

汎用性のあるマルチメディアアート製作装置の実現手法の提案

西本 一志[†] 間瀬 健二[†]

本稿では、汎用的に用いることのできるマルチメディアアート製作装置の実現手法について考察する。従来から多くのマルチメディアアートの作品が製作されている。これらの作品では様々な製作装置が使用されているが、これらのほとんどは特定の作品専用の装置であり、別の作品の製作には基本的に適用できない。そこで、本稿では汎用的マルチメディアアート製作装置の要件について議論し、その実現手法として、合理的機能マッピング手法と共通属性に基づく役割割り付け手法を提案する。さらに、この手法に従って構築したプロトタイプシステムについて紹介する。

A Framework for General Multimedia Art Creation Instruments

Kazushi Nishimoto[†] and Kenji Mase[†]

In this paper, we propose a framework for realizing general multimedia art creation instruments. Recently, a lot of attempts at creating multimedia art have been made with the aid of computer technology. However, we think a system for a certain artwork is usually dedicated only to the artwork. Such a system cannot be applied to any other artwork. In this paper, we discuss the prerequisites of a general multimedia instrument, and propose a framework for achieving a general multimedia instrument, that is, a rational function mapping and a commonly-attributing role allotment. Moreover, we illustrate prototype instruments constructed in accordance with the framework.

1 はじめに

本稿では、汎用的に使用できるマルチメディアアート製作装置（以下 **GMI**(General Multimedia Art Creation Instrument) と略す) の実現手法について検討する。さらに、提案手法に基づき構築したプロトタイプについても示す。

近年、多くのアーティスト達によって、様々なマルチメディアアートが製作されている。しかしながら、これらのほとんどにおいて、ある作品の製作のために用いられているシステムやツールは、その作品の製作のためのみを目的として構築されている。換言すれば、それらのシステムやツール自体がその作品の一部を構成していると言えるだろう。

筆者らは、このような従来のあり方を否定するつもりはない。しかし一方で、今後のマルチメディアアートの普及と発展のためには、特定の作品とは独立な、たとえば音楽におけるピアノのような、より汎用的なマルチメディアアート製作装置が必要なのではないかと考える。従来の方法では、マルチメディアアート作品を製作しようとする、まずその

作品を製作するためのツールの開発製作から取り掛からねばならない。これは、作品の製作者に非常に負荷を強いる。この結果、多くの人々がマルチメディアアートの製作を断念せざるを得なくなっている可能性がある。

そこで本稿では、GMIの実現手法について検討する。本手法は、(1) 1つのGMIはメディア独立な共通属性を操作すること、(2) 各GMIの操作インタフェースへの操作機能マッピングを合理的で容易に把握可能なものとする、(3) 作品全体は、個々の演奏者¹による各共通属性への演奏を統合することによって生成されること、を特徴とする。

以下、第2節では、GMI実現のための要件と実現手法を考察する。第3節では、プロトタイプシステムについて論ずる。第4節はまとめである。

2 GMIの実現手法

GMIの実現のためには、2つの重要な要件があると考えられる。すなわち、操作インタフェースへの機能の不変的かつ合理的なマッピングと、各GMIへの適切な役割の分担である。以下本節ではこの2点について

¹ 「演奏」とは、本来音楽を奏すること、すなわち、ある楽器の操作インタフェースを操作して、音楽的出力を生成する行為を指す言葉である。本稿ではその意味を拡張し、ある機器（本稿ではGMIにあたる）の操作インタフェースを操作して、なんらかの（芸術に関わる）出力を生成する行為を指す用語として用いる。

[†] (株)ATR 知能映像通信研究所
ATR Media Integration & Communications Research
Laboratories
E-mail: {knishi, mase}@mic.atr.co.jp

て考察し、最後に提案手法をまとめる。

2.1 機能マッピング

GMI の操作インタフェースには、その GMI が操作すべき対象を操作するための各種の機能がマッピングされる。この際、どの機能が操作インタフェース上のどこに配置されているかを容易に演奏者が把握でき、かつその配置が変化しないようなマッピング手法をとることが重要であると思われる。

