

# ピアノ演奏に合わせた映像制御システム（仮）

福知山公立大学 情報学部情報学科

32245100 矢野 瑞季

指導教員 橋田光代 准教授

提出日 2025 年 1 月 17 日

改訂日 2025 年 1 月 31 日

— 目次 —

1	はじめに	1
2	関連研究	1
3	システムの概要	1
3.1	システム全体構成	1
3.2	使用機材およびソフトウェア	1
3.3	映像出力およびパフォーマンス環境	1
4	画像生成システム	2
4.1	入力インターフェース	2
4.2	画像生成処理	2
4.3	生成画像の表示および保存	2
5	TouchDesigner を用いた映像制作	2
5.1	TouchDesigner と Processing との比較	3
5.2	制作した映像システムの全体構成	3
5.2.1	入力処理部	
5.2.2	映像生成部	
5.2.3	映像合成・出力部	
5.3	生成画像・歌詞データの利用方法	4
5.3.1	生成画像の表示	
5.3.2	歌詞の表示	
5.4	映像の最終合成および出力	4
6	足キーボードを用いた映像制御について	5
7	実演について	5
8	課題と今後の発展について	5
9	まとめ	5

— 図目次 —

## 1. はじめに

近年、デジタル技術の発展により、音楽と映像を組み合わせた表現が多様化している。特に、ライブパフォーマンスや舞台演出において、音楽の進行に応じて映像が変化する演出は、観客に対して強い没入感を与える手法として注目されている。このような音楽と映像の連動表現は、事前に制作された映像を再生するだけでなく、演奏中の音楽的要素をリアルタイムに反映させることが求められている。

しかし、映像制作や映像演出には多くの時間と手間、ならびに専門的な知識が必要であり、音楽演奏と並行して映像を操作することは容易ではない。そのため、映像演出は演奏とは切り離され、別の操作者によって制御される場合が多いのが現状である。また、演奏者自身が演奏中に映像を制御することを前提としたシステムに関する研究や事例は少なく、音楽表現と映像表現を演奏行為の中で統合する手法は研究事例が限られており、現在も発展途上の段階にある。

さらに、演奏中の操作が複雑である場合、演奏そのものに支障をきたす可能性がある。このため、演奏者の負担を最小限に抑え、演奏の流れを妨げることなく直感的に映像を制御できる仕組みが求められている。このような課題は、音楽表現と視覚表現の一体化を実現する上で重要な検討事項である。

そこで本研究では、ピアノ演奏に基づいて映像をリアルタイムに制御・生成するシステムの構築を目的とする。具体的には、MIDI キーボードから取得したノート番号およびペロシティ情報、ならびに USB フットペダルからの入力を用いて、演奏内容に応じた映像表現を実現する。映像の制御および制作には TouchDesigner を用い、演奏中の入力情報を即時に映像パラメータへ反映させることで、演奏と映像が連動した表現を可能とする。

また、本研究では映像表現の一要素として生成 AI を用いた視覚表現を取り入れる。歌詞を含む楽曲に対しては、Python を用いて入力した歌詞データをもとに画像生成を行い、楽曲内容に対応した映像素材を生成する。一方、歌詞を持たない楽曲に対しては、曲の雰囲気や印象を表す情報をもとに生成 AI を用いて視覚表現を生成する手法を採用している。

さらに、本システムは生成された画像を静止画として映像演出に用いるだけでなく、動画素材を映像として利用することも可能な構成となっている。これにより、楽曲の特性に応じて柔軟に映像表現を選択でき、音楽的要素および言語的要素の双方を含んだ映像演出をリアルタイムパフォーマンスに組み込むことも目標としている。

## 2. 関連研究

あとで書く——\*——\*——\*——\*——\*——\*——\*

## 3. システムの概要

### 3.1 システム全体構成

本研究で提案するシステムは、ピアノ演奏および生成 AI によって作成された映像素材を用いて、演奏内容に応じた映像表現をリアルタイムに制御・出力することを目的としている。本システムは、演奏情報および映像素材生成のための入力を受け取り、TouchDesigner 上で映像制御を行い、最終的にスクリーンへ映像を出力する構成となっている。

