

# すべてのプレイヤーが対等な 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

後藤 真孝 日高 伊佐夫 松本 英明 黒田 洋介 村岡 洋一

早稲田大学 理工学部

{goto, hidaka, matumoto, ykuroda, muraoka}@muraoka.info.waseda.ac.jp

あらまし 本稿では、すべてのプレイヤーが対等な立場でインタラクションし、即興演奏するジャズセッションシステムについて述べる。従来の多くのシステムでは、人間のソロ演奏に対して計算機が他のプレイヤー全員の演奏を伴奏としてまとめて生成していたため、ソロに追従する域を出なかった。我々は、計算機内のプレイヤー同士も人間同様にお互いの演奏を聞き合い反応することができ、さらにジェスチャーも視覚的に交換することができるシステムを提案する。対象はジャズのピアノトリオとし、人間がピアニスト、計算機がベーシストとドラマーを担当する。両計算機プレイヤーは独立したプロセスとして複数の計算機上に実装され、実際にプレイヤーが対等な立場で演奏できた。

## A Jazz Session System for Interplay among All Players

### I. System Overview and Implementation on Distributed Computing Environment

Masataka Goto Isao Hidaka Hideaki Matsumoto Yosuke Kuroda Yoichi Muraoka

School of Science and Engineering, Waseda University  
3-4-1 Ohkubo Shinjuku-ku, Tokyo 169, JAPAN.

**Abstract** This paper presents a jazz session system where each player is independent and can interplay with other players. Most previous systems reacted to only human player's performance with the fixed master-and-servant relationship. Our system enables computer players to listen to other computer players' performance as well as human performance, and to interact each other. Moreover, all players can communicate not only by listening others' performance, but by looking at others' bodies and gestures. In our current implementation, the system deals with jazz piano trio consisting of a human pianist, a computer bassist and a computer drummer. These computer players have been implemented as independent processes on several distributed workstations.

## 1 はじめに

ジャズセッションにおいて重要なのは、すべてのプレイヤーが対等な立場で、お互いの演奏を聞き合いながら即興演奏をすることである。これはインタープレイ (interplay) とも呼ばれ、プレイヤーが主従関係を持たずに、相互に相手の音に耳を傾けて音で反応する。本研究は、このようなインター

プレイを人間と計算機上の複数のプレイヤー間で実現することを通じて、人間と計算機が共に演奏する際のインタラクションの本質を探求することを目的とする。

従来のジャズセッションシステム [1]–[3] や自動伴奏システム [4]–[9] の多くでは、人間のソロ演奏に対して、計算機が他のすべてのプレイヤーの演奏を伴奏としてまとめて生成していた。そこでは、

人間の演奏に追従して適切に伴奏することに重点が置かれていたため、あくまで人間が主であり計算機が従であった。また、計算機が担当するプレイヤーは人間の演奏しか聞けなかったため、計算機のプレイヤー同士はインタラクションできなかった。一方文献 [10] では、マルチエージェントモデルによる計算機のプレイヤー同士のインタラクションに言及しているが、音楽聴取過程の実現が中心であり、実際の演奏の生成はおこなっていなかった。

本稿では、すべてのプレイヤーが対等な関係にあり、計算機が担当するプレイヤー同士もインタラクションできるジャズセッションシステムについて述べる。本システムでは、計算機のプレイヤーは人間の演奏に反応するだけでなく、ソロを弾くなどの自己主張もおこなう。さらに、実際の人間同士のセッションにおいて各プレイヤーが自分以外のすべてのプレイヤーの演奏を聞くように、計算機内のプレイヤー同士も人間の演奏と同様にお互いの演奏に耳を傾け、影響を与え合って演奏することができる。

本稿ではさらに発展形として、各プレイヤーがお互いの姿を見ることができるジャズセッションシステム VirJa Session (Virtual Jazz Session System) を提案する。計算機のプレイヤーが演奏する姿は、コンピュータグラフィックス (CG) により表示される。これにより、人間は音を聞くだけでなく相手の動きも見ることができ、より臨場感のあるセッションをおこなえる。同時に、計算機の各プレイヤーはカメラを通じて人間が演奏する姿を見ながら、通信によって仮想的に、他の計算機プレイヤーの姿やジェスチャー (身体的な合図) も見る。こうして、すべてのプレイヤーは演奏の終りなどを指示するジェスチャーを交換することができ、聴覚と視覚の両方を通じたインタラクションを実現できる。