伝統的な楽器では、その操作インタフェース上の各操作ポジションは、特定の音高の音を出力するという機能を持つ。たとえば、ピアノで左端から 31 番目の白鍵には、“C5”の音高の音を出力するという機能が割り当てられている。したがって、演奏者は C5 の音を演奏したい時には、この左から 31 番目の白鍵を操作することで、随時その目的を達することができる。このように、必要な音高の音に自由にアクセスできるがゆえに、従来の楽器を用いてどんな音楽でも自由に演奏できたのである。もしこのような明瞭な機能のマッピングが存在せず、常に機能の配置が変動しているような楽器があれば、そのような楽器を用いて楽曲を演奏することは、形状が良くわからないブロックを積み上げて塔を建てようとするのと同じ行為であるといえる。しかし、いわゆる新世代楽器やマルチメディアアート製作装置には、各機能の設定位置が曖昧なもの（例えば、テルミン [1] や Video Harp [2] など）や、さらにはどんな機能がマッピングされているかも曖昧なもの（例えば Casiotone CT-647 のアドリブミュージシャン機能 [3] や、Brain Opera における Rhythm Tree [4] など）が多数見られる。

そこで筆者らは、すでに提案している「機能固定マッピング」の考え方 [5] を応用する。すなわち、ある単一のメディアについても通常は複数存在する機能的側面の 1 つに注目し、その側面における各機能を操作インタフェース上の一定位置に固定的に配置する方法である。これにより演奏者は、必要な機能を必要に応じて使用可能となる。

2.2 役割分担

GMI では、ある 1 つの操作によって、同時に複数のメディアを操作できることが求められる。単に 1 つの装置で複数メディアを同時操作することは、たとえば、“Music Plays Images X Images Play Music” [6, 7] などですでに実現されている。この作品では、ピアノを操作インタフェースとしたシステムで、音と映像の両方を同時に生成している。しかし、依然そこにはいくつかの問題がある。

第 1 の問題は、各メディアの取り扱いの非対称性

である。“Music Plays Images X Images Play Music” では、演奏者（坂本龍一氏）は基本的に音楽演奏に集中し、それに伴って、あらかじめシステム構築者（岩井俊雄氏）が与えたアルゴリズムによって、映像が演奏者の意図と関わりなく自動生成されているように見える。つまり、ここでは基本的に音楽が支配的で映像が従属的立場にあり、しかも演奏者が意図的に試みたとしても、この主従関係を逆転させることは不可能と思われる。

このような固定的なメディア間の非対称性の存在は、複数のメディアに対する自在な操作の妨げとなる。一時的に演奏者達の意図により非対称な扱いが発生する場合はあっても良いが、本質的にシステム自体は全メディアを対称に扱える構造と性質を持っている必要があると考える。

第 2 の問題は、メディア間の一定かつ有意な関連性の欠如である。従来システムのほとんどでは、各メディアをどう相互に関連づけるかに関する原則が存在せず、システム毎（作品毎）にそれらの関係がアドホックに決定されていたと思われる。この結果、これらのシステムの汎用性が失われている。

そこで本研究では、メディアに依存しない「共通属性」に着目し、ある 1 つの GMI は、原則としてある 1 つの共通属性を操作する装置とする手法をとる。これにより、各メディアの対称的取り扱いおよび各メディア間の有意な関連性が実現できる。また理想的には、必要なメディアを必要に応じて随時追加／削除することが可能となり、しかも各演奏者の演奏はそのメディアの追加／削除による影響を基本的に受けず、担当する属性に対する演奏表現にのみ集中していれば良い。

