

# 音楽演奏情報による仮想演奏者の CGアニメーション生成

松本 英明      後藤 真孝      村岡 洋一

早稲田大学 理工学部

〒169-8555 東京都新宿区大久保 3-4-1

E-mail:{matumoto, goto, muraoka}@muraoka.info.waseda.ac.jp

**あらまし** 本稿では、仮想ジャズセッションシステム VirJa Session における、仮想のベーシストとドラマーのリアルタイム CG アニメーション生成について述べる。従来の研究では、CG キャラクタは人間の指示に従属的な動作をするだけであった。それに対し、本研究の仮想演奏者は自らの即興演奏に基づいて自律的に動作する。その動作を基本動作の組み合わせで実現し、動作接続部分は前後から補間する手法を提案する。また、基本動作を選ぶ際の曖昧性を解消しながら、演奏音と動作の同期を可能にする手法も提案する。これらの手法は VRML と Java を用いて実装され、演奏音に同期した演奏動作を行なう CG アニメーションを生成する際に、効果的であることが確認された。

## Generating CG Animation of Virtual Players by Musical Performances

Hideaki Matsumoto      Masataka Goto      Yoichi Muraoka

School of Science and Engineering, Waseda University

3-4-1 Ohkubo Shinjuku-ku, Tokyo 169-8555, JAPAN.

E-mail:{matumoto, goto, muraoka}@muraoka.info.waseda.ac.jp

**Abstract** This paper describes a real-time CG animation system which displays the virtual bassist and drummer in our virtual jazz session system, called VirJa Session. In most previous systems, CG characters were directly controlled by human directions. In our system, virtual players can autonomously play their instruments with motions according to their improvisation. The player's movement is generated by combinations of several basic motions. We propose a method which interpolates a transient motion in connecting successive basic motions and a method which synchronizes motions with musical sounds. Experimental results showed that these methods implemented on VRML and Java were effective enough for CG animation of the virtual players improvising their performances.

## 1 はじめに

仮想空間内に存在する仮想生物を、3次元コンピュータグラフィックスによるキャラクタ(以後、CGキャラクタと表記する)で表現する試みが近年盛んになってきた。その場合、人間とCGキャラクタとの円滑なインタラクションのためには、どのようにしてCGキャラクタの柔軟な動作をリアルタイムに生成できるかが、大きな課題となっていた。

一方、音楽情報処理の分野において、計算機に演奏を生成させ、人間とセッション(合奏)をさせようとする試みがある。演奏を生成する計算機は、人間とインタラクションする仮想演奏者と見なすことができるが、従来研究では演奏音のみを考慮し、CGキャラクタによる表現までは考慮されていなかった。音楽に反応するCGキャラクタに関する研究事例[3, 4, 5]もあるが、人間の演奏や指示に対して従属的にCGキャラクタが動作するだけであった。

本稿では、自律的に即興演奏をおこなう仮想演奏者(ベーシストとドラマー)を、CGキャラクタとして表現する際の動作生成手法を提案する。これは、我々の提案する仮想ジャズセッションシステムVirJa Session[1, 2]の一構成要素となっている。VirJa Sessionは、人間-計算機間の視聴覚を通じたマルチモーダルなインタラクションをジャズセッションにおいて実現するシステムである。仮想演奏者は人間の演奏を理解し、その上で自らの演奏を生成して、演奏音と演奏動作のアニメーションを出力する。人間はその出力に反応して、さらに次の演奏に変化を加えていく。

このようなインタラクションを実現するには、リアルタイムにCGキャラクタの多様な動作を生成しなければならない。従って、モーションキャプチャで収集したデータをそのまま利用することは不可能である。そこで本研究では、各演奏音を発音するための短い基本動作を予め多数用意し、それらの基本動作を組み合わせることで、柔軟な動作生成を実現する。その場合、基本動作を接続する際に不連続になるという問題があるが、基本動作をオーバーラップさせてその区間を補間する手法によりこれを解決する。