現在の実装では、本システムはジャズのピアノトリオの演奏を対象とし、人間がピアニスト、計算機がベーシストとドラマーを担当する。そして、テンポが一定で演奏される 4 ビートのジャズスタンダード曲を扱う。計算負荷の分散のために、本システムをネットワークに接続された複数の計算機上に実装した。その際、実装を容易にし拡張性を高くするために、ベーシストとドラマーを独立したプロセスとして実現し、さらにこれらの演奏理解・生成部と CG 表示部も別々のプロセスとした。実験の結果、実際にプレイヤーが聴覚と視覚を通じてインタラクションしながら即興演奏するセッションをおこなうことができた。

## 2 すべてのプレイヤーが対等なジャズセッションシステム

ジャズセッションでは、すべてのプレイヤーがお互いの即興演奏を聞き合い、固定した主従関係を持たずに対等な立場でインタラクションすることが重要である。そこで以下では、まずこのようなジャズセッションのモデルを提示した後、そのモデルをジャズのピアノトリオへと適用する。

### 2.1 ジャズセッションのモデル

すべてのプレイヤーが対等なジャズセッションシステムは、図 1 のようなモデルに基づいている。ここではカルテット (4 人) の場合を例示したが、一般に  $n$  人 ( $n$  は 2 以上の整数) の場合に適用できる。プレイヤーは自分以外のすべてのプレイヤーの演奏を聞き、状況に応じて音で反応したりソロを演奏したりする。

各プレイヤーを人間が担当するか計算機が担当するかはモデルとしては自由である。例えば二人の人間がプレイヤー 1 とプレイヤー 2 を担当するときには、計算機はプレイヤー 3 とプレイヤー 4 を担当する。さらに、すべてのプレイヤーを人間が担当してもよいし、すべてのプレイヤーを計算機が担当してもよい。すべてを人間が担当する前者の場合は、複数箇所にいるプレイヤー間でのセッションなどに有効である。これは、6 で後述するようにコンピュータネットワークを介して演奏情報をやりとりすることで実現できる。すべてを計算機が担当する後者の場合は、様々な個性を持たせたプレイヤーや異なる人が設計したプレイヤー同士を一緒に演奏させるセッションなどに有効である。そこでは、計算機上のプレイヤーがいわば設計者

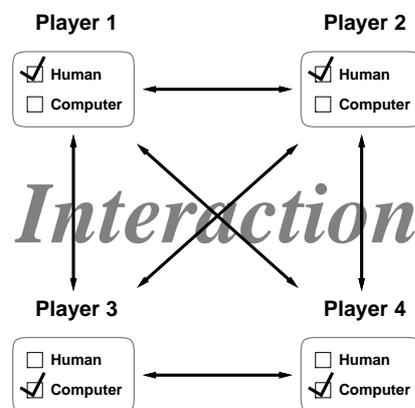


図 1: すべてのプレイヤーが対等なジャズセッションのモデル

の分身となり、それらの分身同士のインタラクションがおこなわれる。

## 2.2 ジャズのピアノトリオへの適用

本研究では、以上のモデルをジャズのピアノトリオへ適用したシステムを実現する(図2)。プレイヤー1のピアニストを人間が担当し、プレイヤー2のベーシスト、プレイヤー3のドラマーを計算機が担当する。

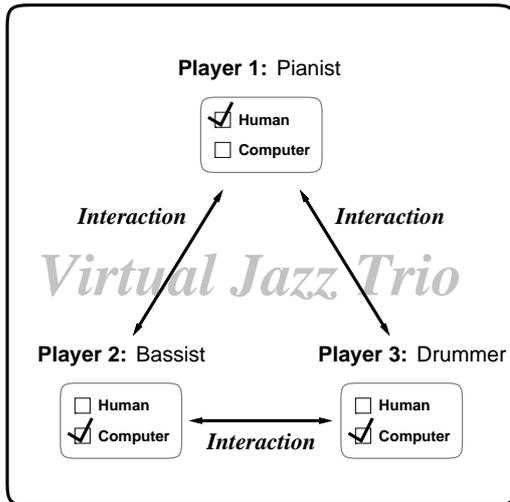


図2: ピアノトリオのセッションのモデル

ここで重要なのは、計算機がプレイヤー2とプレイヤー3を共に担当する場合にも、それらは独立したプレイヤーとして処理されることである。プレイヤー2は、プレイヤー1の人間の演奏だけでなく、プレイヤー3の計算機の演奏にも反応して演奏する。つまり、プレイヤー1の演奏だけを解釈して、プレイヤー2,3の演奏を従属的に決定するのではない。ここでは、プレイヤー2(ベーシスト)とプレイヤー3(ドラマー)が演奏によって影響を与え合うことができる。

## 3 姿の見えるジャズセッションシステム — VirJa Session

図2のモデルを発展させた、各プレイヤーがお互いの姿を見ることができるジャズセッションシステム VirJa Session のモデルを図3に示す。2で述べたプレイヤーは、頭脳以外には演奏を聞くための耳と演奏するための手足だけしか持っていなかった。ここで提案する VirJa Session のプレイヤーは、さらに他のプレイヤーを見るための目と自分の状態を表す体全体を持つ。これにより各プレイヤー

