A REAL-TIME PROCEDURAL MUSIC GENERATION SYSTEM BASED ON AFFECTIVE CONTENT OF LOW-LEVEL VIDEO FEATURES

Filippo GIABELLI (filippo.giabelli@studenti.unimi.it) [1],
Federico AVANZINI (federico.avanzini@unimi.it) (0000-0002-1257-5878) [1], and
Giorgio PRESTI (giorgio.presti@unimi.it) (0000-0001-7643-9915) [1]

[1] *Lab. of Music Informatics (LIM), Dept. of Computer Science*, **University of Milan**, via G. Celoria 18, Milan, 20133 Italy

ABSTRACT

This paper presents a proof of concept for a software architecture designed to generate real-time procedural music based on the affective content of arbitrary video streams, utilizing low-level video features. We achieved this objective by implementing a demonstrative application developed entirely within the Unreal Engine 5.4 environment. The system utilizes MetaSounds for music generation and the Pleasure-Arousal-Dominance (PAD) model to map affective values from imagery to music. Furthermore, it employs asynchronous computation to optimize processing load, ensuring suitability for real-time applications. The work has been validated through performance measurements and empirical testing.

1. INTRODUCTION

In multimedia, both sound and imagery are vital means of communicating emotion, and artists often seek emotional coherence between these two domains. However, in the context of real-time digital multimedia, the unpredictability of on-screen content presents an additional challenge for composers. They often overlook fine-grained visual events, focusing instead on the general mood of the digital environment.

While many studies have explored the affective analysis of music and images independently (even proposing models to convey specific emotions via these media) fewer have attempted to emotionally associate visuals and sound within real-time interactions.

The purpose of this work is to demonstrate the feasibility of an automatic procedural music generation system capable of detecting low-level visual cues in a real-time environment. The system reflects these cues in an evolving soundtrack to accentuate the emotions experienced by the audience.

To achieve this general objective, we developed a prototype serving as a proof of concept. We aimed to satisfy two main conditions in the development: first, ensuring the system was lightweight enough for real-time operation; and second, establishing a valid and explainable affective correlation between visuals and sound. In light of these two conditions, approaches based on large machine learning models were discarded in favor of the use of simple and understandable video features such as color, light, and movement.

The remainder of this paper is organized as follows: Section 2 provides an overview of the state of the art regarding affective computing and procedural music generation. Section 3 details the proposed system architecture, describing video feature extraction, mapping strategies, and implementation within Unreal Engine 5. Section 4 presents the system evaluation, discussing both performance metrics and results from an empirical user study. Finally, Section 5 draws conclusions and outlines potential future developments.

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. STATE OF THE ART

2.1 Adaptive and Procedural Music

Adaptive and procedural music serve as alternatives to linear and pre-composed music, respectively [1]. Historically, adaptive music has been prominent in real-time interactive environments, particularly video games. This application aligns with the non-linear nature of the medium; indeed, the first notable instance dates back to *Space Invaders* (1978). Its soundtrack was minimal yet adaptive in terms of tempo (Beats Per Minute - BPM).

Over the following decades, as technology advanced, so did adaptive music systems. The most notable example is iMuse, designed by Michael Land and Peter McConnell for the game *Monkey Island 2: LeChuck's Revenge*, which demonstrated potential for application in fields beyond gaming [2, 3]. *Unreal* (1998), the game for which the Unreal Engine was originally developed, also utilized an adaptive soundtrack.

Conversely, procedural music, while also employed in video games, with a notable example being *Spore* [4], has historically been explored across a wider range of disciplines. The first related example is the Markov chain system for generating new scores in a specific style, proposed by Brooks et al. [5] in 1957. In the 1960s, certain synthesizers, such as the *Moog*, were capable of creating procedural music by allowing feedback loops between inputs and outputs to generate unpredictable sounds. Perhaps the most significant exploration of procedural music has occurred within domain-specific programming languages, ranging from *MUSIC 5, Csound, Max*, and *SuperCollider*

in the 1970s to various others still in development today [6].

