

公立はこだて未来大学 2025 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2025 Systems Information Science Practice
Group Report

プロジェクト番号/Project No.

17

プロジェクト名

拡張される文字体験のデザイン

Project Name

Designing Expanded Experiences of Written Language

グループ名

グループ 音楽

Group Name

Group Music

プロジェクトリーダー/Project Leader

稲船悠斗 Haruto Inafune

グループリーダー/Group Leader

佐伯世響 Sena Saeki

グループメンバー/Group Member

佐伯世響 Sena Saeki

稲船悠斗 Haruto Inafune

小澤日向 Hina Ozawa

青山遥香 Hanako Hokkai

指導教員

角康之 塚田浩二 渡邊拓貴

Advisor

Yasuyuki Sumi Tukada Koji Watanabe Hiroki

提出日

2026 年 1 月 21 日

Date of Submission

Jan. 21, 2026

概要

本プロジェクトでは、「文字」という存在そのものに着目し、読む・書くといった従来の役割にとどまらない多様な表現の可能性を探究した。活動は「いきもの」「音楽」「調和」の三つの軸に基づいてグループ分けを行い、それぞれ異なる視点から文字表現に取り組んだ。これらの取り組みを通して、文字を情報伝達的手段としてだけでなく、感性や印象を喚起するメディアとして再定義し、新たな体験の創出を目指した。

本グループでは、楽曲の音響的特徴を分析し、その結果を文字の形状・色・動きといったタイポグラフィ表現へと変換することで、曲名の文字を見るだけで楽曲の雰囲気伝える新しい音楽インターフェースを提案・実装する。これにより、聴覚と視覚を統合したマルチモーダルな音楽体験を提供することを目的とする。

本システムは、音楽分析と文字表現の2つから構成される。音楽分析では Python を用い、入力された楽曲データの分析を行う .librosa を中心としたライブラリを用いて、テンポ (BPM) や音圧、スペクトル特性、リズム構造、MFCC(メル周波数ケプストラム係数) などの音響特徴量を抽出した。ジャンル分析には MFCC を特徴量とした k 近傍法 (k-NN) を使用し、ジャンルの推定を行っている。また、抽出した基礎特徴量を統合・統計処理することで、energy, valence, danceability, intensity といった音楽特性指標を独自に算出した。これらの分析結果は JSON 形式に変換され、文字表現で使用される。

文字表現では Unity を用い、音楽分析によって得られた特徴量をもとに文字表現の生成および表示を行った。ジャンル分析結果に基づいたフォントの選択や、音響特性と文字の色や歪み、太さ、動きなどを対応させることで、音楽の印象を視覚化した。

本手法の有効性を検討するため、複数の楽曲を用いたデモンストレーションを行い、成果発表会において第三者からのフィードバックを収集した。その結果、「着眼点が良い」「これまで見たことのない表現で面白い」といった評価が得られ、音楽と文字を結び付ける表現の新規性が確認された。一方で、ジャンルに基づくフォント固定や一定時間ごとの分析により、楽曲固有の特徴や時間的変化が十分に反映されていない点が課題として挙げられた。

これらの結果を踏まえて今後は、文字を見るだけで曲の雰囲気を直感的に把握できるインターフェースの改善のため、ジャンルと楽曲固有の特徴をより両立させて反映させた文字表現の生成を目指す必要があるだろう。

キーワード 文字体験, 音楽, タイポグラフィ, 音楽分析

Abstract

This project focused on text itself, exploring expressive potential beyond its customary roles of “reading” and “writing.” Each group approached typographic expression from different perspectives, based on three core themes: “creatures”, “music” and “harmony”. Through these activities, we aimed to define text not simply as a means of communicating information, but as a medium capable of awakening emotions and impressions, proposing new forms of experience.

This group analyzes the sound features of musical pieces and converts the results into typographic expressions such as character shape, color, and motion. We propose and realize a new music interface that conveys the feel of a song simply by viewing its title characters. This aims to provide a multimodal music experience integrating audio and visual elements.

This system consists of two components: music analysis and textual expression. For music analysis, Python is used to analyze input song data. Using libraries based on librosa, acoustic features such as tempo (BPM), sound pressure, spectral characteristics, rhythm structure, and MFCC (Mel-frequency cepstral coefficients) are extracted. For genre analysis, k-nearest neighbors (k-NN) using MFCC as the feature performs genre estimation. Furthermore, by integrating and statistically processing the extracted basic features, music characteristic indices such as energy, valence, danceability, and intensity are calculated independently. These analysis results are converted into JSON format for use in the textual representation.

For textual representation, Unity was used to generate and display typographic expressions based on features extracted through music analysis. By selecting fonts according to the results of genre analysis and mapping acoustic characteristics to visual parameters such as text color, distortion, thickness, and motion, the system visualizes the impression of music in a visual form.

To test the effectiveness of this approach, we demonstrated it using multiple musical pieces and collected feedback from third parties at a results presentation. The feedback included comments such as “The perspective is excellent” and “It’s interesting with expressions I’ve never seen before,” confirming the originality of this approach that links music and text. However, issues were raised regarding the limited reflection of song-specific features and temporal changes due to genre-based font locking and analysis at fixed intervals.

Based on these results, it will be necessary to aim for generating textual expression that better reflects both genre and song-specific features in order to improve the interface so that users can instantly recognize a song’s feel just by looking at the text.

Keyword experiences of written language, music, Typography, Audio Feature Analysis

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	目的	1
第 2 章	関連研究と専門知識の位置づけ	2
2.1	関連研究	2
2.2	本研究の位置づけ	2
2.3	使用される技術	2
2.3.1	Python	2
2.3.2	Unity	3
2.4	関連する専門科目との関係	3
第 3 章	目標	4
第 4 章	手段, 手法	5
4.1	考案したアイデアとその意義	5
4.2	作成したシステムの構成	5
4.3	音楽分析手法	6
4.3.1	ジャンル分析	7
4.3.2	特徴量分析	8
4.4	文字表現への変換手法	9
第 5 章	結果	11
5.1	完成したシステムの概要	11
5.1.1	システム上での音楽の読み込み	11
5.1.2	システム上での音楽の分析	11
5.1.3	システム上での文字への反映	12
5.2	FutureBass における視覚表現の例	12
5.2.1	特徴量	13
5.2.2	視覚表現	13
5.3	HyperPop における視覚表現の例	14
5.3.1	特徴量	14
5.3.2	視覚表現	15
第 6 章	考察	16
6.1	成果発表から得られたこと	16
6.2	課題と改善点	17
6.3	今後の展望	18

参考文献	19
付録 A 音楽分析で使したパラメータ	20
A.1 基礎音響特徴量 (Librosa によるダイレクト抽出)	20
A.1.1 テンポ・ビート関連	20
A.1.2 エネルギー・ダイナミクス	20
A.1.3 周波数・スペクトル特性	20
A.1.4 メル周波数ケプストラル係数 (MFCC)	21
A.1.5 調性・ハーモニー分析 (Chroma)	21
A.2 正規化と記号	21
A.3 派生指標の計算式	22
A.3.1 Acousticness (アコースティック度)	22
A.3.2 Danceability (踊りやすさ)	22
A.3.3 Instrumentalness (インスト度)	22
A.3.4 Liveness (ライブ感)	22
A.3.5 Speechiness (話し声度)	22
A.3.6 Valence (明るさ・ポジティブ度)	23
A.3.7 Energy (エネルギー)	23
A.3.8 Intensity (激しさ・攻撃性)	23
A.3.9 EDM Signature (EDM らしさ)	23
A.3.10 Kawaii Score (かわいいスコア)	24
A.4 使用パラメータの完全一覧	24
A.4.1 基礎音響特徴量 (直接抽出)	24
A.4.2 派生指標 (計算値)	25
A.4.3 内部計算値 (中間パラメータ)	25
A.5 パラメータ抽出のアルゴリズム概要	25
A.5.1 前処理	25
A.5.2 特徴抽出の順序	25
付録 B システムで利用したフォント一覧	27
付録 C 音楽分析の出力結果	28
C.1 FutureBass	28
C.2 HyperPop	30

第 1 章 はじめに

1.1 背景

現代の音楽体験は、ストリーミングサービスの普及によって「聴く」中心の消費型へと変化してきた。しかし、楽曲に付随する情報の多くは依然としてテキストとして提示されており、ユーザーは曲名を“読むだけ”で、その文字そのものを視覚的・情緒的に味わうことはほとんどない。一方で、タイポグラフィは本来、形状・線の太さ・リズム・色彩といった多様な表現手段を持ち、視覚的なメディアとして豊かな感情喚起力を持つ。

近年、音楽分析技術や音響特徴量抽出の精度が高まったことで、楽曲のテンポ・エネルギー・音色・ジャンルといった情報を定量的に扱うことが可能になった。しかし、それらの豊富なデータは多くの場合レコメンドアルゴリズムや内部処理に利用されるに留まっており、ユーザーが直接触れられる表現として活用されているケースは少ない。

また、UI デザインや VJ（ビジュアルジョッキー）文化の観点から見ても、音楽と視覚表現を結びつける試みは盛んに行われてきたが、その多くは抽象的な映像が中心である。日常的に接触する「文字」を音楽の視覚的インターフェースとして扱う研究は限定的である。

このような現状を踏まえ、本グループは音楽分析から得られた情報を曲名の文字にタイポグラフィのような表現として反映させ、文字を見ると曲の雰囲気が伝わるインターフェースを開発した。