は演奏音以外にも視覚情報を交換することができ、より人間同士のジャズセッションに近付いた、聴覚と視覚の両方による臨場感のあるインタラクションを実現できる。また、VirJa Session の演奏を観客として楽しむ立場からは、すべてのプレイヤーの姿が見えることで、ライブ演奏を見ているときのようにプレイヤーの存在を感じることができる。

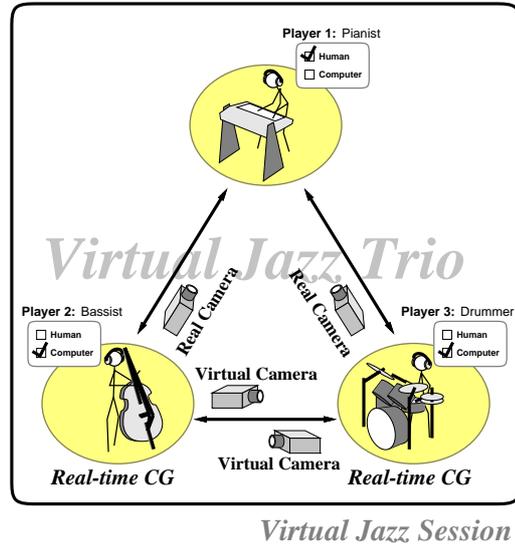


図3: 姿の見えるピアノトリオのセッションのモデル

計算機が担当するベーシストとドラマーの演奏する姿・ジェスチャーは、三次元CGアニメーションによってリアルタイムに表示され、人間は演奏中にこれらを見ることができる。一方、計算機の両プレイヤーは、図3のように実際のカメラにより人間が演奏する姿をとらえ、ジェスチャーを認識する。同時に、通信で実現される仮想的なカメラにより他方の計算機プレイヤーの姿を見る<sup>1</sup>。こうして、各プレイヤーはお互いにジェスチャーを交換することができ、演奏音だけでは伝えにくい情報を介した協調が可能になる。

### 3.1 CGによるプレイヤーの可視化

CGの画面上にはプレイヤーとその演奏する楽器(ベースまたはドラムス)が表示され、プレイヤーは演奏音に合わせた演奏動作をおこなう。さらに両計算機プレイヤーは、他のプレイヤーとのコミュニケーションをとるために、次節で述べる二種類のジェスチャーの動作やうなずく動作をしたり、他のプレイヤーに視線を向けたりする。また曲のビートに合わせて体を揺らし足踏みなどもおこなう。

<sup>1</sup>カメラによるジェスチャー認識の結果に相当する情報を通信する。

なお、CGによってミュージシャンの姿を可視化する試みとしては、文献[11]や文献[12]があげられるが、ギタリストの手の動きの生成が中心となっていた。また、文献[13]では提案だけに留まっていた。

### 3.2 カメラによる視覚を通じたインタラクション

VirJa Sessionでは、テーマ演奏部・各プレイヤーのソロ演奏部・フォーバース(4 verse)部を組み合わせた演奏進行表を事前にシナリオとして与えることができる<sup>2</sup>。そして、シナリオ中の各部の繰り返し回数(何コーラス演奏するか)は、最大回数を制限するだけで固定とはせず、ジェスチャー等を用いた演奏中のインタラクションによって動的に決定できるようにする。一般に人間同士のセッションにおいても、シナリオは事前に打合せして決めておくが、このシナリオ中の各部の繰り返し回数は演奏時に決定されることが多い。

本システムではそのための合図として、次のプレイヤーのソロ演奏部に移るように指示するジェスチャーと、テーマ演奏部に移る(戻る)ことを指示するジェスチャーの二種類を用いる。前者は体を左右に傾けるジェスチャーで、傾けた側にいるプレイヤーにソロを指示する<sup>3</sup>。後者は手で頭を指すジェスチャーにより指示する。

計算機プレイヤーは、人間のこのようなジェスチャーをカメラ画像から認識して判断する。また計算機プレイヤー同士は、通信によりジェスチャーを交換して判断する。一方、人間はCGによって可視化された計算機プレイヤーのジェスチャーを見て判断する。現在の実装では、計算機プレイヤーは、シナリオ中のソロ演奏部やフォーバース部で最大繰り返し回数に達したときに、次へ移ることを人間に知らせるためにジェスチャー動作をおこなう<sup>4</sup>。