また、本研究の仮想演奏者は自律性を持っているため、自分がおこなう即興演奏をどういった演奏動作で表現するか、またその演奏とどのように同期して表示するかを解決する必要がある。例えば、仮想のドラマーがスネアドラムを演奏する動作を生成する場合、それを右手と左手のどちらで叩くのかという曖昧性を解消しなければならない。また、腕を振りあげ、振り下ろし、元の位置に戻すという一連の

動作を表示する際に、演奏音と腕を振り下ろした瞬間を同期させなければならない。そこで、ある演奏音を弾く動作とその次の演奏音を弾く動作のあらゆる組合せに対してコストを定義し、一連の動作におけるコストの和を最小にする組合せを求めることにより、曖昧性を解消する手法を提案する。演奏音と動作の同期に関しては、各演奏音のタイムスタンプ付きの情報を、実際に発音する約一拍前までに演奏生成部から取得し、その一拍の期間内に動作のスケジューリングをする手法によって解決する。

VirJa SessionはLANに接続された計算機上に分散実装されており、仮想演奏者のアニメーションシステムの実行環境には、パーソナルコンピュータ上のVRMLブラウザを用いた。移植性が高くなるように、システムはVRMLとJavaで記述されている。実験の結果、提案した手法が有効であり、仮想のベーシストとドラマーが即興演奏をおこなう動作を、リアルタイムに生成できることを確認した。

## 2 人間と計算機によるセッションシステム—VirJa Session

### 2.1 概要

VirJa Sessionは人間と計算機が演奏情報を介したインタラクションによってセッションを行なうシステムである。VirJa Sessionを構成する以下の機能については、文献[1, 2]において既に報告されている。

#### ●演奏解析・生成機能

仮想演奏者が人間の演奏を理解し、その演奏に対して仮想演奏者の演奏情報を生成する機能。

#### ●演奏入力機能

人間の演奏を入力する機能。

#### ●演奏出力機能

生成された仮想演奏者の演奏を演奏音として出力する機能。

そこで本稿では、演奏情報に加え、視覚的な情報を用いたインタラクションを可能にするために不可欠な以下の機能について述べる。

#### ●仮想演奏者のアニメーション機能

音楽演奏生成機能によって、生成された仮想演奏者の演奏から、CGキャラクタの動作生成を行い、アニメーション表示する機能。

### 2.2 システム構成

VirJa Sessionは人間のピアニスト、計算機による仮想ベーシスト、仮想ドラマーからなるトリオで構成されたジャズセッションを対象とし、LAN上に分散したワークステーション、パーソナルコンピュータ群に実装された(図1)。2.1節で述べられ

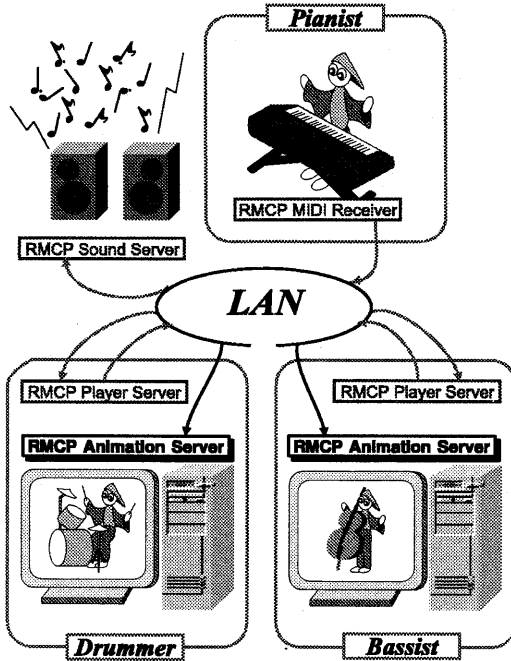


図 1: VirJa Session 構成図

た機能はそれぞれ別のプロセスとして実装され、演奏情報をお互いに通信することでシステムを実現した。演奏情報は MIDI (Musical Instrument Digital Interface) として表現し、各プロセス間の通信には RMCP [3, 6] を用いた。