1.2 目的

本グループでは、楽曲の音響的特徴をタイポグラフィ要素（フォント・色・動き）へ直接対応付けることで、曲名の文字だけで楽曲の雰囲気を伝える新しい音楽インターフェースを提案・実装する。これにより、音楽配信サービスでの楽曲選択時や、ライブ会場でのビジュアル演出において、聴覚と視覚を統合したマルチモーダルな音楽体験を提供することを目的とする。

このアプローチは、従来の抽象的な音楽ビジュアライゼーションとは異なり、日常的に接する「文字」という身近なメディアを活用することで、文字と音の新たな関係性を構築する。タイポグラフィ・UI/UX・音楽可視化の境界領域において新しい表現可能性を提示する試みであり、デザイン研究および音楽インターフェース研究の双方に貢献するものである。

第 2 章 関連研究と専門知識の位置づけ

2.1 関連研究

音楽情報の可視化に関する研究は、音楽の特徴を視覚的に提示することで、楽曲の構造や印象を理解しやすくすることを目的として発展してきた。松原（2015）[4]によると、音楽情報可視化は楽曲を単一の評価軸で捉えるのではなく、複数の特徴量を同時に提示することで、多面的な視点から音楽を理解することを可能にする手法であるとされている。また、可視化は音楽鑑賞や分析における補助的なインタフェースとして機能し、音楽の構造や特徴の把握を支援すると述べられている。これらの指摘は、本研究における「音楽特徴量を視覚的に提示する」という基本方針を理論的に裏付けるものである。

音楽と文字表現を結びつけた代表的な研究として、加藤ら（2015）[1]による TextAlive Online が挙げられる。TextAlive Online は、楽曲と歌詞のタイミング情報を解析し、音楽再生に同期した動的な文字表現（キネティック・タイポグラフィ）を生成することを目的としたシステムである。この研究では、音楽と文字表現を連動させることで、従来の聴覚中心の音楽体験に視覚的要素を付加できることが示されている。

しかし、TextAlive Online は主に歌詞という意味情報を対象としており、音楽そのものの音響的特徴を文字の形状や色、動きに反映することを主眼としていない。

2.2 本研究の位置づけ

松原（2015）[4] が示すように、音楽情報可視化には音楽の特徴そのものを多面的に提示する価値がある。一方、加藤ら（2015）[1] の TextAlive Online は、音楽と文字表現を結び付ける有効な表現手法を示しているものの、歌詞という意味情報に依存した表現が中心であり、音響的特徴量そのものを文字表現へ反映することを目的としていない。

本研究は、これらの先行研究を踏まえ、歌詞に依存せず、音楽分析によって得られた音響特徴量を文字のフォント、色、太さ、動きといったタイポグラフィ要素へ直接対応付ける点に特徴がある。文字を単なる情報伝達手段としてではなく、音楽の印象や雰囲気や視覚的に伝える媒体として位置付けることで、「文字を見るだけで曲の雰囲気が伝わる」新たな音楽可視化手法の提案を目指す。

2.3 使用される技術

本グループでは、楽曲の音響データを解析し、その結果を文字表現として可視化するために、Python と Unity を用いたシステムを構築した。Python は音楽データの分析処理を行い、Unity では分析結果をもとにした文字表現の生成を行う。

2.3.1 Python

Python では、音楽のジャンル分析および音響特徴量の抽出を行った。librosa, NumPy, pydub などのライブラリを使用し、楽曲の波形データから MFCC(メル周波数ケプストラム係数)、テン

ポ、音圧、スペクトル特性、リズム構造などの特徴量を算出した。ジャンル分析には MFCC を特徴量とした k 近傍法 (k-NN) を用い、複数ジャンルへの自動分類を行った。また、抽出した基礎音響特徴量を統計処理し、楽曲の明るさやエネルギー、激しさといった高レベルの音楽特性値を算出した。これらの分析結果は、Unity で扱うことのできる JSON 形式に変換して出力している。

2.3.2 Unity

Unity は、Python により得られた分析結果をもとに、文字表現を生成・表示するために使用した。ジャンルごとに対応するフォントを適用し、音楽特性に応じて文字の色、太さ、大きさ、動きなどを制御することで、楽曲の印象を視覚的に表現する。また、BPM に同期した文字のアニメーションや、音圧に応じた文字の変形などを実装し、音楽と文字の時間的な連動を表現できるようにする。

2.4 関連する専門科目との関係

本グループにおける各種機能の実装では、Python を用いた音楽データの解析および Unity を用いた可視的表現の描画を行った。これらの実装は、本学における講義である「データサイエンス入門」および「データサイエンス演習」、「インタラクティブシステム」において学習した知識・技術を基盤として進められている。

データサイエンス関連の講義では、Python による基本的なプログラミング手法に加え、NumPy や pandas を用いた数値データおよび時系列データの処理、Matplotlib を用いたデータの可視化手法について学習した。また、scikit-learn を用いた機械学習の基礎的手法についても扱われており、分類や特徴量の取り扱いに関する理解を深めた。これらの知識を活用することで、音楽データから特徴量を抽出し、それらを定量的に扱うための分析処理を実装することが可能となった。

一方、「インタラクティブシステム」の講義では、Unity を用いたアプリケーション開発を通じて、視覚表現とユーザインタラクションを組み合わせさせたシステム構築の方法を学習した。本プロジェクトでは、Python によって得られた分析結果を Unity 側へ連携し、文字表現として動的に反映させる必要があったため、これらの講義で得た知識が有効に活用された。

以上のように、本学で学習したデータ分析およびインタラクティブシステムに関する知識を組み合わせることで、本プロジェクトに必要な音楽分析処理と視覚的表現の実装を円滑に進めることができた。

第3章 目標

本グループの最終的な目標は、楽曲の音響特徴を分析し、その結果をタイポグラフィ要素へと変換することで、「文字を見るだけで曲の雰囲気が伝わる」新しい音楽可視化体験を創出することである。

具体的には、Python による音響特徴量の抽出・統計処理と、Unity によるリアルタイム文字表現生成を組み合わせ、ジャンル、エネルギー、音圧、明るさなど多様な特徴を文字のフォント・色・動き・質感として反映させる。これにより、従来の「読む文字」ではなく、音のニュアンスを宿した動的な文字表現を実現し、音楽理解に対する新たなアプローチを確立する。

また、本システムは音楽配信サービスでの楽曲選択支援やライブ演出での視覚表現など複数の応用可能性を想定している。ジャンル分類に依存するだけでなく、曲固有の音響特性を反映することで、同一ジャンル内でも異なる印象を持つ楽曲を適切に差別化できる表現を目指す。

目標達成の評価は、最終発表で得られた第三者フィードバックに基づき行う。「文字を見るだけで楽曲の雰囲気が伝わるか」「音楽体験としての新規性や理解のしやすさが感じられるか」「ジャンルや曲ごとの差異が適切に表現されているか」という観点から肯定的評価を得ることで、本システムが音楽可視化手法として一定の有効性を持つことを示す。

第 4 章 手段, 手法

第 3 章で示した目標を実現するために, 本プロジェクトでは音響分析技術とタイポグラフィ表現の双方からアプローチし, 音と文字を結びつける具体的な方法を構築した. 以下では, 考案したアイデアと, その実現に利用した技術・手法について述べる.

4.1 考案したアイデアとその意義

本グループの中心となるアイデアは, 音楽の音響特徴や感性情報を文字表現に反映させる点にある. 従来, 文字は主に情報を伝える記号として扱われ, その形態や動きが音楽と直接的に結びつくことはほとんどなかった. しかし, タイポグラフィが持つ線の太さ, 丸み, 伸び, 色彩といった多様な表現要素を活用することで, 文字を媒体とした視覚的な感性伝達が可能となる. 李 (2017) [3] は, 文字の形状や線幅が受け手の心理的印象に影響を与えることを示しており, 太い文字は「力強い」, 丸みのある文字は「柔らかい」といった印象を形成することが明らかになっている.

本グループでは, 音響特徴量 (テンポ, エネルギー, 音圧, スペクトル情報など) と文字表現の要素 (フォント, 太さ, 色, 動き) とを対応させることにより, 音楽の複雑な印象を直感的に読み取れる「文字による音の表現」を実現した. たとえば, エネルギーの高い曲では彩度の高い赤系統の色を使う, 音圧が強ければ文字を太く伸ばす, BPM が速い曲では文字の点滅や振動を速くする, というように, 音響特徴の違いを文字の振る舞いとして視覚化している. 色彩が感情に与える影響については, 柳瀬 (1997) [5] により「赤は興奮・情熱」「青は冷静・静寂」といった体系的な整理がなされており, 本研究ではこの知見を活用している (表 4.1 参照).

このアイデアの意義は, 音楽の理解や鑑賞方法を「聴く」だけの体験から, 視覚と言語による多層的な理解へと拡張できる点にある. 松原 (2015) [4] は, 音楽情報の可視化が楽曲を多面的な視点から理解することを可能にすると述べており, 本研究もこの方針に基づいている. さらに, ジャンルごとにフォントを選択する仕組みを導入することで, 「Future Bass は丸く可愛い文字」「Metal は尖った鋭い文字」といった音楽のイメージそのものを視覚化でき, 加えて, 普段は聴覚だけでは気づきにくいジャンル間の意外な共通性を, 視覚を通して直感的に感じ取ることも可能になる.

4.2 作成したシステムの構成

本システムは, 音楽分析と文字表現生成の 2 つの要素から構成される. 音楽分析では, Python および librosa を用いて楽曲ファイルから音響特徴量の抽出とジャンル分類を行い, 得られた分析結果を JSON 形式で出力する. 文字表現生成部では, Unity がこの JSON データを読み込み, 音響特徴に対応した視覚表現を生成・表示する. 本システムはリアルタイム処理ではなくオフライン解析を前提とした設計であり, 分析と表示を明確に分離することで, 各処理の独立性と拡張性を確保している. 図 4.1 にシステム全体の技術構成を示す.

