Tampereen yliopisto

Choudhary Shahzad Shabbir

GENERATION OF MUSICAL PATTERNS USING VIDEO FEATURES

Faculty of Information Technology and Communication Sciences
M. Sc. thesis
June 2019

ABSTRACT

Choudhary Shahzad Shabbir: Generation of musical patterns using video features
M. Sc. thesis
Tampere University
Master’s Degree Programme in Software Development
June 2019

With the growing interest in social media applications, mobile phones have also seen a dramatic improvement in the quality of their cameras. This has caused a surge in the number of videos made by ordinary users, now capable of capturing any scene anywhere. Such videos often suffer from a lack of background music accompanying them. A simple solution is to attach an existing track that is particularly suitable for the video, yet it is also possible to create a completely new one. Research has thus far focused on recommending appropriate tracks for a given video, whereas the concept of automatic music generation is less studied. In any case, the addition of a new music track must rely exclusively on the features of the original video.

In this study, a novel approach has been used to extract data using different video features and generating new music from those features. A desktop application has been designed for this purpose, containing a complete pipeline from importing the video to outputting the final video complemented with new music. To analyze the music quality, a user survey was conducted with roughly 100 participants. The survey contained several distinct videos, each represented in multiple variations with different musical settings. It was revealed that most samples of the newly generated music had enough potential to accompany the video and make it more interesting and meaningful. The results suggest that a more detailed user survey is needed to identify the precise features found appealing by the listeners, exhibiting less variation in musical tempo but more in the instruments applied.

Keywords: video metrics, video features, auralization, sonification, music generation

Contents

List of abbreviations

BPM	Beats Per Minute
CSV	Comma-Separated Values
D2M	Data2Music
GUI	Graphical User Interface
HSL	Hue, Saturation, Lightness
JSON	JavaScript Object Notation
MIDI	Musical Instrument Digital Interface
MP3	MPEG-2 Audio Layer III
MPEG	Moving Picture Experts Group
OpenCV	Open Source Computer Vision
PCM	Pulse Code Modulation

1. Introduction

In recent years, the usage of smartphones has dramatically increased. Among its many everyday uses, one very important feature of a modern smartphone is a good built-in camera. It has certainly become an important aspect in the competition among smartphone makers and the decisions of potential buyers. Since a smartphone equipped with a good camera attracts so much of its owner's attention, the production of user-generated videos is also increasing faster than ever before. Filming a scene anywhere and anytime is just one tap away with the phone almost always in one's reach.

On top of the wide usage of mobile phones and increased number of recorded videos, the trend of using different social media applications and platforms is also gaining greater popularity. Facebook, YouTube, WhatsApp, Instagram, Snapchat and many other such applications have clearly taken over the traditional modes of communication and socializing. The increased usership of these social media giants directly reflects the number of videos filmed and shared by their users.

While user-generated videos sometimes lack appeal without fitting background music, in some cases their original sound is perfectly adequate. There is often no need to change the audio recorded alongside the video if it conveys added value and meaning to the video clip. For instance, if a video is recorded at a concert, in an interview or at a sports event with live commentary, no additional background music is required. Educational material, where the presenter actively refers to the content and provides their own explanations, should also be left unmodified.

Instead, the need for suitable background music arises for videos where the recorded sound does not aid the video in any way, or in other words, when the video requires more support from the sound than it provides. When the clip includes sources of background noise, such as passing cars, gusts of wind or irrelevant conversations, the original audio track can be replaced with a different one altogether. Addition of music is thus relevant for both edited vlogs and unedited footage filmed at various locations. Another appropriate scenario is to add background music to slideshows, composed of static images with a possible use of transitions between them. Such videos often have a simple and relatively popular audio track attached to them or no sound whatsoever, and would surely benefit from a greater variety of music.

One way to accomplish this variety is by synchronizing the video with an existing piece of music, usually stored in a database. Even with a database large enough to contain thousands of soundtracks, it is a complex task to select a single soundtrack that best complements the video in question. Prior research attempted to find this match by applying different algorithms for video analysis. Once the best match was selected, the length of the video was adjusted to fit the audio or vice versa [Foote et al., 2002; Liao et al., 2009; Shah et al., 2014].

The idea of matching an existing soundtrack to the video is clearly promising, but it has its own drawbacks as well. It is challenging to cut out parts of either the audio or the video to compensate for the other, specifically to select the parts worth cutting. Removing fragments from a completed sequence can significantly damage the harmony and smoothness of the mapping, resulting in an unpleasant experience. Moreover, maintaining a repository of soundtracks will likely require additional work, such as acquiring relevant permissions.

The other possible solution appears to be producing completely new music according to the video features. This idea is likewise problematic. Firstly, music composition is an art that cannot be perfectly imitated by an algorithm. Secondly, since the video and audio do not share enough similarities in their features to be mapped against each other in general, automation of this process seems hard to implement.

The crucial task in creating music for videos is the extraction of meaningful features from the video. In some cases, metadata such as geographical location tags of the place where the video has been recorded may be sufficient. More commonly, however, different video features such as human gestures, shot boundaries and camera motion are also utilized, as in the study by Wang and Cheong [2006].

There are some other rather abstract video features, mostly pixel-level metrics such as brightness, contrast, or hue, to be employed for the task. These features are an essential part of every video but they do not convey enough information about the video’s content. Thus, such features do not seem to be of any use for matching with the audio features of existing tracks. However, they could become very handy for generating music from the video, since they often produce quite a big range of values that can be utilized in the process.

Since matching the existing sound track does not always produce the desired result and a lot of work has already been done in this regard, creating new music, as discussed above, appears a more promising idea. The main problem here lies in the selection of the video features and their impact on the resulting music. The video properties could be extracted not only at the pixel level, but also at the frame, shot or scene level. These high-level attributes, such as shot boundaries, can also prove useful as their values are expected to relate more directly to the video.

Once a number of properties are drawn from a video, the resulting data could be then used for music production, but perhaps not directly. For variety in generated music, users could also be given some control to apply different combinations of the acquired features and available musical instruments, together with various preprocessing tools. The music produced by different settings could be further compared and judged on the basis of its variety, harmony and relevance to the original video. The features that apparently produce more fitting music could also be enhanced in some ways, and those that do not seem to make much of a difference could be discarded.

Of course, the criteria for rating produced music are understandably ambiguous. The rating must essentially decide whether the music is good or bad, which is a highly subjective matter dependent on one's taste, mood and background. Defining universal scales and quality metrics should be almost impossible; however, an indirect assessment of the results can still be obtained, for example, through a user study. Distinct music samples related to the same video can be compared by the participants, producing at least a subjective measure of their quality.

To have the aforementioned feature extraction and music generation functionality, a separate tool would be helpful in order to automate this process and provide an interface to interact with. This would enable the user to apply the settings discussed above and analyze the results produced. Ideally, an existing tool may already be able to imitate various musical instruments and generate music using them.

To summarize, the direction of this thesis is threefold: extracting useful data sets from video features, utilizing these extracted data for the generation of new music, and assessing the produced music with respect to the original video. The work thus attempts to answer the following research questions:

- Which video features can be effectively used to generate new music?
- What is required to transform video data into music?
- Is the resulting music aesthetically pleasing and a good fit to the original video?

These tasks are performed by selecting a set of frame-based metrics from video data, using these as inputs to an existing music generation tool and evaluating the resulting music in a survey. In addition, a new application has been developed to provide a smoother conversion experience, making use of the existing tool's capabilities but also adding several necessary routines before and after the generation process.

The thesis addresses these questions in the following way. Chapter 2 presents a review of prior work in the field, including the properties of video data that could be utilized in the addition of new music, the techniques of selecting a suitable audio sequence from a predefined collection of samples and the methods of generating new audio directly from video data. Chapter 3 discusses the foundations of the practical work performed in this thesis: a selection of metrics elicited from user-provided videos and an existing tool for music generation, as well as the motivation for further extending its functionality. After these preliminary studies, Chapters 4 and 5 report on the results attained by the work, namely a new music generation tool with a more centralized set of functions and a number of findings produced by the survey. The thesis concludes with a brief discussion of the findings in Chapter 6, including the restrictions inherent in the work and potential directions of its further advancement, and formulates the conclusions in Chapter 7.

2. Related work

The problem of selecting suitable audio to accompany a given video, instead of its original soundtrack, has been explored unevenly from different angles. The fundamental prerequisite for the process is apparently the extraction of various metrics, or numerical data, from the source video, with the intent of relating them to the new soundtrack. These metrics can range from basic frame-specific parameters to complex attributes of shots or scenes, and in general they can be elicited from any information embedded in the video. This also includes the audio potentially accompanying it.

The role of choosing appropriate music for available video material is fairly significant. This is seen not only in the cases when the background sound is lacking in quality or missing altogether, but also in such operations as movie production. In a way, the composer tasked with providing a soundtrack for a movie is also faced with a video sequence with limited auditory support (dialogues and miscellaneous sounds). What the composer can produce to accompany this video sequence has a notable impact on the ultimate quality of the movie.

When a human is able to participate in the task, the work tends to be done on a fairly abstract level. Composers perceive the intended mood and purpose of individual scenes and try to create musical themes and transitions that would match these intangible characteristics. In doing so, they rely on their own personal style and a familiarity with a great body of existing music. The result is usually a coherent musical collection, with recurring themes persisting across the movie in different variations but always treated in accordance with the scene they happen to cover.

If the process of music production is to be automated, more rudimentary techniques necessarily have to be developed instead. Possibly the easiest way to obtain a new audio track for a particular video is simply to pick one from a sufficiently large collection, i.e. to recommend a suitable track. Given certain criteria that define "goodness of fit" between the audio and the video, the best match can be selected and applied in each case. The exact criteria and their relation to the original video properties can be highly diverse, as seen in the many existing implementations of the procedure.

Instead of the recommendation task, i.e. selecting an existing soundtrack from a predefined set according to its alignment with the video, it is also feasible to generate an audio sequence from scratch. Understandably, soundtrack generation is a more complicated problem than soundtrack recommendation and is not as extensively studied in the literature. However, this process has an even closer relation to the metrics derived from the video, since the soundtrack is constructed from these values alone. Accordingly, generation was regarded as the more relevant technique to this thesis.

A simplified view of these two approaches is offered in Figure 1. In particular, the figure hides the complexity of the procedures needed to generate a soundtrack from scratch, as opposed to merely selecting a match from a prepared collection. The data extracted from the video in both cases can be the same or different, depending on how well they support the selection or generation tasks.

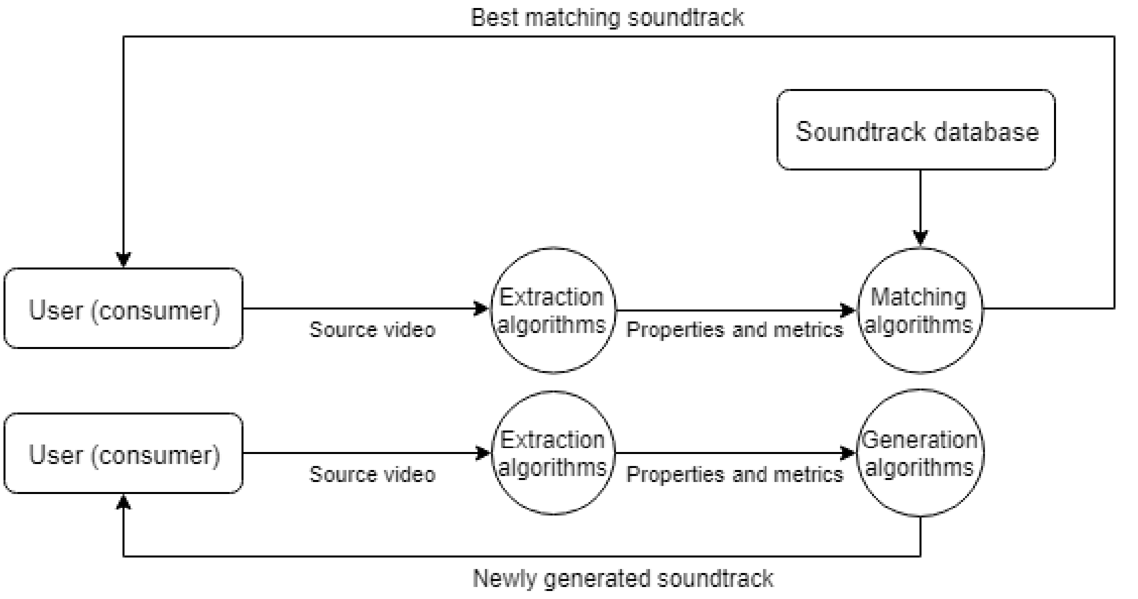


Figure 1. Recommending and generating a soundtrack.

In Figure 1 as well as the following discussion, “soundtrack” is used in the common sense of the word, i.e. the audio material that accompanies a given video sequence. While it is not necessarily a musical product, everyday use of the term (e.g. in movie production) does imply the music attached to a particular scene. For amateur videos, a better expression with the same meaning would be “background music”. From a practical perspective, video playback tools usually refer to an “audio track” that comes with a particular video file, so that the audio and video data represent two components of the same file. In this sense, an audio track is a concrete realization of the soundtrack concept and a specific solution to the problem of soundtrack recommendation (generation).

The remainder of this chapter is structured according to the key concepts mentioned so far. Section 2.1 offers a review of the metrics that may be elicited from video material for various purposes, whether direct frame-specific parameters or more sophisticated quantities. Sections 2.2 and 2.3 focus respectively on soundtrack recommendation and soundtrack generation, the two principal modes of processing metric data for musical purposes. The literature on soundtrack recommendation is notably more extensive and exhibits a wide variety of approaches, including the idea of editing the audio and video components to achieve an even better fit between the two.

2.1. Video features and metrics

No matter how a soundtrack for a video clip is derived, it must necessarily depend on a number of properties drawn from the clip. These can be fairly low-level features such as pixel colours, frame rate, brightness and contrast, shot-specific and scene-specific metrics including camera motion, tempo, and object movement, or even abstract concepts such as emotion.

Apart from video data, the original audio track accompanying the clip can also be inspected for various metrics, including audio energy and tempo. This is presumably a poorer source of information, given that a video signal requires more information to be encoded and occupies a larger share of overall human perception. The audio track's relevance may lie in more abstract concepts, such as detection of arousal and valence in the study by Hanjalic and Xu [2005]. In this paper, sound energy was taken as one of the three components of a model evaluating arousal values for a given video segment.

