社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＨＩ－106　(4)
2003－MUS－ 52　(4)
2003／11／7

SONASPHERE - 動的な3次元インタフェースを用いたインタラクティブな音楽システム

徳井 直生 伊庭 斉志 †,††

昨今の計算機技術の劇的な進歩は，ラップトップコンピュータを楽器として扱い様々な音響処理をリアルタイムに行うという新しい音楽の演奏形態を生み出した．しかし，そうしたライブパフォーマンスでは，実際に演奏者が何をしているのか，観客の側からは分からないことが多い．そこで，本論文は，3次元ビジュアルインタフェースに基づく音楽パフォーマンスシステムを提案する．音声信号の流れなどの音響的なプロセスやそれらの制御関係などをビジュアルとして聴衆に提示することによって，音楽のパフォーマンスに音以外の新しい意味を付加することを試みる．

SONASPHERE - A Kinetically Driven Interactive Music Environment

Nao Tokui† and Hitoshi Iba††

The recent dramatic progress of computer technologies gave birth to a new type of live music performance, in which performers use laptop computers as musical instruments and process digital sound signals in real time. In these live performances, however, it's very difficult for audiences to figure out the processes being conducting by performer. This paper describes our novel live music system with 3D visual interface. This system helps us to attract audiences by visually showing processes on sound signals and control structures to audiences.

1. はじめに

人類の歴史上に登場したテクノロジーは常にアートへと昇華されてきた．中でも音楽は歴史的にテクノロジーとの結びつきが特に強い芸術分野であり，新しい技術の開発とともに新しい形態や制作手法が誕生している [1]．

例えば昨今のコンピュータ技術の進化は，音楽の制作環境を高価な専用ハードウェアからパーソナルコンピュータの上へと移しただけにとどまらず，その演奏形態にも多大な影響を与えた．高性能化したラップトップコンピュータを使った新しいタイプのライブ演奏がその最たる例で，コンピュータの計算能力を駆使して様々な音響処理をリアルタイムに行うといった形のパフォーマンスが登場した．これまで，こうしたコンピュータ技術を前面に押し出したパフォーマンスはごく限られた前衛的な音楽分野でのみ見られたが，いわゆるテクノミュージックなどのポピュラーな電子音楽と結びつくことでその裾野を広げつつある [2]．

しかし，こういった「ラップトップミュージック」のコンサートでは，ステージ上の演奏者がコンピュータの画面をにらみながらひたすらマウスを操作しているといった姿が往々にして見られる．実際にそこで何が行われているのか，演奏者が何をコントロールしているかは不透明で，聴衆の側からは伺い知ることができないことが多い．

そこで本研究では，3次元ビジュアルインタフェースに基づく新しい音楽パフォーマンスシステムを提案する．制御構造や音声信号の流れをビジュアルとして聴衆に提示することによって，音楽のパフォーマンスに音以外の新しい意味を付加することを試みる．具体的には，オーディオファイルの再生やエフェクタなどの機能単位が自律的に振る舞うような仮想3次元空間を考える．それらが相互作用することによって，豊かな音響的効果が生成されるような環境の構築を目指す．

以下，第2章では，研究の背景として，音楽ソフトウェアのインタフェースとコンピュータを使ったライブ演奏の現状を述べ，第3章で関連する研究を紹介する．その後，第4章で，手法を提案し，第5章でソフトウェアへの実装について述べる．第6章で，評価

† 東京大学工学系研究科
Graduate School of Electronics Engineering, The University of Tokyo
†† 東京大学新領域創成科学研究科
Graduate School of Frontier Sciences, The University of Tokyo

と考察を行い，最後にまとめと今後の課題を述べる．

2. 背　　景

2.1 音楽ソフトウェアのインタフェース

パーソナルコンピュータの普及と高性能化は，これまで高価な専用ハードウェアでしか実現されなかった高度な音響処理・楽曲制作を比較的安価なパーソナルコンピュータ上のソフトウェアで行うことを可能にした．1) MIDI／オーディオシーケンサ，2) 波形編集，3) ソフトウェアシンセサイザ，4) 音響処理／エフェクタ，5) 採譜といった機能を持つソフトウェア (あるいは複数の機能を持つ統合型ソフト) が多数市販されている [1]．

