

タイトル	MIDI padを入力としたTouchDesignerの制御
著者	菊地, 慶仁; 西村, 蒼太; KIKUCHI, Yoshihito; NISHIMURA, Souta
引用	工学研究: 北海学園大学大学院工学研究科紀要(24): 25-29
発行日	2024-09-30

MIDI pad を入力とした TouchDesigner の制御

菊地 慶仁* · 西村 蒼太**

TouchDesigner input control with MIDI pad as input

Yoshihito KIKUCHI* and Souta NISHIMURA**

要 旨

本報ではノードベースのビジュアルプログラミング環境である TouchDesigner の入力制御に MIDI 機器を用いた試みについて報告する。TouchDesigner は、音楽ファイルやマイクなどからの音楽データを入力として動画に対するエフェクトを制御してリアルタイムに動画を生成する目的で用いられる。しかし音楽演奏時にその操作や制御などで通常のキーボードやマウスを用いるには問題がある。本報告ではこの問題点に注目し、通常は楽器として用いられる MIDI pad を入力制御の目的で用いた内容について報告する。

1. 序論

TouchDesigner は、DERIVARATIVE が開発・提供しているノードベースのビジュアルプログラミング環境¹⁾で、マイクや楽器などからの音楽データを入力として画像生成を行う目的で用いられることが多い。3D グラフィックスを簡単に扱えるためプロジェクションマッピングやメディアアートなどでの成果も多い。しかし演奏中に操作目的で通常のキーボードやマウスを用いるには問題がある。本報告ではこの問題点に注目し、通常は楽器として用いられる MIDIpad を入力制御の目的で用いた内容について報告する。

2. TouchDesigner でのプログラミング

第 2 章では TouchDesigner とこれを用いるプログラミングについてまとめ、次に本報告での課題をまとめる。

2.1 各種プログラミング形式の特徴及び本研究での課題

2.1.1 テキストベース方式

プログラミングのコーディング方式としては、伝統的なテキストベース方式がある²⁾。これはエディタなどを用いてテキストファイルのプログラムを記述しコンパイラに通してビルドすることで実行させる形式である。ビルドする対象 CPU が開発機のものとは異なっている場合はクロスコンパイルとなる。

テキストベース方式の特徴は、プログラムの定義が非常に単純である点にある。このためプログラムを構成するには、構文規則で規定された予約語とユーザ定義の識別子及び定数などの言語要素を組み合わせて記述を行う。

テキストベース方式では、エディタによる編集機能が高度化されており、キーワードの対応、動的な外部ファイル参照検査、検索機能の高度化、機械的なコード生成が可能となっている。

また最近では LLM を用いた AI によるサポートが充実しつつあり、自動的なコード生成やコメント作成などが可能となっている。

* 北海学園大学工学部電子情報工学科

Hokkai-Gakuen University Faculty of Engineering Department of Electronics and Information Engineering

** 本人希望により非公開（北海学園大学工学部電子情報工学科卒）

Currently affiliation is not revealed by the wish of the person. (Graduated from Hokkai-Gakuen University)

2.1.2 マークアップ言語

プログラムの記述ではないが、タグを使って文書の構造や装飾を指定する方式がマークアップ言語である。代表的なものはHTMLやXMLがある³⁾。

HTMLは、ウェブページの記述に特化したマークアップ言語であり、当初はタグが書式指定の機能のみしか持っていなかったがHTML5になって構造化が可能となった。

XMLの規格自体は、タグの構造化を行う際のルールを決めたものであり、XMLのタグ構造化構文に基づいてタグと意味定義を行った**MLと呼ばれる言語が多数開発されている。

2.1.3 ビジュアルスクリプト

テキストプログラムの構造を部品として配置することでプログラミングを行うビジュアルスクリプトと呼ばれる形式がある。近年では愛玩ロボットであるSONY製aiboのプログラムに用いられるaiboビジュアルプログラミングなどがある⁴⁾。

2.1.4 ノードベースプログラミング

TouchDesignerの特徴としてノードベースプログラミングがある。ノードベースプログラミングはノンコーディングもしくはビジュアルプログラミングと呼ばれている。処理の最小単位はノードと呼ばれる部品で予め用意され、それらを連結して一つのプログラムとしている²⁾。ノードは連続して入力されるデータを処理するフィルタもしくはプロセスとして見ることができる。ノードベースプログラミングは近年増加傾向にありvvvv, Max, Notch等で用いられている。