本稿で扱う仮想演奏者のアニメーションシステムは演奏解析・生成機能で生成された演奏情報を入力とし、出力として仮想ベーシストは 4 弦ベースを演奏する様子が、仮想ドラマーはドラムセットを演奏する様子が表示される。

### 3 仮想演奏者の動作生成

#### 3.1 基本動作の組み合わせ

仮想演奏者のアニメーション表示はインタラク션을考慮しなければならないため、ミュージックビデオで導入されるように、予め用意された画像を切替えて用いる手法は使用できない。そこで仮想演奏者を骨格モデルで表現し、各関節角の変化によって動作を表現する。

骨格モデルをもった CG キャラクタの動作生成には、しばしばモーションキャプチャが使われる。しかし仮想演奏者の動作は、インタラクションによって演奏中に動的に生成される演奏情報から生成しなければならないので、モーションキャプチャで収集したデータをそのまま利用することはできない。そこで本研究では、各演奏音の演奏情報に対応する基本動作を多数用意し、基本動作を組み合わせる

ことですべての動作生成を実現する。

基本動作とは、1 つの演奏音を発音するための一連の動作のことである。例えば、ドラマーがドラムを叩く場合は、腕を振り上げ、ドラムを叩き、元の位置に戻すといった動作であり、ベーシストがベースを弾く場合は、指先を弦の位置まで動かし、弦を弾き、元の位置に戻す動作である。ドラムを叩いたり、ベースの弦を弾いたりといった楽器に触れた瞬間をインパクトと呼ぶ。基本動作は、人間の演奏者の演奏中の動作をモーションキャプチャで収集した関節角変化の時系列データから、基本動作部分を切り出して用意した。

#### 3.1.1 基本動作の接続

演奏情報に基づいて基本動作を選択して組み合わせるだけでは、基本動作間の動作が考慮されないために動作の連続性が欠如する。結果として、基本動作を切り替えたフレーム間で CG キャラクタは全く異なった姿勢をとっていることになる。だからといって基本動作間の動作を新たに用意するのは、基本動作の組み合わせが膨大であるため現実的ではない。そこで、基本動作をオーバーラップさせて、その区間を補間することにより基本動作間の動作を生成する。

基本動作の接続の概念図を図 2 に示す。接続区間の動作は、予め計算によって求めることはできないので、演奏中にリアルタイムに計算しなければならない。毎秒 20 フレームのアニメーションであれば、1 フレームの計算は 50msec. 以下に抑えなければならない。现阶段では画像処理に大幅に時間を費やすので、基本動作の接続は時間のかからない計算が要求される。そこで、接続区間のみ動作の補間を行なう。

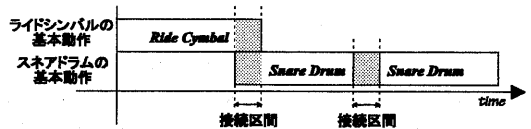


図 2: 基本動作の接続の概念

接続区間の動作を補間する際に、単に前後の中間値をとるのではなく、基本動作 A から基本動作 B までの動作を重み付けを変えながら補間する。接続区間の開始時には基本動作 A が 100%，基本動作 B が 0%，接続区間の終了時には基本動作 A が 0%，基本動作 B が 100% になるような重み付けをする。このことにより、接続区間以外には影響が及ぼすことなく、基本動作の接続が可能になる。

#### 3.2 自律性を持った仮想演奏者の動作

本研究の仮想演奏者は、自分の演奏をどのような動作で表現するか自分で決定するという自律性を持っている。そのため、以下の二つの問題が生じる。

●動作の曖昧性の問題

演奏情報にはどちらの手で演奏するかという情報は含まれないため、曖昧性がある。例えば「スネアドラムを叩く基本動作」には右手で叩く基本動作と左手で叩く基本動作の2つの基本動作が存在する。