In recent years, the intersection of affective computing and procedural generation has seen significant advancements. Modern approaches have evolved from basic state-machine mappings to sophisticated frameworks focusing on emotional targets. Notable examples include systems where users collaborate with autonomous agents to generate affective music [7], and rule-based generative systems explicitly controlled by desired valence and Arousal values [8]. Furthermore, tools like EmoteControl [9] have demonstrated the viability of real-time, interactive manipulation of emotional expression in music, shifting the paradigm toward highly adaptive, user-centric audio engines. For a comprehensive review of these adaptive generation paradigms, we refer to the work of Nie et al. [10]

2.2 Emotional State Model

Precisely defining emotions has always been a complex task, and in the literature many have proposed contrasting views [11].

Among the other models though, one of the most used is the PAD (Pleasure, Arousal, Dominance) model, which is scientifically recognized for its ability to represent human emotions with sufficient accuracy by mapping them into a three-dimensional space [12, 13]. This mapping facilitates a numerical representation of emotional states, which is particularly advantageous for applications relying on affective analysis. The three variables are defined as follows:

- **Pleasure**: The degree of pleasantness of the emotion.
- **Arousal**: The degree of excitement of the emotion.
- **Dominance**: The degree of control felt in the presence of the emotion.

2.3 Affective Perception of Images

Recent literature on affective image analysis and video-conditioned music generation primarily focuses on heavy multimodal neural networks [14]. However, given the scope of this work, such approaches lack parameter interpretability and could increase the computational load to a point where real-time audio rendering is compromised. Furthermore, neural networks often lack direct parameter interpretability. Therefore, we have focused exclusively on research regarding the perception of low-level features. Specifically, we analyzed three computationally efficient features:

- **Movement**: Calculated as the binary image resulting from the difference between two consecutive frames or, for a longer time window, the sum of multiple binary images (*Motion History Image*). Movement has been shown to influence the Arousal value. [15].
- **Lighting**: generally, there are two main lighting settings associated with different moods: *chiaroscuro* and *flat lighting*. The former is associated with sadness, fear, or surprise, while the latter conveys joy, lightheartedness, and warmth [16]. Rasheed et al. [17] proposed two main features which, when multiplied, estimate the lighting type: the *general light level* and the *shadow ratio*. Hee and Cheong [18] later refined this model, calculating the general light level using the median luminance, and the shadow ratio as the proportion of pixels with a luminosity below 0.18 across the entire image.
- **Color**: Ou et al. [19] defined the three color properties *Color Activity (A_c), Color Heat (H_c)* and *Color Weight (W_c)*, that can describe color perception as follows:

$$A_c = -2.1 + 0.06\left[(L^*-50)^2 + (a^*-3)^2 + \left(\frac{b^*-17}{1.4}\right)^2\right]^{\frac{1}{2}},$$
$$R^2 = 0.95.$$
$$H_c = -0.5 + 0.02\,(C^*)^{1.07}\,cos\,(h-50^\circ),$$
$$R^2 = 0.74.$$
$$W_c = -1.8 + 0.04\,(100-L^*) + 0.45\,cos(h-100^\circ),$$
$$R^2 = 0.73.$$

In the above equations, L^*, a^* and b^* are the coordinates in the color space $CIEL^*a^*b^*$, while h and C^* are respectively the *hue angle* and the *chroma*. R^2 is the measure of the mean square error of the model.

Color Activity, Color Heat and *Color Weight* influence *Pleasure, Arousal* and *Dominance*, respectively [20]. This model was validated through a study involving a mixed British and Chinese population.

2.4 Affective Perception of Music

The human brain processes music through its fundamental components: pitch, temporal information, and timbre, each analyzed by a specific cerebral region [21]. The theoretical foundation for mapping these musical parameters to emotional states is extensively documented in the literature. Comprehensive frameworks, such as the one detailed in Juslin and Sloboda's handbook [22], establish robust, scientifically validated links between specific musical factors and human emotional responses. Furthermore, work by Eerola et al. [23] and Schubert [24] highlights how continuous emotional expression in music can be systematically measured, categorizing the relationship between acoustics and affect, while Bresin and Friberg [25] have demonstrated how these parameters can be algorithmically manipulated for the emotional coloring of computer-controlled performances. Building upon these foundational theories (which refer primarily to Western musical culture), the specific perceptual mappings used in our architecture are defined as follows:

- **Pitch**: In music, pitch perception affects mode, melody, and harmony. The mode sets the general mood of a piece; empirical tests confirm that a major mode communicates cheerfulness, while minor modes evoke sadness and melancholy [26]. Melody can convey a sense of stability if its structure emphasizes triadic scale degrees (first, third, and fifth).