2.3 GMI 実現の枠組み

図 1 に、GMI の実現手法の概念を示す。この図は、M 種類の共通属性を操作する GMI 群によって、N 種類のメディアによる出力を生成する、一種のマルチメディアオーケストラの構成例を示すものである。なお、図中 N.A. で示すように、ある共通属性が、必ずすべてのメディアに対応づけられていなければならないわけではない。また、図ではたとえば共通属性 1 が必ず共通属性 2 の上位レイヤであるように描かれているが、このような属性間の上下関係は必ずしも一定一様とは限らず、メディアによっては、属性の上下関係が異なる場合もありうる。しかしながら本手法では、基本的にこのような横割りのレイヤ構造となる。なお、図の下部に Traditional Approach で示すのは従来のアプローチであり、各メディアは他のメディアとは独立に、そのメディア専用の操作ツールにより、そのメディアに関わるす

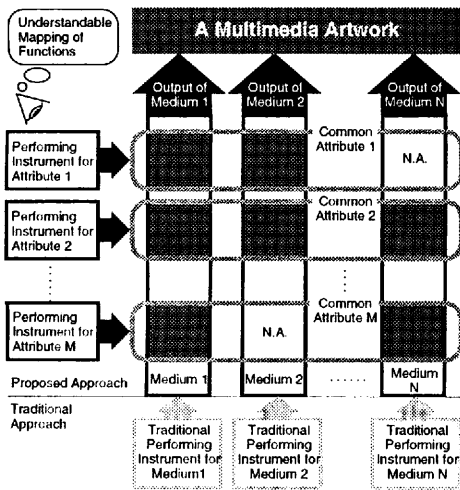


図 1: GMI の実現手法の概念図

すべての属性を操作される、縦割り構造をとる。

3 プロトタイプシステム

3.1 MKS2 の全体構成

以上の GMI の実現手法に基づき、プロトタイプシステム “MusikaScope-2” (以下 MKS2 と略記する) を開発した。図 2 に、MKS2 の全体構成を示す。MKS2 は、4 種類の GMI によって 4 つの共通属性を操作し、最終的に音楽と映像の 2 つのメディアによる出力を生成する。

GMI の操作インタフェースとして、現在は音楽用キーボード、YAMAHA MIBURI²、CCD カメラ、タップシステム³、および YAMAHA 電子ドラム (8 つのパッドと 1 つのフットペダルで構成される) を使用している。

MKS2 で採用した共通属性は、全体的な色彩バランスとその変遷に関わる属性 Global Color (GC)、全体的な緊張感と緩和感の程度に関わる属性 Tension/Relaxation (TR)、局所的な色彩の配置に関わる属性 Local Color (LC)、および最も表層的な構造に関わる属性 Surface Contour (SC) の 4 つである。現在の MKS2 では、キーボードが GC 用、

²MIBURI は衣服型楽器であり、上半身に着用するジャケットの肩、肘、手首に装着された曲げセンサによって姿勢を検出することにより、出力音の各種属性 (音高、ボリュームなど) を制御する。また、演奏者は両手に把持型キーボード (8 つのボタンと 1 つのシーソースイッチを備えている) を持ち、これによって発音タイミングを制御して音楽演奏を行う。

³タップシステムは、薄型のタッチセンサを 1 枚ずつ左右の靴底に装着し、たとえば右足で床を踏むと、右靴底に装着されたタッチセンサからの信号をオーディオ信号として無線伝送し、ワークステーションに入力するものである。

MIBURI が TR 用、CCD カメラとタップシステムが LC 用、そして電子ドラムセットが SC 用の操作インタフェースである。

各操作インタフェースからの出力は、それぞれの共通属性処理モジュールに入力される。これらの処理モジュールは、いずれもワークステーション上 (SGI O2 および INDY) で動作するソフトウェアモジュールである。各モジュールは前後 2 段に分かれている。前段部では、操作インタフェースの機器に依存した形式の共通属性操作データをより一般的な表現に変換し、これを後段部で出力メディアに依存した形式のデータに変換する。したがって後段部は、実際に使用するメディアの数だけ準備する必要がある。現在の MKS2 では、後段部は音楽と映像の 2 種類が存在するが、さらに他のメディアを追加する場合には、そのメディア用のデータ変換処理ソフトをプラグインとして準備し、各モジュールの後段部に追加すれば良い。