●演奏音と動作の同期の問題

各基本動作中に設定されたインパクトと演奏音の開始時刻を同期させる必要がある。例えば、ドラマーが腕を振り上げ、振り下ろして叩き、元の位置に戻す、といった一連の動作の中で、腕を振り下ろして叩くインパクトの瞬間が、演奏音に同期していなければならない。

3.2.1 動作の曖昧性の問題の解決

動作の曖昧性の問題は、基本動作間にコストを定義して、コストを最小にするような演奏動作を一意に選択することで解決する。3.2.2節で述べるように、演奏情報は一拍先まで得ることができるので、一拍先までのあらゆる基本動作の組合せからコストが最小になるような組合せを選択する。

□仮想ベースの基本動作間コストの定義

仮想ベースは、4本の弦を持つアコースティックベースを、左手で弦を押えながら右手で弦を弾く動作をする。左手の指で弦を押さえる位置によって音高が変わるが、違う弦でも同じ音高の演奏音を発音させることが可能である。従って、演奏情報の音高から押さえる位置を一意に決定できないという問題がある。

ここでは、仮想ベースの演奏動作間コストを演奏動作間の指の移動距離で定義する。以下の指の移動をコスト1とする。

- 同一弦上の1音分の移動
- 1つ隣の弦への移動

例えば、E弦のFからA $\sharp$ (Bb)に指が移動する場合はコスト5で、E弦のFからA弦のA $\sharp$ (Bb)に指が移動する場合はコストが1なので、後者を選択する(図3)。

例外として開放弦<sup>1</sup>を用いる場合は、同一弦上ならば移動距離を0とし、実際のベースと同様に開放弦を優先的に利用できるようにコストを設定する。

□仮想ドラマーの基本動作間コストの定義

仮想ドラマーは、バスドラム、スネアドラム、ハイタム、ロータム、フロアタム、ハイハット、ライドシンバル、クラッシュシンバルで構成されるドラムセットを、両手両足を用いて叩く動作をする。左

<sup>1</sup> 左手の指を触れない場合の弦を言う。4本弦のベースの場合、E2, A2, D3, G3(実際の音は1オクターブ低い)がそれにあたる。

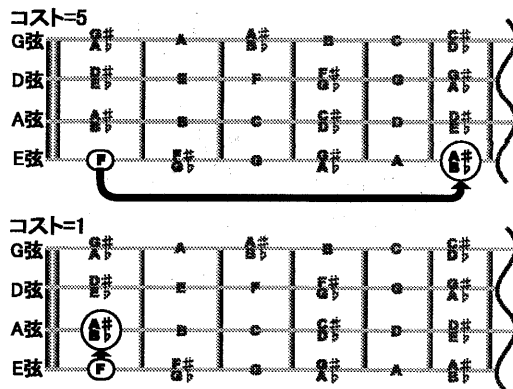


図3: ベーシストのコスト定義例

右の足でバスドラムとハイハットのペダルを踏む動作は曖昧性がないため、基本動作を一意に決定できる。しかし、手に持ったスティックで演奏する場合には、多くの楽器が左右いずれの手でも叩くことが可能である。そのため、例えば「スネアドラムを叩く」という演奏情報のみでは、基本動作を一意に決定できない。

そこでまず、前の基本動作がおこなわれていないときに次の基本動作を始める場合のために、各楽器に対してどちらの手の基本動作を優先的に用いるかを事前に定めておく。これを演奏情報がない状態からのコストとし、ドラムセットの配置からどちらの手で演奏がしやすいかを考慮して、優先的な手のコストを0に設定する。このコストは、演奏開始時や他の演奏者のソロなどでよく用いられる。

次に、前の基本動作の途中で連続して次の基本動作を始める場合のために、各基本動作間のコストを設定しておく。その際、同じ楽器を叩く動作が続く場合、同じ手で叩き続けずに左右の手を交互に使う方が演奏しやすいことを考慮する。例えば、同じ楽器の連打では、右手→右手、左手→左手よりも左手→右手→左手…のコストを小さく設定しておく。

