.2　同期マルチメディア

#### 2.2.1　同期マルチメディアとは

同期マルチメディアとはインターネット上で動画や CG アニメーションに同期して、各種メディアを表示する技術のことである。これを利用することで、動画を配信する際の情

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補うことができる。

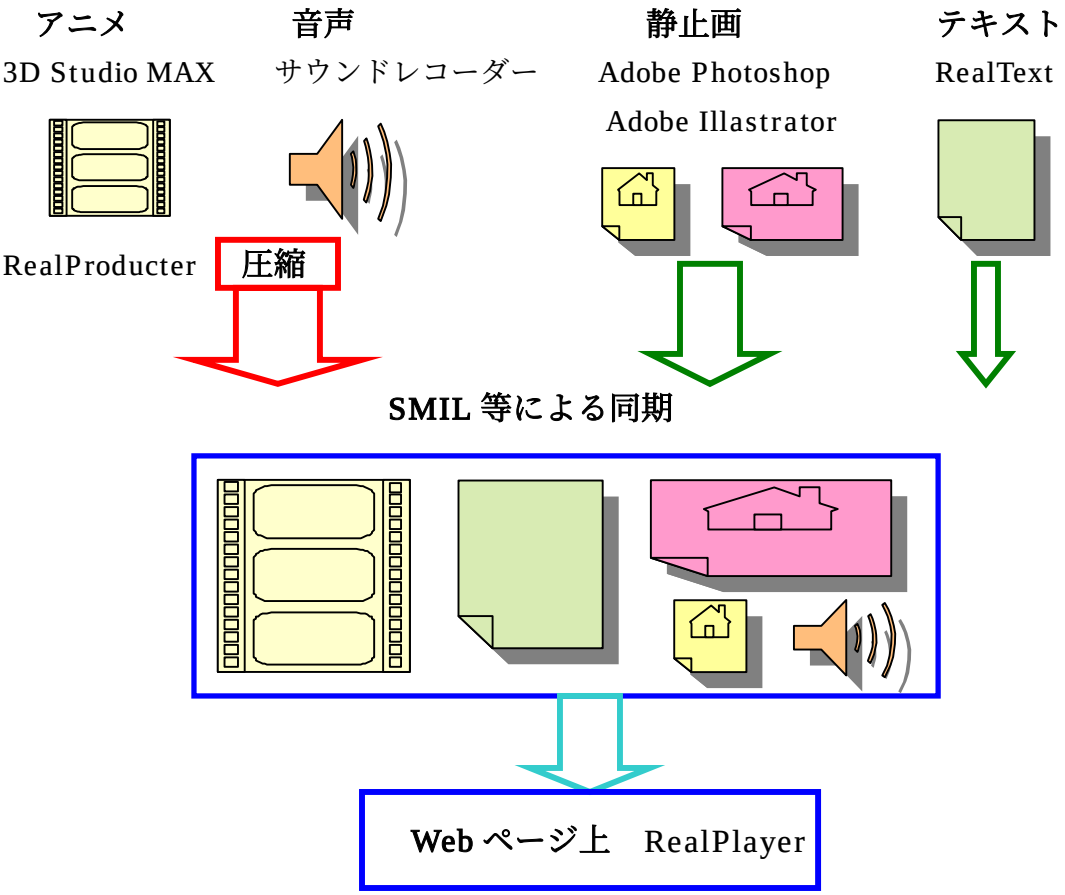


図 2：同期マルチメディアの概要

### 2.2.2　同期マルチメディア言語：SMIL

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)という言語が一般的に用いられる。SMIL は 1998 年に W3C(World Wide Web Consortium)によって SMIL1.0 として勧告されたもので、XML(extensible Markup Language)に依存するマルチメディアプレゼンテーションをレイアウトするスクリプト言語である。サウンドクリップとビデオクリップをタイムベースで同期をとったり、ユーザの接続帯域に応じてコンテンツを配信したりすることができる。[7] [8] [9]


図 3：SMIL の例

図 3 に簡単な SMIL の例を示す。これは、動画（heiretu1.rm）とテキスト(t1.rt)、静止画（L1.jpg）の同期をとったもの。

## 2.3 同期マルチメディアを用いた並列処理教材

同期マルチメディアを用いて、視聴者に分かりやすい並列処理教材を作成していく。並列処理の基礎概念を、アニメーションを通して説明する。

### 2.3.1 並列処理キーワード

並列処理におけるキーワードの説明

(1) 並列処理

1 台のプロセッサで処理を行う(逐次処理)よりも、複数台で同時に処理(並列処理)できれば、短い時間で処理できる。

(2) スケーラビリティ

プロセッサの台数を増やしたときに、台数増加に見合う性能向上が得られる可能性のことである。

(3) 粒度

並列処理の単位となる処理の大きさのことをいう。

(4) 負荷均衡

各プロセッサにかかる負荷をできるだけ均等にすることである。

(5) 同期

各プロセスの行う処理の速度が異なる場合、結果回収に時間的統一性をもたせて処理を終えることである。バリア同期とは、各プロセスの行う処理の処理速度が異なる場合、最も処理速度の遅いものに合わせて、全てのプロセスが処理の境界線に到達するまで一時的に処理を停止し、待つことである。