表 4.1 色と感情的印象の対応関係 (柳瀬, 1997)

色相	赤	黄	緑	青	紫	白	黒
象徴語	用熱	明朗	さわやか	静寂	高尚	清潔さ	不安
	危険	快活	若さ	冷たい	優雅	たよりない	死
	興奮	躍動	平和	涼しさ	魅力	空虚	不気味
	暑い	のどか	新鮮さ	開放的	不良	可能性	陰気
	吸い込まれる	若さ	おだやか	眠り	あでやかさ	軽い	力
	怒り	前進的	安息	知性	女性的	真理	アク
	爆発	ゆるみ	安全	精神	田舎者	虚脱感	重々しさ
	圧力	注意	やすらぎ	深さ	うぬぼれ	清潔	孤独
	故意	スピード	幸福	清潔	孤独	純潔	沈黙
	活動的	軽い	豊かさ	悲しみ	複雑さ	明るい	暗さ
	動き	不安	生命力	失望	不安	冷たい	さみしさ
	派手	危険	健康	平和	上品	派手	地震
	愛	緊張	すっきりした	気持ちよい	永遠	新しい	男性的
	生命	軽率	自然	不安	神秘	味気ない	極限
	勇気	はなやか	平穩	純粹	不吉	理知的	絶対的
	喜び	柔らかい	未熟	安息	エキゾチック	威厳	拒絶
	燃える	愉快	清潔	温和	豪華さ	放心状態	厳粛
	争い	いらいら	静的	安楽	高級	薄情	豪華さ
	強烈	発展	落ち着き	冷酷	冷たい	スポーティー	深さ
	残酷	可愛らしい	涼しい	理知的な	ムード	おごそか	どん底
激しい	さわやか	美しさ	憂うつ	大人っぽい	冷淡	絶望	
積極	弱々しい	情的	広大さ	神聖	無限	冷淡さ	
愛情	活発	希望	デリケート	崇高	無意味	ニヒル	

システム構成(使用技術)

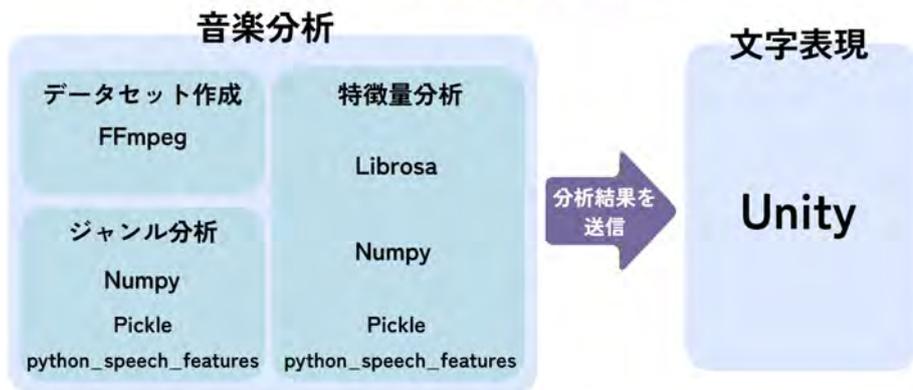


図 4.1 システムの技術構成

4.3 音楽分析手法

本グループでは、楽曲の印象を客観的に扱うために、音響信号処理に基づく特徴量分析を行った。歌詞やテキスト情報は用いず、音そのものが持つ物理的・構造的な特徴のみに着目している点が本手法の特徴である。

4.3.1 ジャンル分析

データセット構築

まず，入力された楽曲データに対して前処理を行い，サンプリングレートやファイルフォーマットを統一した上で解析を実施した．音響特徴量の抽出には Python および librosa を使い，テンポ (BPM)，音圧，スペクトル特性，リズム構造，音色といった基礎的な特徴量を算出した．

ジャンル分析に用いる学習データセットは，本研究で扱うジャンルおよびサブジャンルごとに複数の楽曲を収集し，表 4.2 にある通り，計 20 ジャンルのデータセットを独自に構築した．ジャンルおよびサブジャンルの選定にあたっては，Spotify の公式プレイリストに加え，Chosic が提供する Music Genre Finder を参考にした．Music Genre Finder は，楽曲の音響的特徴に基づいてジャンルやサブジャンルを推定するサービスであり，ジャンルの境界が曖昧な楽曲についても，音楽的傾向を客観的に把握するための補助的指標として活用した．

表 4.2 fugafuga

ジャンル名	サブジャンル名	曲数
Classic		105
EDM	DubStep	66
EDM	ElectroHouse	38
EDM	FutureBass	66
EDM	House	41
EDM	Trap	47
Enka		44
HipHop	Drill	50
HipHop	EmoRap	56
HipHop	OldSchool	21
Jazz		39
Pops	HyperPop	83
Pops	ElectroPop	38
Pops	J-Pop	38
Pops	Pop	106
R & B		33
Rock	Alternative Rock	169
Rock	Classic Rock	46
Rock	Metal	33
Rock	Punk	73

収集した楽曲数は，全ジャンル・サブジャンルを合わせて約 1,100 曲であり，各ジャンルにつき数十曲から百曲程度を含む構成となっている．各楽曲について MFCC（メル周波数ケプストラム係数）を算出し，得られた MFCC の平均値および分布を各ジャンルを代表する音色特徴として保持している．選曲にあたっては，ジャンルごとの音響的特徴が比較的明確に現れている楽曲を中心に採用し，学習データセットの偏りを抑えるよう配慮した．

分析手法

ジャンル分類には、音色の特徴を表す指標である MFCC を特徴量として使用し、k 近傍法 (k-NN) による分類を行った。MFCC は楽曲の音色的傾向を低次元で表現できるため、ジャンルごとの音響的違いを捉えるのに適した特徴量である。

入力楽曲についても学習データと同様に MFCC を算出し、その分布と学習データセット内の各ジャンルの MFCC 分布を比較することで、統計的に最も近いジャンルを推定結果として出力する。k-NN による分類では、距離尺度に基づいて複数の近傍データを参照するため、単一曲に特有のノイズや一時的な特徴に左右されにくく、楽曲全体の音響的傾向に基づいた安定したジャンル推定が可能となる。

4.3.2 特徴量分析

ジャンル分析に加え、本研究では楽曲の印象や雰囲気をより詳細に捉えるため、複数の音響特徴量を抽出し、それらを総合的に分析した。特徴量分析の目的は、楽曲の「明るさ」「激しさ」「リズム感」「音色の質感」といった抽象的な印象を、数値として扱える形に変換し、後段の文字表現に反映させることである。特徴量の抽出には Python および librosa を用い、時間領域・周波数領域の双方から分析を行った。まず基礎音響特徴量として、テンポ (tempo)、音量の平均値および変動幅 (RMS mean / std)、スペクトル重心、スペクトルロールオフ、スペクトル帯域幅、スペクトルコントラスト、ゼロ交差率 (ZCR) などを算出した。これらの特徴量は、楽曲のスピード感、明るさ、音の広がり、ノイズ感といった要素を表す指標であり、ジャンルや楽曲ごとの音響的個性を捉える基礎情報となる。

次に、リズム構造の把握を目的としてオンセット検出を行い、オンセット間隔の平均値および分散を算出した。これにより、ビートの強さや規則性、リズムの複雑さを定量的に評価することが可能となる。特に EDM 系ジャンルや HyperPop など、リズムの密度や変化が印象に大きく影響する楽曲において有効な指標である。

さらに、調性感に関する分析として、キー (key)、モード (長調・短調)、キーの明瞭度や安定度を推定した。これらの特徴量は、楽曲の明暗や感情的な方向性を示す要素として扱い、後述するポジティブ度 (valence) や文字表現の色調選択に利用している。音色分析では、MFCC の平均値および分散を算出し、楽曲全体の音色的傾向とその変化量を捉えた。MFCC は人間の聴覚特性に基づいた特徴量であり、楽器構成やサウンドデザインの違いを反映しやすい。そのため、ジャンル分析のみならず、文字の形状変化や歪み量の制御にも利用している。

これらの基礎特徴量を基に、本研究では「energy」「valence」「danceability」「intensity」などの音楽特性指標を独自に算出した。これらは単一の特徴量ではなく、音量、テンポ、スペクトル特性、調性情報など複数の要素を重み付けして統合することで求めている。例えば、energy は音量や高周波成分、アタックの強さを総合した指標であり、楽曲の力強さや迫力を表す。一方、valence はテンポ、調性、音色の明るさなどから算出され、楽曲のポジティブさや楽しさを表現する。

加えて、EDM 系楽曲に特有のリズム構造を捉えるため、4 つ打ちの規則性やサイドチェイ的な音量変動を検出し、EDM らしさを示す指標として「edm-signature」を算出した。また、FutureBass や J-Pop などに見られる「かわいらしさ」を定量化する試みとして、音色の明るさや高音域成分、ジャンル傾向を組み合わせた「kawaii」指標を導入した。これらの特徴量は楽曲全体を通して一定区間ごとに算出され、時間的な変化も考慮した上で集約される。最終的に得られた特徴量群は JSON

形式で出力され、文字の大きさ、色、形状、動きといった視覚表現の各パラメータへと対応付けられる。尚、パラメータの一覧は付録 A に示す。このように、特徴量分析は音楽の印象を視覚表現へと橋渡しする中核的な役割を担っている。

