TU WIEN Informatics

Master Thesis Proposal

Song-Adaptive Music Visualization with Large Language Models

From Structured Audio Analysis to
Real-Time Shader Control

Sebastian Antes

Student ID: 11905187
Degree Programme: 066 932 Visual Computing
Institute: E193 - Institute of Visual Computing and Human-Centered Technology

May 4, 2026

Advisor: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer

Co-Supervising Assistant: Projektass. Mag.rer.soc.oec. Stefan Ohrhallinger, PhD

Motivation & Problem Statement

Many existing music visualizations react mainly to low-level audio features such as volume, frequency bands, or beats. While this can create dynamic visuals, the result often remains generic and does not reflect the larger musical structure of a song, such as verses, choruses, instrumental parts, or changes in mood and instrumentation.

This thesis investigates whether Large Language Models (LLM) can help create music visualizations that better adapt to the individual piece of music. Therefore, the LLM receives structured audio analysis, including rhythmic, melodic, timbral and section-level information, and uses it to generate a configuration for a real-time WebGPU renderer. The renderer then maps precomputed, time-synchronized audio features to shader parameters during playback.

The central challenge lies in combining the semantic flexibility of LLMs with the precision and reliability required for real-time graphics. This thesis investigates a hybrid approach in which the LLM creates section-aware visualization mappings offline, while the final visual output is generated deterministically by the rendering engine in real-time.

Expected Results

This thesis explores how structured audio analysis and LLMs can be combined to generate song-adaptive configurations for real-time music visualization. In particular, it studies how rhythmic, melodic, timbral, and structural descriptors can be mapped to deterministic shader parameters in a WebGPU-based rendering system. The expected results include:

1. An offline audio analysis pipeline that extracts relevant musical descriptors, including stem-based features, MIDI or pitch-related information, and section-level song structure.

2. A structured intermediate representation of the analyzed song that can be used as input for LLM-based reasoning and configuration generation.

3. A multi-stage LLM prompting pipeline that generates section-aware visualization configurations, including visual layer selection and mappings from audio features to shader uniforms.

4. A real-time WebGPU render engine that applies the generated configurations during playback by mapping precomputed audio features to shader parameters in a deterministic and time-synchronized manner.

5. An evaluation of the generated visualizations in terms of perceived audio-visual alignment, section awareness, aesthetic coherence, timing accuracy, and real-time rendering performance.

The final implementation should demonstrate that LLMs can support the creative mapping between musical structure and visual behavior, while the actual rendering remains deterministic, reproducible, and suitable for real-time execution.

Methodology

The proposed framework addresses the constraints of real-time AI graphics by decoupling semantic orchestration from the rendering loop. The architecture is structured into a three-stage hybrid pipeline: an offline Python-based audio analysis backend, an offline multi-stage LLM configuration generator, and a real-time WebGPU rendering frontend.

Audio Analysis

After audio upload, the FastAPI backend processes the source file through a series of Music Information Retrieval (MIR) techniques. The raw audio is first isolated into discrete instrument stems (drums, bass, vocals, other) using Demucs [RMD23]. The system then analyzes the audio both globally and locally. The global analysis calculates macro-level features, including the global tempo in beats per minute (BPM) and tonal key and mode estimation. The local analysis extracts instantaneous rhythmic, melodic, and timbral attributes, such as spectral bandwidth and derived MIDI pitch tracking, for each individual instrument stem. A subsequent structural analysis segments the composition into temporal sections (e.g. intro, verse, chorus). The system aggregates the localized low-level features into high-level section descriptors, summarizing properties like rhythmic complexity and harmonic stability.