## 3　ボーリングモデルによる並列処理教材の作成

### 3.1　全体の流れ

ボーリングモデルを用いて、並列処理の基本的な概念を説明していく。全体の構成としては、まずメニューがあり、そこから選択されたアニメーションの説明が出て、ユーザの選択によってアニメーションを見ていくことになる。1 つのアニメーションの時間は約 1 分から 3 分ぐらいになっている。複数のメディアを用いることで視聴者に分かりやすく、楽しみながら理解できるものを作成する。

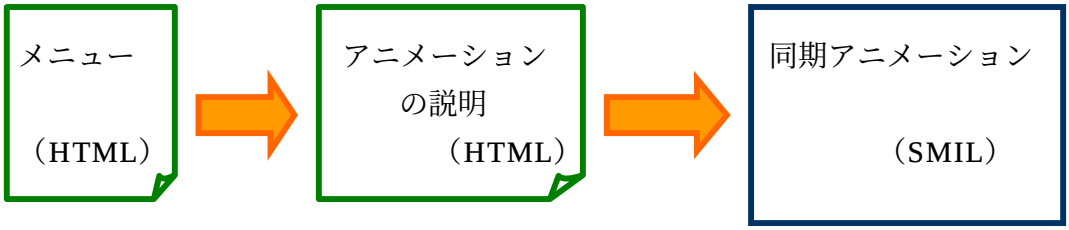


図 4：並列処理教材の構成

### 3.2　3D Studio MAX によるアニメーションの作成

アニメーションの作成には 3D Studio MAX を用いる。3D Studio MAX を使用することにより、3D のキャラクタ、あらゆる種類のオブジェクトやテーマを作成できる。キャラクタに動きを設定して、仮想世界のムービーを作成することができる。

図 4 は 3D Studio MAX の作成画面である。インタフェースは、上部のメニューバー、中央のビューポート、右端のコマンドパネルに分かれている。コマンドパネルよりさまざまなオブジェクトが選択でき、ビューポートで CG グラフィックを作成する。また、オブ

ジェクトはさまざまな変化が可能である。ビューポートの基本は 4 分割されており、左上のウィンドウがオブジェクトの上部から、右上が前部から、左下が左側方向から、右下がユーザ任意の視点から見ることができる。ビューポートもさまざまに変化させることができ、オブジェクトの右側、下部、後方からの視点からも見ることができる。ビューポートも 2 分割や分割なし、ユーザのカスタマイズ等が可能である。[1] [2]

図 5：3D Studio MAX のインタフェース

アニメーションの作成は、まずウィンドウのひとつのビューにオブジェクトを配置する。アニメーションさせたいオブジェクトに対して、アニメートボタンを押した後、オブジェクトを移動させる。すると、途中のフレームは自動に作成されるためフレームごとにアニメーションを作成する必要なく、滑らかなアニメーションを作成することができる。また、カメラなどの機能もあり、ムービーに合わせ自由に画面の切り替えができる。

本研究ではボーリングモデルを作成した。

(1)　人物の作成

人物の作成では細かい動きをつけるため関節ごとにパーツを作成した。まず適当なオブジェクトを配置し、モデルファイヤを使ってそのオブジェクトを変形させて形を作っていく。すべてのパーツを作成したら、それぞれに対して階層を設定する。階層を設定しないと、アニメーションを作る時バラバラになってしまう。スケマティクビューを使って、図 6 のように階層を設定すると、最上階の顔を動かせば、その下流にある耳、首、体、ズボンも一緒に動く。体を動かすと、その下流のオブジェクトは動き、首から上流が動くことはない。