しかし，インタフェースの側から音楽ソフトウェアを分類してみると，現在市場に出回っているソフトウェアのほとんどが次の二つのうちのいずれかのメタファーに依存していることが分かる [3]．その二つとは，

- (広義の) 楽譜
- 既存のハードウェア

である．

周知のように「楽譜」は，時間軸に沿ったピッチ (音高) と強弱として音楽を記述するものである．シーケンサなどで広く使われているピアノロールや波形表示なども広義の楽譜として捉えることができる．サンプリングなどの手法によって，楽譜によらない，楽譜では記述できない音楽の形態が生まれている一方で，それを扱うソフトウェアに関しては実世界のインタフェースに囚われている部分が大きい．

また，ハードウェアを模倣しているものとしては，ソフトウェアシンセサイザなどにその典型を見ることができる．ソフトウェアの自由度が増し，パラメータが多くなるにつれて，表示すべきつまみの数も多くなってしまう，似通った外観を持つソフトウェアが多いといった問題がある．

もちろん，既存のインタフェースの模倣が必ずしも悪いというわけではない．ユーザがそれまでに蓄えている知識をそのまま利用できるという意味で，大きな利点があるのは確かである．しかし，コンピュータを使った新しい音楽，パフォーマンスの形態が生まれつつある中で，そうした新しい表現に適した今までにないユーザインタフェースがあってしかるべきであると考える [2),4)]．

2.2 ラップトップミュージックの手法

上記のような現状の中，逆にラップトップミュージシャンたちはステージ上のコンピュータモニタの前で一体何をしているのだろうか．幸いにも筆者はライブ

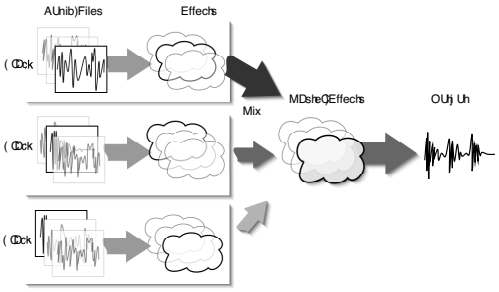

図 1 オーディオファイルを使ったライブパフォーマンスの概念図
Fig. 1 Conceptual Diagram of Live Performances Using Audio Files

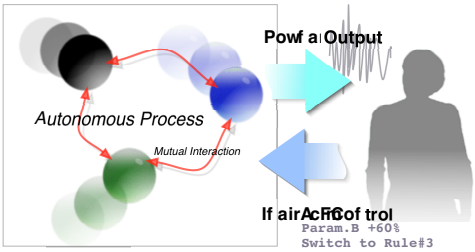

図 2 Generative Music の概念図
Fig. 2 Conceptual Diagram of Generative Music

演奏を行う側として，内外を問わず多くのアーティストのパフォーマンスの中身を観察する機会に恵まれた．こうした観察の結果得ることができた知見として，コンピュータを用いたライブ演奏で広く使われる手法の特徴を以下に示す．

(1) あらかじめレコーディングされた音の組み合わせ
(2) ランダム性の導入

1 はハードディスク上のオーディオファイルを多用する手法で，Ableton 社の Live![5] などが代表的である．読み込んだオーディオファイルにエフェクトをかけてミックスする，タイミング良くファイルを入れ替える，エフェクトのパラメータを調整するといった方法で音楽的な展開を実現する (**図 1**)．オーディオファイルは，全体のテンポに合わせてタイムストレッチなどによって自動的に長さの調整が施されている場合が多く，ある意味では DJ (ディスクジョッキー) 的なライブ方式といえる．

次のランダム性の導入という観点は，現代音楽の文脈ではコンピュータ音楽以前より重要視されている．たとえば，楽譜上に記述された音楽をその通り再現することを嫌い，演奏の「一回性」にこだわったジョン・ケージらは，ある所定の手続きや演奏される環境からのノイズ，あるいはゆらぎを自身の音楽に取り込

んだ [6]．コンピュータを使えばあらかじめ用意された音楽を完璧に再現することが可能であるがゆえに，逆説的にライブパフォーマンスにおいては演奏の一回性をどのように実現するかが重要になる．