4.4 文字表現への変換手法

音楽分析によって得られた数値的な特徴量を文字の視覚的要素へ変換するため、本グループでは音響特徴と視覚表現との間に明確な対応関係を設計した。本研究における文字表現は、単なる装飾ではなく、音楽の特徴や印象を直感的に理解可能な情報表現として機能させることを目的としている。

まず、ジャンル分析の結果に基づきフォントの選択を行った。ジャンルは楽曲全体の雰囲気や音楽的傾向を包括的に表す要素であり、視覚的な第一印象に大きく影響する。文字の形状やスタイルが受け手の印象形成に影響を与えることは、タイポグラフィ研究においても指摘されている（李, 2017） [3]。この知見を踏まえ、本研究ではジャンルごとに異なるフォントを対応づけることで、文字を見た段階で楽曲の方向性や印象を把握できる表現を目指した。次に、音楽の明るさやポジティブさを示す指標である valence の値を、文字の色相および明度に反映させた。色彩が人間に与える感情的影響については、柳瀬（1997） [5] により体系的に整理されており、「赤は興奮・情熱」「青は冷静・静寂」といった一般的傾向が示されている。本研究ではこの知見を基に、valence が高い楽曲ほど明るく暖色系の色彩を、低い楽曲ほど暗く寒色系の色彩を用いる設計とした。また、energy の値は彩度に対応づけ、エネルギーの高い楽曲ほど鮮やかな色調となるようにすることで、音楽の活発さや強度を視覚的に強調している。

音圧や迫力を表す指標については、文字の太さや縦方向の伸びとして反映した。文字の線幅が心理的印象に与える影響について、李（2017） [3] は、太い文字は「力強い」「重い」と評価され、細い文字は「繊細」「軽い」と評価される傾向があることを示している。本研究ではこの結果を踏まえ、音量やダイナミクスが大きい楽曲ほど文字を太く変形させることで、音の強さを視覚的に伝達する設計とした。さらに、intensity が高い楽曲に対しては、文字に歪みや揺れを付加することで、楽曲の激しさや不安定さを表現している。

加えて、楽曲のテンポ（BPM）に同期した文字の点滅や動きを実装した。音楽の時間的変化を視覚表現に同期させる手法は、音楽可視化研究において有効であることが示されており、松原（2015） [4] は、音楽構造と視覚変化を対応づけることで多面的な理解が促進されると述べている。本研究ではこの考え方を応用し、文字の動きをテンポに同期させることで、音楽の流れを視覚的にも体感できる表現を実現した。

以上の対応関係を組み合わせることで、楽曲ごとに異なる文字表現が自動的に生成され、文字を見るだけで音楽の雰囲気や特徴を把握できる可視化を実現した。

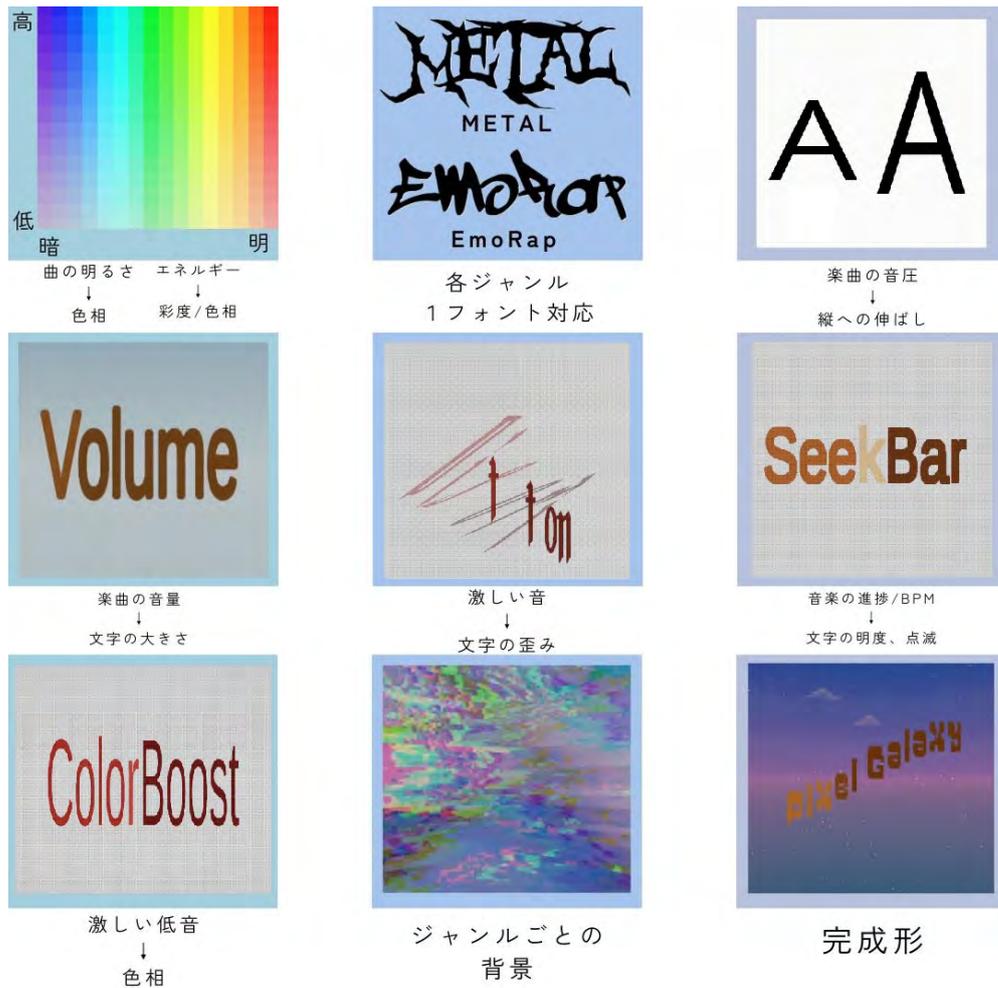


図 4.2 音楽の可視化方法

第 5 章 結果

本システムは、任意の音楽ファイルを読み込み、音響的特徴を分析した上で、その結果を文字表現として可視化する音楽プレーヤーである。音楽の再生に合わせて文字の色・形状・動きが変化し、視覚的に楽曲の特徴を理解できる構成となっている。本章では、複数の楽曲を用いた具体例を挙げ、本手法によってどのような文字表現が得られたかを評価する。

5.1 完成したシステムの概要

5.1.1 システム上での音楽の読み込み

本システムでは、画面上の SELECT ボタン を押すことで、任意の音楽ファイルを読み込むことができる。読み込み可能な音楽ファイルは WAV 形式とし、ファイル名は「楽曲名.wav」の形式を想定している。音楽ファイルが選択されると、自動的に音楽分析処理が開始される。



図 5.1 システムの画面

5.1.2 システム上での音楽の分析

音楽ファイルの読み込み後、Python による音楽分析が実行される。この処理では、MFCC を用いたジャンル分析および、テンポ、エネルギー、明るさなどの音楽的特徴量の抽出が行われる。分析結果として、楽曲のジャンルおよび複数の音楽特性が数値データとして出力され、Unity 側へ送信される。各ジャンルの分類精度は以下の通りとなっている。

尚、実際の JSON 形式の分析結果の内容は付録 C に記載する。

表 5.1 ジャンルごとの分類精度の一覧

ジャンル名	サブジャンル名	分析精度
Classic		0.9905
EDM	DubStep	0.7424
EDM	ElectroHouse	0.7895
EDM	FutureBass	0.7551
EDM	House	0.7317
EDM	Trap	0.5532
Enka		0.8332
HipHop	Drill	0.9
HipHop	EmoRap	0.7679
HipHop	OldSchool	0.7619
Jazz		0.8718
Pops	HyperPop	0.8434
Pops	ElectroPop	0.7368
Pops	J-Pop	0.8158
Pops	Pop	0.9151
R & B		0.7576
Rock	Alternative Rock	0.2899
Rock	Classic Rock	0.7609
Rock	Metal	0.9091
Rock	Punk	0.8356

5.1.3 システム上での文字への反映

音楽分析によって得られた結果は、事前に設計した対応関係に基づき、動的に文字表現へ反映される。楽曲が再生されると、付録 B にある通り分析結果に応じたフォントが自動的に適用され、文字の色、太さ、形状、動きが楽曲ごとに变化する。文字の視覚的变化は楽曲全体の分析結果だけでなく、再生中の時間変化にも対応しており、サビや音圧の高い部分では文字の変形や動きが強調される。これにより、楽曲の盛り上がりやリズムの変化を視覚的に感じ取ることが可能となっている。また、テンポ情報に基づいて文字がリズムに同期して動作するため、文字が音楽と一体となって振る舞うような表現が実現されている。これらの処理はすべて自動で行われ、ユーザは文字を見るだけで楽曲の雰囲気や特徴を直感的に把握できる。尚、具体的な動作例は 5.2 および 5.3 にて分析精度

5.2 FutureBass における視覚表現の例

本節では、本システムの有効性を検証するため、実際の楽曲を用いた表示結果の具体例を示す。対象とするジャンルは FutureBass とし、楽曲には Snail' s House – Pixel Galaxy を使用した。本楽曲は、明るくポップな音色と高いエネルギーを持つ FutureBass の特徴が顕著であり、本システムによる文字表現の変化を確認するのに適した楽曲である。以下では、まず音楽分析によって得ら