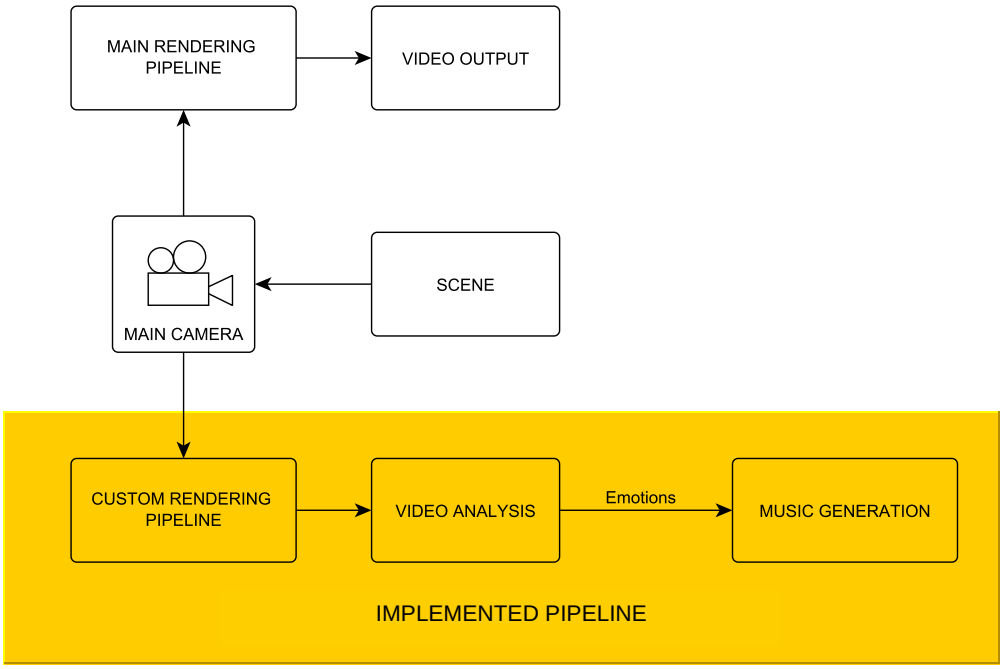


Figure 1. High-level scheme of the architecture for the developed software, embedded into a typical context for Unreal Engine.

Conversely, emphasizing non-triadic degrees (second, fourth, sixth, seventh) instills a sense of instability in the listener [26]. In harmony, pleasant emotions are communicated by consonant intervals and chords, as opposed to dissonant ones [21].

- **Temporal aspects**: The primary temporal components are BPM and rhythmic cells. As expected within the PAD model, higher BPM correlates with higher Arousal [27]. Regarding rhythmic cells, the repeated use of half notes and quarter notes suggests lower Arousal. Conversely, Arousal is enhanced by eighth notes, and even more so by sixteenth notes and dotted or syncopated rhythms [27].

- **Timbre**: Instruments with a lower frequency spectrum and slower attack and decay are associated with sad emotions. In contrast, bright sounds with a fast attack are related to joy or anger; however, if combined with a paced vibrato, they can transmit fear [28].

3. ARCHITECTURE DESIGN

The system was designed for implementation entirely within the Unreal Engine 5.4 environment [29], a choice driven mainly by the availability of the new built-in DSP system, MetaSounds [30]. This ensures seamless integration into any Unreal project.

A high-level scheme of the architecture is shown in Fig. 1. Its core is a parallel rendering pipeline that periodically captures the main camera's output to a texture and analyzes its low-level visual features. It extracts numerical PAD values and sends them to various adaptive MetaSounds sources to generate evolving music. The extraction and usage of PAD values are grounded in the state of the art. The system allows the user to define the frame rate and resolution of the parallel rendering pipeline (maintaining the main camera's aspect ratio). We opted for an iterative design process due to the significant relevance of empirical testing; consequently, the system was refined following initial results.