コンピュータを用いたパフォーマンスにおいて，ある種のランダム性を実現するためには適当なプロセスを意図的に導入する必要がある．こうした目的には，Cycling'74 Max/MSP[7] や PureData (PD)[8], jMax[9], SuperCollider[10] などの音楽プログラミング環境が広く使われている．これらの環境上で作ったソフトウェアを用いることで，あらかじめ用意したサウンドファイルやシーケンスデータの鳴らし方に複雑な変化を持たせることができる．実際には，何らかの確率分布やそれらの適用に関する簡単なルールに基づいてある程度のランダム性を導入することが多く，演奏者は確率分布やランダム性の度合いを制御することで，全体の音楽的な構成を形作る．演奏者がプロセスを変化させ，その結果として現れる生成物に対してさらに反応を返すことで，ゆるやかなフィードバックループが形成される．コンピュータプロセスと人間の間でのある意味ジャムセッションが展開されるのが理想的である．

本研究は，上記二つの手法を念頭にライブパフォーマンスに適したソフトウェア環境の構築を目的とする．**図 3** に示すように，

- オーディオファイルの再生やエフェクトの変化などを，それぞれ自律的なプロセスに任せ，
- 演奏者はアウトプットされる音を聞きながら，プロセス間のルールやパラメータを変えることによって間接的にコントロールすることができる

のがこのシステムの特徴である．プロセスの中身を出来るだけビジュアルとして提示することで，聴衆の興味を引きつけるようなシステムの構築を目指す．さらに，演奏者がパフォーマンスのために自分なりのプロセス，プロセス間のルールを作れるようにするためには，ある種のプログラミング的な要素を取り込むことが当然必要となる．

3. 関 連 研 究

音楽や音を操作するためのインタフェースを視覚的に表現された仮想オブジェクトによって提供しようとする研究は，インタフェース的な側面とプログラミング的な側面のどちらを優先するかによって大きく二つに分けることができる．

3.1 ビジュアル音楽インタフェース

何らかのグラフィックによって音をコントロールする研究は，1977 年に作曲家 Xenakis が考案した UPIC

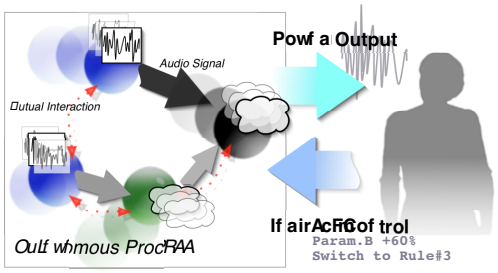

図 3 提案するシステムの概念図
Fig. 3 Conceptual Diagram of Proposed System

(Unité Polyagogique Informatique de CEMAMu) などにさかのぼることが出来る [1]．コンピュータグラフィックスのインタラクティブ性と音楽の関係を追求した例としては，AVE (Audio Visual Environment)[3] などが挙げられる．AVE では,，ユーザのマウスを用いたジェスチャをうまく取り込み，ソフトウェアシンセサイザのアウトプットと抽象的なコンピュータグラフィクスの映像を合理的に結びつけることに成功している．

音楽的にはグラニュラーシンセシス [1] などの比較的高度なシンセサイズ方式なども採用しているものの，ユーザが操作できるのは，あらかじめ定められた特定のパラメータのみで自由度に乏しい．汎用的なシステムというよりは，映像と音を組み合わせたインタラクティブなアート作品という色合いが強い．

ほかにも，Stretchable Music[11] や Tranceducer[12] や Small Fish[13] のように，画面上のオブジェクトを動かすことで音をコントロールするシステムの例がいくつかあるが，いずれの場合もオブジェクトの動きは特定のパラメータと結びつけられているだけで，汎用性・自由度の面では問題が残る．

また実体を持ったインタフェースと CG の柔軟性，双方の利点を取り入れた AudioPad[14] のような新しいインタフェースも開発されている．実際のオブジェクトを用いることでより直感的な操作，複数のユーザの協調などが可能になった．演奏のためのインタフェースとしては非常に優れているが，自律的なプロセスとのインタラクションの実現を目指す本システムとは研究の方向が異なる．