The distinction between less and more abstract features relates closely to the structure of a video, or more generally a movie. The usual approach is to examine a video as a sequence of scenes, which are technically combinations of different shots captured by a camera. Likewise, a single shot comprises multiple individual frames, the smallest units of classification. Accordingly, detecting individual shots or scenes and their boundaries is a problem persistently encountered in the literature. Since the distinction between scenes is mostly semantic, not directly visual, it is clearly difficult to make it with conventional video analysis tools alone.

A clear-cut distinction of features applicable to shots, scenes, and whole movies was given by Zhai et al. [2004]. Figure 2 illustrates the relation between these concepts that is in agreement with the paper's terminology. While this study focused on the classification of scenes into different types, specifically conversation, suspense and action, it also proposed a number of relevant features. The paper used a compound metric built from the intensity of the camera motion, its smoothness, and the audio energy of a given shot. Using these parameters, several finite-state machines were described with the intent of deciding the type of an arbitrary input scene.

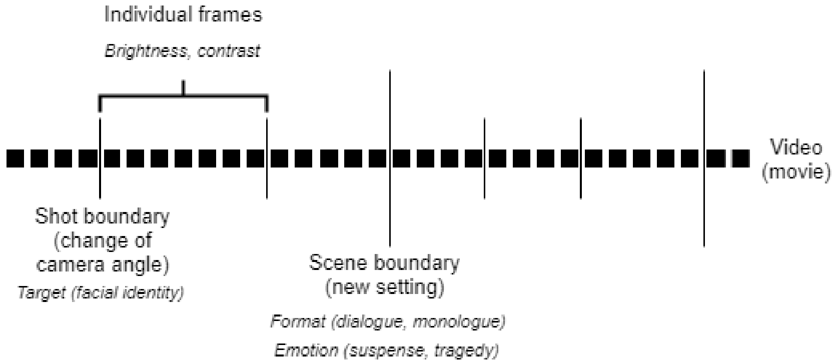


Figure 2. Structural elements of a video.

A study by Kang [2003] attempted to detect emotional features in videos with the aid of hidden Markov models. Three states, namely fear, anger and joy, were manually mapped to the colours, camera motion, and shot rate of a particular video segment. The model was then applied to a sequence of these feature values, effectively a time series, which produced the likelihoods for each of the emotional states. The technique showed adequate recognition rates for a small selection of videos, though it is doubtful whether low-level metrics truly map smoothly to psychological states.

Chen et al. [2004] posed the problem of detecting movie segments based on tempo. This is in itself a compound metric, derived (similarly to the previous work) from shot changes, motion intensity, and audio features. The work used a simple algorithm based on pixel differences to locate shot boundaries, making further use of the motion descriptors of MPEG-7 and audio energy peaks to compute a weighted metric of these three factors.

A hierarchical clustering algorithm was then employed to find the most “interesting” shots, with the restrictions that high-tempo shots not be too close to each other and be separated by low-tempo shots, which acted as story boundaries. The paper attempted to arrange high-tempo shots into a movie trailer of sorts, or to expand them with adjacent scenes to create a somewhat more detailed preview. In the context of soundtrack recommendation or generation, the results of the process may instead be used to change the intensity of the audio track at the appropriate moments.

Scene extraction was further attempted by Truong et al. [2003], on a level of detecting not frame-specific boundaries between shots but shot-specific boundaries between scenes. The article provided a comprehensive description of a scene from the movie and its director’s perspective. The work used colour values in the HSL (hue, saturation, lightness) model, averaged across the entire shot and further normalized to the same scale, before looking for colour changes with an edge detection method. Alternatively, the coherence of adjacent shots was evaluated to find scene boundaries. This method proved to be more accurate, though it still failed to detect certain cases called “punctuation devices”, which some further refinements could handle with mixed success.

A common feature of these studies is their focus on well-developed videos, such as edited clips or fragments of actual movies. It is apparent that high-level metrics, much as their identification is complicated, are still more likely to be found in such videos. However, the majority of videos that are of interest to the average smartphone user probably do not possess the same internal structure and thus the same abstract metrics. Video data that are meaningful for further soundtrack recommendation or generation must therefore be derived from more primitive features.

2.2. Soundtrack recommendation

Among the different ways to pair a given video clip with a suitable soundtrack, selecting a track from a predefined collection is apparently the easiest solution. Multiple studies have used different video-related aspects to suggest an appropriate soundtrack from the available tracks in a database. They mostly rely on a combination of video and audio features, which is interpreted and rated in some way to determine the most suitable soundtrack candidates. Generally, this approach is often referred to as soundtrack recommendation.

Kim and André [2004] discussed the concept of an affective music player, which chose audio tracks to elicit a particular emotional response. For this task, the emotional impact of music itself was to be evaluated, whether through the listener's self-reported perceptions, their physiological reactions, or features of the audio. By evaluating automatically generated music samples, test subjects indicated to the system which physiological factors were related to which types of music. A genetic algorithm then scanned through a pool of random rhythms to determine the ones with the most suitable emotional payload. The focus of the work, however, was on the detection and matching of emotional responses, not on the music generation process itself.

Kuo et al. [2013] proposed a soundtrack for a video by analyzing the relationship between audio and video features using multi-modal semantics. They also used an algorithm to calculate the alignment between the music and video streams. The videos were first analyzed to predict emotions using colour, light, texture and motion factors. After the video analysis, certain low-level (rhythm, timbral texture) and high-level features (danceability, energy, loudness, mode, tempo) were extracted from the available audio tracks. Once identical semantics were found, the alignment algorithm was used to improve the harmony between music beats and video shots. Using the calculated content correlation and alignability, a list of recommended audio tracks was finally proposed for the given video.

These articles also referred to emotions and emotional responses already found in the preceding review of potential metric sources. However, the last paper in particular approached the problem more practically and identified emotions as a helper metric, not as the goal of the whole analysis. It also emphasized the usage of a postprocessing algorithm to improve the alignment between fragments initially seen as suited to each other. This subsequent refinement of the obtained matches, as opposed to a simple one-stage recommendation process, was applied in other studies as well.

For instance, Feng et al. [2010] introduced a framework that taught itself about the similarity of patterns and structures found in online professional videos and their respective background music. Audio and video, being physically independent from each other, required complex mechanisms to define generic matching rules to map audio features such as rhythm, genre and timbre against scene, motion and emotion features of a video. In the paper, two probability models (Gaussian mixture and Markov chain model) were used to filter the associations between audio and video.

Firstly, a shot boundary detection method was used for video segmentation and a colour histogram was created for every frame. The histogram difference was calculated from the differences between two neighbouring frames. The audio track was likewise broken into shots. When the most harmonious fragments were chosen from the music library, they were further adjusted with a warping function to blend better with the video's original audio track, which could possibly include speech.

Once a list of relevant audio tracks was selected for the given video, the dynamic programming approach was used to improve the smoothness of the tracks to fit the video. The cost function to be minimized included two components for the smoothness of adjacent audio shots and their proximity to respective video shots.

Similarly, Yoon and Lee [2007] also used dynamic programming to synchronize music with user-generated videos. Audio features such as note pitch, duration, and velocity were extracted from existing MIDI files and compared to respective video properties, including shot boundaries, camera movement and object movement. Depending on different video features, multiple patterns in the audio track were located that best matched the video. These segments were then mapped to the video, with some synchronizing adjustments that would least affect the sound. The results listed in the paper do not quite determine the efficiency of the technique, but a suspicion is voiced that the matched audio track can still convey the wrong mood.

Liao et al. [2009] approached the issue of soundtrack recommendation by first segmenting professional music videos into small chunks. They used a dual-wing harmonium model [Xing et al., 2005], which is an extension (in fact a restriction) of the neural network class called Boltzmann machines [Larochelle and Bengio, 2008]. The model was trained on a combination of video and audio features, mapping video fragments to points in a multidimensional space. A clustering algorithm was then employed to identify dense groups of points, i.e. related samples, so that the original video clips could be paired with the most closely matching audio fragments.

Video editing was handled more broadly in the recent study by Lin et al. [2017]. They proposed either editing music to match a user-created video or editing the video to match a music track, but not generating new audio tracks from scratch. In their study, segments of video and music were selected and brought together based on their proximity, according to a metric. Experiments showed that suitable soundtracks could be generally found to match user-generated videos; however, these tracks must still come from a previously gathered collection.

Similarly, Foote et al. [2002] approached the production of music videos by having the user select a soundtrack to their liking and matching the source video with it. High-quality audio, especially synchronized with the video material, apparently led to a better reception of the resulting clip. Audio segments were parameterized and analyzed for similarity to each other, with the correlation between segments viewed as a time-specific audio novelty metric. Video clips, on the other hand, were distinguished by their "unsuitability", or the presence of tilt, pan and overexposure; no other segmentation method was used.

Operation of the proposed tool was possible in fully automatic mode by aligning video clip boundaries with the peaks of audio novelty, taking into account the length of the clips and the distance between the peaks. However, an interface was also provided so that a potential user could personally choose the clips to be matched with the soundtrack. This has produced reasonable results, although the authors still considered rhythmic synchronization, i.e. matching video clips with musical beats themselves, and mixing the original audio track with the new one instead of discarding it completely.

Apart from emotional states, similarity of audio and video data as well as synchronization between these two components, valuable information for soundtrack recommendation can be derived from other sources as well. The following studies utilized metadata and techniques that were not applicable to all videos in general, but nonetheless provided interesting results when available.

In particular, Yu et al. [2012] used geographical data obtained from a community-based project called OpenStreetMap. Their system proposed a fitting soundtrack for the given video by looking for suitable mood tags, which were in turn matched to the original video's geotags. However, the actual content of the video or music was not analyzed in any other way, and the conclusions about the outcomes of the study were drawn using very small samples.

Geographical location was also used for the same purpose in a tool named ADVISOR. This system, introduced by Shah et al. [2014], recommended soundtracks for user-specified videos by working on three main aspects. Firstly, it predicted a scene mode based on the user-generated data collected from different sources, including the user's video preferences predicted by online activities such as GPS, listening and search history. Secondly, a heuristic ranking approach was used to predict confidence scores using heterogeneous late fusion. Finally, the proposed video soundtrack was customized to work with the user's device.

Wang et al. [2005] provided an extensive discussion of sports videos in their article. Combining shot-specific and camera-specific video features, "keywords" of audio streams and even related textual commentary, they attempted to fit sports video fragments to already available music clips, which is another instance of soundtrack recommendation. The authors noted that the matching could proceed in both directions. However, the music-centric approach, where video fragments were paired with a fixed audio track, was more complicated due to the need of matching both content and tempo.

To summarize, the problem of soundtrack recommendation is extensively covered in the existing literature. In addition to the metrics already considered, reviewed studies proposed a wide variety of new sources for the recommendation process, including emotional states, audio patterns, camera movements and even geographical data. Importantly, a number of works employed additional procedures to refine the matches identified between audio and video fragments, aiming to create a smoother correspondence between the two.

2.3. Soundtrack generation

The sources reviewed so far show that there are many different techniques to break down a video into features and use them to propose suitable soundtracks. However, the issues of soundtrack recommendation are perhaps not as relevant to this thesis as the generation of principally new music. This particular problem is not widely covered in existing sources, most likely due to the difficulties associated with refining the generated audio track and making it sound more natural.

The process of attaching sounds to an existing object or procedure is commonly called sonification. Hananoi et al. [2016] regarded it as an alternative and an enhancement of visualization techniques. They introduced a tool that converted environmental data, presented in the common format of comma-separated values (CSV), to MIDI. Afterwards, the "composer" was expected to further refine the resulting sound pattern by using an audio editor. The study used several other data sources such as foreign exchange and remote sensing data.

Additionally, O'Sullivan et al. [2017] also converted environmental data, specifically wind turbine output, into a musical form. The collected audio data were normalized into a more harmonious shape without affecting the actual representation of the input. In particular, voltages were mapped to the frequencies of the nearest MIDI notes and amplitudes were quantized to a set of discrete values. Furthermore, recently introduced notes provided feedback to the music generation process, so that new notes formed natural chords with prior ones (a chord in the conventional sense is a grouping of multiple notes, all perceived simultaneously by the ear).

It is worth noting that soundtrack generation does not necessarily require producing individual sounds and applying them strictly to the characteristics of the video (e.g. one sound per frame or one sequence per scene). Some work can also be performed by creating longer chords or themes and linking them with the video in question. Such is the study by Hua et al. [2004], which attempted to create music videos from unedited source material. Their approach relied on locating musical patterns, making use of the self-similarity present in finalized music tracks, and aligning them with appropriate scenes of the video. This is similar to some of the soundtrack recommendation techniques cited above, but a generation element is also present since the result depends exclusively on the source material.

A related "assisted generation" technique was employed by Legaspi et al. [2007] for the purpose of constructing musical pieces that would match a listener's affective labels such as "bright" or "sad". While the objective of the work differs from that of this thesis, the procedure used to generate music is worth considering: it was a genetic algorithm that introduced small random changes to a chord progression, thus creating whole fragments that were consistent with the norms of musical theory. More generally, the work used notes to compose the music, as opposed to individual samples of a digitized sound wave.

2.4. Summary

An overview of the existing literature suggests a wide variety of approaches to the problem of supplying available videos with background music. Most of these techniques fall into the categories of either proposing a suitable, already existing audio track or creating a new one altogether. In both cases, a tight connection with the characteristics of the video is desirable to create an adequate musical representation.

Significant video features appear on several levels, from primitive frame-based characteristics such as brightness to sophisticated shot- and scene-level features such as mood and style. The difficulty in evaluating metrics tends to rise with their abstraction level and the amount of extracted information likewise diminishes. To some extent, high-level features can be recognized and predicted with the aid of low-level ones. The intention of this operation is often to identify "interesting" moments in the video and make use of them in further processing.

Features elicited from video material can significantly aid in the process of soundtrack recommendation. The same high-level features can be used to establish similarities between the video track and candidate audio tracks, so that the most suitable candidate can be chosen, for example, the track with the most similar emotional profile. Given the complexity and disparity between audio and video sources, it is often more feasible to match them in shorter segments, i.e. on the level of shots and scenes. Accordingly, algorithms are needed to reliably detect boundaries between these.

Generating new audio tracks based on an existing video is a more challenging task. One commonly used simplification is the establishment of a mapping between video data and notes of musical instruments, instead of more elementary components such as individual audio samples. This provides a reasonable conversion from original data values to musical notation, which finds a flexible digital representation in MIDI data. The process can be further refined by grouping individual notes into chords and reshaping these to produce more cohesive musical fragments.