3.1 Affective Analysis

The application computes the aforementioned color features, lighting settings, and movement (relative to the previous frame) once for each captured image to extract overall PAD values within a reasonable timeframe. The features and their corresponding affective values are calculated as follows:

- **Activity, Heat, Weight**: These features are calculated on a per-pixel basis, following the equations cited in Section 2.3. The mean value is then computed. The three resulting values are normalized to a 0-1 range and mapped to their respective PAD variables.

- **Lighting**: The lighting setting is detected using the Hee and Cheong model [18], as mentioned in Section 2.3. The result influences the Pleasure variable.

- **Movement**: Movement is determined by computing the ratio of pixels that have changed value relative to the previous frame to the total number of pixels. The resulting value influences the Arousal variable.

3.2 Music Generation

The extracted PAD values are fed into the MetaSounds sources to generate the audio. Drawing upon state-of-the-art research, these affective data modify the score and timbre of non-percussive instruments and manage the vertical mixing of percussive elements to convey the detected emotions. The design of the MetaSounds sources is summarized below:

- **Global Parameters**: The root key is user-defined, allowing control over the overall pitch. Other global parameters are driven by PAD values: BPM is directly proportional to Arousal, the key mode depends on Pleasure, and the global reverb level is determined by the equation: *Reverb Wet Level* $= 1 - A * D$, where *A* is Arousal, and *D* is Dominance.

- **Melody**: The score of the main melody is procedurally generated, with Arousal controlling its primary characteristics. Rhythmic cells are selected from a predefined set based on stability; scale degrees (from 1 to 7) are chosen randomly with weighted probabilities favoring stable or unstable degrees. Furthermore, higher Arousal increases the probability of transposing the current degree up by an octave.

  The timbre for the melody is a sawtooth wave with an AD (Attack-Decay) envelope, where attack time is inversely proportional to Dominance and decay time is inversely proportional to Arousal. Finally, in conditions associated with fear (low Pleasure and Dominance, high Arousal), the notes are slightly detuned.

- **Harmony**: The harmony comprises chords and an arpeggio, crossfading between each other based on the Arousal value (higher Arousal increases the arpeggio volume, and vice versa). The notes played are identical in both cases; however, they are transposed one octave lower and depend on the scale degrees generated for the melody. Specifically, every melody note has a 50% probability of triggering a new chord, for which it serves as the root note. The other degrees are selected from a set of predefined chords based on the Pleasure value: higher Pleasure selects more consonant intervals, whereas lower Pleasure selects more dissonant ones.

  The timbres for the chords and arpeggio differ only in their amplitude envelopes: the former is sustained with a fast attack and release, while the latter uses a fast AD envelope. Both are sine waves with a chorus effect, the depth of which is determined by the equation: *Chorus Depth* $= 1 - A * P$, where *A* is Arousal and *P* is Pleasure. The resulting timbre is processed with distortion (increasing with Dominance) and a low-pass filter whose cutoff frequency is directly proportional to the Dominance value.

- **Bass**: The bass line follows the root notes of the chords but is played two octaves below the melody. Its timbre is a sawtooth wave processed through a low-pass filter with a cutoff frequency directly proportional to Dominance. Its amplitude envelope is sustained with a fast attack and release.

- **Percussions**: Percussive sounds are managed via vertical mixing, crossfading between pre-recorded loops designed to convey specific emotions. The selected emotions are happiness, fear, surprise, and anger, all characterized by high Arousal but varying Pleasure and Dominance levels. The low Arousal cases are associated with the absence of percussions, which generally delivers more calm emotions. For other PAD combinations, no percussion is played.

4. IMPLEMENTATION DETAILS

The core logic of the system, excluding the MetaSounds sources, is implemented within a single Blueprint class intended to be instantiated as a child of the main camera.

Frames captured on a `RenderTexture` are read in parallel, rather than pixel-by-pixel, using the *AsyncReadRT* plugin [31]. The resulting color array is processed by an asynchronous C++ function to calculate visual features without blocking the main execution thread.