各共通属性の操作インタフェースと属性処理モジュールで、1 つの共通属性演奏器が構成される。

各共通属性処理モジュールの音楽用 plug-in および映像用 plug-in からの出力は、それぞれ音楽用/映像用出力生成モジュールにおいて統合処理され、最終的に音楽/映像として出力される。

3.2 共通属性演奏器

3.2.1 GCI: Global Color 演奏器

キーボードでは各鍵が操作ポジションにあたり、どの鍵がどの程度の強さでどれだけの期間打鍵されたかを、MIDI (Musical Instrument Digital Interface) データとして出力する。

GC 属性処理モジュールは、まず入力された MIDI データからどのような和音が入力されたかを決定し、各和音の種類に関する情報 (根音は何か、和音のタイプは何か) を求める。

音楽用 plug-in では、入力される和音の種類とその進行情報から、音楽理論に基づく解析を行い、各時点において 1 オクターブ中に含まれる 12 種類の音がそれぞれどのような機能を持つかを判定する (詳細は文献 [5] を参照されたい)。現段階の実装では、各音がその時点での和音の根音から数えて何度目の音であるかという音程情報 (以下、3 度音、4 度音のように表記する) と、各音が和音構成音、テンション音⁴、あるいはそれ以外の音、のいずれであるかを示す情報が出力される。

映像用 plug-in は、入力された和音の種類に基づ

⁴ある和音において理論的に使用可能な音と判定された音で、かつ和音構成音に含まれない音。

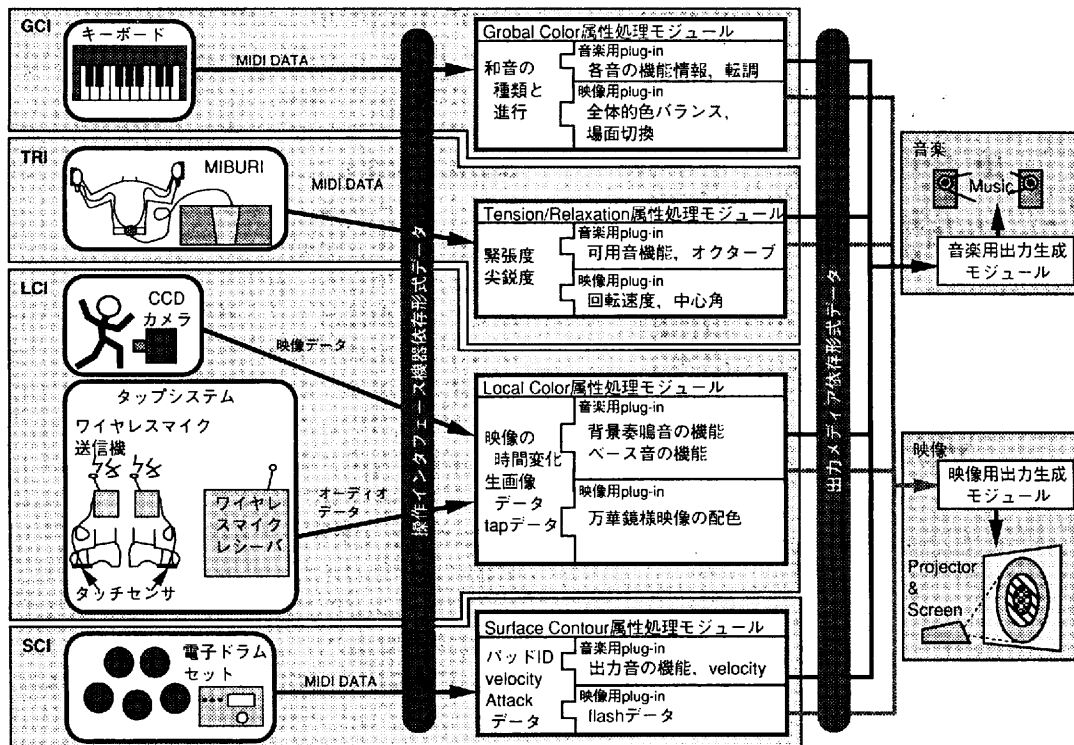


図 2: MKS2 の全体構成

き、全体的色バランスを決定する。すなわち、長調系和音：全体的色バランスを明，短調系和音：全体的色バランスを暗，それ以外の和音：全体的色バランスを不安定，と決定する。同時に和音の進行情報に基づき，転調あり：Scene Change = yes，転調なし：Scene Change = no，と決定する。