システム全体の流れは大きく二つに分かれる。一つは、Python で構築した歌詞・イメージ入力画面から生成 AI を用いて映像素材を生成し、その結果を TouchDesigner に取り込む流れである。もう一方は、MIDI キーボードおよび USB フットペダルから取得した演奏情報を TouchDesigner へ入力し、映像演出を制御する流れである。これら二つの情報が統合されることで、多様な映像表現に対応可能なシステムの構築を目指している。

【システム概要図】

### 3.2 使用機材およびソフトウェア

本システムで使用する主な機材は、MIDI キーボード、USB フットペダル、および処理を行う PC である。MIDI キーボードはピアノ演奏情報の入力装置として使用し、USB フットペダルは演奏中の映像および歌詞の切り替え操作を行うために用いる。

使用ソフトウェアとしては、映像制御および映像制作に TouchDesigner を使用する。また、歌詞や曲のイメージ情報の入力には Python を用い、映像素材の生成には OpenAI の API を利用している。

### 3.3 映像出力およびパフォーマンス環境

本システムによって制御された映像は、スクリーンに対して一画面で投影される。ライブパフォーマンスでの使用を想定し、演奏者のピアノ演奏と映像が同期して提示される構成としている。これにより、音楽の進行や演奏表現に応じて視覚的な変化が生じ、観客に対して音楽と映像が一体となった表現を提示することが可能となる。

パフォーマンス中には、演奏者が USB フットペダルを用いて映像や歌詞の切り替えを行う。フットペダルによる操作を採用することで、手を使わずに映像制御を行うことができ、演奏動作への影響を最小限に抑えている。また、演奏者は演奏の流れに合わせて必要なタイミングで映像を切り替えることができるため、ライブパフォーマンスにおける即時性にも対応可能である。

このような構成により、演奏行為を中心としながらも、演奏者自身が映像表現に直接関与できる環境を実現している。結果として、音楽表現と映像表現が相互に影響し合うライブパフォーマンスが可能となり、演奏行為と同期した

視覚表現を実現している。

## 4. 画像生成システム

本研究では、ピアノ演奏と連動した映像演出に使用する映像素材を生成するため、生成 AI を用いた画像生成システムを構築した。本システムは Python 上で動作する GUI アプリケーションとして実装されており、OpenAI の画像生成 API および GUI ライブラリである Tkinter を用いて構成されている。

ユーザインタフェースでは、最大 20 行までのテキスト入力が可能であり、各行ごとに「歌詞」または「インスト」を選択することができる。歌詞を含む楽曲の場合には歌詞をそのまま入力し、歌詞を持たない楽曲の場合には楽曲の雰囲気やイメージを文章として入力することで、生成 AI による画像生成を行う。生成された画像は PNG ファイルとして保存され、TouchDesigner に取り込むことで映像演出に利用される。

### 4.1 入力インタフェース

入力インタフェースは、Tkinter を用いて実装されており、左側に入力欄、右側に処理状況を表示するテキストエリアを配置した構成となっている。入力欄は最大 20 行まで用意されており、各行にはテキスト入力欄とともに、「歌詞」または「インスト」を選択するためのプルダウンメニューが設けられている。

#### 【インタフェース画像】

利用者が歌詞を任意のタイミングで分割して入力する運用を想定している。各歌詞フレーズは一つのセクションとして扱われ、入力したフレーズに対応する映像素材は、演奏中に USB フットペダルを操作することで切り替えられる。そのため、歌詞の進行と映像の切り替えを同期させるには、歌詞の入力時に想定したタイミングでフットペダルを踏む必要がある。

本研究では、システムの動作確認および評価のためのテスト曲として松任谷由実の「春よ、来い」を使用し、歌詞を複数のフレーズに分割して入力した。入力画面および生成された画像については、参考画像として後述する。また、歌詞を含まない楽曲や楽曲の雰囲気を重視した演出を行う場合には、「インスト」を選択し、生成したい映像のイメージを文章として入力することも可能である。