3.2 ビジュアルプログラミング環境

音楽用のビジュアルプログラミング環境としては，既出の Max/MSP, PD, jMax などが挙げられる．これらのプログラミング環境では，オブジェクトと呼ばれる機能単位をパッチコードという線で結んでいくことで音響処理，制御の流れを記述する，音楽用のビジュ

アルプログラミング環境である．アルゴリズムとそれを制御するユーザインタフェースが同じレベルで記述され，ユーザは数値を表示，入力するナンバーボックスや，ダイアルなどの基本的なインタフェースオブジェクトを使って自分なりのインタフェースを作ることができる．様々な処理を簡便に実現することが可能であるが，実際にどういった処理が行われているのか，処理がどのように変化しているのかを，プログラムのビジュアルからとらえるには，そのプログラミング環境に精通する必要がある．

4. 提 案 手 法

前節の背景から，システムを設計する際に特に留意する点として，下の項目が導き出された．すなわち，コンピュータ上で行われている音楽的なプロセスの

(1) 視認性
(2) プログラミング可能性
(3) 自律性/流動性

の三つである．

ここではまずこれらの要求に適したソフトウェアの形態を考える．プロセスの変化が聴衆から視覚的に見て取れるようにするためには，通常の音楽ソフトウェアのインタフェースのようにパラメータの数値の変化で示すのではなく，実際に何かが動くようなより直接的な表現が望ましい．言い換えると，その「何か」の位置とプロセスのパラメータを結びつけることで，動きがプロセスの変化を示すような機構が実現される．また，そうしたオブジェクトを組み合わせ，それぞれの動きを制御するような仕組みがあれば，視覚的なプログラミングが可能になるはずである．

前述の 1，2 の要求に基づいて，グラフィックで表現されるインタフェース部のプロトタイプを作成した (**図 4**)．ここのプロトタイプを用いて，3 の自律性および流動性を，直感的に分かりやすい形で取り込む方法を探った結果，物理空間のメタファーに基づいて，空間およびオブジェクト間に仮想的な力を定義するという方法を採用した．具体的には，重力や空気抵抗などの物理法則が支配するとして空間を定義し，オブジェクト間にはオブジェクトの持つ電荷に応じたクーロン力が働くものとした．また，オブジェクト間の接続には，バネモデルを適用した．

4.1 SONASPHERE システム

以上の考えに基づいて実際にソフトウェアとして実

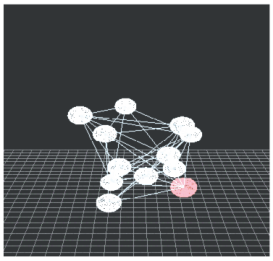

図 4 ユーザインタフェースのプロトタイプ
Fig. 4 Prototype User Interface

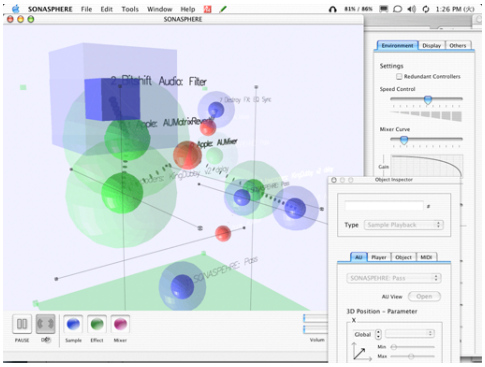

図 5 SONASPHERE システムのスクリーンショット
Fig. 5 Overview of SONASPHERE

装したシステムが，SONASPHERE である (**図 5**)．☆

SONASPHERE では，基本的な処理の機能単位を仮想的な三次元空間上に浮かぶ球体として表現し，それらを順につなげることで全体の処理を記述する．この基本モジュールは，Max にならい「オブジェクト」と呼ぶこととし，仮想三次元空間は「シーン」と呼ぶ．三次元の空間を仮定したのは，オブジェクトの位置によってコントロールできるパラメータの数を増やしつつ，ユーザが直感的に理解できるような視覚表現を求めたためである．いわば，SONASPHERE をプログラミング環境として捉える視点からの要求と，インタフェースとして見る視点からの要求のバランスを取った形となった．