3.2.2 TRI: Tension/Relaxation 演奏器

MIBURI は、把持型キーパッドの8つのボタンのいずれかが押された時点の手首，肘，肩関節の角度情報と，どのボタンが押されたかに関する情報を，MIDI データとして出力する。

TR 属性処理モジュールは，まず入力された MIDI データから TRI 演奏者の姿勢を判定し，これに基づき緊張度を求める。すなわち，両ひじ伸展状態：緊張度 low，両ひじ直角屈曲：緊張度 medium，両ひじ完全屈曲：緊張度 high，と判定する。同時に，把持型キーパッドのどのボタンが押されたかに応じて鋭度を求める。すなわち，人差し指のボタン押下：鋭度 lowest，中指のボタン押下：鋭度 low，薬指のボタン押下：鋭度 high，小指のボタン押下：鋭度 highest，と判定する。

音楽用 plug-in では，緊張度に応じ使用する音の機能の範囲を決定する。つまり，緊張度が高いほ

ど，より不協和な音も使用可と決定する。一方鋭度に応じ，使用する音のオクターブ増分を決定する。すなわち，鋭度 lowest, low：オクターブ増分 0，鋭度 high, highest：オクターブ増分 1，と決定する。

映像用 plug-in では，緊張度に応じ後述する万華鏡様映像の回転速度を決定する。すなわち，緊張度 low：静止，緊張度 medium：低速回転，緊張度 high：高速回転，と決定する。一方鋭度に応じ，万華鏡様映像を生成する元となる扇型映像の切り出し中心角度を決定する。すなわち，鋭度 lowest：中心角度は大，鋭度 low：中心角度は中，鋭度 high：中心角度は小，鋭度 highest：中心角度は極小，と決定する。

3.2.3 LCI: Local Color 演奏器

CCD カメラの操作ポジションは，CCD 素子の各画素であり，結局撮影された映像が操作入力情報となる。またタップシステムの操作ポジションは，左右のタッチセンサであり，タッチセンサからワイヤレスマイクシステムを介してのオーディオ信号が操作入力情報となる。

LC 属性処理モジュールは，まず CCD カメラから入力される映像情報の時間変化を計測する。すなわ

ち、画像をいくつかのセグメント（現在の実装では10セグメント）に分割し、各セグメントにおける画像の輝度変化を計測し、いずれかのセグメントで所定の閾値以上の輝度変化が生じた場合、そのセグメントにあらかじめ割り当てられているセグメントIDを出力する。同時に、CCDカメラから得られる画像をそのまま出力する。一方、タップセンサからのオーディオ信号からタップ演奏の有無を判定する。すなわち、オーディオ信号がある所定の閾値よりも大きな値を取ったとき、そのオーディオ信号が左右いずれのタッチセンサから得られたかに応じ、右の場合は tap = R、および左の場合は tap = L を出力する。

音楽用 plug-in は、輝度変化の大きいセグメントID 情報に基づき、作品背景において奏鳴させる音（伴奏に相当する）の機能を決定する。現在の実装では、セグメントID が1の場合に root 音、2の場合に5度音、3の場合に3度音、4の場合に7度音、…のように、和音構成音だけを対応づけている。また、tap = L の場合にはベース音機能 = root を、また tap = R の場合にはベース音機能 = 5th を出力する。

映像用 plug-in は、入力される生のカメラ映像をそのまま出力する。

3.2.4 SCI: Surface Contour 演奏器

電子ドラムセットは、あるパッドないしフットペダルが叩かれた際、そのパッドに割り当てられているMIDI ノートナンバーと、どの程度の強さで叩かれたかを示す velocity 値を出力する。