### 4.2 画像生成処理

入力されたテキストのうち、空でない行のみを対象として、生成 AI による画像生成を行う。各行は順番に処理され、OpenAI の画像生成 API を用いて 1 行につき 1 枚の画像を生成する。本研究では、高解像度の映像演出を想定し、生成される画像サイズを横 1792 ピクセル、縦 1024 ピクセルに設定している。生成された画像データは、Base64 形

式で受信され、Python 上でデコード処理を行った後、画像ファイルとして保存される。「春よ、来い」では、1 番の歌詞に加えて前奏および間奏を含めた構成を想定し、楽曲全体を 7 つのセクションに分割した。

図〇に示す画像生成画面において「生成開始」ボタンを押すと、入力された各行に対応する画像が順に生成され、合計 7 枚の画像が作成される仕組みとなっている。また、画像生成処理の進行状況やエラー情報は GUI 上に逐次表示され、ユーザが処理の状態を確認できるようになっている。

### 4.3 生成画像の表示および保存

画像は 1 枚ずつ PNG ファイルとして指定フォルダに保存され、TouchDesigner 側で画像ファイルとして読み込むことで、映像演出に利用される。また、入力されたテキストのうち、「歌詞」として指定された行のみを抽出し、テキストファイルとして保存する機能を実装している。このテキストファイルは TouchDesigner 側で読み込まれ、映像演出時に一行ずつ順番に表示されることを想定している。一方で、「インスト」が選択された行についてはテキストファイルには保存されず、映像上にも歌詞として表示されない仕様となっている。これにより、歌詞を持たない楽曲に対しても、本システムを用いた映像演出を行うことが可能となっている。

#### 【ファイル内画像】 【テキストファイル内画像】

## 5. TouchDesigner を用いた映像制作

TouchDesigner は、Derivative 社によって開発された、リアルタイム映像処理およびインタラクティブコンテンツ制作のためのビジュアルプログラミング環境である。ノードベースの操作体系を特徴としており、映像、音声、センサ入力、MIDI などの多様なデータを組み合わせて処理・制御することができる。そのため、ライブパフォーマンスやインスタレーション、舞台演出など、リアルタイム性が求められる表現分野で広く利用されている。

TouchDesigner では、映像処理を行う TOP (Texture Operator)、数値や制御信号を扱う CHOP (Channel Operator)、三次元空間での描画を行う SOP (Surface Operator) や COMP (Component) など、複数のオペレータを接続することで処理フローを構築する。これにより、プログラムコードを直接記述することなく、視覚的に処理内容を設計できる点が特徴である。

本研究では、TouchDesigner を映像制御および映像制作の中核として用い、MIDI キーボードや USB フットペダルから取得した演奏情報、ならびに生成 AI によって作成された画像およびテキストデータを統合的に処理する。これにより、ピアノ演奏の内容に応じた映像表現をリアルタイムに制御・出力することを可能としている。

本研究において TouchDesigner は初めて使用したツール

であり、主に YouTube や公式ドキュメント、参考サイトなどの資料を参照しながらシステム構築を行った。ノードの接続によって処理の流れを視覚的に把握できる点は、複雑な映像制御を行う本研究において有効であった。

## 5.1 TouchDesigner と Processing との比較

当初は、映像制作および演奏連動システムを Processing で構築していた。Processing とは、MIT メディアラボを起源とする Java をベースとしたプログラミング環境であり、簡潔な記述によって図形描画やアニメーション、入力処理を行うことが可能であるため、ビジュアル表現やインタラクティブアートを中心に広く利用されているソフトウェアである。しかし、映像表現が複雑化し、複数の映像素材やエフェクトを同時に扱うようになるにつれて、描画処理および入力処理の負荷が増加した。その結果、映像の表示に遅延が生じるなど、リアルタイム性の面で課題が見られた。