次節ではプログラムの形式に加え実行時も含めたTouchDesignerの特徴についてより詳しく述べる。

2.2 TouchDesigner プログラミングの特徴

TouchDesignerで処理の基本となるノードはOperatorと呼ばれ機能ごとにグループ化されている。プログラミングは、これらのOperatorを実体化させ、必要なパラメータを設定し、各Operatorの入力と出力をワイヤーで連結（パッチングもしくはワイヤリングと呼ばれる）して最終出力先のOperatorに接続することで完了する⁵⁾。

TouchDesignerで用意されているノードは以下のようにグループ化されている⁶⁾。

- ・COMPs (Components) : UI, プログラムの構造構築, 3Dシステム構築用
- ・TOPs (Texture Operator) : 静止画や映像データなど2D画像処理用
- ・CHOPs (Channel Operator) : 数値処理, 演算, デバイス入出力用
- ・SOPs (Surface Operator) : 3Dオブジェクト処理用
- ・MATs (Material) : 3Dのマテリアルとシェーダー処理, SOPと共に使用
- ・DATs (Data Operator) : スクリプト処理, データ格納, デバイス入出力
- ・Custom : 独自開発のオペレータの登録

各ノードにはプレビュー画面が付属しており出力動画をリアルタイムで確認可能で、ノード毎に加工される画像を確認しやすい⁵⁾。

TouchDesignerではビルドを行わない常時実行形式でプログラムが実行される。ノードがエディタ上に呼び出された時点で一つのプロセスとなる。ノードはTouchDesignerが用意しているが、Proエディションを使っている場合はユーザー定義ノードを作成することもできる⁵⁾。

TouchDesignerはノード内部での処理やノード間でやり取りされるデータの形式、エラー処理などの対応が予めノードに組み込まれているためにテキストプログラムをコンパイルする際に起こる構文エラーや実行時のエラーを回避しやすい。そのために安定して動作することが可能で大規模なプロジェクトやインストールでも利用されている。またターゲットマシン向けにコンパイルすることはないために開発時の実行内容と最終的にパフォーマンスする際の実行内容の違いはない。また最後までパラメータを調整しやすい。

実行されている内容をリアルタイムにモニターできるため、インタラクティブなコンテンツやライブパフォーマンスに適している。

しかしながら各ノードの実行はTouchDesignerが管理して隠蔽しているため、並列実行されるノードの最適化や実行順序などの制御は行いにくい⁵⁾。

2.3 本報告での課題

TouchDesignerでは幾つかのノードをひとま

とめにしてグループ化(階層化)することが可能となっている。このことによってより具体的な機能単位でプログラムを取り扱うことが可能となる。

TouchDesigner は、非常に多彩な機能を持っているがライブパフォーマンスを行う際には、操作をしてノードの接続レベルで再編成することは無理で基本的には予め用意したプログラムもしくはノードグループを実行させることになる。

パフォーマンス中にノードグループを選択するには、曲の音程やアルゴリズムによる選択が考えられるが、操作者の意図による感覚的な選択を実現する必要がある。またその選択方法も一般的な PC 上の UI であるマウスやキーボードでは十分なスピードで指示できるとは考えられない。GUI 上での指示選択や演奏中の楽器と似た機材を TouchDesigner の制御専用用いる、もしくは楽器の一部機能を指示のために活用することが考えられる。

TouchDesigner が入力として扱うことができる楽器は MIDI インターフェースを持っているものであればほぼ全て対応が可能でありこれらの機器を入力用として用いることが可能と考えられる。

本研究での課題をまとめると以下の 2 項目となる。

- 1) MIDI 機器を楽曲演奏ではなく TouchDesigner の制御のために用いるために MIDI 機器からの入力をフィルタして特定

の入力があったかどうかを判定するノードグループを作成する。

- 2) MIDI 機器からの入力を上記ノードグループに渡して TouchDesigner からの画像出力を変化させるルーチンを作成する。

3. MIDI 機器からの入力のフィルタ方法

前節での課題について、本報告では以下に述べる方法を用いた。本報告では MIDI 機器として inMusic 社製 MIDI pad AKAI professional MPD218⁷⁾を用いた(図 1)。PC との接続はシリアルポート (USB) を用いて行われる。この MIDI pad の特定のパッドが押し下げられた状態を検知して予め設定したノードグループを機能させることを試みる。図 2 にノードの接続状況を図に示す。