計算機の担当するベーシストとドラマーは視線を持っており、シナリオ中の各部において終りに近付くと、再び繰り返すか次に移るかを判断するために頻繁に他のプレイヤーへ視線を向ける<sup>3</sup>。人間がジェスチャーを計算機プレイヤーに伝えたいと

<sup>2</sup>例えば次のようなシナリオに沿って演奏できる: テーマ演奏部 ⇒ ピアノソロ演奏部 ⇒ ベースソロ演奏部 ⇒ フォーバース部(ピアニストとドラマーが交互に4小節間ソロを演奏) ⇒ テーマ演奏部。

<sup>3</sup>三人のプレイヤーはお互いに向き合うように三角形の各頂点に配置され、この配置はあらかじめ与えておく。

<sup>4</sup>まだ実現はされていないが、例えば人間のピアノソロを計算機プレイヤーが低調だと判断したときには、計算機の方から次のベースソロ演奏部へ移るジェスチャー動作をするように設計してもよい。

ときには、視線が自分を向いている間におこなわなければならない。ジェスチャーを受けとって了解した計算機プレイヤーは、相手に対して確認の意味でうなづく動作を返す。

## 4 分散環境での実装

提案したシステムを実現するには、大別して以下の六種類の処理が必要となる。

1. 計算機プレイヤーが演奏を理解・生成する処理(計算機プレイヤーの頭脳)
2. 演奏全体のテンポを管理する処理
3. 計算機プレイヤーの演奏を人間が聞こえるように音として出力する処理
4. 人間の演奏を計算機プレイヤーに伝えるために入力する処理
5. 計算機プレイヤーのCG画像を表示する処理(計算機プレイヤーの体全体)
6. カメラにより人間のジェスチャーを検出する処理(計算機プレイヤーの目)

以上の処理を単一プロセスで実現するのは、計算量が多いために現実的でない。そこで本研究では、これらの各処理を担当する複数のプロセスをサーバ・クライアント・モデルに基づいて実装し、演奏情報などをお互いに通信することでシステムを実現した。これは、

- 各プロセスはそれぞれに特化した処理に専念すればよいため実装が容易になる。
- 6で後述するように拡張性が高く、提案したモデルを実現する上で適している。

という利点がある。

現在の実装では、ネットワーク(Ethernet)に接続された3台のワークステーション(SGI Indigo2 Impact/Extreme)上でシステムは動作する。演奏の入力と出力はMIDI(Musical Instrument Digital Interface)によりおこない、プロセス間の通信には次に述べるRMCP(Remote Music Control Protocol)[13]を用いた。

### 4.1 RMCP

RMCPは、MIDIとLAN(Local Area Network)を融合した分散協調システムにおけるサーバ・クライアント間の通信プロトコルであり、UDP/IP上のコネクションレスのプロトコルである。これはシンボル化された音楽情報をネットワークを通じて伝送するために設計された。本実装においてRMCP

は、次の四種類の情報を RMCP パケット<sup>5</sup>として伝送する役割をする。

- 演奏情報 (MIDI メッセージ)
- ビート情報 (四分音符の時刻と現在のテンポ)
- ジェスチャー情報 (体を左に傾けた, 体を右に傾けた, 手で頭を指した)
- アニメーション生成情報 (ジェスチャー動作指示, うなづく動作指示, 視線方向指示)

クライアントが RMCP パケットをすべてのサーバへ向けて送信(ブロードキャスト)することにより、演奏情報などをネットワーク上の様々なサーバで共有して同時に活用できる<sup>6</sup>。さらに、サーバを単に追加するだけで新たな機能を実現できるので、システムの実装や拡張が容易になる。

RMCP は文献 [13] の時点ではリアルタイム性を全く確保していなかったが、本実装ではジャズセッションをリアルタイムにおこなうために、RMCP パケットのヘッダー部にタイムスタンプ<sup>7</sup>を付加できるようにプロトコルを拡張した。これによりサーバは、タイムスタンプの時刻前に到着したパケットをその時刻通りに処理することができ、それらの時間順も保証できるようになった。

前述の六種類の必要な処理を担当する RMCP サーバ・クライアントの一覧を表 1 に示す。名前に“Server”が含まれるものが RMCP サーバである。これらのサーバ・クライアントの運用例について 4.2, 4.3 で順に説明する。

## 4.2 すべてのプレイヤーが対等なジャズセッションシステムの実装

すべてのプレイヤーが対等なジャズセッションシステムを RMCP により実装した構成図を図 4 に示す。これは、表 1 の 1.~4. で構成される。前述した RMCP パケットが伝送する四種類の情報のうち、ここでは演奏情報とビート情報が用いられる。

<sup>5</sup> RMCP が通信に用いるパケット。演奏情報を伝送する場合には、MIDI メッセージをパケット中に含む。RMCP パケットはクライアントからサーバへ片方向に送信され、サーバからクライアントへの返答は原則としてない。