4.1.1 オブジェクト

オブジェクトは，その役割から大きく四つに分けられる．

- サンプルオブジェクト
  ハードディスク上のサンプリングされたオーディオファイルを再生する．ループの有無やループの始点，終点などの設定ができる．

☆ SONASPHERE という名前には，音の塊がそれぞれに意志を持って動き回っている，音の生態圏といったイメージを込めた．

- エフェクトオブジェクト
  オーディオ信号に音響効果を加えるオブジェクト．リバーブ (残響効果) や エコーなど，複数の効果を用意した (**表 1**)．5 で述べるように，プラグインを付け足すことで，ユーザは望みの音響効果をシステムに追加することができる．
- ミキサーオブジェクト
  オーディオ信号の経路を一つにまとめるオブジェクト．逆に一つの経路を複数に分けるインタリーバ (interleaver) も用意した．これによって，複数の信号をまとめて一つのエフェクトをかける，あるいは異なるエフェクトをそれぞれ施した信号を再度一つにまとめるなど，柔軟なルーティングが可能になる．
- アウトプットオブジェクト
  信号を外部機器に出力するオブジェクト．一つの出力先に複数の信号をまとめて出力することが一般的なので，アウトプット／オブジェクトにはミキサーの機能も持たせてある．

オブジェクトには，質量と電荷という特性を持ち，それに応じた力を空間あるいは他のオブジェクトから受ける．こうしたオブジェクトが，シーンの中で一つあるいは複数の有向グラフを形成することで，全体の処理が定められる．この有向グラフの辺，すなわちオブジェクト間のつながり (信号の経路) は，小さな円を円柱上に並べたグラフィックを対応するオブジェクトの間に描くことで表している．信号の流れる方向は，円の流れで表現される．

オブジェクトの動きとオーディオ信号に出力される音を結びつけるために，いくつかの仕掛けを設けてある．まずその一つが，オブジェクト間を流れる信号の大きさは，距離の二乗に反比例して小さくなるという仮定である☆．例えば，ミキサーのオブジェクトに他のオブジェクトを近付けたり遠ざけたりすることで，直感的にミックスのバランスをコントロールすることができる．

もう一つの仕掛けが，オブジェクトの三次元空間上の位置をそのオブジェクトのエフェクトパラメータと関係づけることができるというもので，これによってオブジェクトの動きがパラメータの変化へと直接的につながる．具体的には，パラメータが取りうる値の全範囲あるいは任意の一部の範囲を適当な座標軸の座標値と関係づける仕組みを実装した (**図 6**)．例を挙げると，ローパスフィルタのカットオフ周波数を，ローパスフィルタオブジェクトの Y 座標 (高さ) に割り当てることで，空間内での上下運動によってフィルタの開閉をコントロールすることが可能になる (**図 7**)．加えて，オブジェクトの位置の情報を OpenSound Control (OSC)[15] ☆☆で出力することも可能であり，他の OSC 対応アプリケーションのユーザインタフェースとして使うこともできる．

オブジェクトの動きをある平面上，あるいは線上に限定することで，特定のパラメータを固定することも可能である．動きに制限のあるオブジェクトには，動ける面また線が表示される．全方向への動きが制限され，空間上の一点に固定されたオブジェクトについては，球から立方体へと形を変えることで，その状態を表現している (**図 8**)．

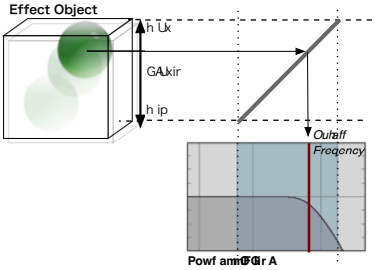

図 6 パラメータと座標の関係付け (ローパスフィルタの例)
Fig. 6 Relationship between Parameter and Coordinate Value (Eg. Lowpass Filter)

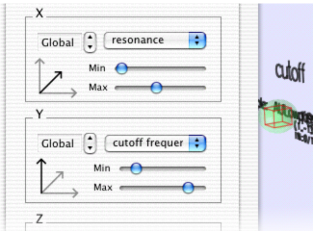

図 7 パラメータ割り当てウィンドウ
Fig. 7 Window for Parameter Assignment

表 1 用意したエファクタの例
Table 1 Samples of Default Audio Effectors

Type	name
Delay	Simple, Multitap,
Filter	Lowpass, Bandpass, Highpass
	Low Shelf, High Shelf
EQ	Parametric, Graphic
Misc.	Reverb, Peak Limiter