While other features are computed following the equations described above, the implementation of lighting detection warrants specific mention, as the presented model required adaptation to our context. We applied the equation: *lighting key* $= L - sh$, where *L* is the general light level and *sh* is the shadow ratio. The result can be positive or negative and modifies the Pleasure value accordingly.

Additionally, it is important to note the remapping applied to Color Activity, Color Heat, and Color Weight. Since the referenced study did not specify the output range for these metrics, we experimentally determined scaling factors to optimally distribute the values within the normalized 0-1 interval. In particular we found a range from -1.786497 to 3 for Activity, from -1 to 1.5 for Heat, and from 0 to 2.5 for Weight. We then used these reference to compute the normalization.

Finally, the extracted PAD values are transmitted to the MetaSounds sources, where they are interpolated with the previous values over a user-adjustable duration.

5. EVALUATION

For all tests, the analysis texture resolution was set to 72x128 pixels, with a refresh rate of 2 FPS and a 2-second interpolation time for the extracted parameters. These values were deemed a suitable compromise, ensuring accuracy while maintaining sustainability on standard consumer-grade hardware.

5.1 Performance Test

The performance evaluation was conducted using a sample 3D Unreal project emulating the game Pong. The test environment consisted of a Windows 10 workstation equipped with an *AMD Ryzen 7 5700G* CPU and an *NVIDIA GeForce RTX 2060* GPU.

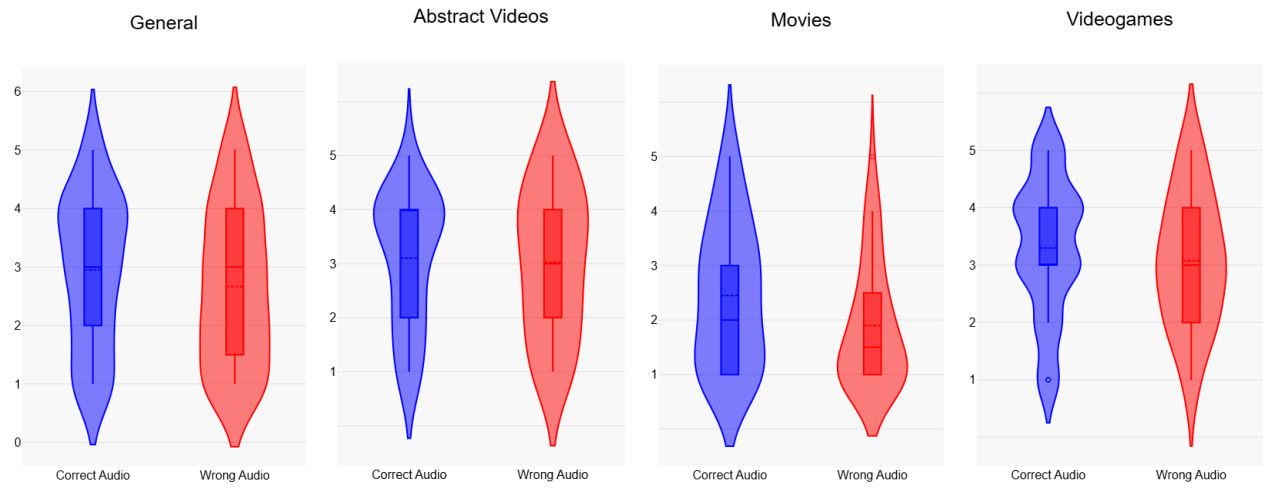


Figure 2. Violin plots of the response distributions, with correct and wrong audio. The leftmost plot represents responses across all video categories, the three remaining plots represent responses for each of the three video categories.

We compared the system's performance with the application both active and inactive, measuring FPS, CPU usage, and memory usage.

The results were favorable: activating the real-time audio generation introduced a negligible performance overhead. Profiling via Windows Process Monitor showed a constant 120 FPS in both states, with CPU and memory usage deltas consistently below 1% (averaging 28% CPU and 60% memory usage overall).