これらを考慮し、前の基本動作から次の基本動作へのコスト表を経験的に作成した。本研究では、15×14のコスト表を作成した<sup>2</sup>。例としてスネアドラムに関する部分を抜粋したものを表1に示す。例えば、1拍の間にスネアドラムを4回連続して叩く動作が続く場合には、以下のようなコストになる。

$$\text{左手} \rightarrow \text{左手} \rightarrow \text{左手} \rightarrow \text{左手} \quad (2+2+2=6)$$

$$\text{左手} \rightarrow \text{右手} \rightarrow \text{左手} \rightarrow \text{右手} \quad (1+0+1=2)$$

そして、この場合は後者の基本動作列が選択される。

<sup>2</sup> スネアドラム、ハイタム、ロータム、フロアタム、ハイハット、ライドシンバル、クラッシュシンバルの7種類の楽器を左右で叩く組合せ(14通り)があるので、基本的には14×14となっている。ただし前の基本動作に限り「演奏情報なし」の場合があるため、15×14となる。

表 1: 仮想ドラマーの演奏動作間のコスト表 (抜粋)

前の基本動作	次の基本動作	
	SD(右手)	SD(左手)
SD(右手)	3	0
SD(左手)	1	2
演奏情報なし	9	0

### 3.2.2 演奏音と動作の同期の問題の解決

演奏音と動作の同期の問題は、演奏情報を表現する際に演奏音の発音時刻のタイムスタンプを付け、そのタイムスタンプの時刻に、演奏動作のインパクトを同期させることによって解決する。タイムスタンプ付きの演奏情報の通信に関しては、RMCPの時間管理機能を活用することで容易に実現できる。

そこで、演奏生成部からタイムスタンプ付きの演奏情報を、一拍前までに取得しておくようにする。そして、その一拍の間にコストを用いたスケジューリングをおこない、そのタイムスタンプの時刻よりも前に各基本動作を開始する。例えば、基本動作が始まってから腕を振り上げて下ろし、インパクトの時刻になるまでに 200msec. の時間があれば、発音時刻の 200msec. 前から基本動作を開始すればよい。

## 4 実装

仮想演奏者のアニメーションシステムを LAN に接続されたパーソナルコンピュータ (MMX Pentium 200MHz) 上に実装した。

アニメーションシステムは移植性が高くなるように、VRML と Java で記述し、VRML ブラウザ<sup>3</sup> 上に実装した。VRML には Interpolator と Time-Sensor を組み合わせて、線形補間によるアニメーションを実現する機能がある。基本動作をこの機能の活用によって実現することも検討したが、基本動作を接続する手法が実現困難なため、本研究では計算した関節角を各関節の Transform ノードの rotation フィールドに直接与えることによって、仮想演奏者の動作を表現している。

処理の流れを図 4 に示す。入力された演奏情報を基に、動作の曖昧性の解消の手法を用いて基本動作を選択する。演奏情報のタイムスタンプを基に演奏音との同期するようなタイミングで、選択した基本動作に対応するモーションキャプチャで収集した関節角のデータを、各関節の rotation フィールドに与える。このとき、基本動作が重なる接続区間であれば、基本動作の接続手法を用いて、計算した関節角のデータを各関節の rotation フィールドに与える。

<sup>3</sup> VRML Browser は日本アイ・ビー・エムの RVE Browser を使用した。RVE Browser は QoS 制御が可能である [7, 8, 9]。

VRML ファイルには、仮想演奏者の骨格構造と体及び楽器のポリゴンデータが用意されている。仮想演奏者は骨格構造の各関節の rotation フィールドに与えられた関節角の変化により動作する。結果は、VRML ブラウザ上に CG キャラクタのアニメーションとして表示される。