☆ 反比例の係数はユーザが調整することができる．

☆☆ 音楽情報通信プロトコルのひとつで，TCP/IP，UDP などのネットワークを通して送受信される点に特徴がある．

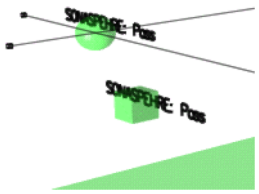

図 8 オブジェクトの動きの制限
Fig. 8 Constraints on Object Movement

4.1.2 シ ー ン

オブジェクトを配置する仮想空間「シーン」は，周囲を壁に囲まれた有限の広がりをもった三次元空間として定義されている．オブジェクトの座標値とパラメータとの間の対応付けから，シーンのサイズを変えることで，パラメータ変化の急激さを変えることができる．☆

シーンに存在するオブジェクト全体の動きを統御するために，シーンにはいくつかの特性を持たせてある．それらは，オブジェクトに働く「重力」「空気抵抗」などで，三次元空間の表現として，いずれも自然な定義である．それぞれの力の大きさは可変で，ユーザが自由にコントロールできる．空気抵抗を増やして動きを穏やかにしたり，重力によって床面で繰り返し弾ませるといった，間接的な制御が可能になる．

4.1.3 リ ン ク

オブジェクト間の接続「リンク」は，上述のように円柱上のグラフィックで表される．ユーザは，マウス操作で任意のオブジェクトを結ぶことができる．リンクの自動生成を許すモードでは，ユーザの明示的な操作によって生成されるリンクとは別に，オブジェクト同士の衝突によってもリンクが自動的に作られる．この際の音声信号の流れる方向は，オブジェクトが持つ電荷によって決定される．デフォルトでサンプルオブジェクトのポテンシャルは高く，逆にアウトプットは低く設定してあるため，オーディオファイルのプレイバックからアウトプットへの信号の流れが生まれる．

さらに各リンクには，ユーザが定義できる複数の特性がある．その一つがバネ係数で，標準の長さからの変位に応じて伸縮方向に力が働く．この性質を上手く用いることで，リンクは信号の流れるパスとしてだけではなく，オブジェクトの動きを制御する仕組みとしても動作する．こうしたオブジェクト間のローカルなルールが組み合わされて，全体としては複雑な振る舞いが生み出される．

☆ シーンの大きさをかえても，オブジェクトの動きのスピードは影響を受けないため

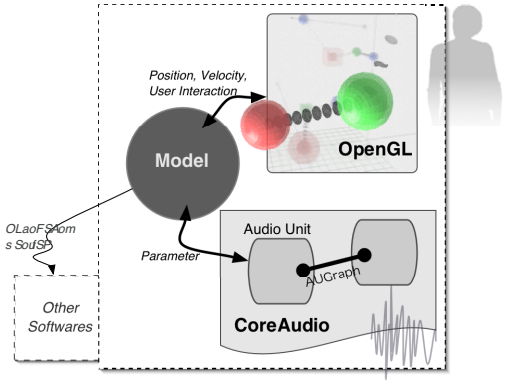

図 9 システム構成図
Fig. 9 System Architecture

5. システム実装

Apple Computer の OS X 上で動くソフトウェアとして実装した．OS X の標準的な API，プログラミング言語である Cocoa[16] と Objective-C を用いている．3D グラフィックに関しては，OpenGL を利用した．

オーディオ面では，OS X で新たに採用された Core-Audio アーキテクチャ[17] に準拠した作りとなっている．Core Audio アーキテクチャは，OS X 上で採用したオーディオ環境の総称である．OS レベルでオーディオ処理をサポートしているため，レイテンシを低く抑えつつ高音質のオーディオソフトウェアを作ることができるとされている．

実装された SONASPHERE システムは，AUGraph API を用いた Audio Unit のネットワークという形で音響処理を実現している．すなわち，三次元空間上に球で表された各オブジェクトと Audio Unit とが，一対一対応している．同様にオブジェクト間の接続は，Audio Unit 間の接続，すなわち AUGraph の一つのパスに相当する．サンプルオブジェクトでのオーディオファイルの再生は，Audio Unit のコールバック関数を用いている．