5.2 Empirical User Study

The empirical evaluation involved two surveys, with software updates implemented between sessions following an iterative design approach. The study employed a within-subjects design. Video clips were chosen in order to present a variety in color palettes, lightning and movement. Presentation order of the audio versions was randomized, and participants were blind to the matching condition. Participants were predominantly university students (aged 20-30), of whom a 35% have to do with procedural music, music production or multimedia production in their jobs (25%) or as a hobby (10%). To ensure consistency in the contextual setting, all evaluations were conducted in a quiet environment using standard studio headphones.

The first test presented 12 clips drawn from four video games (*Spore*, *Outer Wilds*, *Gris*, and *Jotun*). For each game, the same video content was presented, but three audio variations were generated: one based on the original video clip, one based on a time-reversed version, and one based on a color-inverted (negative) version.

20 participants were asked to rate the following question on a Likert scale of 1 to 5: "To what extent does the audio content emotionally reflect the video content?". The results are reported in Table 1.

The results suggested that the affective model required improvement, particularly regarding color analysis logic. Consequently, the system was updated prior to the second survey.

Table 1. Results of the first empirical test. Prefix *neg* stands for *negative*, prefix *rev* stands for *reverse*. Note: Mean ratings are reported for heuristic comparison; non-parametric tests were used for significance.

Rank	Mean Rating	Label
1	3.75	*neg* Outer Wilds
2	3.79	*neg* Gris
3	3.45	*neg* Spore
4	3.35	*neg* Jotun
5	3.25	*rev* Spore
6	3.14	Spore
7	3.10	*rev* Gris
8	3.00	Gris
9	2.95	Outer Wilds
10	2.75	*rev* Jotun
11	2.60	Jotun
12	2.10	*rev* Outer Wilds

First, we refined the chiaroscuro computation. The original equation yielded maximum values for completely dark screens, which does not accurately represent high chiaroscuro. We therefore applied a penalty to the chiaroscuro value when the shadow ratio exceeded an experimentally determined threshold. In particular we used a value of 0.6, meaning if more than 60% of the pixels have a luminosity under 0.18 we decrease chiaroscuro value. The penalty linearly increases as the shadow ratio increases beyond that threshold. Second, we incorporated two additional color features: brightness and saturation, which influence Pleasure and Arousal, respectively, according to Wang and Ji [32]. These variables are context-dependent, implemented via a leaky integrator defined by the equation: $y(n) = \lambda^* y(n-1) + (1-\lambda)^* x(n)$, where λ represents the leaky integrator memory (user-defined). We allowed users

Table 2. Results from the second empirical test. Label (*abs*) stands for *abstract*, label (*mov*) stands for *movie* and label (*vg*) stands for *videogame*. Prefix *wrong* indicates the wrong audio version.

Rank	Mean Rating	Label
1	3.80	*wrong* abs 2
2	3.40	abs 2
3	3.35	vg 1
4	3.25	vg 2
5	3.15	*wrong* vg 1
6	3.00	*wrong* vg 2
7	2.90	mov 1
8	2.80	abs 1
9	2.65	*wrong* mov 1
10	2.25	*wrong* abs 1
11	2.00	mov 2
12	1.15	*wrong* mov 2

to toggle brightness and saturation analysis based on their desired perceptual outcome.

Additionally, drum loops were re-recorded to better evoke the corresponding emotions, and the mapping parameters for Color Activity, Heat, and Weight were fine-tuned. In particular the new ranges considered for the normalization were: -1.5 to 2.5 for Activity, -1 to 1.7 for Heat, and -1.3 to 2 for Weight.

The second test involved 20 new participants. The stimuli included video games, movies, and abstract videos, totaling 12 unique clips. Each clip was presented twice: once with the correctly generated audio and once with audio generated from an unrelated clip, purposely chosen to have a low visual features correlation with the shown video (which can be considered as a case of random PAD values). For this test, brightness and saturation were enabled with a leaky integrator memory (λ) of 0.95. The results are shown in Table 2.

The Wilcoxon signed-rank test on the aggregated data showed statistical significance (p ¡ 0.001). However, a categorical breakdown revealed variance: movies (p ¡ 0.001) and video games (p ¡ 0.05) showed significant alignment, whereas abstract clips did not (p = 0.555). This aligns with the data in Table 2, where the clip "abs 2" received higher ratings with the mismatched audio. We observed that this specific clip featured a predominant black background. This suggests the model may need to account for the spatial relevance of specific screen areas (saliency). The lower ranking of movies as a category reflects the model's inability to process high-level semantic content.