SC 属性処理モジュールは、MIDI ノートナンバーに基づき、どのパッドないしペダルが操作されたかを判定し、対応するパッドID を出力する。現在の実装では、ペダルのパッドID = 0 であり、またパッドは横一列に配置しているので、最も左に配置されたパッドのパッドID を1とし、以下右方向にパッドID を1ずつ増すように設定している。併せて velocity 値をそのまま出力する。さらに、velocity がある所定の閾値以上であった場合に、Attack = strong を出力する。

音楽用 plug-in は、パッドID に基づき、出力する音の機能を決定する。現在の実装ではパッドID=0 は root 音に、パッドID=1 が2度音に対応し、以下順に3度音、4度音、…と対応している。これは機能固定マッピングの考えに従っている。また、各パッドの外縁部（リム）も独立な操作ポジションとして使用可能である。リムには、強い緊張感を表現する機能の音、すなわち非常に不協和な音が割り当てられる場合があるが、これは TRI からの可用音

機能指定に依存して変化する（第3.3.1節参照）。また、同時に入力された velocity 値を、併せてそのまま出力する。

映像用 plug-in は、Attack = strong であった場合に flash = on を出力する。パッドID ならびに velocity 値は使用しない。

3.3 各メディア用出力生成モジュール

以上の処理の結果を統合して、最終的に各メディアにおける出力を生成する。

3.3.1 音楽用出力生成モジュール

本モジュールは、GC 属性処理モジュールからの各音の機能情報と、TR 属性処理モジュールからの可用音情報を元に、まず各時点で使用可能な音を決定する。

たとえば、ある時点において各音の機能情報が D: root(C), E: 2度音(T), F: 3度音(C), G: 4度音(T), A: 5度音(C), B: 6度音(T), C: 7度音(C), と与えられているとする。ここに(C)は和音構成音で、この状況において最も協和な音であり、(T)はテンション音でやや不協和な音である。上記以外の音は「アウト音」であり、非常に不協和な音となる。

この状況で、LC 属性処理モジュールから背景奏鳴音機能として3度音が指示された場合、実際の演奏音として“F”を出力する。また、SC 属性処理モジュールから4度音が指示された（左から3つめのパッドの中央が叩かれた）場合は、“G”が出力される。なお、TRI から非常に不協和な音も可用な音として指定された場合、リム部には原則アウト音をマッピングする。実際には各パッド中央に割り当てられている音の半音下の音が割り当てられる。たとえば、上記の例では、左から3つめのパッドのリムから出力される音は G \flat である。TRI からの指定がそれ以外の場合は、リム部にはパッド中央と同じ音が割り当てられる。

さらにオクターブ増分が1の場合、出力する音を1オクターブ上げる。

以上の処理により、各時点における具体的な演奏音が決定される。この結果得られる音情報をMIDI データとしてMIDI sound module に入力し、最終的な出力音を得る。

3.3.2 映像用出力生成モジュール

本モジュールは、Iamascope[8] と呼ばれる電子万華鏡システムを流用している。Iamascope では、CCD カメラ画像から扇形の部分映像を切出し、これを鏡像反転させながら連続的に円状に配置することにより、カメラ画像の変化に応じて実時間的に変化

する万華鏡様の映像を生成する。したがって、カメラの前で演奏者が様々に動くことによって、生成される映像の変化を制御することができる。MKS2においては、LC属性処理モジュールから出力されるCCDカメラ画像を用いて、万華鏡様映像を生成する。さらに各GMIからの入力によって、万華鏡映像を変化させる。

GC属性処理モジュールからの全体的色バランス情報に基づき、万華鏡様映像全体のRGBバランスを変化させる。現在の実装では、全体的色彩感が明である場合はRGB均等バランスとし、暗である場合には青色を強調したバランスとし、不安定である場合にはランダムにバランスを決定する。またScene Change = yesの場合には、万華鏡様の映像の生成手法を切り替える。上述の万華鏡様映像の生成手法は、鏡を2枚用いた場合の万華鏡映像生成であるが、この他にも鏡を3枚用いた場合など、いくつかの生成モードがある。そこでScene Change = yesが通知されると、これらの生成モードのうちいずれか1つをランダムに採用し、シーン変更を行う。