加えて，Audio Unit を単位オブジェクトとして使うことで，以下のようなメリットを享受することが出来る．

- サードパーティ製のプラグインの利用
  OS 標準のプラグインを利用することで，第三者が製作したプラグインをそのままシステム内で利用することが可能になる．

図 10　SONASPHERE を用いたライブパフォーマンス
Fig. 10　Live Performance using SONASPHERE

- 実装の面の簡便性
  厳密な規格が定められているので，互換性を保ちつつ新しいオブジェクトを実装することが容易である．

6. 評価と考察

提案手法の評価の一環として，実装したシステムを用いてこれまで多くのライブパフォーマンスを行ってきた (**図 10**)．実際のパフォーマンス時には，液晶画面をペンで直接操作できるワコム社製のタブレットを用い，操作している画面と同じものを観衆に見えるようにプロジェクタで投影した (**図 11**)．**図 12** は，ライブで使用したシーンファイルの一部を例示している．ここでは，左右の端の声とリズムのオーディオファイルを再生し，中央のアウトプットから出力している．例えば，声のオブジェクトが中央のアウトプットに近づくと，同じ電荷を持つローパスフィルタのオブジェクトが反発して下に下がり，フィルタがかかることによってリズムの音量が小さくなるといった仕組みが実現されている．

表 2 に，ライブパフォーマンス後に行ったアンケートの結果☆を示す．投影された映像によってことによってパフォーマンスへの理解が深まったと答えた人が全体の 70 パーセントを超えたことから，プロセスを可視化して観衆に提示するという当初の目的は十分に果たされたと考えられる．一方で，質問 2 からは，演奏者の操作と具体的なプロセスの変化との関係性の把握に難しさを感じていることがわかった．

その最大の理由としては，オブジェクトの見かけ上の区別が付きにくい点が挙げられる．現時点では，オブジェクトはエフェクト，アウトプットといった大まかな区分で色が分けてあるのみで，細かい区別はオブジェクトの横に文字で表示される Audio Unit の名前を読まなければならない仕様になっている．そのため，動いているオブジェクトの種類を知ることは，そのシーンを作った人にとっても難しい．今後は，機能の視覚化についてより分かりやすいビジュアル表現の方法を探る必要がある．

同様にアンケート時に募った意見でも，視覚的なバリエーションのなさを指摘する声が多かった．現状では，球と立方体を中心とするシンプルな表現にとどめているが，上記のオブジェクトの区別の問題も含めて今後の課題としたい．

より広い層からの意見を募るため，現在，WWW

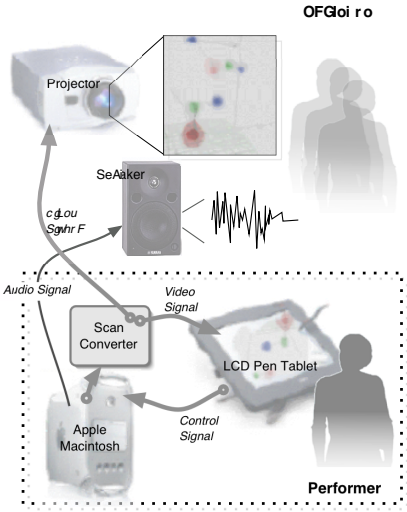

図 11　ライブパフォーマンス時のシステム全体図
Fig. 11　The Whole of Live Performance System

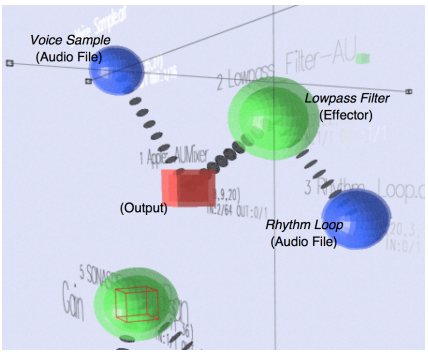

図 12　ライブで使用したシーンの一部
Fig. 12　A Part of Scene File used in Live Performances

☆ 2003 年 9 月 20 日渋谷 UPLINK にて．40 人の観客のうち 23 人から回答を得た．

表 2 ライブパフォーマンス後に実施したアンケートの結果
Table 2 The Result of Questionnaire after Live Performance