<sup>6</sup> 文献 [14] では分散環境での実装の際のプロセス間通信モデルをいくつか考察しているが、RMCP のモデルは正確にはどれにも該当せず、文献中の「多対多」モデルと「黒板」モデルの中間に位置する。つまり、RMCP はコネクションレスであるために「多対多」モデルで議論されていたような全プロセス間のリンクは必要でなく、ブロードキャストで情報を共有するために「黒板」モデルのような黒板を管理するプロセスは存在しない。プロセスが黒板を介さずに直接通信するという意味では「多対多」モデルを受け継いでいるが、全プロセス間で情報を共有するという意味では「黒板」モデルを受け継いでいる。

<sup>7</sup> 現在の実装では 1ms 単位の絶対時刻とした。ただし、NTP(Network Time Protocol)等により、事前に計算機間で時刻を合わせておく必要がある。

表 1: RMCP サーバ・クライアント一覧

処理内容	サーバ・クライアント名
1. 演奏理解生成	RMCP Player Server
2. テンポ管理	RMCP Beat Provider
3. 演奏音出力	RMCP Sound Server
4. 演奏入力	RMCP MIDI Receiver
5. CG 出力	RMCP Animation Server
6. カメラ入力	RMCP Camera Analyzer

人間や計算機のプレイヤーの演奏情報は、図の中央のネットワーク (Ethernet) 上に RMCP パケットとしてブロードキャストされる。このパケットをすべてのサーバが受信することで、演奏情報を共有できる。これは、全プレイヤーがお互いの演奏を聞き合っ共有することに相当する。

人間がピアニストとして MIDI 楽器の鍵盤を弾くと、その演奏情報 (MIDI メッセージ) は、RMCP MIDI Receiver によりネットワーク上に RMCP パケットとしてブロードキャストされる。ベーシストとドラマーを担当する二つの RMCP Player Server は、この演奏情報を受信しながら (演奏を聞きながら)、自分の演奏情報をブロードキャストする (自分の楽器を演奏する)。この際、他方の RMCP Player Server の演奏情報も同様に受信して考慮する。これらの演奏情報を含む RMCP パケットを RMCP Sound Server が受信すると、パケット中の MIDI メッセージを MIDI 楽器に送ることにより実際の音として出力する。両 RMCP Player Server はテンポに合わせて演奏を理解・生成しなければならないが、これは RMCP Beat Provider が送信するビート情報に基づいて演奏することで実現する。

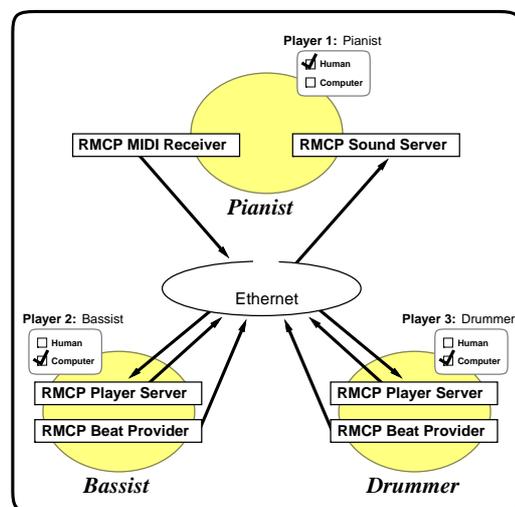


図 4: RMCP によるジャズセッションシステムの構成図

#### 4.2.1 RMCP Player Server

ベーシスト用とドラマー用のそれぞれの RMCP Player Server は、他の二人のプレイヤーの演奏情報と自分用のビート情報を RMCP パケットとして受信し、一拍ごとにその内容を過去の演奏情報と共に分析する。そして、分析結果に基づいて次の演奏情報を決定し、それを実際に鳴る時刻の一拍前までにブロードキャストする。この際、各 RMCP パケットには、それが実際に処理されるべき時刻を表すタイムスタンプを付加する。

本サーバは大きく分けて、演奏理解部と演奏決定部、演奏生成部で構成される。演奏理解部では、各プレイヤーの演奏をその楽器の演奏特性に応じた方法で分析して、プレイヤーの主張度を求める。演奏決定部では、次の主張度等を予測しながら、自分がその後の演奏においてどれくらいの主導権を握るかを決定する。その際、シナリオ中のどの部分を現在演奏しているかも考慮する。最後の演奏生成部では、音程・音量・リズムそれぞれについてのパターンのデータベースから、決定した主導権を実現できるようなパターンを選出して組み合わせ、実際の演奏情報 (MIDI メッセージ) を生成する。これらの具体的な実現上の課題や解決法については、文献 [15] に詳しい。