れた特徴量について述べ、その後、それらがどのように文字の視覚表現として反映されたかを示す。



図 5.2 FutureBass の楽曲をシステムに読み込ませた際の視覚表現結果

5.2.1 特徴量

Snail's House – Pixel Galaxy を本システムに読み込んだ結果、以下のような音楽特徴量が得られた。尚、実際の JSON 形式の分析結果の内容は付録 C に記載する。テンポは約 152 BPM と比較的高速であり、FutureBass に特徴的な軽快で疾走感のあるリズムが確認された。danceability は 0.69 と高めの値を示しており、一定の規則性を持ったビート構造が存在することが分かる。音色に関しては、spectral centroid (約 3247) や spectral rolloff (約 6927) が高い値を示しており、高音域成分が豊富で明るい音色を持つ楽曲であることが示唆される。また、timbre-brightness および timbre-complexity がともに 1.0 と最大値であることから、キラキラしたシンセサウンドと多彩な音色変化が強く表れている。感情的特徴を示す valence は 0.77 と高く、楽曲全体として明るくポジティブな印象を持つことが分かる。energy も 0.82 と高く、音量やアタックの強さを伴ったエネルギッシュな楽曲であると評価できる。一方で intensity は 0.77 と比較的高いものの極端ではなく、激しさよりも楽しさや躍動感が強調された楽曲であることが示されている。調性分析ではキーが E minor と推定されたが、mode-confidence は約 0.51 と中程度であり、明るい音色やコード進行の影響により、長調的な印象も併せ持つ楽曲であると考えられる。ジャンル分析の結果、楽曲は EDM / FutureBass に分類され、近傍ジャンルとして HyperPop や ElectroHouse が挙げられた。これは、本楽曲が FutureBass を基調としつつ、ポップ性やエレクトロ要素を含むことを示しており、分析結果は人間の聴感による印象とも概ね一致している。

5.2.2 視覚表現

読み込まれ、分析された音楽は図 5.2 のように視覚化された。本楽曲はジャンル分析の結果に基づき FutureBass 用に設定したフォントが適用され、全体として丸みのあるドットのフォントが適応された。これは、FutureBass に特徴的なポップさや親しみやすさや電子音感を強調するためである。文字の色彩は、楽曲の valence および energy の値を反映し、明度および彩度の高い、総じて

エネルギーの強い橙色の配色となった。文字の動きに関しては、約 152 BPM のテンポに同期した周期的なアニメーションが適用されており、キックやビートに合わせて文字が動き、BPM に合わせて文字が点滅する表現が確認できた。これにより、文字が音楽と同じリズムで「音楽にのっている」ような印象を与えている。

5.3 HyperPop における視覚表現の例

本節では、FutureBass とは異なる音楽的特徴を持つジャンルとして HyperPop を取り上げ、本システムによる表示結果を示す。対象楽曲には、lilbeshramko - lost woods! を使い、強い歪みや高い intensity を持つ音楽に対して、文字表現がどのように変化するかを確認した。以下では、音楽分析によって得られた特徴量と、それに基づく文字の視覚表現について述べる。

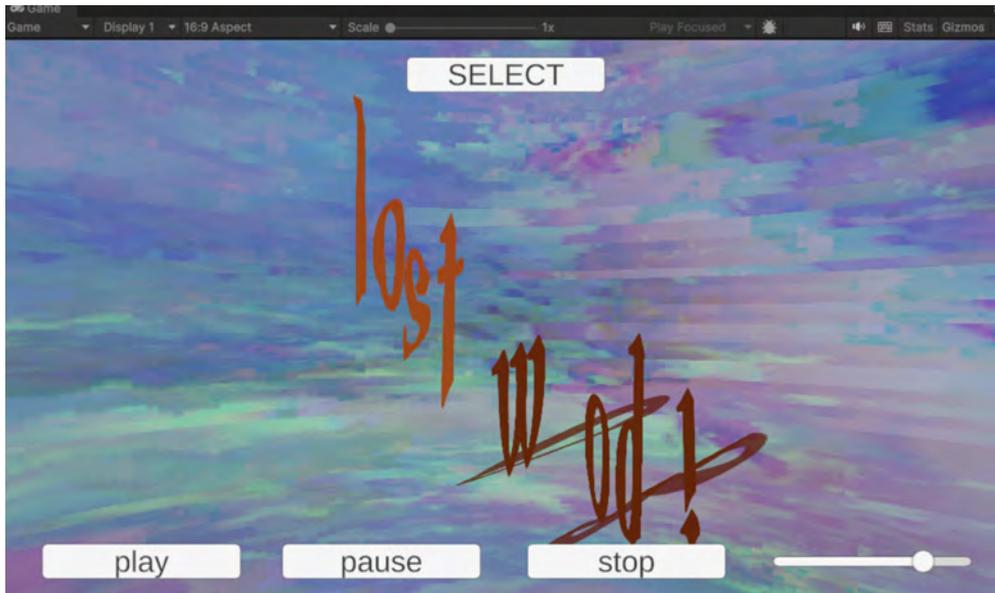


図 5.3 HyperPop の楽曲をシステムに読み込ませた際の視覚表現結果

5.3.1 特徴量

lilbeshramko - lost woods! を本システムに読み込んだ結果、HyperPop ジャンルに特徴的な音楽特徴量が得られた。尚、実際の JSON 形式の分析結果の内容は付録 C に記載する。テンポは約 172 BPM と非常に高速であり、FutureBass と比較してもより過激で切迫感のあるリズム構造を持つことが分かる。onset-mean および onset-std も高く、リズムが細かく分断され、予測しづらいビート構造を持つ楽曲であることが示されている。音圧に関する指標である rms-mean は 0.33, rms-std は 0.23 と大きな値を示しており、音量変化が激しく、ダイナミクスの強い楽曲であることが確認された。さらに hpss-ratio が 0.95 と非常に高いことから、パーカッション成分が支配的であり、アタック感の強いサウンドが楽曲全体を構成していると考えられる。音色面では、spectral centroid や rolloff が高い値を示しており、高音域成分が非常に強調された鋭い音色を持つことが分かる。また、timbre-brightness および timbre-complexity がともに 1.0 であることから、歪みや加工を含む複雑な音色変化が頻繁に現れる楽曲であることが示唆される。感情的特徴を示す valence は 0.81 と高く、ポップで明るい印象を持つ一方で、intensity は 0.99 と極めて高い値を示している。これは、楽曲が明るさと激しさを同時に内包する、HyperPop 特有の誇張された表現

を持つことを反映している。ジャンル分析の結果、本楽曲は Pops / HyperPop に分類され、近傍ジャンルとして ElectroPop が挙げられた。分析結果は、強い歪みや極端なリズム構造を持つという聴感的印象とも整合している。

5.3.2 視覚表現

読み込まれ、分析された HyperPop 楽曲は、FutureBass の場合とは大きく異なる視覚表現として可視化された。ジャンル分析の結果に基づき、鋭く個性的な印象を持つフォントが適用され、文字全体に強い存在感が与えられている。文字の色彩は、energy および valence が高い値を示したことから、彩度の高い鮮やかな配色となった。加えて、本楽曲は intensity が極めて高いため、文字の色は赤となっている。加えて、曲で激しい音が鳴ったときは明度と彩度共に高くなり、HyperPop 特有の過剰さや刺激的な印象が視覚的に強調されている。文字の形状および動きには、歪みや揺れが加えられており、約 172 BPM の高速テンポに同期した細かくせわしない動作が確認できた。FutureBass のような周期的で安定した動きとは異なり、リズムのばらつきに合わせて文字が予測不能に変形する点が特徴的である。また、音量変化が大きくパーカッション成分の強い楽曲であることから、音圧の高い瞬間では文字の変形量が増加し、視覚的にも「暴れる」ような挙動を示した。これにより、明るさと激しさが同時に存在する HyperPop の音楽的特徴が、文字表現として明確に表れている。

第 6 章 考察

6.1 成果発表から得られたこと

成果発表を通して、表 6.1 の通り聴衆から「着眼点が良い」「これまで見たことのない表現で面白い」といった評価が寄せられた。これらのコメントから、音楽と文字を結び付ける表現自体の新規性が一定程度評価されたといえる。また、音楽に合わせて文字の色や大きさがリアルタイムに変化する視覚表現については、「曲を聴きながら見ていて楽しい」、「視覚と聴覚の両方から理解できて分かりやすい」といった意見があり、視覚情報が音楽の雰囲気把握に寄与していることが確認された。

表 6.1 最終成果発表によって得られた肯定的なフィードバック

肯定的な意見
面白い取り組みだと思った
Netflix で人の興味でサムネが変わるみたいなのを聞いたことがあるが、それに近い実際に取り入れられたらうざいかもかもしれないけど面白そうだなと思った
音楽の特徴に応じた文字変化なので、拡張性あって興味深かった
みたことないもので面白かった
着眼点がいいと思いました
とっても素晴らしいと思った。
とてもおもしろいものを見させて頂きました
レベルが高くてすごいと思います。
面白かったからもっといろんなジャンルのデモを見たい
文字が曲が進むによって色が変わるのが面白いです。文字のサイズが音量によって変わるのも聞いていて楽しい気持ちになりました
実際に聴覚と視覚で見ることができたので理解しやすかったです。
聴覚に訴えるものに視覚を増やしたのは面白かった
面白いインタラクションだった。今後世の中に出たらいいな
分析した分類がとても細かく分けられていて、しっかりとできていた。
実際に曲との連動を見た時に自分のイメージと近い物ですごいと思った。
1つの単語で曲を表現するようにするのが新しい発想でおもしろかったです。デモを見ることができたので理解がしやすくていい経験になりました。
音による歪みが面白かったです。

一方で、表 6.2 の通り、「ジャンルごとの音響特性と曲ごとの特性の関連が分かりにくい」という意見もあり、これは現在のシステムがジャンルごとにフォントを固定していることに起因すると考えられる。これにより、文字表現が楽曲固有の音響特徴よりもジャンルの印象に依存してしまっている可能性がある。また、音楽を開始から一定時間ごとに区切って分析している点について、「前後の区間の影響を考慮することで、より楽曲全体の印象を表現できるのではないか」という意見も