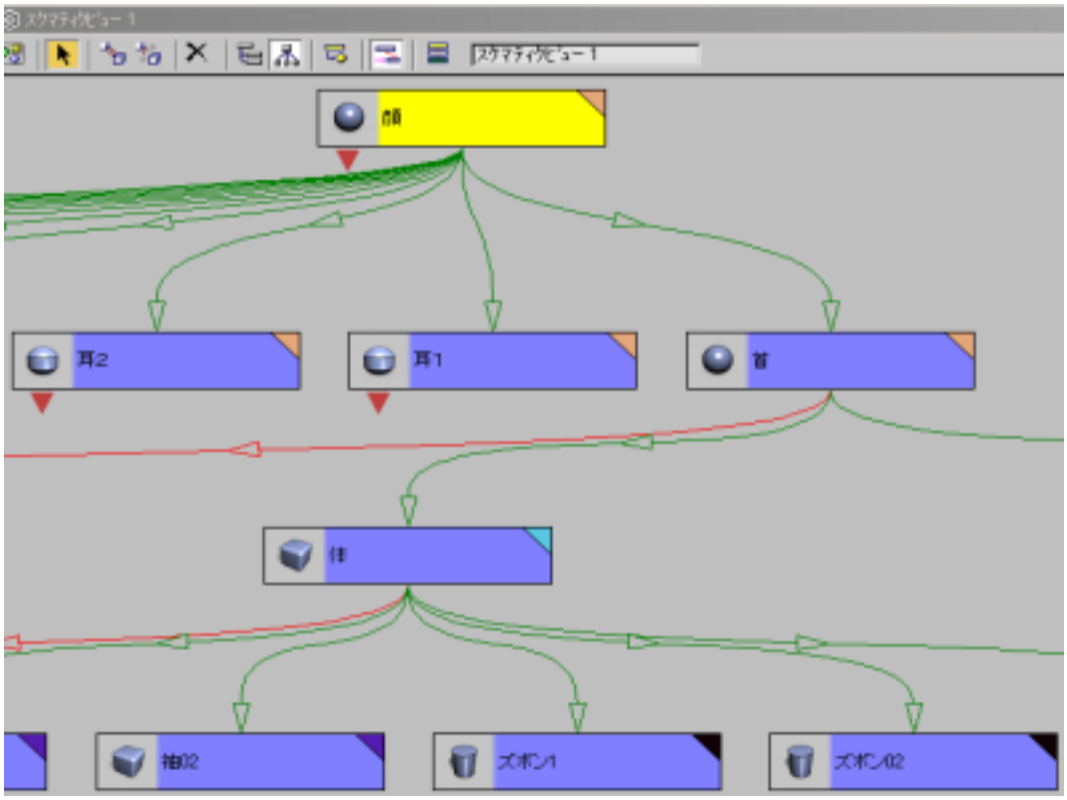


図 6：スケマティクビュー

(2)　ピンの作成

ピンは、まずスプライン外形線を作成する。スプライン外形線は、図 7 の右側のように横から見た状態でのピンの半分の外形線を作る。次にその外形線をスピンさせ完成させた。

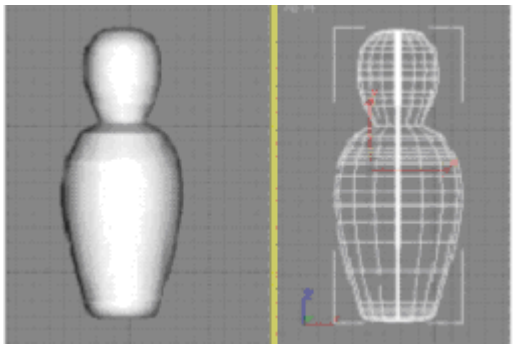

図 7：ピンの作成

(3) アニメーション

動画を作る場合はアニメートボタンを押す。これを押していないといくら動かしてもアニメーションにはならない。押し忘れて画像だけかわってしまうという失敗もある。アニメートボタンは押すと色が赤く反転する。アニメートボタンを押した後、物体の移動を完了させたいフレームまで下のバーを移動させ、そのフレームで物体を移動させる。