#### 4.2.2 RMCP Beat Provider

RMCP Beat Provider は、各拍の位置 (四分音符の時刻) を表すビート情報を順番に RMCP パケットとしてブロードキャストする。このパケットは、タイムスタンプとして四分音符の時刻を持ち、実際の時刻の一拍前までに送信される。RMCP Player Server は、この時刻に基づいて、他のプレイヤーの演奏を分析し自分の演奏音の時刻を決定する。

現在の実装では演奏のテンポは一定としたため、本サーバは常に同一間隔のビートの情報を送信する。ベーシストとドラマーのそれぞれに RMCP Beat Provider を用意してあるが、両者は同じ内容のビート情報を送信する<sup>8</sup>。

### 4.3 姿の見えるジャズセッションシステムの実装

姿の見えるジャズセッションシステム VirJa Session を RMCP により実装した構成図を図 5 に示す。4.2 に対し、二つの RMCP Animation Server と一つの RMCP Camera Analyzer が追加されて

<sup>8</sup>演奏開始時に両者の RMCP Beat Provider は同期をとる。

いる。RMCP Camera Analyzer を二つ用意する実装も考えられるが、現時点では正面から人間のジェスチャーを検出すれば十分なので、一つのビデオカメラを両計算機プレイヤーが共用している。以上の追加に伴い、RMCP パケットとして、演奏情報とビート情報に加えてジェスチャー情報とアニメーション生成情報も伝送される。ジェスチャー情報は、RMCP Camera Analyzer から両 RMCP Player Server へ送信されるか、両 RMCP Player Server 間で通信される。後者は、仮想的なカメラを通じて他方の計算機のジェスチャーを見ることに相当する。アニメーション生成情報は、プレイヤーの可視化用に演奏音以外の動作情報を伝達するために用いられ、RMCP Player Server から対応する RMCP Animation Server に送られる。これは、計算機プレイヤーの頭脳から体全体への動作指令に相当する。

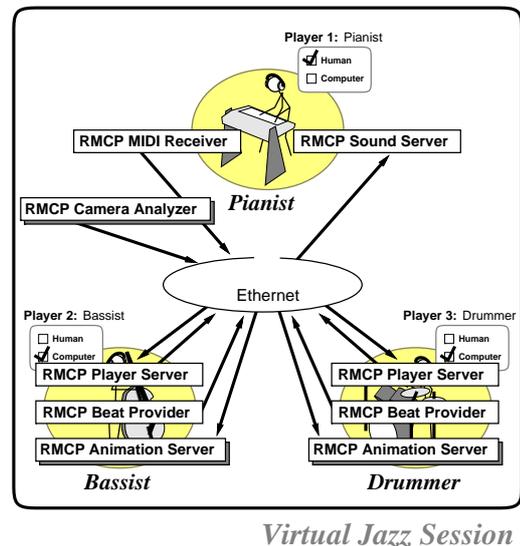


図 5: RMCP による姿の見えるジャズセッションシステムの構成図

VirJa Session のために、4.2 で述べた RMCP Player Server を次の三点で拡張する。

- RMCP Camera Analyzer や他方の RMCP Player Server が送信するジェスチャー情報を受信し、シナリオ中の各部の繰り返し回数を動的に決定する。そして、確認のためのうなずく動作をアニメーション生成情報として送信する。
- ソロ演奏部やフォーバース部で最大繰り返し回数に達したときに、次へ移るためのジェスチャー情報を送信すると共に、そのジェスチャーに相当するアニメーション生成情報も送信する。

- シナリオ中の各部において終りに近付くと、頻繁に他のプレイヤーへ視線方向を切替えるためのアニメーション生成情報を送信する。また、他のプレイヤーの主張度が高いときなどに視線を向けることもおこなう。

#### 4.3.1 RMCP Animation Server

RMCP Animation Server は、演奏情報とアニメーション生成情報、ビート情報の RMCP パケットを受信し、その内容に応じてリアルタイムに変化するプレイヤーの姿を三次元 CG で表示する。対応する RMCP Player Server から演奏情報を受信すると、その音を自分の楽器で演奏する動作を生成する。アニメーション生成情報を受信すると、その指示に応じてジェスチャー動作やうなづく動作、視線を向ける動作を生成する。また、RMCP Beat Provider から受信したビート情報により、体を揺らしたり足踏みをする動作を生成する。

本サーバは大きく分けて、プレイヤー共通の RMCP 処理部と画像生成部、プレイヤー個別の動作決定部とモデリングデータから成る。RMCP 処理部では、パケット中のタイムスタンプの時刻に実際の動作が起きるように、時間管理等をおこなう。動作決定部では、ベーシストの場合には、左手のポジションと右手で弾く弦を、前の音からの指の移動量などを考慮しながら決定する。ドラマーの場合には、足の動作は演奏情報から一意に決定し、手でどの打楽器を叩くかは曖昧性があるため動的に決定する。これらの処理の詳細に関しては、紙面の都合上、別の機会に報告する。