TouchDesigner は、GPU を活用したリアルタイム処理に優れており、Processing で問題となっていた映像表示のタイムラグを軽減できる点が大きな利点である。

また、ノードベースの操作体系により、映像の合成やエフェクト処理を視覚的に設計・調整できるため、デザイン性の自由度が高く、表現内容に応じた柔軟な映像制作が可能である。この点を示すため、Processing で制作したシステムと、TouchDesigner で制作したシステムの映像例をそれぞれ図○および図○に示す。Processing による映像表現は、主に図形描画を中心とした構成となっており、表現の変更やエフェクトの追加にはソースコードの書き換えを要する。一方、TouchDesigner による映像表現では、複数の映像素材やエフェクトを組み合わせた構成が可能であり、色彩表現や動きのバリエーションにおいて、より高いデザイン性を実現している。

この比較から、TouchDesigner を用いることで、演奏内容に応じた視覚的に豊かな映像演出を柔軟に設計できることが確認できる。さらに、MIDI 入力や外部デバイスとの連携が容易であり、ピアノ演奏情報やフットペダル操作を映像制御に直接反映できる点も、本研究の目的に適している。

これらの理由から、演奏と映像をリアルタイムに統合する本研究において、TouchDesigner は有効な制作環境であると判断した。

## 5.2 制作した映像システムの全体構成

本システムにおける映像制作は、TouchDesigner を用いたノードベースの構成によって実装した。

図○に、本研究で制作した映像システム全体のネットワーク構成を示す。本システムは、大きく分けて「入力処理部」「映像生成部」「映像合成・出力部」の三つの要素から構成されている。【TouchDesigner 内のシステム構成図】

### 5.2.1 入力処理部

入力処理部では、MIDI 信号などの外部入力に加え、LFO CHOP や Noise CHOP を用いて周期的またはランダムに変化する制御信号を生成し、それらを映像の動きやエフェクト強度の制御に利用している。

MIDI 信号を入力として、各鍵盤に対応するノート番号を水平方向の位置情報 (tx) に変換し、これを基準として鍵盤から発生する光エフェクトの x 座標を決定している。また、ベロシティの値はスケール値 (size) に変換され、鍵盤から発生する光や炎の大きさに加え、煙状エフェクトの強度にも用いられる。さらに、LFO (CHOP) や Noise (CHOP) を利用することで、入力画像の動きや桜吹雪、炎の揺らぎといった時間変化を伴う映像表現を実現している。

### 5.2.2 映像生成部

映像生成部では、TOP および Geometry COMP を用いて鍵盤および各種エフェクトの描画を行っている。鍵盤表現は、まず一つの矩形 (Rectangle TOP) を基に白鍵を複製して配置し、その上に位置を調整した黒鍵用の矩形を重ねる構成とした。

さらに、押鍵時の視覚的フィードバックを表現するため、白鍵および黒鍵それぞれに対して、押された場合のみ発光する矩形オブジェクトを別途用意し、MIDI 入力に応じて表示・非表示を制御している。この構成により、演奏中に実際に押された鍵盤のみが発光する仕組みを作っている。

#### 【鍵盤が光る画像】

図○に示すように、ピンク色の鍵盤から飛び出す光および鍵盤上で発光する炎のエフェクトは、Circle TOP を基に構成している。これらのエフェクトは、押鍵時に生成され、鍵盤が押されている時間に比例して縦方向に伸びる構造とした。これにより、鍵盤を長く押し続けるほど、より長い光が飛び出す視覚表現となる。

光および炎の出現位置は、各鍵盤に対応する位置情報 (tx) によって制御されており、エフェクトの大きさや強さは MIDI ベロシティ値を変換したスケール値 (size) によって決定される。また、炎の揺らぎや光の不規則な動きには Noise CHOP や LFO CHOP を用い、時間的に変化する制御信号を付加している。

さらに、Blur TOP, Level TOP, Base COMP, Feedback Edge TOP の各パラメータを調整することで、円形オブジェクトの輪郭を拡散させ、発光感や残像を強調し、光や炎のように見える表現となるよう工夫している。