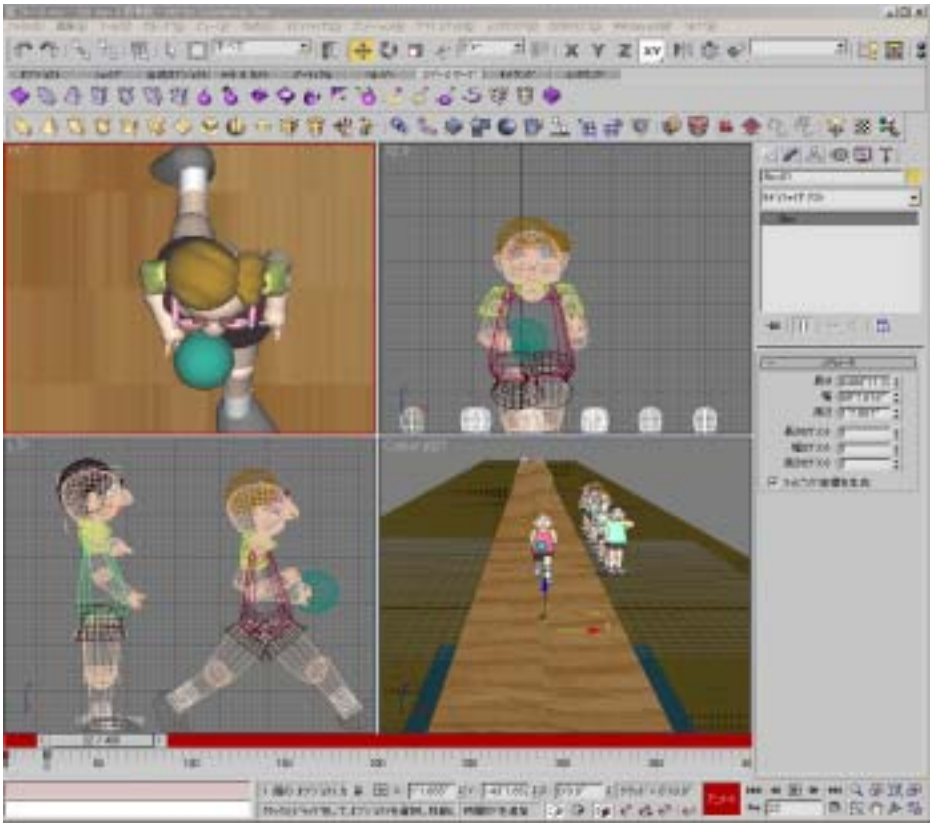

図 8：アニメート画面

人物は、動きにあわせて関節等も動かしていき、リアルな動きを目指した。カメラを設定し、人物やボールの動きにあわせて画面がきりかわるようにカメラ自体もアニメートさせていった。

(4) レンダリング

このソフトの場合さまざまなメディアに出力することができる。ここでは avi 形式で出力する。メニューバーにあるレンダリングを選択する。図 9 の共有パラメータの中の時間出力で出力するフレームの範囲を決定する。この場合は 0 フレームから 280 フレームまで出力することになる。また最下にあるビューポートで選択された画面が出力される。この場合は Camera021（アニメートしたカメラ）の画面がアニメーションとして作成される。


図 9：レンダリング

## 3.3 音声やテキストの同期

SMIL 言語を用いて CG アニメーションに、静止画、テキスト、音声等を同期させた。

同期画面のレイアウトは図 10 のようになっている。左側にメインメディアとして CG アニメーション、右側にテキスト、左下に静止画を出し、音声を加え説明する。[7] [9]


図 10：同期画面のレイアウト

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。

(1) 静止画の作成

静止画は jpg 画像を使用した。画像の作成には Adobe Illustrator、Adobe Photoshop 等を用いた。

(2) テキストの作成

テキストには、RealText を用いた。RealText では文字表示を時間指定できる。図 11 の場合は、開始時間が 1 秒、次の文字表示が 17 秒となり、一番上の行の duration=”0：18”が全体の時間を表す。拡張子は、.rt として保存する。

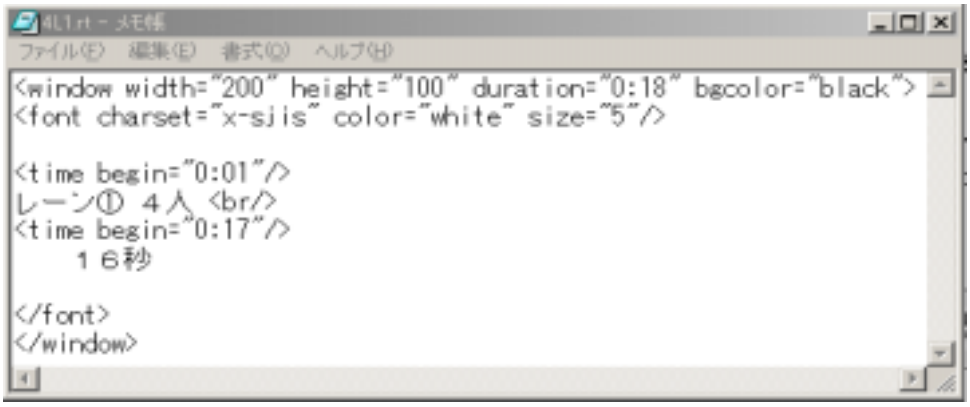


図 11：RealText の例

(3) 音声の作成

音声を付加するため RealAudio 形式のサウンドを使用した。RealAudio の作成には、Windows に標準で付属しているサウンドレコーダーを使用した（図 12)。これを使用して音声を録音し、いったん＊＊.wav 形式に保存する。