#### 4.3.2 RMCP Camera Analyzer

RMCP Camera Analyzer は、演奏する人間の上半身を正面からビデオカメラで撮影して解析し、二種類のジェスチャーを検出する。そして、この結果得られるジェスチャー情報をブロードキャストする。

本サーバは、頭部位置推定部とその頭部位置を用いるジェスチャー検出部から成る。頭部位置推定部では、まず事前に取り込んだ背景画像との差分を二値化し、 $x$  軸 (水平軸) 方向のヒストグラムを求める。次に、ヒストグラム中で最大値をとる  $x$  座標を頭部の  $x$  座標 ( $H_x$ ) とし、最大値自体を頭部の  $y$  座標 ( $H_y$ ) と推定する。ジェスチャー検出部では、 $H_x$  が左右に大きく変動すれば、体をその方向へ傾けるジェスチャーをしたとみなす。また、( $H_x, H_y$ ) 周辺の色が大きく変化すれば、手で頭を指すジェスチャーをしたとみなす。

## 5 実験結果

実際に VirJa Session を運用してジャズセッションをおこなった。実験では、ジャズピアノ歴 5 年 (ピアノ歴 21 年) のピアニストが、4 ビートのジャズのスタンダード曲「Take the “A” Train」を即興演奏した。シナリオとしてはベースソロ演奏部やフォース部 (ピアノとドラムスのソロ) を含むものを設定し、演奏用の鍵盤および音源にはシンセサイザー (Korg 01/W FD) を用いた。

実験の結果、ピアニストが主導権を握るだけでなく、他のプレイヤーも主導権を握ってソロ演奏などをおこなうといった、従来のシステムに比べて全プレイヤーがより対等なセッションを実現できた。演奏音だけのセッションと比較すると、相手の姿が見えることで臨場感が増し、ピアニストは実際にライブ演奏をしているときに近い感覚を得られた。また、プレイヤー間でのジェスチャーを介した視覚的なインタラクションも達成できた。

本実装に用いた RMCP がジャズセッションシステムの通信プロトコルとして有効に機能することを確認する一環として、異なる計算機上にあるクライアントからサーバへ RMCP パケットが到達するまでの遅延時間を測定した<sup>9</sup>。その結果、平均値 0.30ms、標準偏差 0.06ms、最小値 0.28ms、最大値 1.24ms であった。これから、RMCP は速度的に MIDI に遜色なく情報伝送できており、ブロードキャストやタイムスタンプ機能を活用しながら効果的に実装できることがわかった。

## 6 拡張性に関する考察

提案したモデルを分散環境で実装したことによって、どのように拡張性が高くなったかを具体的に考察する。

### 図 1 のモデルの実現

2.1 で述べたモデルにおいて、各プレイヤーを人間と計算機のいずれが担当してもよいことを述べた。例えば、図 2 のモデルにおいてベーシストの担当を計算機から人間に変更するには、図 4 の計算機担当用 (ベーシスト用) の二つのプロセス (RMCP Player Server, RMCP Beat Provider) を人間担当用の二つのプロセス (RMCP MIDI Receiver, RMCP Sound Server) に単に置き換えるだけでよい。ピアニストの担当を人間から計算機に変更する場合も、人間担当用を計算機担当用 (ピア

<sup>9</sup>MIDI note on メッセージを 10000 回伝送して測定した。

ニスト用に新たに実装) に逆に置き換えればよい。また、プレイヤーを新たに追加する場合には、人間担当用か計算機担当用のプロセスを追加し、各 RMCP Player Server の演奏理解部を新たなプレイヤーにも対応するように修正すればよい。以上は VirJa Session の場合にも同様に適用できる。

#### テンポ変化への対応

現在は演奏テンポを一定としたため RMCP Beat Provider を用いているが、テンポが変化する演奏に対応するよう拡張するには、これを他のプレイヤーの演奏のテンポ変化に追従できるような RMCP Beat Tracker に置き換えるだけでよい。

#### 複数箇所にいるプレイヤー間でのセッション

今回は一箇所の実験したが、RMCP によりネットワークを利用して実装したことで、複数箇所にいるプレイヤー間にも自然に拡張できる。この場合には、各プレイヤーのいる LAN 間で RMCP パケットを中継する。