3. Preliminaries of music generation

In order to supply background music for user-provided videos, it was necessary to implement a number of concepts related to the discussion in the previous chapter. A selection of suitable video metrics had to be extracted from source videos and mapped to sound patterns. This mapping can occur in two principal ways: it can relate videos to particularly suitable, already existing audio tracks, thus "recommending" them for every video, or it can be utilized to generate new music that would, according to the metrics, be an adequate fit for the original video.

With the relative lack of prior work on generating music, the intent of this study was precisely to provide a way of equipping original videos with new music depending exclusively on each video's characteristics. The practical part of the work involved selecting the metrics to be calculated for a given video, transforming them into an audio track to be used with the video and attempting to evaluate the quality of the resulting music.

The current chapter is divided into subsections discussing particular arrangement and preparation issues for the first two of these objectives. Specifically, Section 3.1 deals with the metrics ultimately extracted from user-provided videos, which had to be computed using custom code due to a lack of uniform processes in prior work. At the same time, a suitable existing tool was utilized for the subsequent task of music generation: an overview of that tool's functionality, as well as its shortcomings and the motivation for further work in the same direction, is presented in Section 3.2.

3.1. Extracted metrics

The most basic video metrics characterize individual frames, so that the entire video yields a sequence with as many values as there are frames in the video. This approach is less sophisticated than the usage of shot- or scene-based metrics, which could potentially assign a single value to a whole sequence of frames. However, frame-based metrics are more reliable in the sense that they are always computable; in the context of arbitrary user-provided videos, as opposed to specifically crafted professional ones, more abstract concepts such as scenes will not necessarily be meaningful.

The metrics discussed in this section are fairly simple implementations of basic video and audio properties, not using particular filtering or preprocessing algorithms. More complex metrics in the context of video quality can be found in a study by Mendi et al. [2011], as well as a detailed overview by Loke et al. [2006].

3.1.1. Metrics from visual data

The following expressions for the metrics rely on the representation of individual video frames as matrices of fixed dimensions, with each element viewed as a tuple of the corresponding pixel's red, green and blue colour components:

$$F^{(k)} = \left(\left\langle R_{ij}^{(k)}, G_{ij}^{(k)}, B_{ij}^{(k)} \right\rangle\right)_{i=1, j=1}^{w,h}. \tag{3.1}$$

Here, $F^{(k)}$ is the matrix corresponding to frame k, while w and h are the frame's width and height in pixels, or alternatively the number of its columns and rows. Each value of the matrix is a non-negative integer no greater than 255. The total number of frames in the video will be denoted by N.

The simplest metric under consideration is undoubtedly brightness, the intensity of light observed in a frame. The total brightness of the frame is effectively the sum of each pixel's colour components:

$$Br^{(k)} = \sum_{i=1}^{w} \sum_{j=1}^{h} \left(R_{ij}^{(k)} + G_{ij}^{(k)} + B_{ij}^{(k)}\right). \tag{3.2}$$

Brightness values can be further scaled by dividing them by the number of pixels in the frame, i.e. $w \cdot h$, producing the average brightness per pixel. However, such scaling is not required for any of the metrics considered here: further processing was able to provide both "horizontal", removing values at either end of the sample, and "vertical" filtering, removing values above or below certain thresholds. Audio generation is also based on the relative magnitudes of the input data, which are not affected by taking averages.

The brightness metric is a measure of intensity in itself, and can thus be used to determine the intensity of the corresponding audio track. In practice, high brightness values may correspond to louder notes of the track's instruments, or perhaps a single instrument that ought to be emphasized.

A related metric can be given the name of "contrast", though it refers to the difference between adjacent images, not the contrast of the image itself. The total contrast includes the differences between the colour components of the same pixels, examined in two successive video frames:

$$Ct^{(k)} = \sum_{i=1}^{w} \sum_{j=1}^{h} \left(\left|R_{ij}^{(k)} - R_{ij}^{(k-1)}\right| + \left|G_{ij}^{(k)} - G_{ij}^{(k-1)}\right| + \left|B_{ij}^{(k)} - B_{ij}^{(k-1)}\right|\right). \tag{3.3}$$

The first frame has no "previous" frame to be compared to, so it is convenient to take $Ct^{(1)} = 0$. This metric is not equivalent to the difference between adjacent brightness values, since it accounts for the magnitude of per-pixel differences, even if they happen to be negative.

The contrast metric is a measure of change, and it can also enforce a certain degree of change in the audio domain. Sequences of low contrast values correspond to continuous sounds, and thus longer notes, while high values should translate into short notes of varying frequency.

Like the brightness metric, this measurement is rather volatile and serves only as a very raw indicator of change between frames. A more sophisticated algorithm, drawn from [Lienhart, 1998], suggests first grouping the pixels into a number of bins depending on their colour components, then adding up the differences between the respective bin counts in adjacent frames. This method only detects changes in colour values that are comparable to the “width” of the bin and thus move a given pixel from one bin to another. Accordingly, it is applicable to the problem of shot boundary detection.

The calculation can be expressed with the following sum:

$$Ctb^{(k)} = \sum_{x=1}^{N_b} \sum_{y=1}^{N_b} \sum_{z=1}^{N_b} \left| bn^{(k)}(x,y,z) - bn^{(k-1)}(x,y,z) \right|, \tag{3.4}$$

where $bn^{(k)}(x,y,z)$ is the number of pixels in frame k such that

$$x-1 \le R^{(k)} \cdot \frac{N_b}{256} < x, \quad y-1 \le G^{(k)} \cdot \frac{N_b}{256} < y, \quad z-1 \le B^{(k)} \cdot \frac{N_b}{256} < z\,. \tag{3.5}$$

Bins are established separately for each of the three colour channels, hence the need for the three-variable notation. N_b is the number of bins in every dimension, typically a small power of 2 such as 8. The sum of all bin counts is the total number of pixels in the frame, $w \cdot h$.

The values produced by this calculation are not directly compatible with the “raw” contrast values, since they reflect the number of changed pixels, not the actual magnitude of changes. However, sufficiently large changes will also alter the pixel distribution between bins, so both metrics will increase or decrease for the same frameset. There is no need to establish a common scale between them, as long as they are separately handled during further processing.

3.1.2. Indirect use of metric data

In practice, the contrast metric naturally exhibited some relation to other frame-dependent statistics, especially the per-pixel contrast that is basically a crude version of the same calculation. However, the correlation between the respective values was not too strong. It is possible to use the bin counts as an additional metric in itself, but it functions perhaps more intelligently as a derived metric. That is, these values are not directly used as data, but only to change the values of other metrics appropriately.

More precisely, the original purpose of shot detection was utilized for this task. It was assumed that, when a large number of pixels moved from one bin to another, the video shot likely changed. This can be reflected by a shift in the values of other metrics around the same point, and thus in the music produced for the shot. Accordingly, each new shot is accompanied by a musical change that would hopefully be highlighted to the listener.

Preliminary experiments indicated that a suitable threshold for a shot change is a shift in 30% of the frame’s pixels, and a reasonable duration for a shot is at least 2 seconds. When these conditions were met, the “change point” frame numbers were captured so that other metrics could be altered around these points. The values, of course, depend significantly on the character and quality of the video in question. In the current setting they were derived by repeatedly generating music with different settings for the same sample of video data. Using shorter shot lengths or lower thresholds for bin counts resulted in a rapid increase in the number of detected shots, and thus in the loss of perceived emphasis made on a particular shot.

The exact modification applied to a given metric can also be chosen in many ways. It makes sense to "boost" the values immediately at the start of the shot, emphasizing the change, before allowing them to return to the original metric-defined ranges. However, more noticeable results were achieved by increasing or decreasing all of the shot's values at once, shifting them by a fraction of the whole value range. This can be expressed as follows:

$$Br^{*(k)} = Br^{(k)} \pm \max_{1 \le i \le N} Br^{(i)} \cdot 0.1, \quad c_j \le k < c_{j+1}, \tag{3.6}$$

where c_j and c_{j+1} are the frame numbers corresponding to two successive change points, i.e. shot boundaries. The direction of the shift can be chosen randomly, depending on whether the original values are low or high enough: this may also be necessary to prevent the new values from exceeding the original data's extreme values. Also, only even-numbered or odd-numbered shots can be modified to avoid highlighting every one of them.

If emphasizing the entire shot is unproductive, and instead only the first few values of each shot should be altered, the same shifting mechanism can be used with an additional linear or exponential term:

$$Br^{*(k)} = Br^{(k)} \pm \max_{1 \le i \le N} Br^{(i)} \cdot 0.1 \cdot \max\left(0, 1 - \frac{k - c_j}{20}\right); \tag{3.7}$$

$$Br^{*(k)} = Br^{(k)} \pm \max_{1 \le i \le N} Br^{(i)} \cdot 0.1 \cdot \exp\left(\frac{c_j - k}{20}\right). \tag{3.8}$$

The constant 20 in these expressions can be adjusted to control the rate at which the magnitude of the shift decays. It may also be made dependent on the shot length, $c_{j+1} - c_j$.

3.1.3. Metrics from other sources

To complement the video features listed above, an audio-based metric was introduced. This metric refers to sound energy, or the intensity of the sound at a given moment in time. Since audio is generally recorded as a change in sound pressure, this intensity is conceptually different from analogous video metrics: a static video image still represents a certain input, while a fixed-value audio sample corresponds to silence, a lack of input.

Moreover, standard audio encoding forms cannot be used in the calculations directly: an audio sample differs in length from a video frame, because digital audio is sampled at far higher rates than typical video framerates. Since previous metrics provide one value per frame, it would be desirable to reduce the amplitudes to the same rate. This can be done by averaging the values belonging to the same video frame:

$$Amp^{(k)} = \frac{1}{S}\sum_{i=1}^{S}|A_i|, \tag{3.9}$$

where S is the ratio between the audio sampling rate and the video framerate, or, alternatively, the ratio between the total number of audio samples and video frames. Absolute values are taken to account for formats that represent amplitudes (A_i) as signed integers, where deviations from zero in either direction are direct equivalents of the sound's volume. This is the convention adopted in most representations of raw audio via pulse code modulation (PCM), except for the 8-bit variety where the values are traditionally unsigned. Such samples must be first converted to signed values by subtracting a constant from each of them.

The amplitude metric is a direct equivalent of brightness in the audio domain, and accordingly another useful indicator of note loudness. However, similarly to other audio-based data it is only meaningful as long as the audio was recorded together with the video. The newly generated audio may replace the original, hopefully retaining the same peaks and oscillations, or be mixed with it so that both sources are audible. If, instead, the audio track used for metric extraction is added later, no meaningful correlations can be expected between the corresponding video and audio segments, unless the track was itself automatically generated.

Finally, a "joint" metric can be derived from the values of all other metrics considered above. From a data perspective, this is a redundant operation since the new metric will fully depend on already existing values. However, in music generation it is often convenient to have an extra audio source (and thus data source), and the dependency on other audio sources is not at all easily perceived. A simple technique to give equal weight to all the original metrics is to rescale them into values between 0 and 1, dividing each number by the maximum of the corresponding metric, and adding up the resulting fractions for each frame. In the case of four metrics, for example, the "joint" sum would then be a fractional value distributed between 0 and 4.

The selection of metrics to be elicited from a given video can be constrained by performance issues. While the analysis of audio samples and single-frame pixel arrays tends to be fast, comparisons between two frames and other operations needed in contrast calculations can be rather slow in certain implementations. Thus, the general strategy of computing all the metrics at once can be reconsidered by choosing only some of them, as long as they are assumed to suffice for further music generation. Table 1 provides a brief summary of the metrics covered in this section and their interpretations.

Metric	Notation	Value range	Meaning
Brightness	Br	$[0;255\cdot 3\cdot wh]$	Sum of all pixel values
Contrast	Ct	$[0;255\cdot 3\cdot wh]$	Differences of pixel values between consecutive frames
Contrast, with bins	Ctb	$[0;wh]$	Same, but calculated as the differences between bin counts
Amplitude	Amp	$[0;2^{bit\_depth-1}]$	Frame-based average of audio intensities
Joint	N/A	$[0;M]$	Average of M other metrics (may be weighted)

Table 1. Metrics extracted from user-provided videos.

Each of the metrics described above thus produces a list of values, one per video frame. As covered in the following section, an existing tool is then utilized to generate audio out of the metric data. For this purpose, the values are passed to the tool, just like other sources of input data, within a single file.

3.2. Data2Music tool

Once the intended video features were extracted, a need arose of a system where these data could be imported and utilized. The tool should be able to use these data for music creation.

In this study, an existing tool called “Data2Music” will serve the purpose of music generation. It is a Web-based auralization tool that accepts data in a specific JSON format with timestamps along with variable names and numeric values. The Data2Music tool (D2M) has been already run on different datasets collected from multiple sources such as physical activity tracking and weather information [Middleton et al., 2018]. A certain effort was applied to studying the tool’s input data, features and functions, which are briefly described in the following subsection. The Musicalgorithms tool, by the same contributor, is an online implementation with closely related functionality [Musicalgorithms].

Although other auralization tools exist, they are still relatively rare and the D2M tool has been chosen as the most flexible and the easiest to analyze. In particular, the synthesis toolkit in C++ [STK] is an example of a more extensive, yet also less directly applicable instrument. While the lack of a GUI in its samples can be remedied with a suitable environment, the decision to focus on raw audio waveforms is a complication for music generation purposes. The most similar tool in terms of functionality and purpose (sonification of data) is perhaps the recent TwoTone application [TwoTone]. However, it is so recent that it was not available at the beginning of the thesis work.

3.2.1. Input and functionality

Input data for the tool can be gathered from any source, as long as they contain some variables that change over time. A minimal example of the JSON schema follows:

```
{"timestamp": 1521679315892, "feature": "contrast",
"value": 0, "parameters": {"system": "track1"}}
```

The `feature` field (together with the track name) serves to integrate all the values of the same metric under one name, with the actual data stored in the `value` field. The `timestamp` values allow the interpretation and visualization of the data as a time series (the format corresponds to Unix timestamps with milliseconds). When the data are not periodic or their actual timestamps are inconvenient for the auralization process, it is appropriate to replace these with artificial values. Notably, the input data must be presented as independent JSON objects, one per line, without wrapping them into an array.