得られた。この指摘は、楽曲のメロディ構成やセクション境界を考慮せず分割を行っていることに起因していると考えられる。

表 6.2 最終成果発表によって得られた肯定的なフィードバック

肯定的な意見
曲の特徴を捉えて文字デザインが変わるのは斬新でした。しかし根本的な需要が分かりませんでした。画面を見ているならその曲の MV 等を見たいし、曲を選んでいる際はサビをちょっと聞く分で十分だと感じました。
音響特性はジャンルごと？曲ごと？（ジャンル内でも結構音響特性が違っているようだったが、音響特性は曲後に作成？それだと、ジャンルに分けた意味はどこにある？）
今後の話として、提案する文字体験の関係が分からなかった（別の新しい文字体験を提供するのか、文字体験の補強か、提案できなかった文字体験の提案なのか？）
曲の印象を文字表現に反映させることによって、どのようなことを達成したいのかよく分からなかった。
曲のジャンルの定義がよく分からず、ジャンルが曲に 1 対 1 に対応しない曲もあるように感じ、耳で聴く印象は違うにもかかわらず文字表現が似通ってしまうこともある気がした。
曲を時間で区切って文字表現に反映している、ということだったが、区切られた部分がその後に来る部分に少し影響すれば、よりその曲の印象を表していると言えると思った。

最終報告で得られたフィードバック結果（表 6.3）から、肯定的な意見が 74% を占めることが確認された。この高い評価率は、本研究で提案した音楽と文字を結び付ける表現手法が、聴衆に対して効果的に機能していることを示唆している。

表 6.3 最終成果発表によって得られたフィードバックの数と割合

	数	割合 (%)
肯定的な意見	17	74
批判懐疑的な意見	6	26

6.2 課題と改善点

6.1 の結果を踏まえ、本研究には以下の三点が主な課題として挙げられる：

1. 想定される使用場面の整理
2. 曲単体の特性のわかりにくさ
3. 分析時の音楽の区切り方

まず、「曲単体の特性のわかりにくさ」は、楽曲内部の変化や構造的特徴を可視化に反映し切れていない点に起因している。VJ 表現やインタラクティブ映像メディアの研究では、音楽を時間的に適切に分割し、その変化に応じて視覚情報を提示する手法が一般的に用いられている [7]。このような時間的分割を導入することで、楽曲各区間の特徴を抽出・可視化しやすくなると考えられる。

さらに、分析時の区切り方は、単に一定時間で分割する方法に加えて、拍構造やセクション境界など楽曲の音楽的構造に基づいた動的な分を検討する必要がある。これにより、楽曲の盛り上がりや転調といった構造的な変化を、より正確に文字表現へ反映できる可能性がある。

また、先行研究では、音楽の印象特徴と視覚情報（映像）を統合する方法が提案されており、音楽の感性や構造情報を視覚表現へ応用する試みが行われている [8]. これらの知見は、本研究における文字表現の改善にも活かせると考えられる.

6.3 今後の展望

本研究で開発したシステムは、普段の音楽体験に加え、ライブイベントにおける VJ（ビデオジョッキー）などの表現手法への応用可能性を示唆している. インタラクティブ映像メディア表現に関する先行研究では、音楽と映像をリアルタイムに連動させる表現が、観客の没入感やライブ体験の一体感を高める重要な要素であることが報告されている [7]. このような背景から、音楽の特徴を直感的に把握できる視覚情報は、表現選択の支援として有効であると考えられる.

また、音楽と視覚情報を統合する研究では、音楽の印象特徴に基づいて視覚的表現を生成する手法が提案されており、音楽鑑賞体験の拡張に寄与している [8]. これらの研究は、音楽と映像を感覚的に結び付けることの重要性を示しており、本研究が目指す「音楽の印象を文字表現として可視化する」という方向性と一致している.

したがって、今後の研究では、楽曲の時間構造やセクション境界に基づいた分析方法の導入、そして視覚的表現との連携により、文字可視化表現の表現力と汎用性を向上させることが求められる.

参考文献

- [1] 加藤 潤, 中野 智康, 後藤 真孝, “TextAlive Online: Live Programming of Kinetic Typography Videos with Online Music”, Proceedings of the First International Conference on Live Coding, 2015.
- [2] 新編 色彩科学ハンドブック, 第3版, オーム社.
- [3] 李 志炯, 「実験心理学的手法による文字の太さと印象の関係の検討ーひらがなとカタカナを中心にー」, 学位論文, 千葉大学大学院工学研究科, 2017.
- [4] 松原 正樹, 「音楽情報可視化による多面的視点の獲得」, 可視化情報学会誌, 2015.
https://www.jstage.jst.go.jp/article/jvs/35/136/35_8/_article/-char/ja/
(参照日 2025 年 12 月 15 日)
- [5] 柳瀬 徹夫, 「色のイメージ (色彩感情)」, 可視化情報学会誌, Vol. 17, No. 64, pp. 18–22, 1997.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, MIT Press, USA, 1998.
- [7] 本村 健太, 「インタラクティブ映像メディア表現の考察」, 電子情報通信学会論文誌 A, Vol. 95, No. 7, pp. 595–599, 2012.
- [8] 深井 遥, 入地 浅子, 清木 康, “Semantic and emotional music & image integration with impression-based correlation computing”, Proceedings of the IASTED International Conference on Advances in Computer Science and Engineering (ACSE 2009), pp. 7–14, 2009.

付録 A 音楽分析で使したパラメータ

本研究では、Librosa を用いて抽出した低レベル音響特徴量から、楽曲のムードや激しさを示す高レベル指標 (Valence, Energy, Intensity など) を計算し、可視化に反映する。本章では、基礎音響特徴量を先に定義したうえで、正規化と派生指標の計算式を示し、最後に使したパラメータの完全一覧を記載する。

A.1 基礎音響特徴量 (Librosa によるダイレクト抽出)

以下の指標は Librosa ライブラリから直接抽出される低レベル音響特徴量である。

A.1.1 テンポ・ビート関連

- **Tempo [BPM]** : オンセット強度関数から自己相関法により推定される拍数. 50~300 BPM の範囲で、ハーフテンポ (半分の BPM) をダブルテンポで補正する。
- **Onset Mean** : オンセット検出関数 $E(n)$ の時間平均. ビート明瞭度の指標。
- **Onset Std** : オンセット関数の標準偏差. リズムの不規則さを示す。

A.1.2 エネルギー・ダイナミクス

- **RMS Mean** : 短時間フーリエ変換 (STFT) によるフレーム単位 RMS 値の時間平均. 全体的な音量レベル。
- **RMS Std** : RMS 値の標準偏差. ダイナミクスの広さを示す。
- **HPSS Ratio** : ハーモニック・パーカッシブ音源分離 (HPSS) 後の打楽器 (パーカッシブ) 成分と調和成分の RMS 比。

A.1.3 周波数・スペクトル特性

- **Spectral Centroid [Hz]** : スペクトル重心周波数. 音色の「明るさ」を示す指標。
- **Spectral Rolloff [Hz]** : スペクトルエネルギーの 85% が集中する周波数. 高周波成分の多さを示す。
- **Spectral Bandwidth [Hz]** : スペクトルセントロイドの周囲のエネルギー分散。
- **Spectral Contrast [dB]** : 各周波数帯域 (デフォルト 7 帯域) のピークとバレーの対比. 複雑さや質感を示す. 各帯域の値を時間平均し、全帯域の平均を採用。
- **Zero Crossing Rate (ZCR)** : 波形が 0 を横切る回数の比率. 音声やノイズの有無を示す。

A.1.4 メル周波数ケプストラル係数 (MFCC)

- **MFCC Mean [13-dim]** : 13 次元の MFCC 特徴量を時間軸で平均したもの. 音色の平均的な質を表す.
- **MFCC Std [13-dim]** : MFCC の時間軸標準偏差. 音色の時間的変動度合い.
- **Timbre Brightness** : MFCC の第 1~3 係数から推定される音色の明るさ.
- **Timbre Complexity** : MFCC 全体の標準偏差の平均. 音色の時間的揺らぎ・複雑さ.

A.1.5 調性・ハーモニー分析 (Chroma)

- **Chroma [12-dim]** : 12 音階 (C, C#, D, ..., B) 各音のエネルギーを表す 12 次元ベクトル. 時間平均値を採用.
- **Key** : Chroma の最大エネルギー音を調として判定. C~B の 12 種類.
- **Mode** : Major/Minor の長短調判定. 各調の長 3 度・短 3 度を含む三和音スコアを比較し, 10% 以上の差で判定.
- **Key Strength** : 最強 Chroma コンポーネントと全体の比率. 調の確信度.
- **Key Stability** : Chroma 特徴量の時間変動度合い. 値が大きければ調が安定.