TR属性処理モジュールからの中心角度情報に応じ、抽出する扇型の中心角 θ を、極小：20°、小：30°、大：45°、極大：60°とする。また回転速度情報に応じ、扇型部分の切り出し部位を、静止：固定、低速：扇型の中心を中心としてゆっくり回転、高速：速く回転、とする。

SC属性処理モジュールからのflashデータに基づき、flash = onが入力された場合は、最終的に出力する映像の全体の輝度を一時的に増し、映像全体が一瞬輝くようにする。

3.4 議論

MKS2は、今のところ提案手法に完全に則って構築されているわけではない。たとえばGCI演奏者による入力は「和音(chord)」である。これは当然「音楽」に依存しており、メディア独立な操作が実現されていない。これは「キーボード」という伝統的楽器を流用しているためで、本来はGCI専用の操作インタフェースを開発する必要がある。この点、TRIは本手法の考え方をかなりよく具現化している。特に、「緊張度」は、明らかに音楽にも映像にも依存しないメディア独立な属性である。また、この緊張度と手の姿勢(肘の曲げ角度)のマッピングは非常に直観的であり、良い機能固定マッピングの例となっている。

またMKS2では、ある演奏者によるある属性への演奏が、他の演奏者の演奏に直接影響する。したがって、熟練したGCI演奏者が良い演奏を行えば、それはLCI/SCI演奏者の演奏を底上げするという

良い影響を与える。一方、ある演奏者の犯した失敗も、そのまま全演奏者の演奏に直接的に影響するため、従来のシステム以上に注意力と緊密な相互作用を求められる。

4 おわりに

本稿では、汎用的マルチメディアアート創作ツール実現のための要件について検討し、操作インタフェースへの操作機能の合理的で不変的なマッピングの必要性和、各ツールへの適切な役割分担の必要性を指摘し、メディア独立な共通属性を演奏する演奏器群による、横割り構造を持つ枠組みを提案した。さらに、この枠組みに基づき、4種類のGMIを持つプロトタイプシステムMKS2の構成について述べると共に、その特質について議論した。これまでのところまだごく小規模の実験的な演奏しか試みていないが、今後は、MKS2のための作品を創作し演奏を行うことにより、本手法の評価を行いたい。

謝辞

本研究の機会を与えて下さった(株)ATR知能映像通信研究所の酒井保良会長、中津良平社長に感謝致します。またシステム構築にあたり多大なご貢献をいただいた、Dr. Sidney Felsに感謝致します。

参考文献

- [1] 持田康典, 青木栄一郎: 楽器とコンピュータ, 情報フロンティアシリーズ7, 共立出版, 1994.
- [2] 長嶋洋一, 橋本周司, 平賀 譲, 平田圭二編: bit別冊 コンピュータと音楽の世界, 共立出版, 1998.
- [3] カシオ計算機株式会社: CasiotoneCT-647取扱説明書
- [4] Paradiso, J., A.: Electronic music: new ways to play, IEEE SPECTRUM, pp.18-30, Dec., 1997.
- [5] 西本一志, 渡邊 洋, 馬田一郎, 間瀬健二, 中津良平: 創造的音楽表現を可能とする音楽演奏支援手法の検討 — 音機能固定マッピング楽器の提案, 情報処理学会論文誌, Vol.39, No.5, pp.1556-1567, 1998.
- [6] <http://www.aec.at/prix/1997/97gnI-images.html>
- [7] <http://sakamoto-iwai.aec.at/>
- [8] Fels, S., Mase, K.: Iamascope: A Musical Application for Image Processing, Proc. of The IEEE Third International Conference on Automatic Face and Gesture Recognition (FG'98), pp.610-615, April 1998.