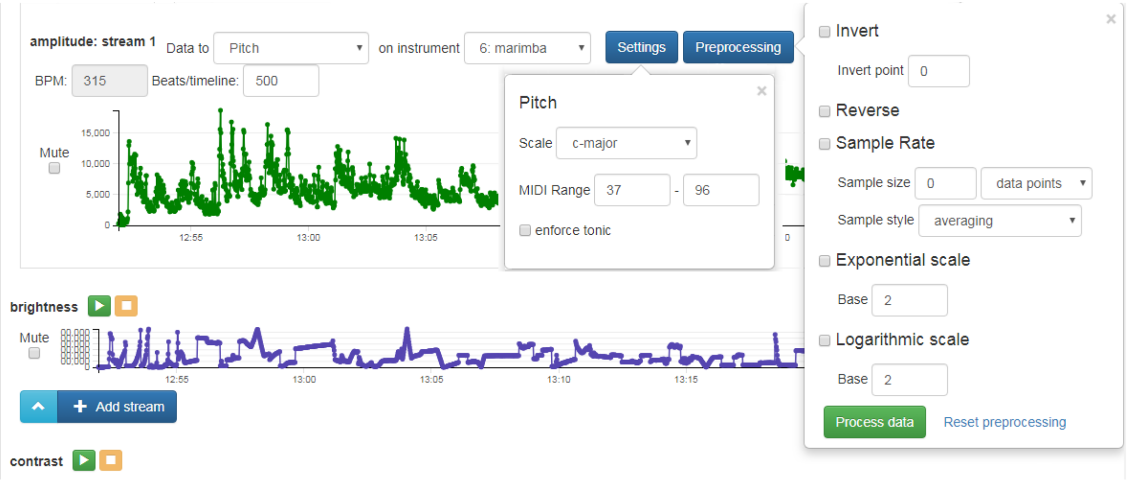
Once the data are imported into the system, they are stored in a database for later retrieval. The database used for this purpose is CouchDB [CouchDB]. The given input data are displayed to the user in the form of a visualization stream using the JavaScript visualization library D3 [D3], respecting the density of timestamps found in the data.

Auralization of the data happens primarily through 8 different musical instruments, namely the piano, guitar, cello, flute, vibraphone, marimba, strings (violin) and drums. Each individual metric has its own stream mapped to one of the available instruments. However, the user can select the same instrument many times for different features in the same dataset, assigning a separate stream to each. Likewise the same feature can have multiple instruments mapped against it, with various settings applied to each instrument if necessary.

Apart from the auralization, the tool also creates a visual preview of both the main data source for each metric and all streams associated with it, plotting data points against time in different colours. This feature provides a convenient overview of the data currently in use and aids in a few simple preprocessing operations.

The burden of additional processing of the input data is shifted from the user to the interface. This can be very convenient in smoothing out certain unwanted features of the data, which can be perfectly natural for their initial source but inconvenient for the creation of enjoyable, harmonious music. The user is thus given access to several potent functions that can reshape the data before the auralization process.

The available preprocessing operations include reverse, which reorders the timestamps of the data points from last to first, and invert, which converts high values into low and vice versa (these can be seen as horizontal and vertical inversions of the data stream, respectively). Threshold values can be set to filter out particularly small or large values in the data, which is also elegantly accomplished by zooming into the dataset’s visualization window. Effectively, the data range becomes only the part visible in the interface at any given moment. This is often desirable to avoid sudden, abrupt changes in the music resulting from extremely low or high values occurring in the data. Horizontal thresholding may be used in a similar fashion, cutting out the first or last values of the stream. Furthermore, values can be subjected to logarithmic or exponential scaling or subsampled, making use of a value range's median, minimum or maximum values. The whole range of the options is demonstrated in Figure 3. In particular, the "amplitude" name in the top-left corner comes from the `feature` field of the input.



amplitude: stream 1 Data to Pitch on instrument 6: marimba

BPM: 315 Beats/timeline: 500

Pitch

Scale c-major

MIDI Range 37 - 96

□ enforce tonic

□ Invert

Invert point 0

□ Reverse

□ Sample Rate

Sample size 0 data points

Sample style averaging

□ Exponential scale

Base 2

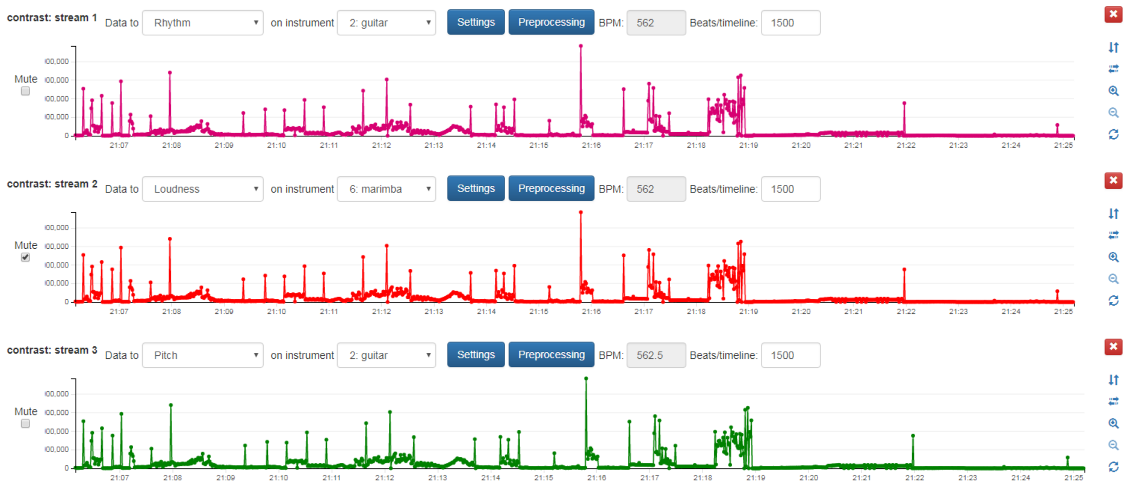
□ Logarithmic scale

Base 2

Figure 3. Preprocessing facilities of the D2M tool.

The generated music does not fully correspond to the original data: in fact, the sampling mechanism only chooses data points around the timestamps where a new note is required, so that both highly dense and sparse data can still be processed. If too many data values are ignored during the sampling, the application can be used to reduce the sample rate of the dataset. This may also aid in denoising the data.

In addition to these preprocessing capabilities, the tool offers a number of functions directly related to auralization and to its MIDI output. Apart from general settings applied to every data sequence at once, every stream has its own controls that may be used to override general settings. The most important of these are undoubtedly the track used to generate sounds (i.e. the respective metric), its instrument and tempo, expressed in terms of beats per minute or beats per track; the entire track can also be configured to have an arbitrary duration, which affects the resulting tempo. Preprocessing operations can likewise be applied to individual streams. Figure 4 shows the tool's representation of three different streams at once, mapped to different properties and instruments.



contrast: stream 1 Data to Rhythm on instrument 2: guitar BPM: 562 Beats/timeline: 1500 Mute □

contrast: stream 2 Data to Loudness on instrument 6: marimba BPM: 562 Beats/timeline: 1500 Mute ☑

contrast: stream 3 Data to Pitch on instrument 2: guitar BPM: 562.5 Beats/timeline: 1500 Mute □

Figure 4. Data representation in the D2M tool.

For most instruments, it is possible to choose whether the data values are responsible for the volume, height (pitch), duration or rhythmic pattern of the resulting notes. Only one of these can be chosen at a time, but the desired effect can usually be obtained by reusing the same instrument and data multiple times. The notes are further adjusted in accordance with a scale, such as the C major scale, which has a profound effect on the resulting music. To create a smoother musical sequence from discrete data points, the tool is also capable of grouping adjacent notes into chords, with a random process initiating from some base note to create more or less complex chords.

Preliminary versions of the generated music can be immediately tested and compared to each other, since the tool supports MIDI playback. This can be especially convenient when experimenting with a new dataset and trying to select suitable instruments to auralize it. Individual streams or combinations of them can be played simultaneously: this encourages the user to try out various arrangements of settings and instruments, while retaining the freedom to filter out unsuitable musical sequences.

Finally, the note sequence generated from the provided data and settings can be exported as a MIDI file. Importantly, the current session can also be saved and imported at a later point, restoring the data and the settings used within the session. This is often useful, since finding the data in the database and restoring the desired settings from defaults can be somewhat tedious. Recent sessions are displayed to the user on the front page of the tool.

3.2.2. Limitations

Practical use of the tool indicated that its strong support for MIDI conversion was nonetheless somewhat unintuitive and difficult to leverage for an inexperienced user. When more than one metric was implemented, there was a problem in figuring out how to represent distinct video features in one JSON file, so that all of them would impact the generated audio in the originally desired way. Values generated from these different metrics had to be combined so that each metric corresponded to an audio "track" (i.e. a lone component of the actual soundtrack). Later, a single track could be mapped to a single instrument, a number of them, or muted if it failed to produce suitable music.

The flexibility of the tool's settings and their significant impact on the result should be undoubtedly appreciated. It is beneficial to have the ability to generate vastly different audio tracks, even from the same data source, and complexity is a natural price to pay for this opportunity. However, this complexity may be somewhat mitigated by a certain simplification of the settings. In particular, various "presets" could be implemented, containing settings that generally result in adequate soundtracks. Novice users could then rely on these presets at first, changing one or two settings from their base values to experiment with the results, while more experienced users would be still free to use the entire settings palette.

Moreover, the generation attempts that a user can perform with the current tool are somewhat hindered by the extra tasks required before and even after the auralization process. The tool cannot extract metrics from the user's videos, and this task must be performed manually. While the result may be exported as a MIDI file, this is not a very common format, and the user would likely want to convert it to a more widespread alternative. Since the generated audio track is supposed to replace the original track of the video, the original video should be edited as well. This is not supported at present.

In an ideal scenario, most of this processing should be automatic. The user should be able to insert a video file into the tool's interface and choose some presets or settings for the kind of processing desired for the video. The tool should then use this video to export metric data to JSON and employ this JSON collection to generate the MIDI file. Likewise the MIDI file could be then converted into a more common format using existing codecs, such as MP3. Finally, this MP3 should be synchronized with the video, replacing its original audio track, and the video shall be transferred to a user-specified destination. The current tool provides a diverse set of features to convert JSON data into MIDI and offers significant control over the process, but it does not establish any "bridges" to the actions commonly needed before and after this conversion.

For the convenience of the end user, it was considered important to automate this whole process. The focus of this thesis accordingly shifted from adding new features on top of the existing tool to enhancing it with an additional interface. The goal was to create a light desktop application that would perform the necessary "preprocessing" and "postprocessing" tasks, at the same time falling back on the functionality of the current tool for the auralization process itself.

With the new tool, users should be able to accomplish all of the aforementioned tasks from selecting a video clip to getting the new video with the generated audio already embedded in it. Unlike the current tool, the new application should also be able to present results and previously created video files in some proper ordering. It should also be possible to save or export the same video with different audio settings. For instance, one music video may be generated using only the brightness values of a video and another may be exported using the brightness and also the contrast values.

To avoid unnecessary implementation work, it was also crucial to select appropriate software tools to perform some of the required conversions. Extracting metrics from source videos is not performed by the D2M tool, nor by any other common application, so this operation had to be implemented separately. However, the transformation from MIDI to a different audio format, as well as the replacement of a video file's soundtrack, are of course reliably implemented in existing tools. Those were utilized to further automate the operations of the D2M tool, with the unfortunate constraint that the end user must also acquire and configure these additional instruments.

A more detailed overview of the proposed tool's implementation can be found further in Chapter 4. This also discusses the additional software used in the conversion process and the extent to which D2M's interfaces have been utilized.

4. Music generation tool

As discussed in Section 3.2.2 above, the D2M tool provides an extensive array of functions to support soundtrack generation from a wide variety of input data. However, these functions can be somewhat complex in actual use and fail to support several common actions in the "video to music" conversion process, such as extracting data from the video and refitting it with a new audio track. A need was observed for a new interface that would integrate these missing functions with D2M's own routines. The entire functionality of the tool is fairly extensive, so priority was given to the most commonly used and easily realized features. The current chapter covers the implementation of a desktop application that allows the user to select a video for conversion. The application generates music from it and applies it to the chosen video.

The application has three primary functions. Firstly, it analyzes videos and extracts multiple metrics from them. Secondly, it uses the metric data together with user-specified settings to generate a MIDI file. Finally, the MIDI is converted into a more common audio format and inserted into the video originally chosen for analysis. The new tool thus provides a complete "pipeline" of operations, from extracting video metrics all the way to generating and reattaching the new audio. Figure 5 illustrates this sequence, with circles corresponding to processes and arrows indicating the data flow between them, according to the conventions for data flow diagrams outlined by DeMarco [1978].

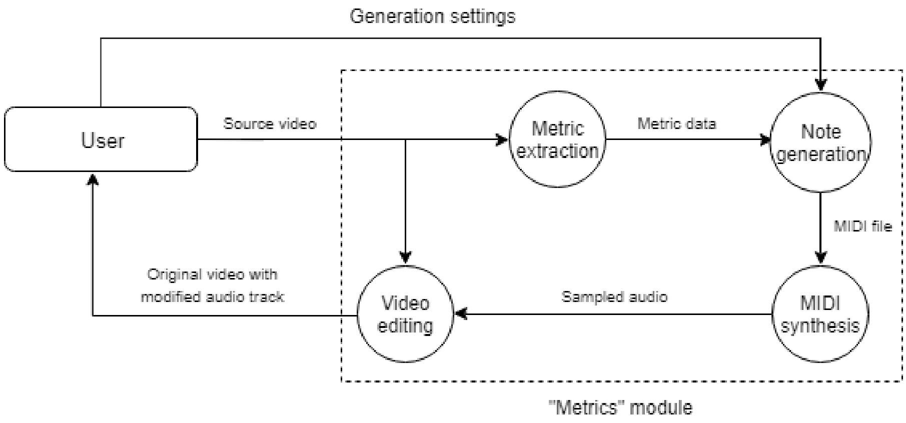


Figure 5. Processing stages for a user-provided video.