問 1. 画面の投影によって，パフォーマンスへの理解は	
ア）深まった	17
イ）あまり変わらない	5
ウ）悪くなった	0
問 2. ライブ時の演奏者の操作の内容は	
ア）よく分かった	4
イ）何となく分かった	16
ウ）全く分からなかった	3
問 3. 音に映像が追従する VJ 形式の パフォーマンスと比較して	
ア）面白い	16
イ）どちらともいえない	6
ウ）面白くない	0

のサイト上でもベータ版の無償公開を行っている☆. 2003 年 3 月以降，2000 を越えるダウンロードを記録し，多くの音楽製作関連の雑誌でもレビュー等を受けた ☆☆. OS X 上で動く音楽ソフトウェアが少ないという現状も，多くのユーザからの需要につながったと考えられる.

7. おわりに

本稿では，3 次元インタフェースに基づく音楽パフォーマンスシステムについて，その背景とシステムの内容を中心に述べた．相互に作用するオブジェクトを仮想空間上で組み合わせることで，興味深い音響効果を生み出すとともに，そのプロセスを観衆に提示することに成功した．一方で，前節に述べたような視認性などにおける問題点が明らかになった.

本システムが，映像と音を組み合わせたライブパフォーマンスの環境として普及するには，ツールとしての透明性が必要である．現行の球とリンクを用いた表現は，音に施されるプロセスを映像的に表現する方法の一つであり，他の形状，物理モデルを用いることは十分に考えられる．好みの映像表現をインタフェースとして自由に使えるような枠組みをユーザに提示することが望ましい.

今後は，明らかになった問題点の改善に取り組みつつ，音楽ビジュアルプログラミング環境としての充実を図る．他のオブジェクトの動作を制御することに特化したオブジェクトや，全体の動きのテンポを制御するオブジェクトなどを追加することによって，音と映像が真に結びついた新しい形のパフォーマンスを実現する環境を提供したいと考えている.

☆ http://www.sonasphere.com/

☆☆ リートーミュージック Sound&Recording 誌や PRIMEDIA Electronic Musician 誌など

参 考 文 献

1) 長嶋洋一, 橋本周司, 平賀譲, 平田圭二: コンピュータと音楽の世界 - 基礎からフロンティアまで, 共立出版 (1998).
2) 久保田晃弘: デジタル表現の四つの特徴 ポスト・テクノ (ロジー) ミュージック序論, ポスト・テクノ (ロジー) ミュージック, 大村書店 (2001).
3) Levin, G.: Painterly Interfaces for Audiovisual Performance, Master's thesis, MIT Media Lab (2000).
4) Gentner, D. and Nielsen, J.: The Anti-Mac Interface, *COMMUNICATIONS of the ACM*, Vol. 39, No. 8 (1996).
5) : Ableton, http://www.ableton.com/.
6) マイケルナイマン: 実験音楽—ケージとその後, 水声社 (1992).
7) : Cycling'74, http://www.cycling74.com/.
8) : PureData, http://www.pure-data.org/.
9) : jMax, http://www.ircam.fr/produits/logiciels/jmax-e.html.
10) : SuperCollider, http://www.audiosynth.com/.
11) Strickon, J., Rice, P. W. and Paradiso, J.: Stretchable music with laser rangefinder, *SIGGRAPH 98 Conference Abstracts and Applications*, ACM Press (1998).
12) Kram, R.: System Models for Digital Performance, Master's thesis, MIT Media Laboratory (1998).
13) Furukawa, K., Fujihata, M. and Münch, W.: .
14) Patten, J., Recht, B. and Ishii, H.: Audiopad: A Tag-based Interface for Musical Performance, *Proceedings of the 2002 Conference on New Insterfaces for Musical Expression (NIME 02)* (2002).
15) Wright, M. and Freed, A.: OpenSound Control: A New Protocol for Communicating with Sound Synthesizers, *ICMC 97* (1997).
16) Apple Computer: Cocoa Documentation, http://developer.apple.com/Cocoa/ (2001).
17) Apple Computer: Audio and MIDI on Mac OS X, http://developer.apple.com/audio/ (2001).