Multi-Stage LLM Configuration Generation

To convert raw quantitative data into deterministic rendering parameters, the summarized section descriptors are fed into a modular, multi-stage LLM prompting pipeline. This approach uses suitable models that match the cognitive requirements of each sub-task:

- **Stage 1 (Aesthetic Interpretation):** Translates raw structural metrics and MIDI data into a free-form creative prose description of the sonic progression of the song, unconstrained by syntax formatting.
- **Stage 2 (Direction Planning):** Reviews the artistic description to isolate the most relevant musical features per section and selects compatible visual layout goals. High-level descriptions of the available visual effects are provided with the prompt to guide the decision making.
- **Stage 3 (Structured Compilation):** Transcribes the high-level direction plan into a strictly validated `configuration.json` file that complies with a predefined machine-readable schema. To enforce runtime safety and valid JSON syntax, the output is parsed using Pydantic schema validation [Pyd26]. If parsing fails due

to syntax errors or missing fields, the system executes an automated programmatic retry loop to regenerate and correct the configuration before final handoff. To mitigate context window bloating and optimize token efficiency, this stage implements a dynamic prompt chaining strategy. Instead of exposing the LLM to the entire shader catalog manifest, the pipeline contextually injects detailed specifications only for the specific visual assets selected in Stage 2. This injected metadata contains definitions of each shader's uniform parameters, alongside semantic heuristics defining optimal cross-modal pairings (e.g., matching high-frequency timbral features with rapid visual noise scales). The final generated JSON explicitly maps precomputed time-series audio features to target graphics uniforms, defining deterministic range normalization bounds and temporal smoothing coefficients for real-time execution.

Deterministic WebGPU Real-Time Rendering

The frontend application executes the precompiled visualization configuration in sync with the audio playback clock. During initialization, the engine preloads the referenced shader catalog and features. This catalog acts as a shared backend/frontend contract using JSON manifests that define a strict uniform interface, allowing the engine to be runtime-polymorphic across modular WGSL fullscreen fragment shaders, instanced note paths, and compute-driven particle pipelines. To guarantee stable runtime performance and avoid dropping frame rates during section transitions, graphics pipelines are prewarmed on the GPU. Transitions between track sections alter uniform mappings and layer opacities rather than forcing runtime shader recompilation. On each frame tick, the engine samples the precomputed audio features at the current playback timestamp, applies the JSON-specified smoothing filters, and binds the values directly to the GPU uniforms, targeting fluid, high-frame-rate visual execution. Specifically, a specialized UniformBridge subsystem parses the active configuration, normalizes and smooths the time-series features per frame, and scatters the values into a predefined uniform namespace via a fixed slot map.

Evaluation

The system will be evaluated using a mixed-methods approach. Qualitative feedback from users will be collected to evaluate perceived audio-visual alignment, aesthetic coherence, and whether section changes are visually meaningful. In addition, quantitative measurements such as rendering performance and correspondence between musical events and visual parameter changes will be analyzed. Where possible, the LLM-generated configurations will be compared against simpler baseline visualizers based on generic audio-reactive mappings. To make this comparison methodologically robust, the study will use standardized full-length songs and a within-subject design in which participants rate both baseline and section-aware outputs. In line with established music-visualization evaluation practice, perceived affective alignment will be measured with valence-arousal ratings (e.g., Affective Slider), complemented by Likert-scale ratings for alignment, transition quality, and overall experience. On the systems side, frame-time stability, dropped-frame ratio,

and transition-window performance will be logged to verify that prewarmed pipelines and runtime uniform remapping maintain real-time behavior without recompilation spikes. Together, these measures assess both perceptual quality and technical reliability of the proposed framework.

Related Work

Traditional music visualization frameworks predominantly rely on mapping low-level acoustic or MIDI features directly to real-time graphics variables [LSM22, GOB21]. While these direct mapping frameworks successfully generate fluid, audio-reactive motion, they are fundamentally context-blind, reacting only to instantaneous signal changes rather than reflecting the overarching musical narrative, arrangement shifts, or structural intent of a composition. Incorporating larger musical forms can resolve this limitation, though music structure analysis presents its own complexities due to its inherently subjective, hierarchical, and multi-layered nature [NMW+20]. To address this gap, early systems explored semantic enrichment via multimodal generative modeling [LZF+21], motivating the need for an abstract orchestration layer that can translate high-level structural analysis into cohesive, section-aware visual mappings.