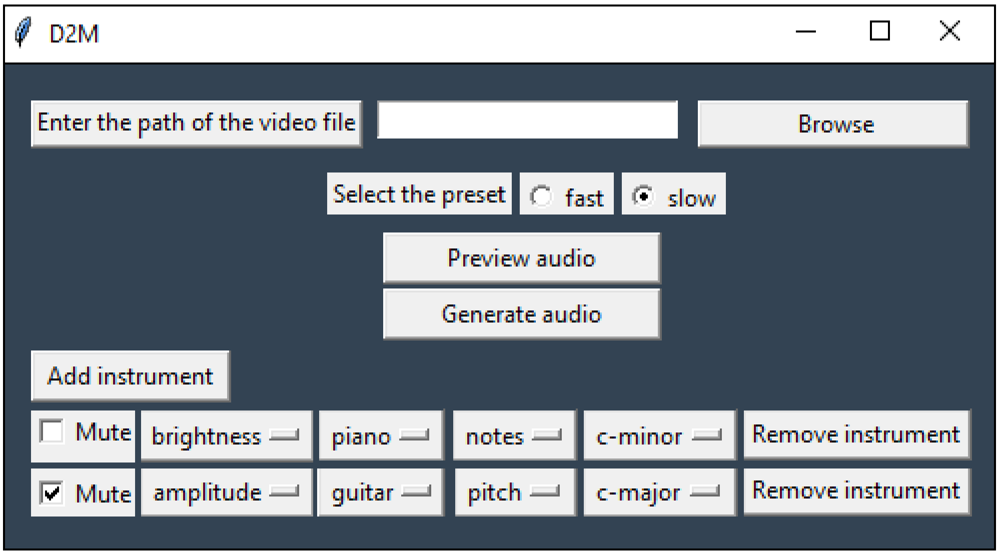
The application is written in Python. This programming language was chosen for several reasons: its general ease, wide compatibility with various libraries, availability of importable modules for performing side tasks, as well as smooth integration between different sources. All the necessary libraries could be easily installed in the same project and virtual environment, ruling out possible name conflicts and "leftovers" that could cause difficulties with later projects. The graphical interface of the tool was also implemented as a simple Python module, without facing the complications found in other languages when transitioning from console applications.

Feature extraction has been facilitated by the freely available OpenCV library [OpenCV]. While its computer vision features are extensive enough in themselves, the advantages most relevant to the current study were the ability to read various video formats and the convenient interfaces to multiple programming languages. However, the library's functions have been used exclusively to parse video files and interpret individual frames as pixel arrays. Further calculations needed to derive metrics from pixel values were implemented within this thesis.

Video analysis is by far the most time-consuming operation, scaling both with the dimensions of the video and its frame count (i.e. duration and frame rate). The speed is also affected by implementation choices: the Python realization of OpenCV, in spite of the helper modules used by it, is still not as fast as the ones in other programming languages. At the beginning of the development work, the library was only available for the 2.7 version of Python. Several weeks later, however, a noticeably more efficient release was produced for the 3.6 version, thus further establishing the choice of language used.

Once the metrics are extracted, they are stored in temporary JSON files for later use. If a video is detected to have such previously saved materials related to it, the same video need not be analyzed again.

A single track is available by default, so that at least some sound can always be produced by the tool. However, the user is able to freely add and remove tracks. While the total number of metrics is restricted to five, it may be needed to use the same data source for multiple instruments. This is achieved by creating additional tracks and mapping them to the same metric. Figure 6 demonstrates the layout of controls for multiple tracks.



Select the preset ○ fast ● slow

□ Mute | brightness | piano | notes | c-minor

☑ Mute | amplitude | guitar | pitch | c-major

Figure 6. Data representation in the new tool.

For compactness, the controls for each track occupy one row in the interface, so that creation and deletion of tracks results simply in appearing or disappearing rows of elements. These controls include the track's original metric, instrument, musical scale, control parameter (the value adjusted by the data, such as the notes' height, duration, volume or rhythm) and a few postprocessing options, such as shifts added to the common beat pattern. A track may also be muted, so its sound will not be included in the resulting MIDI file. Muting a track is a convenient way to test how well the sequence sounds without it, while keeping its settings readily available instead of deleting and recreating the track from scratch.

Since the process of MIDI generation tends to involve multiple trials before a suitable result is obtained, the application features a preview function. This operation constructs the MIDI file with the current settings and plays it via an external player without attaching the audio track to the original video. The relation between the sound and the video is thus missing, but it is often not needed for the first few trials. Initially, the user can focus just on creating a harmonious musical track without the overhead of inserting it in the video every time. This insertion would also cause a copy of the video to be created, which can be fairly large depending on the length and resolution of the file. Such copies are highly redundant, since the audio takes up only a small fraction of the entire filesize, unless a downscaled copy of the video is first produced for preview purposes.

The settings specified by the user are integrated in a JSON file that is used, together with the video data, for MIDI creation. The process happens via D2M's functionality in a similar fashion: in the original tool, implemented as a Web application, the data were uploaded to a database and joined with a settings file to provide input for the generation process. In the current application, a settings file of the exact same structure is utilized together with the extracted data. D2M's mechanisms are thus operated in the same manner; the only difference is that no database is used to record the data. The settings file has the following JSON structure:

```
{"source": "track1", "bpm": "150", "duration": "96",
"instruments": ["piano", "guitar", "cello", "flute",
"vibraphone", "marimba", "strings", "drums"],"variables":
{
    "amplitude":{"muted": true, "streams": {"amplitude:
    stream 1":
    {
        "muted": false, "bpm": "150", "bpt": "1500",
        "instrument": "1", "dataTo": "notes", "settings":
        {
            "controls": "notes", "scale": "c-minor",
            "enforceTonic": false, "midiRangeMin": 20,
            "midiRangeMax": 83
        },
        "thresholds":
        {
            "horizontal":
            {
                "on": false, "filterType": "outer", "max":
                131616321, "min": 0, "filterOption":
                "filter", "filterValue": ""
            },
            "vertical":
            {
                "on": false, "filterType": "outer", "max":
                "Sun May 13 2018 13:27:53", "min": "Sun May
                13 2018 12:13:28", "filterOption":
                "filter", "filterValue": ""
            }
        },
        "y_range": {"max": "131616321.00", "min": "0.00"}
    }}}
},
"variableFilters": {"amplitude": true}}
```

The first few top-level components of this schema are global properties specifying the source track, the intended tempo (in beats per minute) and duration of the audio track. The `variables` object has a greater impact on the generation settings: it features one component for every metric in the input data, such as `amplitude` in the example, and describes its track-specific settings in detail. Apart from the tempo settings, it determines the musical instrument "voicing" the data (the `instrument` field), what property of notes the data are responsible for (the `controls` field), the musical scale of the notes (the `scale` field) and the range of MIDI notes to be used for the current track (the `midiRangeMin` and `midiRangeMax` fields, which are also dependent on the instrument).

Similarly, the `thresholds` component is a reflection of D2M's horizontal and vertical filtering options. The `horizontal` part specifies the lower and upper bound of the filter as the actual values of the data, in this case equal to the actual limits in the `y_range` field (meaning that no filtering is applied). The `vertical` part uses timestamps as filter parameters, excluding data points that do not fit into the specified timeframe.

Once an appropriate MIDI track has been generated, it should be attached to the video, replacing its original audio track. This operation is also performed with the aid of external software. Since video formats typically require PCM audio data, whether in raw or compressed form, the note-based MIDI representation must first be synthesized into a PCM arrangement. This can be achieved with several freely available tools; for Windows, a review of the existing options suggested the VLC media player as the simplest solution [VLC]. Recent versions of VLC support an audio codec called FluidSynth, which can be used to perform the MIDI conversion. The codec also adds playback support to VLC, meaning the program can additionally be used in the preview mode of the tool. FluidSynth is also available as a standalone application; like other freeware tools of the same nature, however, it is much easier to install on Linux or similar systems than on Windows.

Importantly, a sound font file is required for any operations related to MIDI playback or conversion. Sound fonts contain different instrument palettes compatible with the MIDI specification and define their exact sounds. This means that the same MIDI sequence, even with all other factors being equal, can sound differently depending on the sound font used by the playback tool (just like the same text can have various appearances depending on the font used for it). A freely available sound font was used for all music generation tasks in this thesis, so that no inconsistencies arose from the usage of different fonts.

Once a PCM representation of the audio is obtained, typically in MP3 format, it can directly replace the original audio track of the video. This operation is best performed by FFmpeg, a freeware tool for various video manipulations [FFmpeg]. In the current work, FFmpeg is in fact utilized for two purposes: apart from replacing the audio track, it also extracts the original track so that audio-based metrics can be calculated from sound data.

Given the structure of the application and the tasks performed by it, the implementation was confined to two modules. The “main” module rendered the user interface of the tool, collected various inputs from the user and displayed status updates about the conversion sequence. The “metrics” module conducted the actual analysis of the selected video and handled the necessary calls to other helper tools. In Figure 5, this module is outlined by a dashed rectangle, taking inputs and returning outputs to the “main” module outside of the rectangle.

In particular, it first called the FFmpeg tool to extract the original audio track, if any, from the video, so that audio-based metrics could be computed on its basis. The video handle for the file chosen by the user was also stored and reused, so that the video could be repeatedly opened and examined frame by frame. Once the necessary metrics were calculated, the corresponding inputs could be passed to the D2M tool for MIDI generation.

Upon obtaining the MIDI file, the “metrics” module passed it to the VLC player to produce a PCM representation of the data. The MP3 codec was used as a widely supported and compact format of audio storage, with a fixed bitrate value of 128 kbps, although other options are also perfectly possible. The resulting file was again returned to FFmpeg, this time to be added to the initial video in place of its original audio track.

This operation concludes the processing sequence initiated by the application. In practice, a copy of the video with the new audio track is created as a temporary file and the user is prompted to save the result. If a location is chosen, the temporary file is renamed, or effectively moved to the specified location; if not, the file is lost.

Thus, the “main” module is mostly dedicated to user interface tasks, while the functions of the “metrics” module are split between processing video data and establishing an ordered sequence of calls to various helper applications, providing them with appropriate intermediate products to arrive at the final result. Figure 7 demonstrates the actual workflow performed by the new music generation tool, which can be seen as a clarification of the “pipeline” presented earlier.

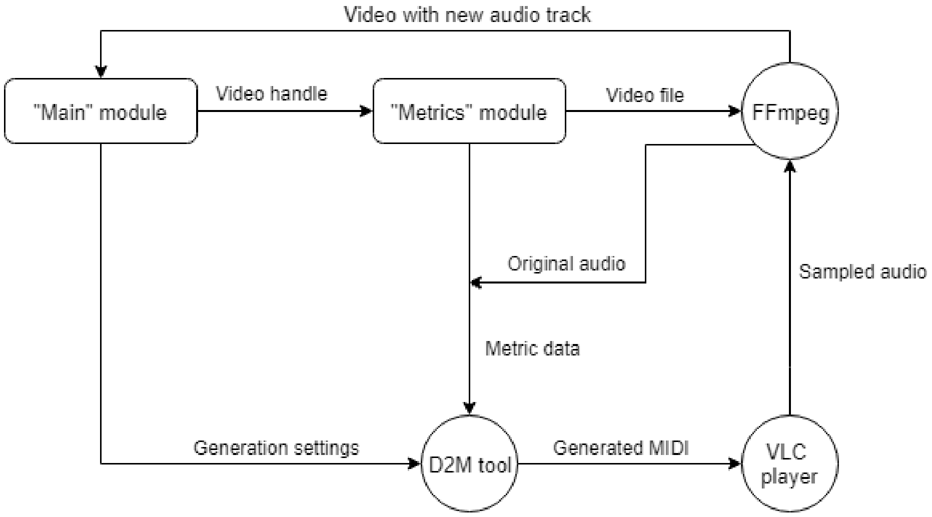


Figure 7. Detailed workflow of the application.

5. Music evaluation

As an art form, music is appreciated subjectively by every individual, and it appears virtually impossible to develop any objective criteria of its quality. At the same time, since the auralization processes described above approach music from an algorithmic perspective, not as a work of art, it becomes desirable to find a way to evaluate the performance of these algorithms, and thus potentially to improve them. The only tool that seemed appropriate for this task, at least in some approximate form, was a user survey.

The shape of the questionnaire became evident quite soon: potential respondents should be provided with a selection of videos, including their original audio tracks, and "replacement" videos where the same visual material is accompanied by generated audio. Comparing these would allow respondents to evaluate the generated track both on its own and with respect to the original. To elicit more information, several generated tracks could be offered for every original video. The differences between these generated tracks would hopefully be reflected in the participants' respective evaluations. Ultimately, the survey questions were to attempt to determine the quality of the generated audio and decide whether it was more appropriate than the original.

One immediate issue with this setting is that it had to be very limited in terms of the number of videos and their lengths. Increasing the number of videos and variations per video would have rapidly increased the number of survey questions, which tends to result in fewer responses received. The same adverse effect would have occurred upon increasing the duration of the videos, and thus of all their variations.

Unfortunately, the video materials used in the survey were themselves liable to subjective interpretation. Just like the generated music, the actual videos, their subject, quality and style could be perceived very differently by different respondents. Even if specifically asked to focus on the music in their evaluations, they could still exhibit a certain bias depending on their preferences for the videos in question. By introducing a fixed set of video and audio tracks, the survey attempted to isolate at least one of these two subjective components while allowing the other to vary to some extent. In this setting, individual liking or disliking expressed towards a given video would hopefully be consistent across the evaluation and allow the music-related preferences to stand out.

5.1. Video materials

Early experiments with the auralization tool used a number of unrelated videos, as well as various combinations of metrics and MIDI settings aimed exclusively at producing "reasonable", coherent music. This was often a difficult task, since a sequence of notes generated from data points is likely less harmonious than a musical piece composed by a human. Moreover, even when a generated sequence is smooth enough, it is still not guaranteed to match the video it was drawn from. However, there were also occasions when the new audio track provided a remarkably good fit with the video: it could still be perceived as artificial, perhaps, but nonetheless a close match to the graphics.

The survey implemented in this thesis was oriented exactly at this “best case” scenario. Instead of selecting purely random videos and arbitrary settings for music generation, survey samples were chosen so as to provide maximum diversity and quality of the music. The objective was to select several distinct videos, iterate through various configurations and parameters to produce diverse, adequately sounding audio tracks for each, and allow respondents to evaluate a number of these “prime examples”. The audio samples, thus, were subjectively better than randomly generated tracks, but still perfectly achievable given the initial videos and a suitable configuration of settings (in this case, the MIDI generation settings of the D2M tool).