図 12：サウンドレコーダー

次に、これをストリーミングできるフォーマットに変換しなければならない。つまりエンコードしなければならないので RealProducer を使って、＊＊.rm 形式ファイルに変換した（図 13)。

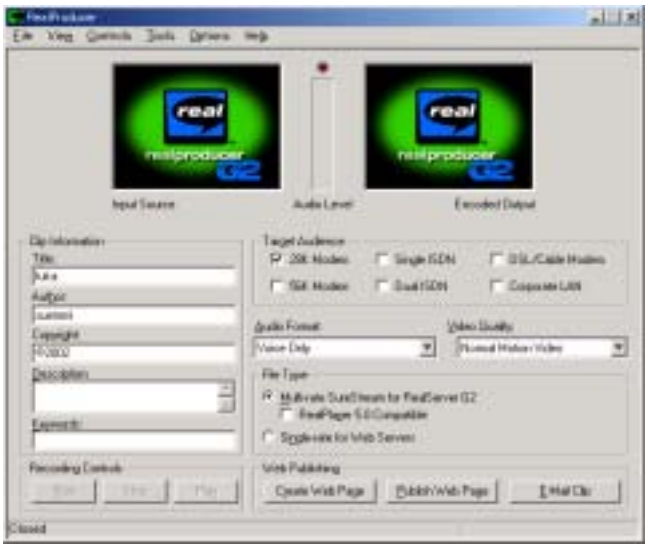

図 13：RealProducer

＊＊.wav 等の音声ファイル以外にも、＊＊.avi 形式のムービーファイルも同じ要領で変換できる。

## 4　並列処理キーワードのアニメーション表現

### 4.1　ボーリングモデルを用いたキーワードの表現

ボーリングモデルを用いて並列処理キーワードを表現する。レーンをプロセッサ、ボールを投げる人を仕事としてアニメーションを作成した。（図 14）


図 14：ボーリングモデル

キーワードにたいするアニメーションを表 1 に示す。

表 1：並列処理キーワードに対するアニメーション

| キーワード | アニメーション |
|---|---|
| 逐次処理 | 1 レーンで全員投げる |
| 並列処理 | レーン数を増やし分かれて投げる |
| 負荷均衡 | レーンごとの人物の能力が均等になるように分かれて投げる |
| 同期 | 自分の列の 1 人目（2 球）が投げ終えても、他のレーンの人が投げ終わるのを待って、一斉に 2 人目が投げ始める |

アニメーションのパターンは 4 パターンあり、人物の能力が同じ場合と異なる場合を作成した。能力は人物の動くスピードや、ボールの速度の違いによるもの。

能力が同じ場合は全員同じ人物で、4 人の場合と 10 人の場合を作成した。能力が異なる場合で負荷均衡と同期を表現した。アニメーションのパターンと含んでいる概念を表 2 に示す。概念を含む場合は○か◎、含まない場合は×としている。◎は理想的な場合を示す。

表 2：アニメーションのパターン

| 能力 | 人数 | 逐次処理（1 レーン） | 並列処理　（2，4 レーン） | | |
|---|---|---|---|---|---|
| | | | 負荷分散 | スケーラビリティ | 同期 |
| 同じ | 4 人 | ○ | ◎ | ◎ | × |
| | 10 人 | ○ | ○ | ○ | × |
| 異なる | 10 人 | ○ | ◎ | ◎ | × |
| | | × | ◎ | ○ | ○ |

## 4.2　人物の能力が同じ場合（4 人）

能力が同じ 4 人の場合で、並列処理とはどういうものかを説明した。(図 15) 初めに逐次処理の例として、1 レーンで 4 人全員がプレイする。(図 16)続いて、並列処理の例として、2 レーン、4 レーンに分かれてプレイする。2 レーンでは 1 レーンに 2 人ずつ、４レーンでは 1 人ずつと、レーンごとの人数が同じになる場合を作成した。(図 17)


図 15：能力が同じ 4 人の場合


図 16：逐次処理の例


図 17：並列処理の例

結果は 1 レーン 16 秒、2 レーン 8 秒、4 レーン 4 秒となり、レーン数に見合った速度向上が得られる。

また最後にスケーラビリティと粒度の説明が出てくる。


図 18：最後の画面

### 4.3　人物の能力が同じ場合（10 人）

10 人では、負荷の違いによって、どのような結果になるのかを説明した。10 人の場合、2 レーンではちょうど 5 人ずつに分けることができ、負荷は等しい。（図 19）4 レーンでは 2 人か 3 人で同数に分けることはできない。つまり、負荷は等しくない場合を表現した。（図 20）