#### 【鍵盤から出る光・炎の画像】

楽曲「春よ、来い」では、桜の花びらが舞い落ちる情景を表現するため、桜吹雪エフェクトを映像演出として用いている。桜の花びらの画像は Movie File In TOP により読み込み、複数の Transform TOP を用いて複製・配置した。

各花びらの移動には LFO CHOP で生成した周期的な数値を用い、Translate X および Y に適用することで、左右に

揺れながら落下する動きを再現している。また、Rotate パラメータにも LFO による制御を加えることで、花びらが回転しながら降ってくるような視覚表現とした。

さらに、各 Transform TOP に適用する LFO の位相や値をわずかにずらすことで、花びら一枚一枚の動きを変化させ、単調さを抑えた自然な桜吹雪の表現を実現している。

桜吹雪エフェクトの表示・非表示は Switch TOP により制御しており、指定した USB フットペダルが押された時のみ花びらが散る演出が発生する構成とした。この仕組みにより、演奏者が任意のタイミングで情景演出を切り替えることが可能となっている。なお、本手法は桜の花びら以外の画像にも容易に適用することができる。【桜の花びらの画像】

煙エフェクトは、USB フットペダルの指定位置が踏まれた際に発生する演出として実装している。このエフェクトでは、Circle TOP を用いて煙の基本形状を生成し、MIDI 信号から取得したベロシティ情報をスケール (size) の制御にのみ使用している。発射位置には鍵盤位置に対応する tx は用いず、画面の右端および左端に固定した位置から煙が噴き出す構成とした。

煙は画面上部から下方向へ発射され、白色を基調とした表現によって煙の質感を表現している。この仕組みにより、鍵盤を強く打鍵した場合には煙のサイズが大きくなり、弱い打鍵では控えめな煙が生成されるなど、演奏の強弱が視覚表現に直接反映される演出を目指している。

【煙エフェクトの画像】

### 5.2.3 映像合成・出力部

映像合成・出力部では、Composite TOP や Over TOP を用いて、入力処理部および映像生成部で作成した各映像要素を統合する構成としている。鍵盤、光や炎、煙、桜吹雪などの映像はそれぞれ独立したレイヤーとして管理され、Over TOP によって段階的に合成される。

この構成により、各映像要素を個別に調整・制御しながら、全体として一つの映像としてまとめることが可能となっている。最終的に統合された映像は 1 番最後に配置した Over TOP 出力され、外部モニターやスクリーンへの表示を前提とした映像信号として扱われる。

## 5.3 生成画像・歌詞データの利用方法

### 5.3.1 生成画像の表示

生成画像の表示には Movie File In TOP を用い、生成 AI によって作成した画像ファイルを TouchDesigner に読み込んでいる。読み込まれた画像は Transform TOP を通して拡大・縮小および位置調整を行い、Scale X および Y に LFO CHOP から生成した周期的な数値を適用することで、画像が緩やかにズームイン・ズームアウトを繰り返す動きを付加している。これにより、静止画像であっても映像としての動きを持たせることが可能となっている。

また、画像には Level TOP を用いて全体をやや暗くする処理を加えている。これは、鍵盤から発生する光や炎のエフェクト、および歌詞テキスト表示を視覚的に強調することを目的としており、背景画像としての役割を明確にするためである。

生成画像の切り替えは、USB フットペダル操作によって行われる。フットペダルからの入力値をトリガーとして、Python によるスクリプトを実行し、指定したフォルダ内に保存された画像ファイルを順に Movie File In TOP に読み込む構成とした。これにより、演奏中でも演奏者自身が足操作のみで表示画像を切り替えることが可能となっている。

画像切り替え時には、LFO の位相をリセットする処理を行い、ズーム動作が毎回同じ状態から開始されるようにしている。これにより、画像切り替え直後の映像の動きが安定し、視覚的な違和感を抑える工夫を行っている。