Ultimately, three videos were selected for the survey. The first was more of a “vlog” sample, filmed during a holiday trip at a popular destination. The second was a “nature” or “travel” video, shot from a car travelling along a highway. The third was a slideshow automatically generated from a set of static pictures: in this case, the original audio track was discarded since it had been merely chosen from a set of samples available to the slideshow tool. Figure 8 provides screenshots from each of the videos.

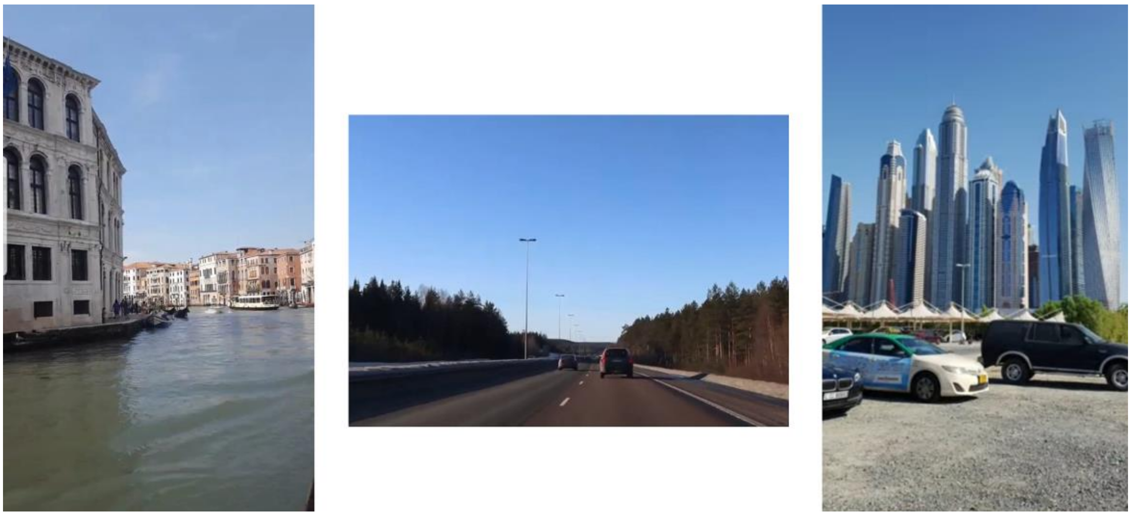


Figure 8. Video samples used in the survey.

This selection of material can appear rather restricted and questionable. However, the main motive behind this choice was the range of use cases to which automated music generation may be applicable. There is little use in considering professionally edited videos, where sound and music effects have already been edited by experts and automatic generation is unlikely to produce better results. It is also pointless to analyze music videos, where the entire content is shaped around a given piece of music and replacing it would be senseless. Likewise public events, where speeches are given, or teaching sessions should certainly retain their original audio content.

On the other hand, in a “travel” video it is reasonable to replace the original sound, such as the monotonous hum of the car engine or gusts of wind, with an appropriate audio track. For a vlog, depending on how much it has been edited, there may be unwanted noise from the streets or transport, which is also worth cutting out. Finally, in the case of a slideshow there can be no “original” audio in the first place; besides, the generated audio does not suffer from the same licensing issues as an imported track, and may in fact compete with it in quality. With rich images used for the slideshow, the product can be quite similar to an actual video and require comparable audio support.

To increase the variety available to survey respondents and possibly identify the distinguishing features of good audio, three distinct variations were provided alongside each video with differing parameters. The participants of the survey were not aware of these differences between variations, apart from the ones they could perceive by listening to the music. The general premise was that each video had a slow, moderate and fast audio track, which were set apart by the BPM value (beats per minute) used in the generation process. Similar values were used for the corresponding presets in all three videos. A brief listing of each variation’s essential settings is found in Table 2. To simplify referencing, video variations in Table 2 and the following discussions will be numbered continuously from 1 to 9 (variations 1-3 belong to the first video and so on).

Video and variation	**Instruments**	**Tempo (beats per minute)**	**Other notes**
Original video 1 (holiday “vlog”, 30 seconds, original sound)			
Variation 1	Piano, guitar, cello, flute, strings	90	Same settings for all variations except the changes in tempo; C minor scale
Variation 2		170	
Variation 3		250	
Original video 2 (car trip, 53 seconds, original sound)			
Variation 4	Piano, flute, strings	100	Continuous strings
Variation 5	Piano, flute, strings	170	Blues piano scale; continuous strings
Variation 6	Guitar, cello, flute, vibraphone, drums	240	Different scales on every instrument
Original video 3 (slideshow of static images, 57 seconds, no sound)			
Variation 7	Strings, piano	110	Continuous strings; C minor scale
Variation 8	Guitar, vibraphone, drums	170	C major scale on the guitar
Variation 9	Piano, cello, flute, marimba	250	All instruments in C major

Table 2. Settings applied to video variations.

As marked in the last column of Table 2, the resulting variations were also edited by altering other settings, with the aim of getting as much diversity as possible. In particular, a given variation was provided with fewer or more instruments, various note scales for each instrument and distinct helper beat patterns. Some postprocessing tricks were likewise applied in order to enhance the music obtained from the video data.

In particular, a small change to the note sequences produced a continuous, uninterrupted sound of string instruments, which is more natural for them than a discrete beat-like sequence often generated by the D2M tool. More precisely, the MIDI playlist files output by the tool as an intermediate step were modified and the note length of the corresponding instruments extended, so that no pause would remain between successive notes of the strings. By default, the note length was shorter than the beat length governed by the tempo.

The impact of this change can be observed in Figure 9, which represents the waveforms of the three audio variations actually generated for the slideshow video of the survey. The beat pattern clearly visible in the second track is partly compensated by the faster tempo in the third one and by the “extension” of strings in the first.

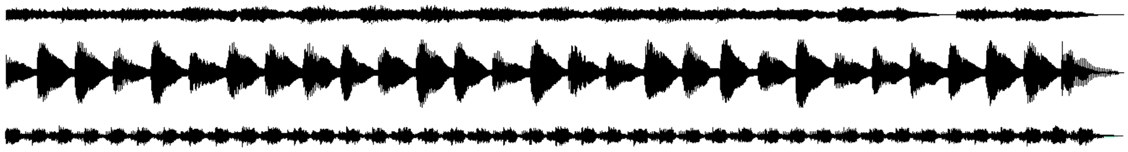


Figure 9. Effects caused by different music generation settings.

The resulting audio tracks all appeared substantially different from each other. The original settings used to generate them were recorded and stored together with the audio files themselves, so that the defining features of each track could be recalled later. The intent of the survey was mostly to observe which tracks were found particularly successful and fitting by the respondents, and thus to determine which features may be responsible for that.

The duration of the survey videos was between 30 and 55 seconds. This was an important criterion in the selection of the material. Videos had to be long enough to exhibit some diversity, both in the graphical setting and in the audio domain, and at the same time short enough to be quickly watched and compared. With three variations per each original video, extending a single video by 1 minute would have increased the survey completion time by at least 4 minutes. This would have likely had a negative impact on the number of responses collected.

5.2. Survey implementation

For each video and each variation, the survey respondents were asked to evaluate the generated audio track in three ways: whether the audio was enjoyable to listen to, in itself, whether it formed a good match with the underlying video, and whether it was a better addition than the original audio track. (In the case of the slideshow, where no original audio was available, this third question was omitted for all variations.)

Responses to these questions were collected on a 7-point scale, with the statements for the extreme values and the midpoint adjusted depending on the question. The precise texts of the questions and possible answer choices are listed in Table 3. For example, the comparison between the newly generated and the original audio suggests that the new audio is “much better than the original audio” (value of 7), “much worse than the original audio” (value of 1) or “as good as the original audio” (value of 4). Seven distinct values appeared to be a reasonable level of granularity: five values would have given only two possible ways of characterizing agreement or disagreement, while nine or ten would have resulted in very small differences between adjacent values.

<table>
<tr><th>Question text</th><th colspan="6">Possible choices</th></tr>
<tr><td>How pleasant is the audio in Variation N to listen to?</td><td colspan="2">1 (very uncomfortable and annoying)</td><td colspan="2">2–6</td><td colspan="2">7 (very pleasant to the ear)</td></tr>
<tr><td>Is the audio in Variation N better than the original AUDIO?</td><td colspan="2">1 (the original audio is much better)</td><td colspan="2">2–6</td><td colspan="2">7 (the new audio is much better)</td></tr>
<tr><td>How well does the audio for Variation N suit the original VIDEO?</td><td colspan="2">1 (doesn’t suit at all)</td><td colspan="2">2–6</td><td colspan="2">7 (suits perfectly)</td></tr>
<tr><td>If you answered less than 5 to the last question, what exactly did not fit?</td><td>Speed (tempo) of the music</td><td colspan="2">Instrument selection</td><td colspan="2">Synchronization with changes in the video</td><td>Something else</td></tr>
</table>

Table 3. Evaluation questions of the survey.

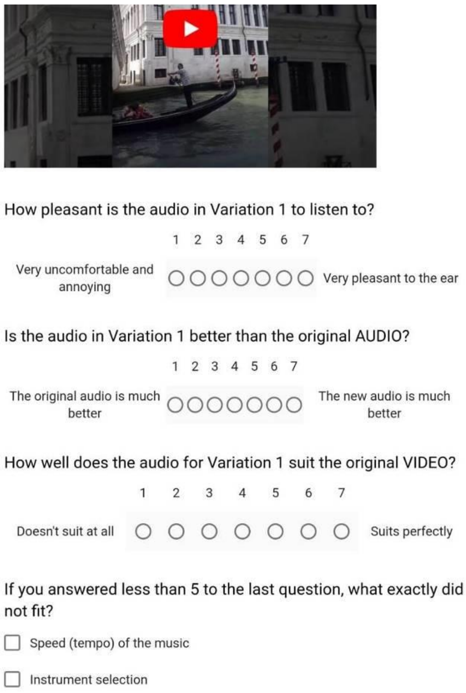
Each variation was also accompanied by a number of checkboxes, allowing the respondent to specify why the corresponding audio track was a poor fit to the video. Respondents were instructed to answer this question only if the corresponding response was less than 5, but this restriction could not be enforced by the survey system itself. Possible reasons included poor choice of instruments, lack of synchronization with events in the video, tempo and a generic “other” option. Unlike the previous questions, multiple answers were accepted in this case. This question was meant to elicit likely reasons behind unsuccessful cases of music generation that could be further addressed. An open-ended field for general comments about the survey, placed at the very end of the questionnaire, also served the same purpose.

Apart from these, the first section of the survey collected several simple background details about the respondents, specifically their age group (in 10-year bands), gender and region of origin (the exact values are found in Table 4). These questions thus preceded the actual video evaluations. It was hoped that this information would exhibit some further relationships between the respondents and their evaluation of the music. For example, the audio generated by the tool is mostly based on instruments and scales belonging to the European musical tradition. Representatives of other cultures, depending on their background, may be unfamiliar with such music and perceive it differently than those who have long lived in the "Western" musical environment. Of course, a limited analysis of such relationships could only be possible with a sufficient quantity of survey responses.

Question text	**Possible choices**							
Age group	15 years or under	16-25 years	26-35 years	36-45 years	46-55 years	56-65 years	Over 65 years	
Gender	Male		Female			Other		
Place of origin	North America	South America	Europe	Africa	West Asia	South Asia	East Asia	Other

Table 4. Demographic questions of the survey.

The survey was implemented as a questionnaire hosted on Google Forms, accessible only via a specialized link to avoid receiving uninvited responses. The survey link was then distributed to potential respondents. This link was identical in every case, i.e. there was no unique identifier attached to it that would help identify individual responses. Figure 10 demonstrates a fragment of one of the survey's pages.



How pleasant is the audio in Variation 1 to listen to?

	1	2	3	4	5	6	7	
Very uncomfortable and annoying	○	○	○	○	○	○	○	Very pleasant to the ear

Is the audio in Variation 1 better than the original AUDIO?

	1	2	3	4	5	6	7	
The original audio is much better	○	○	○	○	○	○	○	The new audio is much better

How well does the audio for Variation 1 suit the original VIDEO?

	1	2	3	4	5	6	7	
Doesn't suit at all	○	○	○	○	○	○	○	Suits perfectly

If you answered less than 5 to the last question, what exactly did not fit?

□ Speed (tempo) of the music

□ Instrument selection

Figure 10. Questions asked about individual variations.

Video material for the survey was publicly hosted on a YouTube channel. All the videos were ultimately selected from the author's own collections, to avoid potential copyright issues, and could therefore be published without restricting access. Each video included its number and variation in the title, to avoid confusion for the respondents, and featured a brief description of the scenes shown in the video. However, the actual musical settings and features used in the generation of each variation were not disclosed.

While links to the videos could be inserted into the questionnaire, it was also possible to embed the videos themselves into the survey. Respondents would still be able to navigate to the actual video if they liked, but by default a smaller version of the video, without fullscreen capacity, appeared directly next to the questions. This served to remove the need of switching between different tabs or applications, which could be rather distracting for mobile users, and the preview was still large enough to capture the graphical details of the scene and associate them with the audio playing simultaneously.

Participants were invited to take part in the survey by distributing its link in several Facebook groups, as well as through private messages. Responses were accepted for a total of 20 days, giving potential respondents enough time to go through the survey, although not all of them received the link at the exact same time. Due to the way posts are displayed and removed from view on Facebook, as well as the usage of different channels of communication, it is hard to evaluate the precise number of users that were able to see the invitation (with respect to the total size of the targeted groups). A rough estimate is 200 potential respondents.

5.3. Results

The survey attracted a total of 101 responses, or about half of the distributed invitations. Most of the answers included some data for every survey question, with just 5 respondents skipping some of the questions. Missing values in those observations were handled afterwards for those methods that could not operate with missing data. Additionally, one response contained no answers except for the first three demographic questions; this entry was excluded from the analysis.

Most participants were split between two age groups, which had been formed in 10-year intervals: 38 respondents were between 16 and 25 years old, while 58 were aged 26-35. There were additionally two responses from the 36-45 group and two from the "under 16" group. 75 respondents were male and 24 female, with one missing value; the "other" gender option was not chosen by any of the participants.

Geographically, survey respondents were primarily split between South Asia (50 replies) and Europe (29 replies). Other less represented areas were East Asia (10), North America (2), Africa and West Asia (1 each). Seven respondents chose the "other" option for this question.