A.2 正規化と記号

派生指標の計算で頻繁に使用する正規化関数と記号を統一的に定義する.

$$\begin{aligned}
 N_{\text{tempo}} &= \min\left(\frac{\text{Tempo}}{200}, 1.0\right), \\
 N_{\text{centroid}}^{(8k)} &= \min\left(\frac{\text{SpectralCentroid}}{8000}, 1.0\right), \\
 N_{\text{centroid}}^{(5k)} &= \min\left(\frac{\text{SpectralCentroid}}{5000}, 1.0\right), \\
 N_{\text{centroid}}^{(3k)} &= \min\left(\frac{\text{SpectralCentroid}}{3000}, 1.0\right), \\
 N_{\text{rolloff}} &= \min\left(\frac{\text{SpectralRolloff}}{6000}, 1.0\right), \\
 N_{\text{onset}}(a) &= \min\left(\frac{\text{OnsetMean}}{a}, 1.0\right), \\
 N_{\text{zcr}}(b) &= \min\left(\frac{\text{ZCR}}{b}, 1.0\right), \\
 N_{\text{loud}} &= \min(20 \cdot \text{RMSMean}, 1.0), \\
 N_{\text{dyn}} &= \min(10 \cdot \text{RMSStd}, 1.0), \\
 \text{HPSS}_{\text{perc}} &= \min\left(\frac{\text{HPSS Ratio}}{5.0}, 1.0\right), \\
 \text{ModeBoost} &= \begin{cases} +0.15 & (\text{Mode} = \text{Major}) \\ -0.15 & (\text{Mode} = \text{Minor}) \end{cases}
 \end{aligned}$$

A.3 派生指標の計算式

以下の指標は基礎特徴量から計算される高レベル指標である。すべて $[0.0, 1.0]$ の範囲に正規化される（ただし極端な曲は 1.0 を超える場合もある）。

A.3.1 Acousticness (アコースティック度)

打楽器が少なく、スペクトルが暗めで、テンポが極端に速くないほど高くなる。

$$\text{Acousticness} = (1 - 0.5 \cdot \text{HPSS}_{\text{perc}}) \cdot (1 - N_{\text{centroid}}^{(8k)}) \cdot (1 - 0.3 \cdot N_{\text{tempo}})$$

A.3.2 Danceability (踊りやすさ)

理想的なダンステンポ 110~130 BPM へのマッチング度とオンセット強度を平均。

$$\text{Danceability} = \frac{T_{\text{score}} + N_{\text{onset}}(2.0)}{2}$$

$$T_{\text{score}} = \begin{cases} \frac{\text{Tempo}}{110} & (\text{Tempo} < 110) \\ 1.0 & (110 \leq \text{Tempo} \leq 130) \\ \frac{200 - \text{Tempo}}{70} & (\text{Tempo} > 130) \end{cases}$$

下限 0 でクリップ。

A.3.3 Instrumentalness (インスト度)

ボーカルの有無を推定。ハーモニック比率とボーカル帯域（2000 Hz 前後）からの周波数ズレで判定。

$$\text{Instrumentalness} = 0.4 \cdot h_{\text{ratio}} + 0.6 \cdot f_{\text{deviation}}$$

ここで $h_{\text{ratio}} = \min(\text{RMS}_{\text{harm}}/(\text{RMS}_{\text{mean}} + \epsilon), 1.0)$ で、 $f_{\text{deviation}} = 1.0$ if $|\text{SpectralCentroid} - 2000| > 1000$ else $|\text{SpectralCentroid} - 2000|/1000$ 。

A.3.4 Liveness (ライブ感)

ライブ録音特有の背景ノイズ、高周波ノイズ (ZCR)、ダイナミクスの広さを平均。

$$\text{Liveness} = \frac{\text{Ratio}_{\text{bg}} + N_{\text{zcr}}(0.1) + N_{\text{dyn}}}{3}$$

ここで $\text{Ratio}_{\text{bg}} = \min(\text{RMS}_{10\%}/(\text{RMS}_{\text{mean}} + \epsilon) \times 10, 1.0)$ は RMS 下位 10% の背景ノイズ比。

A.3.5 Speechiness (話し声度)

摩擦音や子音の多さ (ZCR) と発音の区切れ (Onset) を平均。

$$\text{Speechiness} = \frac{N_{\text{zcr}}(0.03) + N_{\text{onset}}(5.0)}{2}$$

A.3.6 Valence (明るさ・ポジティブ度)

テンポ, 周波数領域特性, 調性, 音色の明るさを加重平均.

$$\text{Valence} = 0.2 + 0.3 \cdot N_{\text{tempo}} + 0.3 \cdot N_{\text{centroid}}^{(5k)} + 0.2 \cdot N_{\text{rolloff}} + \text{ModeBoost} + 0.1 \cdot \text{Brightness}$$

A.3.7 Energy (エネルギー)

音量, ダイナミクス, アタック強度, 周波数スペクトルの鋭さを均等に平均.

$$\text{Energy} = \frac{N_{\text{loud}} + N_{\text{dyn}} + N_{\text{onset}}(5.0) + N_{\text{centroid}}^{(3k)}}{4}$$

A.3.8 Intensity (激しさ・攻撃性)

高速テンポ, 手数 (オンセット密度), 不規則さ, 極端なダイナミクス, 高周波の鋭さ, 打楽器優位, スペクトル複雑さを加重合成. EDM のような規則的なビートが強い場合は若干減衰.

テンポ由来スコア:

$$T_{\text{int}} = \begin{cases} 0 & (\text{Tempo} < 140) \\ \frac{\text{Tempo}-140}{30} & (140 \leq \text{Tempo} < 170) \\ \min(1.0 + \frac{\text{Tempo}-170}{80}, 2.0) & (\text{Tempo} \geq 170) \end{cases}$$

統合スコア:

$$\begin{aligned} I &= 0.22 \cdot T_{\text{int}} + 0.18 \cdot \min\left(\frac{\text{OnsetMean}}{3.0}, 1.5\right) + 0.20 \cdot \min\left(\frac{\text{OnsetStd}}{2.0}, 1.0\right) \\ &+ 0.12 \cdot \min(15 \cdot \text{RMSStd}, 1.5) + 0.18 \cdot \min\left(\frac{\text{SpectralCentroid}}{4000}, 1.5\right) \\ &+ 0.05 \cdot \min(2.0 \cdot \text{HPSS_Ratio}, 1.5) + 0.05 \cdot \min\left(\frac{\text{SpectralContrast}}{15}, 1.0\right) \end{aligned}$$

EDM ペナルティ:

$$\text{Penalty}_{\text{EDM}} = 0.25 \cdot \text{EDM_Signature} \cdot \left(1 - \min\left(\frac{\text{OnsetStd}}{2.0}, 1.0\right)\right)$$

最終出力:

$$\text{Intensity} = \text{clip}_{[0,1]}(I \cdot (1 - \text{Penalty}_{\text{EDM}}))$$

A.3.9 EDM Signature (EDM らしさ)

ビートの規則性 (自己相関スコア) とサイドチェイン・ポンピング (RMS の BPM 周波数成分) を 6:4 で合成. リズムの不規則さ (Chaos) で若干減衰.

$$\text{EDM_Sig} = \min(1.0, 0.6 \cdot B_{\text{strength}} + 0.4 \cdot P_{\text{ratio}}) \cdot \left(1 - 0.3 \cdot \min\left(\frac{\text{OnsetStd}}{2.0}, 1.0\right)\right)$$

A.3.10 Kawaii Score (かわいいスコア)

MFCC 平均ベクトルと共分散行列を用いて、ジャンル分析同様に事前に構築された「かわいい楽曲」データベース内の最近傍距離から計算。

A.4 使用パラメータの完全一覧

A.4.1 基礎音響特徴量 (直接抽出)

ダイナミクス・リズム関連

- Tempo [BPM] (推奨範囲: 50~300, ハーフテンポ補正あり)
- RMS Mean [linear scale] (全フレームの平均)
- RMS Std [linear scale] (全フレームの標準偏差)
- Onset Mean [a.u.] (オンセット検出関数の平均)
- Onset Std [a.u.] (オンセット検出関数の標準偏差)
- Zero Crossing Rate (ZCR) (0~1, 単位なし)
- HPSS Ratio (パーカッション成分 RMS / ハーモニック成分 RMS, ≥ 0)

周波数・スペクトル特性

- Spectral Centroid [Hz] (スペクトル重心, $0 \sim sr/2$)
- Spectral Rolloff [Hz] (85% エネルギー周波数, $0 \sim sr/2$)
- Spectral Bandwidth [Hz] (周波数分散, ≥ 0)
- Spectral Contrast [dB] (7 帯域, 各帯域の平均値, ≥ 0)
- Spectral Contrast Mean [dB] (7 帯域の全体平均)

音色・ケプストラル特性

- MFCC Mean [0~12] (13 個の係数, 各々実数値)
- MFCC Std [0~12] (13 個の係数, 各々 ≥ 0)
- MFCC Covariance [13 × 13] (共分散行列, Kawaii スコア計算用)
- Timbre Brightness [0, 1] (MFCC 第 1 係数から導出)
- Timbre Complexity [0, 1] (MFCC 全体の標準偏差から導出)

調性・ハーモニー分析

- Chroma [12-dim] (12 音階のエネルギー, 各々 ≥ 0)
- Key (C, C#, D, D#, E, F, F#, G, G#, A, A#, B のいずれか)
- Mode (Major または Minor)
- Key Strength [0, 1] (最強 Chroma 成分の比率)
- Mode Confidence [0, 1] (Major/Minor スコアの相対比)
- Key Stability [0, 1] (調の安定性)

A.4.2 派生指標（計算値）

Spotify 型指標（7つ）

- Acousticness [0, 1]
- Danceability [0, 1]
- Instrumentalness [0, 1]
- Liveness [0, 1]
- Speechiness [0, 1]
- Valence [0, 1]
- Energy [0, 1]（場合によっては 1.0 超過）

独自指標

- Intensity [0, 1]（激しさ・攻撃性，場合によっては 1.0 超過）
- EDM Signature [0, 1]（ビート規則性とポンピング，場合によっては 1.0 超過）
- Kawaii [0, 1]（かわいいスコア）