6. DISCUSSION AND APPLICATIONS

While current procedural audio systems often rely on pre-composed adaptive logic that requires manual authoring of triggers and states, this proof-of-concept demonstrates an alternative paradigm for Human-Computer Interaction. By utilizing visual buffers as a direct affective interface layer, interactive media can dynamically generate emotionally congruent audio without explicit programmatic cues. This establishes a direct, unmediated link between the visual output and the audio engine. Such an approach holds significant potential for real-time applications, including live visual-to-audio performances, sensory substitution tools for accessibility, and adaptive gaming, where the software acts as an autonomous affective agent responding to unpredictable visual stimuli.

Drawing a biological parallel to the mechanisms of emotional response to music [33], this architecture does not attempt to mimic the higher cortical functions required for semantic or episodic appraisal of a scene. Instead, it operates similarly to a brainstem reflex: it directly couples low-level sensory inputs (visual features) to low-level acoustic properties to induce immediate, pre-cognitive affective responses.

Results from the performance test, which show the emotional coherence between visual and auditory stimuli, support the objective of proving the feasibility of a procedural music generation system based on real-time affective analysis of video streams has been achieved. However this evalutation should be regarded as preliminary and needs refinements. Specifically, it lacks strict baselines: comparing the generated audio solely to mismatched audio limits the ability to conclusively validate the specific visual-to-audio mapping. Also, the lack of significance for abstract clips and the unexpectedly high ratings for some color-inverted stimuli suggest that the affective mapping may not generalize equally across visual materials.

7. CONCLUSIONS, FUTURE WORKS AND LIMITATIONS

The current prototype may be further refined to enhance affective alignment, for instance by employing a *Saliency Map* [34] approach. Such a method would prioritize central narrative events rather than the overall mood, which was the focus of this study. Another way to improve the model would consist of expanding the timbral palette.

As a proof-of-concept, the current evaluation relied on mismatched-audio comparisons. In order to respond to the limitations highlighted above, future robust evaluations must include strict baselines, such as continuous constant-mood audio, random PAD signal generation, or pure silence, to isolate the specific contribution of the visual-to-audio mappings.

The unexpected results obtained for abstract clips, also discussed above, hint at a methodological limitation on the current system realization, and merit further investigations. Future work will employ controlled ablation studies on individual visual features to understand potential confounding variables in the color analysis pipeline.

Further technical validation will require deep profiling of audio-specific metrics, including buffer underruns and end-to-end latency measurement.

The proposed prototype has been made publicly available through a dedicated git repository. [1] Video clips used for

[1] gitlab.di.unimi.it/filippo.giabelli/realtime-soundtrack-generation-from-low-level-video-features.git

the second test (correct audio only) are also available. [2]

8. REFERENCES

[1] C. Plut and P. Pasquier, "Generative music in video games: State of the art, challenges, and prospects," *Entertainment Computing*, vol. 33, p. 100337, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952119300795

[2] K. Collins, *Game sound: an introduction to the history, theory, and practice of video game music and sound design*. Mit Press, 2008.

[3] M. Z. Land and P. N. McConnell, "Method and apparatus for dynamically composing music and sound effects using a computer entertainment system," May 24 1994, uS Patent 5,315,057.

[4] J. Kent., "Procedural music in spore," in *Game Dev Conference*, 2008.

[5] F. P. Brooks, A. L. Hopkins, P. G. Neumann, and W. V. Wright, "An experiment in musical composition," *IRE Transactions on Electronic Computers*, vol. EC-6, no. 3, pp. 175–182, 1957.

[6] P. Galanter, "Generative art theory," *A companion to digital art*, pp. 146–180, 2016.

[7] F. Morreale and A. De Angeli, "Collaborating with an autonomous agent to generate affective music," *Computers in Entertainment (CIE)*, vol. 14, no. 3, pp. 1–21, 2016.

[8] I. Wallis, T. Ingalls, E. Campana, and J. Goodman, "A rule-based generative music system controlled by desired valence and arousal," in *Proceedings of 8th international sound and music computing conference (SMC)*, 2011, pp. 156–157.