図 19：2 レーンの場合図


図 20：4 レーンの場合

結果は 1 レーン 39 秒、2 レーン 20 秒、4 レーン 13 秒となり、2 レーンでは 2 倍になったが 4 レーンでは 3 倍にしかならなかった。負荷が等しくないと理想的な結果は得られないことを説明した。

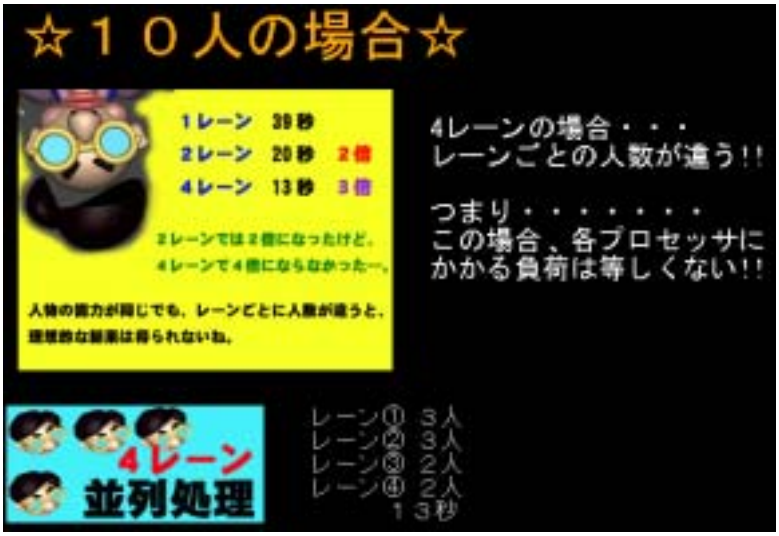


図 21：まとめの画面

### 4.4　負荷均衡

人物の能力が異なる場合で、負荷均衡のアニメーションを作成した。人物は 10 人で、それぞれ能力が異なるように作成した。動きとボールのスピードの速い方が、能力が高いとした。人物の能力は図 22 のようになっている。


図 22：人物の能力

この場合、人物の能力が等しい場合と違ってレーンごとの人数が等しくても、それに見合った速度向上が得られるわけではない。能力の低い人ばかりが集まってしまうと、そのレーンだけプレイしている時間が長くなってしまうことになる。そうなると、他の空いているレーンがもったいないし、全体に理想的な時間の短縮は望めない。そこで、負荷均衡という並列処理の重要な概念を、レーンごとの能力がなるべく均等になるように分かれて投げることで表現した。分かれ方は表 3 のようになっており、人物の番号は上の図の番号に対応している。

表 3：レーンの分かれ方

| レーン数 | レーン | 人物の番号 |
|---|---|---|
| 2 レーンの場合 | レーン① | 1　4　6　8　10 |
| | レーン② | 2　3　5　7　9 |
| 4 レーンの場合 | レーン① | 2　4　5 |
| | レーン② | 1　3　6 |
| | レーン③ | 7　9 |
| | レーン④ | 8　10 |

結果は 1 レーン 87 秒、2 レーン 43 秒、4 レーン 22 秒となる。各プロセッサにかかる負荷をできるだけ均等にしたことで、2 レーンで 2 倍、4 レーンで 4 倍と理想的な結果が得られる。


図 23：まとめの画面

## 4.5 同期

人物の能力が異なる場合で、同期のアニメーションを作成した。人物の分かれ方は負荷均衡の場合（表 3）と同様にした。自分の列の前の人（2 球）が投げ終えても、他のレーンの人が投げ終わるのを待っていて、一斉に次の人が投げ始めることで、同期を表現した。


図 24：同期アニメの 1 人目


図 25：2 人目

## 5　ストリーミング配信と考察

### 5.1　ストーミング配信

実際にストリーミングによるコンテンツの配信を行った。4 章の 4.2 で作成したコンテンツを Web サーバにとストリーミングサーバに置き、Web ページからリンクすることにより、ダウンロード配信とストリーミング配信する。

ストリーミングサーバには 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 サーバでは端末側でデータをすこしずつダウンロードしながら、その完了を待たずに届いたものから順に再生するようにした。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、途切れなく見ることができる。遅ければ、ダウンロード終了を待って再生するため時間がかかるが、映像はきれいに再生できる。

モデムや ISDN などのナローバンドと、ADSL などのブロードバンドでの再生状態を比較した。初期バッファ時間と再生状況を調べた。(表 4) コンテンツの容量は 740KB と 6MB である。