【読み込み画像の画像】

### 5.3.2 歌詞の表示

歌詞の表示には Text DAT および Text TOP を用いている。画像生成システムによって事前に作成された歌詞テキストファイルを Text DAT として読み込み、各行を 1 フレーズの歌詞として扱う構成とした。

歌詞の切り替え処理は、USB フットペダルからの入力をトリガーとして実行される。フットペダル操作により Count CHOP の値を更新し、その値を現在表示する歌詞行のインデックスとして利用している。Count CHOP の値が歌詞の総行数を超えた場合には、インデックスを 0 に戻すことで、歌詞表示が先頭に戻るよう制御している。

表示する歌詞は、取得したインデックスに対応する Text DAT の行を参照し、その内容を Text TOP に反映することで画面上に描画される。この処理により、歌詞を一行ずつ順番に切り替えて表示することが可能となっている。

なお、歌詞の切り替えタイミングは、生成画像が切り替わるタイミングと同一のトリガーによって制御されており、歌詞表示と生成映像が常に同期するよう設計されている。これにより、画像に対応した歌詞が表示されるようになる。

また、Text TOP では文字の色、大きさ、フォントを調整し、背景映像やエフェクトとの視認性を考慮したデザインとしている。これにより、映像演出の一要素として歌詞を表示させている。

【歌詞の表示画像】

## 5.4 映像の最終合成および出力

最終合成段階では、鍵盤に連動するエフェクト、背景映像、歌詞表示など、複数の映像要素を一つの画面として整理・統合することを目的としている。複数の Over TOP を用いて各レイヤーの重なり順を調整し、重要な演奏情報に基づくエフェクトが視覚的に埋もれないよう構成した。

具体的には、鍵盤や光・炎などの演奏に直接対応する要素を前面に配置し、桜吹雪や生成 AI による画像を背景として配置することで、画面全体の情報量と視認性のバランスを取っている。完成した映像は 1 番最後に配置した Over TOP へ出力され、ライブパフォーマンスにおける最終的な映像表現としてモニターまたはスクリーンに提示される。

【完成画像】

## 6. 足キーボードを用いた映像制御について

【USB フットペダル・ピアノ配置の画像】本研究では、演奏中に映像演出を直感的に制御する手段として、3 入力 of USB フットペダルを用いた足操作による映像制御システムを導入した。手を使用せずに映像演出を操作することで、ピアノ演奏への影響を最小限に抑えつつ、演奏と映像を同期させることを目的としている。

フットペダルは、歌詞入力画面において区切られたタイミングと同じタイミングで操作する必要があるため、演奏中に自然に踏める位置として、ピアノの下部・左足側に設置した。一方で、ピアノ本来の足ペダル（ダンパーペダル等）は通常通り自由に使用できるようにし、演奏表現を干渉しない構成とした。

演奏時のフットペダルを踏むタイミングを間違わないようにするため、楽譜には踏むタイミングを事前に記載し、該当箇所にシールを貼ることで視覚的に把握できるよう工夫した。演奏者はこの目印を基に、演奏中に適切なタイミングでフットペダルを踏むことで、映像演出を制御する。

3 つのフットペダルにはそれぞれ異なる役割を割り当てている。「春よ、来い」の演奏で用いた楽譜とシールの色について図〇で説明する。

【楽譜】

【シールの色、ペダルの位置、役割 対応付けの図】

一番左のペダルは、歌詞および生成画像を切り替えるための操作に用いられ、楽曲の進行に合わせて映像と歌詞を更新する。中央のペダルは、桜の花びらエフェクトの表示・非表示を制御するために使用し、楽曲の雰囲気に応じた情景演出を行う。一番右のペダルは、煙エフェクトを発生させるための操作に割り当てられており、演奏の盛り上がりに合わせて視覚的アクセントとして機能する。

このように、足キーボードを用いた映像制御により、演奏者自身が演奏の流れに沿って映像演出を能動的に操作することが可能となり、演奏と映像が一体となった表現を実現している。

——途中です——

## 7. 実演について

## 8. 課題と今後の発展について

## 9. まとめ

### 参考文献