[9] A. Micallef Grimaud and T. Eerola, "Emotecontrol: an interactive system for real-time control of emotional expression in music," *Personal and Ubiquitous Computing*, vol. 25, no. 4, pp. 677–689, 2021.

[10] G. Nie and Y. Zhan, "A review of affective generation models," *arXiv preprint arXiv:2202.10763*, 2022.

[11] P. E. Ekman and R. J. Davidson, *The nature of emotion: Fundamental questions.* Oxford University Press, 1994.

[12] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[13] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, pp. 261–292, 1996.

[2] https://bit.ly/44shXOu

[14] Y. Baveye, C. Chamaret, E. Dellandréa, and L. Chen, "Affective video content analysis: A multidisciplinary insight," *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 396–409, 2018.

[15] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, pp. 174–184, 2002.

[16] H. Zettl, *Sight, Sound, Motion: Applied Media Aesthetics, 3rd ed.* Wadsworth, 1998.

[17] Z. Rasheed, Y. Sheikh, and M. Shah, "On the use of computable features for film classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 52–64, 2005.

[18] H. L. Wang and L.-F. Cheong, "Affective understanding in film," *IEEE Transactions on circuits and systems for video technology*, vol. 16, no. 6, pp. 689–704, 2006.

[19] L.-C. Ou, M. R. Luo, A. Woodcock, and A. Wright, "A study of colour emotion and colour preference. part i: Colour emotions for single colours," *Color Research & Application*, vol. 29, no. 3, pp. 232–240, 2004.

[20] R. M. A. Teixeira, T. Yamasaki, and K. Aizawa, "Determination of emotional content of video clips by low-level audiovisual features: A dimensional and categorial experimental approach," *Multimedia Tools and Applications*, vol. 61, pp. 21–49, 2012.

[21] J. Warren, "How does the brain process music?" *Clinical Medicine*, vol. 8, no. 1, pp. 32–36, 2008.

[22] P. N. Juslin and J. Sloboda, *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[23] T. Eerola, A. Friberg, and R. Bresin, "Emotional expression in music: contribution, linearity, and additivity of primary musical cues," *Frontiers in psychology*, vol. 4, p. 487, 2013.

[24] E. Schubert, "Measuring emotion continuously: Validity and reliability of the two-dimensional emotion-space," *Australian Journal of Psychology*, vol. 51, no. 3, pp. 154–165, 1999.

[25] R. Bresin and A. Friberg, "Emotional coloring of computer-controlled music performances," *Computer Music Journal*, vol. 24, no. 4, pp. 44–63, 2000.

[26] E. Lindström, "Impact of melodic organization on perceived structure and emotional expression in music," *Musicae Scientiae*, vol. 10, no. 1, pp. 85–117, 2006.

[27] A. Fernández-Sotos, A. Fernández-Caballero, and J. M. Latorre, "Influence of tempo and rhythmic unit in musical emotion regulation," *Frontiers in computational neuroscience*, vol. 10, p. 80, 2016.

[28] J. C. Hailstone, R. Omar, S. M. Henley, C. Frost, M. G. Kenward, and J. D. Warren, "It's not what you play, it's how you play it: Timbre affects perception of emotion in music," *Quarterly journal of experimental psychology*, vol. 62, no. 11, pp. 2141–2155, 2009.

[29] EpicGames, https://www.unrealengine.com/en-US, (Retrieved: 22/03/2025).

[30] ——, https://dev.epicgames.com/documentation/en-us/unreal-engine/metasounds-the-next-generation-sound-sources-in-unreal-engine, (Retrieved: 22/03/2025).

[31] N. Chalkley, https://nicholas477.github.io/blog/2023/reading-rt/, (Retrieved: 25/03/2025).

[32] S. Wang and Q. Ji, "Video affective content analysis: A survey of state-of-the-art methods," *IEEE Transactions on Affective Computing*, vol. 6, no. 4, pp. 410–430, 2015.

[33] P. N. Juslin and D. Västfjäll, "Emotional responses to music: The need to consider underlying mechanisms," *Behavioral and brain sciences*, vol. 31, no. 5, pp. 559–575, 2008.

[34] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 2002.