The most important information obtained from the survey is undoubtedly the grade distribution for the variations, i.e. the actual ratings given by the respondents. Table 5 contains a complete listing of the grade counts for every variation, including only the first question about the enjoyability of the audio in itself. The word “pleasantness” will be used further in the text to be consistent with the phrasing of the question. The table lists the number of times each grade was given, as well as the total number, mean and median value of the grades. The lowest scores in each category are highlighted in italics and the highest in bold.

Source	1	2	3	4	5	6	7	Total	Average	Median
Var. 1	8	7	13	14	22	14	22	100	4.650	5
Var. 2	5	12	6	19	24	15	19	100	4.660	5
Var. 3	9	7	14	14	22	11	22	99	4.556	5
Var. 4	5	5	12	15	28	16	18	99	4.778	5
Var. 5	*10*	*7*	*11*	*15*	*26*	*10*	*20*	*99*	*4.515*	*5*
Var. 6	13	6	12	9	18	17	25	100	4.640	5
Var. 7	**4**	**3**	**7**	**17**	**24**	**17**	**28**	**100**	**5.170**	**5**
Var. 8	6	10	8	16	20	21	19	100	4.730	5
Var. 9	11	7	7	19	20	18	18	100	4.560	5

Table 5. Grade distribution for pleasantness of the audio.

The highest grades were obtained by variation 7, a slow-paced track for the slideshow video (mean value 5.17). The lowest grades belonged to variation 5, a medium-paced track for the nature video (mean value 4.515). Since 4 was the middle value of the scale for all questions, roughly corresponding to “neither better nor worse” or “neither good nor bad”, the overall quality ratings are slightly better than average.

A brief correlation analysis of these ratings shows a moderate level of agreement between the evaluations of individual variations, which means that individual respondents were likely to give consistently higher or lower grades to every variation. The entire set of correlation coefficients is provided in Table 6, with each value indicating the correlation between the grades of the variations in the corresponding row and column.

Since the basic Pearson correlation is not entirely appropriate, given the discrete (albeit ordinal) nature of the variables, polychoric correlations were evaluated instead. These regard the discrete values as “cutoff points” of originally continuous variables that the correlation would normally apply to [Drasgow, 1986]. The range of such coefficients is the same as for the Pearson coefficient, from -1 to 1. Missing values in the data were replaced with the median of their respective variables.

Var.	1	2	3	4	5	6	7	8	9
1	1.00	0.58	0.54	0.57	0.55	0.64	0.52	0.62	0.58
2	0.58	1.00	0.65	0.50	0.68	0.62	0.52	0.58	0.61
3	0.54	0.65	1.00	0.47	0.62	0.61	0.56	0.58	0.63
4	0.57	0.50	0.47	1.00	0.51	0.50	0.63	0.49	0.53
5	0.55	0.68	0.62	0.51	1.00	0.66	0.50	0.57	0.62
6	0.64	0.62	0.61	0.50	0.66	1.00	0.54	0.63	0.69
7	0.52	0.52	0.56	0.63	0.50	0.54	1.00	0.48	0.54
8	0.62	0.58	0.58	0.49	0.57	0.63	0.48	1.00	0.61
9	0.58	0.61	0.63	0.53	0.62	0.69	0.54	0.61	1.00

Table 6. Correlation coefficients between the pleasantness grades.

The next question of the survey dealt with the differences between the original and the newly generated audio track. Table 7 lists the corresponding grades, using the same conventions as before. Since the slideshow video did not feature an audio track of its own, only the six variations of the other two videos are included.

Source	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Total**	**Average**	**Median**
Var. 1	14	11	1	12	19	16	27	100	4.670	5
Var. 2	9	11	6	13	22	17	22	100	4.670	5
Var. 3	11	11	8	17	13	18	22	100	4.520	5
Var. 4	**11**	**4**	**9**	**9**	**23**	**14**	**29**	**99**	**4.889**	**5**
Var. 5	*12*	*12*	*8*	*12*	*15*	*18*	*22*	*99*	*4.495*	*5*
Var. 6	14	10	7	12	17	18	22	100	4.500	5

Table 7. Grade distribution for audio quality with respect to the original audio.

In the absence of the third video's variations, the fourth variation now received the highest grades, while the fifth remained the worst rated. Grades are generally consistent with the previous question's values. As in the previous case, the second video has attracted somewhat more diversity in responses than the first.

Finally, the third evaluation-related question of the survey asked about the alignment between the new audio track and the original video material. According to the original question posed in the survey, "alignment" here refers to the suitability of the audio to the video: the matches in the temporal structure of the two components, their moods, styles and their synchronization with each other. The original audio track, conversely, was not considered in this question. The answers are summarized in Table 8, again employing the same highlights.

Source	1	2	3	4	5	6	7	Total	Average	Median
Var. 1	11	12	12	11	21	12	21	100	4.390	5
Var. 2	10	11	8	19	21	13	16	98	4.357	5
Var. 3	12	10	9	13	22	12	22	100	4.470	5
Var. 4	9	9	7	17	21	14	22	99	4.636	5
Var. 5	*15*	*9*	*11*	*17*	*21*	*7*	*19*	*99*	*4.182*	*4*
Var. 6	15	7	6	17	20	16	19	100	4.440	5
Var. 7	**4**	**4**	**8**	**15**	**27**	**14**	**28**	**100**	**5.110**	**5**
Var. 8	10	7	13	11	14	24	21	100	4.680	5
Var. 9	15	4	10	17	15	16	22	99	4.505	5

Table 8. Grade distribution for audio alignment with the original video.

In accordance with the previous results, variation 7 was rated significantly higher than any other, while variation 5 was still ranked the worst. Figure 11 represents the same data in a more condensed graphical form.

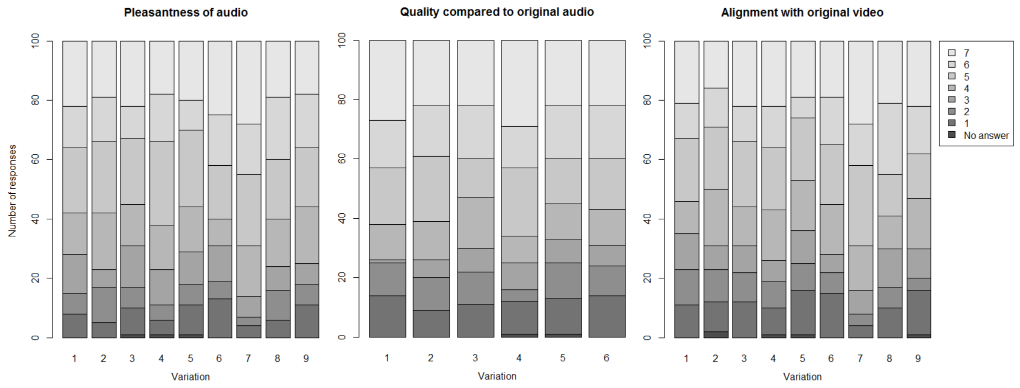


Figure 11. Quantitative responses separated by grade and variation.

The success of the seventh variation is easily explainable: it appears to be a fitting combination of a slow tempo, perhaps aptly corresponding to the transitions in the slideshow, and a harmonious alignment of just two instruments resulting in something vaguely similar to a sonata. The piano was restricted to discrete sounds, while string chords were artificially extended to produce an uninterrupted sound pattern without rapid transitions. Especially the ending of the fragment could have resembled the conclusion of a movement in a chamber music piece. However, this arrangement was the result of many experiments and is not easily attainable for every source video.

The drawbacks of the fifth variation are not so evident. Its lower grades are relatively close to those of variations 3 and 6, which are both fast-paced musical tracks. Details in Table 2 indicate that it used a typical instrument palette without percussion represented by the vibraphone or marimba; these instruments generally seemed to dominate others and create a distracting beat pattern. A possible explanation, at least for variations 5 and 6, may be their usage of unconventional musical scales for some instruments and thus an even more prominent lack of harmony and order in the music. Apart from these considerations, it is also likely that the car trip shown in the corresponding video was associated with a calm and relaxed setting, not an intense "action" scene as the faster variations might have suggested.

One division deliberately introduced and maintained between the variations was a distinction between tempo, namely slow, moderate, and fast tracks. This was done in an attempt to evaluate tempo as a quality factor of the music. The results apparently indicate that slow tracks were rated somewhat higher than fast ones. However, this claim is not supported by statistical testing: when comparing the same evaluation category and the mean grades of each video's variations via the one-tailed, two-sampled Welch's t-test, the only statistically significant difference was observed between variations 7 and 8 (with p-values of 0.0360 for pleasance and 0.0481 for alignment), as well as variations 7 and 9 (with p-values 0.0078 and 0.0117 respectively).

An attempt was also made to determine the reasons for poor matches between the original video and the generated audio track. This was performed by asking the respondents to choose their reasons for giving a low grade to each variation's perceived match. Similar questions could have been asked about the other two components evaluated in the survey, but it was considered that good and bad matches would be more concretely identifiable, as opposed to the more abstract concept of music quality. Introducing additional questions would also risk receiving fewer complete responses.

Table 9 presents the distribution of factors named by respondents with respect to every variation. Note that the total counts do not agree with the number of responses any more, since these questions were optional and multiple selections were allowed (two or three options were chosen by most survey participants). The dependency on the low grade of the corresponding variation was not enforced, apart from the question text.

Source	**Tempo**	**Instruments**	**Synchronization**	**Other**
Var. 1	29	39	19	14
Var. 2	35	33	20	12
Var. 3	40	24	21	13
Var. 4	17	32	17	14
Var. 5	30	34	17	17
Var. 6	27	32	20	11
Var. 7	13	28	13	12
Var. 8	20	33	17	9
Var. 9	28	26	16	14
Total	239	281	160	116

Table 9. Factors mentioned as detracting to the alignment between audio and video.

The results show that the number of complaints about tempo is notably higher for faster tracks, suggesting that music should preferably have a slower pace. This pattern only breaks for variations 5 and 6, where a similar number of mentions is nonetheless observed. However, a bigger issue is evidently posed by the instruments chosen for the tracks. These tended to include most of the options from the original 8-instrument palette, so apparently a completely different set of instruments must be considered.

Finally, the survey included a field intended for free-form comments at the very end. 25 responses were collected from this field. Some explicitly mentioned that the music did not align well with the videos, while a few expressed their satisfaction with the survey. One participant noted that the pitch and other settings of the music should gradually change over time, instead of being constant throughout the entire variation. Another comment was made that alignment should be searched between places depicted in the video and the music, not so much between the instruments or other settings.

This idea was also extended in another, more detailed suggestion, which reasoned that at least the tempo should be related to the setting of the video. If the video depicts a calm and soothing entity, such as a body of water, the tempo should accordingly be slow. Ideally, the entire character of the music should change dynamically depending on the changing circumstances in the video.

According to this respondent, the choice of instruments is also crucial in creating a unified listening experience. However, some instruments were not rendered according to their natural sounding: for example, string instruments are played continuously, with a number of bowing techniques to further modify their sound, which cannot be smoothly replicated by auralizing data. Percussion instruments apparently represent an exception to this rule, since they usually produce discrete beats and their precise arrangement is not so significant in the musical composition. The tempo and beat patterns of these instruments could potentially also be altered in response to changes in the video.

6. Discussion

Results of the survey conducted to evaluate the generated music indicate that it may accompany suitable videos, such as vlogs and slideshows. The primary issues identified with the music are apparently its tempo and instrumental arrangement, and more generally a mismatch between the nature of the music and the scenes shown in the video. At the same time, a number of simple modifications to the resulting sound can reduce the appearance of discrete beats and provide a smoother rendition to some instruments, which makes the music more natural.

It can be almost universally claimed that a bigger pool of respondents and a wider target audience would have yielded more trustworthy results. However, more concrete advancements are also clearly feasible. Now that the initial results are available, a more concrete evaluation of the same kind could be performed, focusing on the specific features that made generated music attractive to listeners. In particular, the tempo of the audio tracks should be reduced, especially if the corresponding videos are conducive to it, and perhaps a completely different selection of instruments is required for the arrangements.

Of course, just because a particular video variation has been well received does not mean that its properties are always useful for automatic music generation. However, the number of videos required to identify such properties with any precision is far too large to fit into an initial survey. At best, once such "candidate" properties are discovered, a more detailed study can be launched into them.

Instead of providing fixed musical tracks for evaluation, it would be interesting to let the users themselves generate a number of appropriate tracks (using the same tool), select the ones they are particularly satisfied with and study the characteristics of these "favourites". The question then becomes not whether the generation process can result in good music, but whether an individual can generate music that they appreciate, which is surely the practical aspect of the problem.

The generation of music has been based on multiple metrics, as many as four or five depending on the user's preferences, but all of them were fairly simple frame-based characteristics. On the one hand, the development of more sophisticated metrics would have diverted the attention from other objectives of this thesis, and the videos used for the survey were unlikely to have a meaningful division into shots or scenes. On the other hand, the search for more abstract video properties, even when applied to the selected videos, could have yielded better audio tracks: if not directly, these metrics could have been used to provide a "flavouring" for the underlying frame-specific data.

As a result, the connections between the metrics currently used and the resulting music remain hard to observe. Exact mappings between low-level characteristics and musical chords could not be expected, of course, but the music still requires persistent editing by a human listener before it establishes an acceptable match with the video. Results of the survey suggest that even such well-crafted alignments can be given average ratings.

In general, music generated from video data can be likely improved by structuring it in accordance to the principles of musical theory. The idea of using notes played by well-known instruments is already a step forward from directly mapping data values to audio amplitudes, the basic components of PCM audio. However, the resulting notes can still be adjusted and grouped into more natural chords after the initial conversion. Perhaps even the entire musical composition can be reshaped depending on the user's preferences, taking on a particular mood or musical style. A side effect of this process will be the loss of exact correspondence between music and video data, but since an aesthetical concern is pursued rather than a scientific one, such discrepancies may be justified.

In principle, these improvements are already somewhat similar to the work of a human composer discussed earlier in the context of movie soundtrack production. However, it appears that videos of such scope represent a relatively small population. Most videos ordinarily filmed and shared by everyday users do not have the same structural complexity as actual movies. Quite often they lack even well-defined shots and scenes, aiming only to create uninterrupted footage of an event or phenomenon. For such unstructured videos, the advantages provided by high-level metrics will likely be insignificant.