Concurrently, broader artificial intelligence research establishes a clear design pattern where LLMs operate as orchestrators that decouple high-level semantic planning from deterministic execution environments [SST+23, KMT+23]. Rather than directly synthesizing end-artifacts, these systems leverage LLMs to generate intermediate structured configuration schemas [GCM+25] or executable procedural scripts [SHD+24, HIJ+24, ZWH+24]. This paradigm has been successfully applied to control complex 3D scenes, procedural assets, and graphics-executable code libraries.

In the domain of music visualization, recent frameworks use LLMs to continuously synthesize or refine raw graphics code on the fly [YS25], or utilize high-level Music Information Retrieval descriptors to guide frame-by-frame generative diffusion pipelines [HWR25]. While these approaches successfully deliver context-aware creative graphics, executing on-the-fly shader compilation [YS25] or real-time frame diffusion [HWR25] introduces significant computational overhead, rendering constraints, and unpredictability.

In contrast to these existing paradigms, the framework proposed in this thesis does not use LLMs to continuously generate pixels or mutate shader source code at runtime. Instead, it connects these separate domains by utilizing a multi-stage offline LLM pipeline to produce constrained, section-aware schema configurations, leaving the real-time execution entirely to a deterministic, high-performance WebGPU rendering engine.

Bibliography

[GCM+25] Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models, February 2025. arXiv:2501.10868 [cs].

[GOB21] Max Graf, Harold Chijioke Opara, and Mathieu Barthet. An Audio-Driven System For Real-Time Music Visualisation, June 2021. arXiv:2106.10134 [cs].

[HIJ+24] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM Agent for Synthesizing 3D Scene as Blender Code, March 2024. arXiv:2403.01248 [cs].

[HWR25] Jenny Huang, Christoph Johannes Weber, and Sylvia Rothe. An AI-driven Music Visualization System for Generating Meaningful Audio-Responsive Visuals in Real-Time. In *Proceedings of the 2025 ACM International Conference on Interactive Media Experiences*, pages 258–274, Niterói Brazil, June 2025. ACM.

[KMT+23] Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. An LLM Compiler for Parallel Function Calling, 2023. Version Number: 3.

[LSM22] Hugo B. Lima, Carlos G. R. Dos Santos, and Bianchi S. Meiguins. A Survey of Music Visualization Techniques. *ACM Computing Surveys*, 54(7):1–29, September 2022.

[LZF+21] Yufan Li, Jinggang Zhuo, Ling Fan, Zhe Wang, and Harry Jiannan Wang. Semantically Enriched Music Visualization via Multimodal Color Generation. In *NIME 2021*, Shanghai, China, June 2021. PubPub.

[NMW+20] Oriol Nieto, Gautham J. Mysore, Cheng-i Wang, Jordan B. L. Smith, Jan Schlüter, Thomas Grill, and Brian McFee. Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications. *Transactions of the International Society for Music Information Retrieval*, 3(1):246–263, December 2020.

[Pyd26] Pydantic Team. Pydantic, 2026. version: 2.13.3.

[RMD23] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *Icassp 23*, 2023.

[SHD+24] Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3D-GPT: Procedural 3D Modeling with Large Language Models, May 2024. arXiv:2310.12945 [cs].

[SST+23] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, December 2023. arXiv:2303.17580 [cs].

[YS25] Kamer Ali Yuksel and Hassan Sawaf. AI Co-Artist: A LLM-Powered Framework for Interactive GLSL Shader Animation Evolution, 2025. Version Number: 1.

[ZWH+24] Mengqi Zhou, Yuxi Wang, Jun Hou, Shougao Zhang, Yiwei Li, Chuanchen Luo, Junran Peng, and Zhaoxiang Zhang. SceneX: Procedural Controllable Large-scale Scene Generation, December 2024. arXiv:2403.15698 [cs].