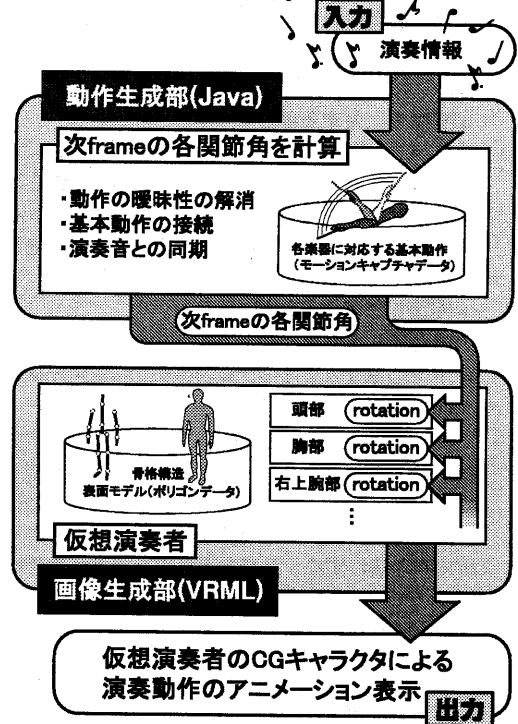


図 4: 処理の流れ

## 5 実験

実装したシステムを用いて、演奏情報の RMCP パケットを LAN 上の他のワークステーションより受信しながら仮想ベーシスト、仮想ドラマーを表示させた。その結果、図 5 に示すアニメーション表示画像を得ることができた。フレームレートは最大で毎秒 20 フレーム程度を得ることが出来た。

このフレーム数は、一般的なアニメーションに劣るが、仮想演奏者のアニメーション表示としては十分に機能しているといえる。フレーム数はマシン性能に依るところが大きく、また VRML ブラウザを使用している関係上、マシンの性能を最大限使用する最適化を行なうことが不可能なため、フレーム数を増やすことは困難になっている。今後、マシンの性能の向上によりこれらの問題は解決できると考えられる。

提案する手法の有効性を確認するために、仮想ドラマーに対して以下の実験を行なった。

●実験 1

スネアドラムを 16 分音符で 4 回連打する演奏情報を送信する。テンポは 120bpm である。

●実験 2

スネアドラムと 16 分音符で 4 回連打する演奏情報と同時にライドシンバルを 8 分音符で 2 回連打する演奏情報を送信する。テンポは 120bpm である。

上記の実験において、以下の結果を得た。

●実験 1 に対する結果

左手→右手→左手→右手で演奏音と同期してスネアドラムを叩くアニメーション映像を得ることができた。このときに、上腕の関節角の変化を記録した。関節角の変化を記録した結果を図 6 に示す。

●実験 2 に対する結果

演奏音と同期して左手で 4 回スネアドラムを、右手で 2 回ライドシンバルを叩くアニメーション映像を得ることができた。

図 6 からは、基本動作を接続する手法が接続区間以外に影響を与えることなく、基本動作を接続していることがわかった。ドラマーの動作から、動作の曖昧性の手法によって同じ楽器の連打に対して同じ手で叩き続けずに、左右の手を交互に使う動作が選択されていることがわかった。また映像と音声は違和感を感じない程度に同期していたので、演奏音と動作の同期の手法も有効であることがわかった。

## 6 おわりに

本稿では、演奏情報から自立性を持つ仮想演奏者の動作を生成する手法について述べた。動作は基本動作の組み合わせによって表現し、基本動作の接続、動作の曖昧性の解消、演奏音との同期に関する手法が有効であることをベーシストとドラマーの CG により確認した。