Improvements in music quality are also attainable through a more extensive analysis of the underlying video material. It is expected that modern methods of video analysis, including a wide array of machine learning techniques, can greatly enhance the understanding of a given video's semantic content. Knowing who or what is depicted in the video, or perhaps evaluating the entire video's mood, type and purpose, can likewise shape the music based on the video in question. These are exactly the high-level video metrics mentioned in the literature: while less tangible and informative by themselves, they represent an important complement to more primitive and abundant video features.

Given the complexity of discovering such metrics, it is also feasible to delegate some of this burden to the actual user of a music generation solution. They will likely be able to easily determine the desired mood of the music and the nature of the video they are about to process. These details, together with a few other high-level settings, can greatly aid the generation process in producing relevant music. In particular, the mood setting can impact the choice of some instruments over less preferable ones or alter the tempo of the music. Similarly, knowing the type of the video helps in shaping the overall structure of the music, the rate and suddenness of transitions inside it, and so on.

These changes can eventually be integrated into the new music generation tool developed within this thesis. However, at present there are more immediate ways to improve it. Although the centralization of the “video to music” conversion has been enhanced with the introduction of a dedicated application, it is still unable to perform all the required operations on its own. The generation routines of D2M are now present as a dependency in the new tool, and there is a reliance on other products that the potential user must still install and configure. As a result, the original workflow of uploading a data file to D2M and retrieving a MIDI track has been made only slightly more fluent.

The tool can be expanded to include more functionality already performed by the D2M application. In particular, the preprocessing operations may definitely provide more flexibility in data manipulations to the user, even if a direct translation of metric data into music has been sufficient so far. Some of the filtering operations were more necessary in D2M’s context to separate multiple datasets within the same timeline, whereas the new tool only imports a single video’s data at any given moment. However, filtering by threshold values and rescaling should be equally useful in either tool.

The ideal end result of further implementation is apparently the entire transfer of the D2M tool into the new desktop application, or perhaps vice versa. It is not clear where the integrated product should reside. On the one hand, adding the video analysis routines to the D2M application, currently realized as a Web tool, would represent less development work, especially since the existing interface can be expanded to accommodate this additional operation. Moreover, any extra tools such as a MIDI synthesizer can be installed on a single server, instead of burdening every potential user. On the other hand, bringing D2M’s full functionality to the desktop application would remove the need to upload videos to the server, which is wasteful given how little data are ultimately extracted from them. Hopefully the computation of metrics will also take less time as the constant improvements in hardware continue.

If the tool is not realized as a server solution after all, some research will be needed to determine other “helper” applications that could assist with MIDI synthesis and audio replacement. At present the user is constrained to use a single solution, although the application can be easily extended to interact with other tools: in most cases only a valid command-line call is required, featuring the name of the tool and the settings it may potentially take. This approach, however, still does not permit all the required functionality to be integrated in a single software product.

7. Conclusions

The current thesis has dealt with the subject of generating musical patterns on the basis of video features. More precisely, it has attempted to solve the problem of providing background music to existing videos, e.g. filmed by an everyday smartphone user and passed to a software tool, in cases where the original audio is missing or of such poor quality as to have little practical value.

The thesis has focused on finding answers to three research questions: what video features can be used fruitfully to generate music on their basis, or "convert" videos to music; how exactly video data can be mapped to musical elements and form a complete musical composition; and whether the music generated in this manner is enjoyable to the listener and true to the original video.

Firstly, the current work has used relatively simple frame-based metrics, such as brightness, contrast and audio amplitude, as the defining features used for music generation. These were fairly easy to compute and provided a large volume of data for further processing. Evaluation results indicate that these metrics established a basic level of alignment between the video and the generated music, but were not successful in conveying the potential mood or style of the video. More sophisticated metrics, perhaps shot-based or scene-based features, are clearly needed to perform this function.

Secondly, the conversion from video data to music requires a selection of metrics drawn from a particular video and a procedure to transform them into musical elements. This general statement permits great flexibility for concrete solutions of the problem: the assortment of metrics found in video data is fairly broad and a number of algorithms may be used to relate them to building blocks of an audio track. This thesis has effectively focused on extracting frame-specific characteristics of the video and assigning them to musical notes played by different instruments, using an existing software tool and further extending it to allow finer adjustments to the conversion process. However, other implementations relying on different video metrics and different mappings from metrics to music could address the same research question equally well.

The decision to generate the music instead of a different approach, such as finding a suitable existing audio track, has been encouraged by the existence of a relevant software product and by the relative lack of development in this field. While the concept of soundtrack recommendation has received extensive treatment in the literature, soundtrack generation, an admittedly more complex task, only features in a handful of studies. A careful inspection of earlier research was also needed to identify appropriate metrics to be drawn from videos.

New interfaces have been added to the existing software tool to facilitate interactions with the generation mechanism developed earlier. While the tool supported generation of MIDI files from arbitrary input data, based on a selection of preprocessing techniques and user-specified settings, all other operations were left to the user of the tool. The thesis has implemented additional modules responsible for extracting numeric data from original videos, converting the resulting MIDI tracks into PCM formats, and restoring the generated music in place of the original audio track. As a result, the user may now utilize a single application that performs the entire music creation process at once, albeit with the use of several extra third-party tools needed for various audio conversion tasks.

Thirdly, a user survey has been performed and roughly a hundred of responses analyzed in order to evaluate the quality of generated music. Respondents were offered a selection of three original videos, each accompanied by three distinct audio variations created using different settings, and asked to evaluate the quality of the audio and of its alignment with the original video.

Results of the survey suggest that the audio quality was slightly above average, but there were numerous complaints about the selection of instruments and tempo utilized in the music generation process. More generally, the setting and mood of the videos were not adequately reflected by the music, resulting in poor alignment between the two. This is explained by the lack of high-level video features employed in the generation and the absence of an in-depth semantic analysis of the source videos.

The most typical outcome of the music generation task, as implemented in this thesis, is that a large sequence of primitive musical elements can be produced to closely match several elementary features of the original video. However, these features are not the ones directly observed and perceived by the viewer; instead, certain less tangible concepts define the nature of the video and thus the music that would be appropriate for it. A more successful generation tool would have to perform better at capturing and utilizing these concepts.

While automating the generation process entirely is an ambitious research goal, leaving enough room for human intervention can also be quite desirable. Ideally, a human “composer” could review the output of the generation tool and apply changes to particular notes, chords and variations, which the tool could potentially learn from. Even if such specialized input is not available, an ordinary user’s creative streak may be put to good use. The user may first be asked to specify the appropriate mood, style and technique for the current music generation task. After the sequence is complete, separate fragments of the music can be likewise manually reviewed, stored for later use or “recomposed” in a different way if the user finds them unsatisfactory. In this way, a relatively simple and natural form of user input, similar to the interactions arising when editing texts, images or videos, can significantly improve the generation process.

References

[Chen et al., 2004] Hsuan-Wei Chen, Jin-Hau Kuo, Wei-Ta Chu, and Ja-Ling Wu. 2004. Action movies segmentation and summarization based on tempo analysis. In: *Proceedings of the 6th ACM SIGMM international workshop on multimedia information retrieval (MIR'04)*, 251-258.

[CouchDB] Apache CouchDB. URL http://couchdb.apache.org [Accessed 18th May, 2019]

[D3] D3.js – Data-Driven Documents. URL https://d3js.org [Accessed 18th May, 2019]

[DeMarco, 1978] Tom DeMarco. *Structured analysis and system specification*. Yourdon Press, 1978.

[Drasgow, 1986] Fritz Drasgow. Polychoric and polyserial correlations. In: S. Kotz and N.L. Johnson (eds.), *Encyclopedia of Statistical Sciences*, vol. 7. New York, Wiley, 1986, 68-74.

[Feng et al., 2010] Jiashi Feng, Bingbing Ni, and Schuicheng Yan. 2010. Auto-generation of professional background music for home-made videos. In: *Proceedings of the 2nd international conference on Internet multimedia computing and service (ICIMCS'10)*, 15-18.

[Foote et al., 2002] Jonathan Foote, Matthew Cooper, and Andreas Girgensohn. 2002. Creating music videos using automatic media analysis. In: *Proceedings of the 10th ACM international conference on multimedia (MM'02)*, 553-560.

[FFmpeg] FFmpeg. URL https://ffmpeg.org [Accessed 18th May, 2019]

[Hananoi et al., 2016] Shunsuke Hananoi, Kazuki Muraoka, and Yasushi Kiyoki. 2016. A music composition system with time-series data for sound design in next-generation sonification environment. In: *2016 International Electronics Symposium (IES)*, 380-384.

[Hanjalic and Xu, 2005] Alan Hanjalic and Li-Qun Xu. 2005. Affective video content representation and modeling. In: *IEEE Transactions on Multimedia*, 7 (1), 143-154.

[Hua et al., 2004] Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. 2004. Automatic music video generation based on temporal pattern analysis. In: *Proceedings of the 12th ACM international conference on multimedia (MM'04)*, 472-475.

[Kang, 2003] Hang-Bong Kang. 2003. Affective content detection using HMMs. In: *Proceedings of the 11th ACM international conference on multimedia (MM'03)*, 259-262.

[Kim and André, 2004] Sunjung Kim and Elisabeth André. 2004. Composing affective music with a generate and sense approach. In: *Proceedings of Flairs 2004 – Special Track on AI and Music, AAAI Press*.

[Kuo et al., 2013] Fang-Fei Kuo, Man-Kwan Shan, and Suh-Yin Lee. 2013. Background music recommendation for video based on multimodal latent semantic analysis. In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6.

[Larochelle and Bengio, 2008] Hugo Larochelle and Yoshua Bengio. 2008. Classification using discriminative restricted Boltzmann machines. In: *Proceedings of the 25th international conference on machine learning (ICML'08)*, 536-543.

[Legaspi et al., 2007] Roberto Legaspi, Yuya Hashimoto, Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao. 2007. Music compositional intelligence with an affective flavor. In: *Proceedings of the 12th international conference on intelligent user interfaces (IUI'07)*, 216-224.

[Liao et al., 2009] Chao Liao, Patricia P. Wang, and Yimin Zhang. 2009. Mining association patterns between music and video clips in professional MTV. In: *Proceedings of the 15th international multimedia modeling conference on advances in multimedia modeling (MMM'09)*, 401-412.

[Lienhart, 1998] Rainer Lienhart. 1998. Comparison of automatic shot boundary detection algorithms. In: *Proc. SPIE 3656, Storage and Retrieval for Image and Video Databases VII.*

[Lin et al., 2017] Jen-Chun Lin, Wen-Li Wei, James Yang, Hsin-Min Wang, and Hong-Yuan Mark Liao. 2017. Automatic music video generation based on simultaneous soundtrack recommendation and video editing. In: *Proceedings of the 25th ACM international conference on multimedia (MM'17)*, 519-527.

[Loke et al., 2006] Mei Hwan Loke, Ee Ping Ong, Weisi Lin, Zhongkang Lu, and Susu Yao. 2006. Comparison of video quality metrics on multimedia videos. In: *2006 International Conference on Image Processing*, 457-460.

[Mendi et al., 2011] Engin Mendi, Coskun Bayrak, and Mariofanna Milanova. 2011. A video quality metric based on frame differencing. In: *2011 IEEE International Conference on Information and Automation*, 829-832.

[Middleton et al., 2018] Jonathan Middleton, Jaakko Hakulinen, Katariina Tiitinen, Juho Hella, Tuuli Keskinen, Pertti Huuskonen, Juhani Linna, Markku Turunen, Mounia Ziat, and Roope Raisamo. 2018. Sonification of musical characteristics: a path guided by user engagement. In: *24th International Conference on Auditory Display (ICAD 2018)*, 35-41.

[Musicalgorithms] Music Algorithms. URL http://musicalgorithms.org/3.2 [Accessed 16th May, 2019]

[OpenCV] OpenCV. URL https://opencv.org [Accessed 18th May, 2019]

[O'Sullivan et al., 2017] Mark O'Sullivan, Bruno Srbinovski, Andriy Temko, Emanuel Popovici, and Hugh McCarthy. 2017. V2Hz: music composition from wind turbine energy using a finite-state machine. In: *2017 28th Irish Signals and Systems Conference (ISSC)*, 1-6.

[Shah et al., 2014] Rajiv Ratn Shah, Yi Yu, and Roger Zimmermann. 2014. ADVISOR – personalized video soundtrack recommendation by late fusion with heuristic rankings. In: *Proceedings of the 22nd ACM international conference on multimedia (MM'14)*, 607-616.

[STK] The Synthesis Toolkit in C++ (STK). URL https://ccrma.stanford.edu/software/stk [Accessed 16th May, 2019]

[Truong et al., 2003] Ba Tu Truong, Svetha Venkatesh, and Chitra Dorai. 2003. Scene extraction in motion pictures. In: *IEEE Transactions on Circuits and Systems for Video Technology*, 13 (1), 5-15.

[TwoTone] TwoTone Data Sonification. URL https://twotone.io [Accessed 16th May, 2019]

[VLC] VLC: Official site. URL https://www.videolan.org [Accessed 18th May, 2019]

[Wang and Cheong, 2006] Hee Lin Wang and Loong-Fah Cheong. 2006. Affective understanding in film. In: *IEEE Transactions on Circuits and Systems for Video Technology*, 16 (6), 689-704.

[Wang et al., 2005] Jinjun Wang, Changsheng Xu, Engsiong Chng, Lingyu Duan, Kongwah Wan, and Qi Tian. 2005. Automatic generation of personalized music sports video. In: *Proceedings of the 13th ACM international conference on multimedia (MM'05)*, 735-744.

[Xing et al., 2005] Eric P. Xing, Rong Yan, and Alexander G. Hauptmann. 2005. Mining associated text and images with dual-wing harmoniums. In: *Proceedings of the 21st conference on uncertainty in artificial intelligence (UAI'05)*, 633-641.

[Yoon and Lee, 2007] Jong-Chul Yoon and In-Kwon Lee. 2007. Synchronized background music generation for video. In: *Proceedings of the international conference on advances in computer entertainment technology (ACE'07)*, 270-271.

[Yu et al., 2012] Yi Yu, Zhijie Shen, and Roger Zimmermann. 2012. Automatic music soundtrack generation for outdoor videos from contextual sensor information. In: *Proceedings of the 20th ACM international conference on multimedia (MM'12)*, 1377-1378.

[Zhai et al., 2004] Yun Zhai, Zeeshan Rasheed, and Mubarak Shah. 2004. A framework for semantic classification of scenes using finite state machines. In: *Image and Video Retrieval. CIVR 2004. Lecture Notes in Computer Science*, 3115, 279-288.