A.4.3 内部計算値（中間パラメータ）

実装では以下も計算・保存される場合がある：

- Harmonic RMS（HPSS 後のハーモニック成分 RMS）
- Percussive RMS（HPSS 後のパーカッシブ成分 RMS）
- ACF (Autocorrelation Function)（テンポ推定・EDM 検出用）
- Beat Strength（自己相関による 4 つ打ちビートの強さ）
- Pump Ratio（RMS エンベロープの BPM 周波数成分比）
- Tempo Score Candidates（ハーフ/ダブルテンポ候補の ACF スコア）

A.5 パラメータ抽出のアルゴリズム概要

A.5.1 前処理

1. オーディオファイルを pydub で読み込み（MP3, WAV, OGG 等の拡張子自動判定）
2. float32 正規化（-1.0 ~ +1.0 範囲）
3. 必要に応じてモノラルに変換（ステレオの場合は平均）
4. Librosa 標準サンプリングレート 22050 Hz にリサンプリング

A.5.2 特徴抽出の順序

1. HPSS で信号をハーモニック/パーカッシブに分離
2. RMS, ZCR, スペクトル特性（Centroid, Rolloff, Bandwidth, Contrast）を抽出
3. Onset Strength エンベロープを計算

Designing Expanded Experiences of Written Language

4. Onset Strength と自己相関から Tempo を推定（ハーフテンポ補正付き）
5. Chroma 特徴量を抽出し，Key・Mode・Key Strength を判定
6. MFCC を抽出し，MFCC Mean/Std, Brightness, Complexity を計算
7. MFCC 共分散行列を計算（Kawaii スコア用）
8. 派生指標（Acousticness, Danceability, . . . , Kawaii）を順に計算

付録 B システムで利用したフォント一覧

本章では、表 B.1 にて実際に使用したフォントの対応関係およびダウンロード元を示す。

表 B.1 システムで実際に利用したフォント

ジャンル名	サブジャンル名	フォント名	ダウンロード元
Classic		こころ明朝体	https://fontopo.com/?p=520
EDM	DubStep	DubStep Heroes Font	https://www.fontspace.com/dubstep-heroes-font-f28851
EDM	ElectroHouse	Electro Shackle Font	https://www.fontspace.com/electro-shackle-font-f25374
EDM	FutureBass	213chan-nonbit	https://213chan.booth.pm/items/6007632
EDM	House	House Music Font	https://www.fontspace.com/house-music-font-f22014
EDM	Trap	Freedom Font	https://www.fontspace.com/freedom-font-f14832
Enka		玉ねぎ楷書「激」	https://booth.pm/ja/items/2929647
HipHop	Drill	Street Toxic Font	https://www.1001fonts.com/street-toxic-font.html
HipHop	EmoRap	Graffogie	https://www.dafont.com/graffogie.font
HipHop	OldSchool	FAST BLAZE Font	https://www.1001fonts.com/fast-blaze-font.html
Jazz		Theatrical Font	https://www.1001fonts.com/theatrical-font.html
Pops	ElectroPop	Digital Desolation Plus	https://ukiyomoji.booth.pm/items/52655
Pops	HyperPop	叛逆明朝	https://www.fontspace.com/hangyaku-font-f31195
Pops	J-Pop	ラノベポップ	https://www.fontna.com/blog/lightnovel.html
Pops	Pop	Nikkyou Sans Font	https://www.fontspace.com/nikkyou-sans-font-f31053
R & B		Stupid Hoe Font	https://fontmeme.com/fonts/trash-hand-font/
Rock	Alternative Rock	FZ イモケンピ	https://zone108.main.jp/s/jp-004imokenpi.php
Rock	Classic Rock	ClassicRock	https://mikannnoki-font.booth.pm/items/164890
Rock	Metal	Brutal Tooth	https://www.dafont.com/brutal-tooth.font
Rock	Punk	PUNK KID	https://www.1001fonts.com/punk-kid-font.html

付録 C 音楽分析の出力結果

—内容—

C.1 FutureBass

```
{
  "input": "C:\\Users\\senas\\Music\\Pixel Galaxy.wav",
  "features": {
    "tempo": 151.99908447265626,
    "hpss_ratio": 0.4009886085987091,
    "rms_mean": 0.23760288953781129,
    "rms_std": 0.126602441072464,
    "spectral_centroid_mean": 3246.95068359375,
    "spectral_rolloff_mean": 6926.83251953125,
    "spectral_bandwidth_mean": 2945.4658203125,
    "spectral_contrast_mean": 21.962621688842775,
    "onset_mean": 1.4175939559936524,
    "onset_std": 1.9030723571777344,
    "zcr_mean": 0.15356555581092835,
    "key": "E",
    "mode": "minor",
    "key_strength": 0.11424659937620163,
    "key_stability": 0.0,
    "mode_confidence": 0.5149572491645813,
    "mfcc_mean": [
      -56.16892623901367,
      47.63121795654297,
      3.286742925643921,
      16.155216217041017,
      2.4417195320129396,
      12.445194244384766,
      4.422144412994385,
      7.573495864868164,
      1.6607940196990967,
      4.812981605529785,
      1.5144915580749512,
```

Designing Expanded Experiences of Written Language

```
        6.2953104972839359 ,
        3.5544581413269045
    ],
    "mfcc_std": [
        130.1034698486328 ,
        22.020721435546876 ,
        16.543689727783204 ,
        13.458980560302735 ,
        9.737207412719727 ,
        9.319643020629883 ,
        9.543845176696778 ,
        8.371296882629395 ,
        8.614139556884766 ,
        8.424400329589844 ,
        7.990137577056885 ,
        8.226887702941895 ,
        9.261144638061524
    ],
    "timbre_brightness": 1.0 ,
    "timbre_complexity": 1.0 ,
    "acousticness": 0.44027793407440188 ,
    "danceability": 0.697262167930603 ,
    "instrumentalness": 0.9246031641960144 ,
    "liveness": 0.8681507110595703 ,
    "speechiness": 0.6417593955993652 ,
    "valence": 0.7728156447410584 ,
    "energy": 0.8208796977996826 ,
    "intensity": 0.7748234868049622 ,
    "edm_signature": 0.5023044943809509 ,
    "kawaii": 0.19321449100971223
},
"genre": {
    "genre_id": "EDM/FutureBass" ,
    "genre": "EDM/FutureBass" ,
    "neighbors": [
        {
            "genre": "EDM/FutureBass" ,
            "count": 2
        }
    ]
},
```

```
{
  "genre": "Pops/HyperPop",
  "count": 2
},
{
  "genre": "EDM/ElectroHouse",
  "count": 1
}
]
}
}
UnityEngine.Debug:Log (object)
MusicController/<AnalyzeSelectedTrackAsync>d_40:MoveNext () (at Assets/Scripts/M
System.Runtime.CompilerServices.AsyncTaskMethodBuilder`1<PythonMusicAnalyzer/Analy
PythonMusicAnalyzer/<AnalyzeAsync>d_8:MoveNext () (at Assets/Scripts/PythonMusic
UnityEngine.UnitySynchronizationContext:ExecuteTasks ()
```

C.2 HyperPop

```
{
  "input": "C:\\Users\\senas\\Music\\lost woods!.wav",
  "features": {
    "tempo": 172.265625,
    "hpss_ratio": 0.9461076259613037,
    "rms_mean": 0.32774117588996889,
    "rms_std": 0.22540058195590974,
    "spectral_centroid_mean": 3371.16845703125,
    "spectral_rolloff_mean": 7077.22216796875,
    "spectral_bandwidth_mean": 2960.7685546875,
    "spectral_contrast_mean": 21.72615623474121,
    "onset_mean": 1.7557158470153809,
    "onset_std": 2.2848143577575685,
    "zcr_mean": 0.1728711575269699,
    "key": "E",
    "mode": "minor",
    "key_strength": 0.11222778260707855,
    "key_stability": 0.0,
    "mode_confidence": 0.48223480582237246,
    "mfcc_mean": [
```

```
    -40.263301849365237,  
    45.053592681884769,  
    11.551462173461914,  
    16.104660034179689,  
    -4.379321575164795,  
    0.041356224566698077,  
    0.39658617973327639,  
    -2.4178476333618166,  
    -1.994977593421936,  
    7.179126262664795,  
    -2.9708547592163088,  
    3.554302215576172,  
    -1.8836995363235474  
  ],  
  "mfcc_std": [  
    108.88218688964844,  
    37.7170524597168,  
    27.431333541870118,  
    24.01548194885254,  
    17.7528018951416,  
    17.19170379638672,  
    18.472217559814454,  
    14.604540824890137,  
    14.538694381713868,  
    11.261651992797852,  
    10.777191162109375,  
    9.849833488464356,  
    10.311758041381836  
  ],  
  "timbre_brightness": 1.0,  
  "timbre_complexity": 1.0,  
  "acousticness": 0.3884967267513275,  
  "danceability": 0.6370316743850708,  
  "instrumentalness": 0.8263858556747437,  
  "liveness": 1.0,  
  "speechiness": 0.6755715608596802,  
  "valence": 0.8106685280799866,  
  "energy": 0.8377857804298401,  
  "intensity": 0.9882760047912598,
```

Designing Expanded Experiences of Written Language

```
    "edm_signature": 0.46939554810523989,  
    "kawaii": 0.30378204584121706  
  },  
  "genre": {  
    "genre_id": "Pops/HyperPop",  
    "genre": "Pops/HyperPop",  
    "neighbors": [  
      {  
        "genre": "Pops/HyperPop",  
        "count": 4  
      },  
      {  
        "genre": "Pops/ElectroPop",  
        "count": 1  
      }  
    ]  
  }  
}
```