表 4：初期バッファ時間

| | ナローバンド | | ブロードバンド | |
|---|---|---|---|---|
| コンテンツ容量 | ダウンロード | ストリーミング | ダウンロード | ストリーミング |
| 740KB | 30～40 秒 | 10～20 秒 | 10 秒以内 | 10 秒前後 |
| 6KB | 30～60 秒 | 20～30 秒 | 10～20 秒 | 10 秒以内 |

ナローバンドでは、ダウンロード配信ではアニメの途中で何度もバッファリングが行われる。スピードストリーミング配信の方がバッファリングは少なかったが、アニメーションの動きが悪かった。ダウンロード配信の方がきれいに再生できる。ナローバンドでは全体的にコンテンツの状態はよくなかった。ナローバンドでの状態の悪さから、環境に対応させたコンテンツの作成が必要だと感じた。

ブロードバンドでは、どちらでもスムーズに見ることができた。再生開始までの時間は小容量のコンテンツではあまり変わらないことが判明した。大容量コンテンツになるとストリーミングの方が早く快適である。また、多くの同時アクセスに対してはストリーミング配信の方が早く、スムーズに、快適な状態でコンテンツを見ることができる。

また、Web サーバによる配信でも HTTP によって疑似ストリーミング配信を行うことができる。擬似ストリーミングとは、メタファイルとコンテンツを Web サーバに一緒に置いて配信する方法で、Web サーバではストリーミングソフトの費用を発生することなく Web 上でコンテンツファイルを配信することが可能である。この技術は大量のストリームを同時配信する大規模なサイトでは不向きであるが、多数の小さな Web サイトの場合には簡単に低コストで済む。この方法では、ストリーミングになってはるが回線速度も認識せず、コンテンツのコピーもされてしまうといったことが言える。今回はコンテンツ自体そんなに大きくなく、1 分程度のものなので、ダインロードでも疑似ストリーミングでもブロードバンド回線ではあまり、差はないと思われる。

## 5.2 並列処理教材としての評価と考察

作成した並列処理教材の考察を行う。

ボーリングモデルを用いての並列処理の概念の説明ということで、表現の仕方がポイントだった。レーンの数や人物の能力をかえることで、負荷分散や同期などを表現することができた。また、テキストや音声を同期させてはいるものの、数分のアニメのなかで、分かりにくいことや、見落としてしまうことはある。その場合、もちろん再度見てもらっても良いが、もう一度見直したりすることは面倒だったりする。そこで、アニメを見る前に人物の説明やレーンの分かれ方を説明する画面を出すことで、この問題を少しでも減らすことができると思う。また、重要な部分は表示する時間を長くしたり、アニメの最後にまとめとして結果を加えることで分かりやすくなった。全体的な構成はよかったとの意見をもらった。しかし、処理時間に関して、アニメの途中で出される処理時間の表示が短かっ

たようで、もう少し強調した方がよかったとの意見もあった。また作成したコンテンツは音声が常にあるわけではなくところどころ音がない場面がある。そのため、しばらく音の無い部分では少し退屈してしまうこともある。そういった部分にボールの音やなんらかの効果音つけた方がよかった。効果音をつけることで、少しの間でも音のない画面を見るよりは視聴者にとっては快適であることが分かった。

ボーリングモデルで、あまりかわいいとは言えないが変なキャラクタを作成し、音声等を加えることで、堅苦しくなく、気軽に見てもらえるものとなった。

## 6　おわりに

本研究では、同期マルチメディアを用いて視聴者が楽しみながら理解できるような並列処理教材の作成を目指した。この並列処理教材は、多くの人に並列処理という分野に少しでも興味を持ってもらいたいと思い作成した。ボーリングモデルで並列処理の概念をどう表現するか考え、より分かりやすいコンテンツを作成することに重点を置いた。使うメディア、配置等を考え、見やすい画面になるよう作成した。CG アニメーションに音声等を同期することで、アニメーションにある劣化情報を補うという点で効果があると思われる。また、同期技術を複数使用することで、多種のメディアを同時に表示することができ、そのテーマについて必要な情報を画面スペースの許す限り付加できることが分かった。アニメーションや音声などは見ていて飽きないなどコンテンツとして効果的であることも分かった。この教材を見て、並列処理に興味を持ってもらえれば、幸いである。

今後の課題として、今回は初心者を対象としたコンテンツを作成したが、様々なユーザに対して個別のものを配信していくことも考えられる。並列処理教育システムとして、マルチメディアシステムと e-Learning を組み合わせ、分かりやすく効率的に教育を行うシステムの構築があげられる。e-Learning を用いた教育システムには、WBT(Web based training)、PC を用いたリアルタイム遠隔授業、オンデマンドによる自由な時間の受講、ストリーミング配信などがあげられる。各ユーザのレベルに合わせて学習が進められたり、作成した並列プログラムの動作を可視化したアニメを同期マルチメディアを用いストリーミング配信をするといったものがある。また、SMIL での各メディアの相性や、快適なストリーミングに堪える新しいメディアの使用も検証できる。またユーザの環境を考え、快適なコンテンツの配信も課題である。

**謝辞**

本研究の機会を与えて下さり、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大塚さん、末冨さん、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] 3D Studio MAX チュートリアル, KINEXIT, 1997.