- 1) MIDI 機器を TouchDesigner 中で処理対象として扱うために MIDI Device mapper を用いて、接続されている MIDI 機器を TouchDesigner の入力に Mapping する。
- 2) MIDI Device mapper からの入力信号を取得して選別するために Midi in CHOP にリンクする。MIDI In CHOP は MIDI デバイスから MIDI イベントを受信して、ワークステーション (ビルトイン・ソフトウェア) に MIDI イベントの内容を読み込んで、MIDI ファイル中のスコアを解釈し数列として出力す



図 1 MIDI pad 全景

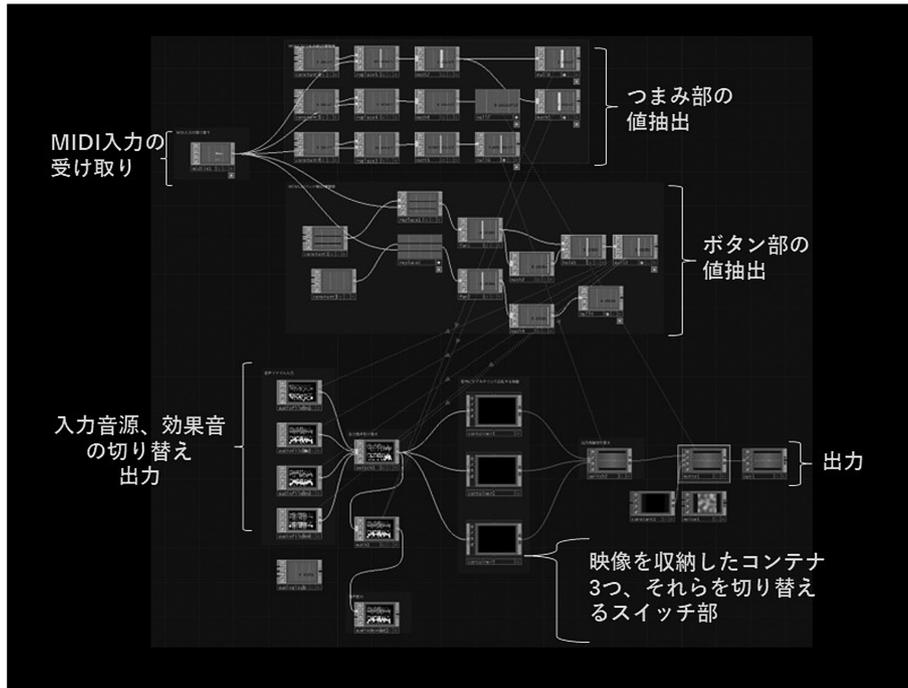


図2 ノード接続構造の全体図

る⁶⁾。MIDI 機器のつまみ部やパッド部はチャンネルとして認識され、個別のつまみやパッドはそれぞれ ID を持つ。MIDI pad には 16 個のパッドがあり MIDI in CHOP からはパッド類はチャンネル 10 でパッド名は ch10N01～ch10N64 までが割り当てられる。パッドは 3 つのパッドバンクで切り替え可能だが MIDI in CHOP からは全てのパッドが常時モニタ可能となっている。

- 3) MIDI In CHOP から出力されるパッド名を予め constant CHOP で作成しておく。ID 名は ch10N01～ch1053 までを生成している。この値と MIDI In CHOP から押し下げられるパッドのチャンネル名の出力を replace CHOP で比較して 0/1 の値で出力する。
- 4) MIDI pad のコントロール・ノブ（つまみ）からの入力も別のノードグループで処理している。コントロール・ノブは 128 レベルのレベル出力設定が可能である。コントロール・ノブの値の期待される値はパッドの処理と同じく constant CHOP で設定され、replace CHOP に送られ比較した後に math CHOP に送られ必要な演算を行う。
- 5) 試作したシステムでは入力は 5 種類の動画ファイルを audiofileinCHOP で指定している。これらの入力動画は switchCHOP に送