現在、様々な人が設計した世界中の仮想演奏者をダウンロードして仮想演奏者とのセッションが楽し



図 5: 仮想ベーシスト, 仮想ドラマーの表示例

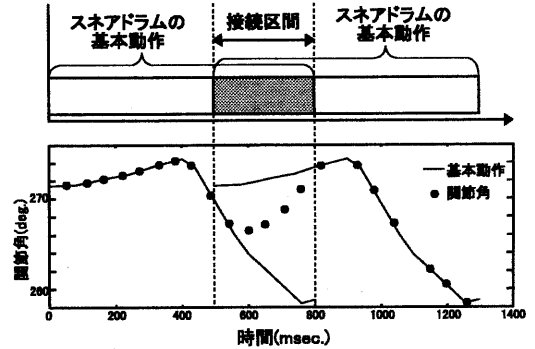


図 6: 動作の接続例 (右上腕部 z 軸回転)

めるシステム [10, 11] の構築が演奏音に関して進められている。今後は、その CG アニメーションに関しても本研究に基づいて実現していく予定である。

### 謝辞

本研究に対し、有益な御協力を頂いた根山 亮氏、緒方 美音子氏、渋谷 薫氏、大森 哲也氏、また、VRML ブラウザの開発者の視点から大変有益な御助言を数多く頂いた、日本アイ・ピー・エム 東京基礎研究所の皆様へ感謝する。本研究の一部は、情報処理振興事業協会「独創的ソフトウェア育成事業」による。ここに記して謝意を示す。

### 参考文献

- [1] 後藤 真孝, 日高 伊佐夫, 松本 英明, 黒田 洋介, 村岡 洋一: 「すべてのプレーヤーが対等なジャズセッションシステム I システムの全体構想と分散環境での実装」, 情報処理学会 音楽情報科学研究会研究報告, 96-MUS-14-4, Vol.96(1996).
- [2] 日高 伊佐夫, 後藤 真孝, 村岡 洋一: 「すべてのプレーヤーが対等なジャズセッションシステム II ベーシストとドラマーの実現」, 情報処理学会 音楽情報科学研究会研究報告, 96-MUS-14-5, Vol.96(1996).
- [3] 後藤 真孝, 橋本 祐司: 「MIDI 制御のための分散協調システム—遠隔地間の合奏を目指して—」, 情報処理学会 音楽情報科学研究会研究報告, 93-MUS-4-1, Vol.93, No.109(1993).
- [4] 後藤 真孝, 村岡 洋一: 「音楽に踊らされる CG ダンサーによるインタラクティブパフォーマンス」, 日本ソフトウェア科学会 コンピュータソフトウェア, Vol.14, No.3, pp.20-29(1997).
- [5] 村井 克之, 山内 康晋, 土井 美和子: 「音楽を用いた対話的 3 次元 CG アニメーション—ミュージックドリブン CG の試作—」, 情報処理学会 ヒューマンインタフェース研究報告, No.74-1(1997).
- [6] 後藤 真孝, 根山 亮, 菊地 淑晃, 村岡 洋一: 「RMCP:Remote Media Control Protocol—時間管理機能の拡張と遅延を考慮した遠隔地間の合奏—」, 情報処理学会 音楽情報科学研究会研究報告, 97-MUS-21-3, Vol.97, No.67(1997).
- [7] 吉田 亮: 「RVE システムによるマルチユーザ環境」, 情報処理学会第 55 回全国大会 2AA-5(1997).
- [8] 谷口 雅昭: 「RVE ブラウザにおける QoS 制御」, 情報処理学会第 55 回全国大会 2AA-7(1997).
- [9] QoS-able VR System, <http://www.tr1.ibm.co.jp/projects/s7730/QoSVRsys2/index.htm>
- [10] 根山 亮, 後藤 真孝, 村岡 洋一: 「WWW 上の異なる計算機環境で動作する音楽セッションシステム—VirJa Session on WWW へ向けて—」, 第 54 回情報処理学会全国大会, 7J-5(1997).
- [11] 根山 亮, 後藤 真孝, 村岡 洋一: 「WWW 上で公開可能な分散音楽情報処理システム—ダウンロードした Java アプレットのための安全なローカル通信方法—」, 日本ソフトウェア科学会第 14 回大会, B4-1(1997).