#### 様々な個性を持つ計算機プレイヤー間でのセッション

RMCP をプラットフォームとして、様々な人が各自の思いを込めた RMCP Player Server を実装すれば、それらのプレイヤーを接続しているいろいろな形態のセッションがおこなえる。つまり、計算機上のプレイヤーを介した、それらの設計者同士の間接的なインタラクションも考えられる。この場合、RMCP Player Server だけを選択・交換すれば、他のプロセスを変更する必要はない。

## 7 おわりに

本稿では、すべてのプレイヤーが対等なジャズセッションシステム、およびその発展形である姿に見えるジャズセッションシステム VirJa Session の、全体構想と RMCP による実装について述べた。本システムでは、プレイヤーは固定した主従関係を持たずに即興演奏でき、演奏音に加えてジェスチャーという視覚情報を介してインタラクションすることが可能になった。また、RMCP を用いて分散環境で実装したことにより、単一プロセスですべてを実現する場合に比べて拡張性が非常に高くなった。

本稿で提案したモデルと実装方法は様々な発展の可能性を持っている。そこで今後は 6 で述べたような拡張を実際におこなっていく予定である。他にも、画面上に仮想のライブステージを設置し、その中でピアニストの CG を含めた全員の姿を可視化することも予定している。

## 謝辞

本システムの実験に協力して頂き貴重なコメントを頂いた緒方美音子氏、浅谷薫氏、大森哲也氏、黛慎一郎氏に深く感謝する。

## 参考文献

- [1] 和気 早苗, 加藤 博一, 才脇 直樹, 井口 征士: テンション・パラメータを用いた協調型自動伴奏システム: JASPER, 情報処理学会論文誌, Vol.35, No.7, pp.1469-1481 (1994).
- [2] 近藤 欣也, 片寄 晴弘, 井口 征士: 音楽情報から奏者の意図を理解する伴奏システム— JASPER++ —, 第 46 回情処全大, 7Q-8 (1993).
- [3] 渡辺 和之, 西嶋正子, 柿本 正憲, 村上 公一: ニューラルネットワークを用いたジャズセッションシステム — ニューロミュージシャン —, 第 44 回情処全大, 4R-5 (1992).
- [4] Roger B. Dannenberg: *An On-Line Algorithm for Real-Time Accompaniment*, Proc. of the 1984 ICMC, pp.193-198 (1984).
- [5] Barry Vercoe: *The Synthetic Performer in the Context of Live Performance*, Proc. of the 1984 ICMC, pp.199-200 (1984).
- [6] Bridget Baird, Donald Blevins, and Noel Zahler: *The Artificially Intelligent Computer Performer on the Macintosh II and a Pattern Matching Algorithm for Real-time Interactive Performance*, Proc. of the 1989 ICMC, pp.13-16 (1989).
- [7] 堀内 靖雄, 藤井 敦, 田中 穂積: 複数の人間と協調して演奏するシステム, コンピュータソフトウェア, Vol.12, No.5, pp.63-71 (1995).
- [8] 日高 伊佐夫, 後藤 真孝, 村岡 洋一: ジャズの独奏の変化に対応する自動伴奏システム, 情処研報, Vol.95, No.19, 音楽情報科学 95-MUS-9-2 (1995).
- [9] Isao Hidaka, Masataka Goto, and Yoichi Muraoka: *An Automatic Jazz Accompaniment System Reacting to Solo*, Proc. of the 1995 ICMC, pp.167-170 (1995).
- [10] 金森 務, 片寄 晴弘, 新美 康永, 平井 宏, 井口 征士: ジャズセッションシステムのための音楽認識処理の一実現法, 情報処理学会論文誌, Vol.36, No.1, pp.139-152 (1995).
- [11] H. Katayose, T. Kanamori, K. Kamei, Y. Nagashima, K. Sato, S. Inokuchi, and S. Simura: *Virtual Performer*, Proc. of the 1993 ICMC, pp.138-145 (1993).
- [12] 亀井 克之, 佐藤 宏介, 片寄 晴弘, 井口 征士: 実写画像の編集と手の 3 次元モデルとによる人の動作アニメーションの生成, 情報処理学会論文誌, Vol.36, No.2, pp.374-382 (1995).
- [13] 後藤 真孝, 橋本 裕司: 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 —, 情処研報, Vol.93, No.109, 音楽情報科学 93-MUS-4-1 (1993).
- [14] 長嶋洋一, 由良泰人, 藤田泰成, 片寄晴弘, 井口征士: マルチメディア生成系におけるプロセス間情報交換モデルの検討, 情処研報, Vol.95, No.74, 音楽情報科学 95-MUS-11-10 (1995).
- [15] 日高 伊佐夫, 後藤 真孝, 村岡 洋一: すべてのプレイヤーが対等なジャズセッションシステム II. ペーシストとドラマーの実現, 情処研報, Vol.96, 音楽情報科学 96-MUS-14-5 (1996).