られパッド入力の処理から送られる値を選択用の入力として、どれか一つのファイル入力を選択され、音声及び動画の処理ノードへ出力する。その後は音声の調整や画像のエフェクト切り替えなどに前述のパッドやコントロール・ノブから取得した値が入力されて処理される。

4. 実験と考察

4.1 実験結果

図3に実際に作成したプログラムを実行している状況の動画からの画像を掲載する。今回のプログラムではコントロール・ノブ3個を利用して映像の切り替え、映像の暗転、ボリューム調整に利用している。またパッドの入力で音声の選択を行えるようにした。

また音楽以外の通常 MIDIpad の使用方法である効果音の発音もパッドで行えている。

4.2 考察

当初の目標としていた映像・音源の Touchdesigner 内での MIDI pad 制御は達成でき実用性があることが分かった。midi の入力系統

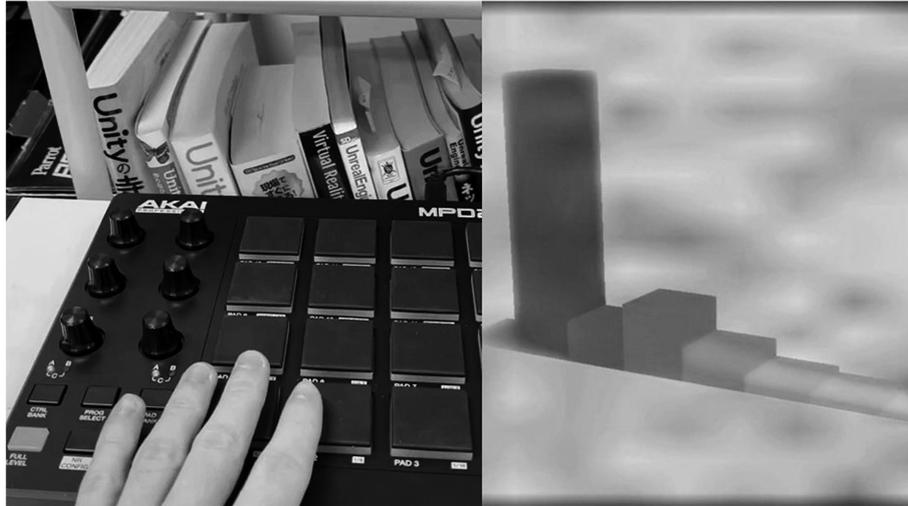


図3 実演状況 (動画からの抜粋)

に応じた出力の変化 (コントロール・ノブの入力に対しての連続的な変化) も行えたため MIDI pad という入力デバイスの特性も活かされたと考えられる。

今回は MIDI pad での入力しか試していないが、他のデバイスに対応する拡張を行いライブシーン等でも扱いやすいプログラムを作成することが課題となる。

5. 結論

本研究では MIDI Ipad を通常の楽器として用いる以外の直感的な画像処理デバイスとして用いることを目的として以下の項目での報告を行った。

- 1) MIDI 機器を入力としてその入力内容を選別するメカニズム constant CHOP と replace CHOP を用いて実現した。
- 2) 提案した手法でアプリケーションを作成し実際に動画処理を行った。コントロール・ノブも合わせて制御に用い、MIDI 機器を演奏以外の目的として使用できることを確認できた。

今後の課題としては、

- 1) TouchDesigner でサポートされているデバイスからの操作に可能とさせる。
 - 2) 各種デバイスを組み合わせた複合的なユーザーインターフェースの開発。
- などの模索をしていく必要がある。

参考文献

- 1) “TouchDesigner”, <https://derivative.ca/>
- 2) About TouchDesigner/ 特徴, <https://note.com/toyoshimorioka/n/nb435551187f1>
- 3) マークアップ言語とは— Markup Language の種類とその歴史, <https://mynavi-creator.jp/blog/article/what-is-mark-up-language>
- 4) aibo ビジュアルプログラミング, https://aibo.sony.jp/fan/visual_programming/
- 5) TouchDesigner 基礎, <https://lecture.nakayasu.com/touchdesigner>
- 6) TouchDesigner 日本語ドキュメント, <http://ted-kanakubo.com/touchdesigner-jp/>
- 7) AKAI Keypad MPD218, <http://akai-pro.jp/mpd218/>