[2] 3D Studio MAX ユーザガイド, KINEXIT, 1997.

[3] 日高功雄：速習 Web テクニック FLASH5, 技術評論社, 2001.

[4] 大澤光：インターネットストリーミング, 共立出版, 2000

[5] リアルネットワーク社：http://www.jp.real.com.

[6] ストリーミング HOWTO：
http://www2h.biglobe.ne.jp/^hnakamur/technolab/stream.htm

[7] SMIL ファイルを作成する：http://homepage.nifty.com/sound-movie/r smil.htm

[8] 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.

[9] 末冨俊樹：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.

[10] 田村智樹：同期マルチメディアを用いた研究室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.

[11] 小畑健二：シンクロナイズド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修士論文, 2001.

[12] 川口智也：マルチメディアを同期させた研究キーワードの作成, 立命館大学理工学部情報学科卒業論文, 2001.

## 付録　SMIL ソースファイル

(1)　アニメ 4.2 の SMIL ファイル　(heiretu1.smi)

```
<smil>
<head>
<layout>
<meta name="title" content=" ” 卒業研究"/>
<meta name="author" content="onaga"/>
<meta name="copyright" content="(C)2002 onaga"/>

<root-layout height="500" width="680" background-color="black"/>
<region id="text" top="10" left="20" width="600" height="100" z-index="1"/>
<region id="movie1" top="80" left="20" width="320" height="240" z-index="2"/>
<region id="photo1" top="350" left="10" width="220" height="100" z-index="3"/>
<region id="photo2" top="350" left="10" width="220" height="100" z-index="4"/>
<region id="photo3" top="350" left="10" width="220" height="100" z-index="5"/>
<region id="photo4" top="80" left="20" width="320" height="240" z-index="6"/>
<region id="text1" top="100" left="360" width="310" height="350" z-index="7"/>
<region id="text10" top="100" left="360" width="310" height="350" z-index="8"/>
<region id="text2" top="350" left="240" width="200" height="100" z-index="9"/>
<region id="text3" top="350" left="240" width="200" height="100" z-index="10"/>
<region id="text4" top="350" left="240" width="200" height="100" z-index="11"/>
<region id="photo5" top="80" left="350" width="320" height="240" z-index="12"/>
<region id="photo6" top="350" left="500" width="130" height="100" z-index="13"/>
<region id="text5" top="350" left="10" width="680" height="100" z-index="14"/>
</layout>
</head>

<body>
<par>
<video src="heiretu1.rm" region="movie1" />
<textstream src="t1/t1.rt" region="text" fill="freeze"/>
<textstream src="t1/1.rt" region="text1" fill="freeze"/>
<textstream src="t1/2.rt" region="text10" begin="0:17" fill="freeze"/>
<textstream src="t1/13.rt" region="text2" fill="freeze"/>
<textstream src="t1/14.rt" region="text3" begin="0:18" fill="freeze"/>
```

```
<textstream src="t1/15.rt" region="text4" begin="0:27" fill="freeze"/>
<textstream src="t1/3.rt" region="text5" begin="0:35" fill="freeze"/>
<img src="photo/L1.jpg" region="photo1" fill="freeze"/>
<img src="photo/L2.jpg" region="photo2" begin="0:17" fill="freeze"/>
<img src="photo/L4.jpg" region="photo3" begin="0:27" fill="freeze"/>
<img src="photo/41.jpg" region="photo4" begin="0:33" fill="freeze"/>
<img src="photo/s.jpg" region="photo5" begin="0:35" fill="freeze"/>
<img src="photo/kf.jpg" region="photo6" begin="0:18" fill="freeze"/>

</par>
</body>
</smil>
```

(2)　テキストファイル　(1.rt)

```
<window width="310" height="350" duration="0:17" bgcolor="black">
<font charset="x-sjis" color="white" size="5"/>
<time begin="0:00"/>
1 レーンで全員投げる!<br/><br/>
<time begin="0:03"/>
これは、<br/>
<time begin="0:04"/>
１台のプロセッサで問題を処理していく
<font color="yellow">
<time begin="0:06"/>
逐次処理
</font>
の例
</font>
</window>
```