From Movement to Music, a Computational Creativity Approach

Edgar Meireles
edgar.meireles@tecnico.ulisboa.pt
Instituto Superior Técnico Lisboa, Portugal
January 2021

ABSTRACT

Creativity in itself is a hard to define human only ability. The creative process, is then, something researchers aim to fully define and recreate in artificial ways. In our work we go through the historical and cultural definitions of creativity, and some theories of the creative process.

Computational Creativity is a field of Artificial intelligence that seeks to create a program able of creative thinking, as well as creative acting.

In this document we present an attempt of making a creative system capable of human body motion capturing and music generation based on what is captured. We present our approach to this problem with a cross domain analogy between the visual to the musical domain.

We ended making three different motion to music approaches and evaluate them further along in this thesis.

Questionnaires were made for this evaluation. The results proved to be good as to evaluate if we matched our goals proposed on this thesis. Our system was considered creative and able to generate music by more than 72% of our evaluators. We also confirmed an association between the movement and the music with over 56% of common adjectives when categorizing the movement and the music.

There are an almost infinite different ways to solve the initial problem. Our proposition is nevertheless, in our point of view, an interesting and successful approach.

INTRODUCTION

The work we present belongs to the field of computational creativity. Computational creativity is a field that studies means to make a computer program act in a way that is considered creative, or produce something that can be considered creative. There are many historical events in which humanity shows the capability to be creative and in the 21st century can be an enduring and survival skill [11]. This capability is one of the characteristics that allowed humans to evolve and make artifacts that helped us get on the top of the food chain, improve the way we live, and, is one of the abilities that makes us constantly create new things, as explained by Margaret Boden in [1].

There is not and accepted connection theory between music processing and dance association for human beings. Darwin proposed that music and dance might have evolved for these courtship displays as a species needed to find a way to select a better mate [7].

But there is indeed a connection between both of these art forms. As these are art forms they are generally associated with creativity. As we studied the subject we found interesting to develop a system capable of doing the inverse of this association. Trying fist to process the movement or dance and generate music with it.

This thesis propose a way to develop a creative system able to process videos of people moving and generate music associated with it. Our main problem to solve is so this movement to music process.

Our main goal is for the program to be able to capture what we see into what we listen and both of them to make sense, be considered music and a creative object when combined and perceived by our audience that later evaluate our system.

Document structure

In this paper we first introduce the related work our project was based on. Than we explain the five parts of our project: Video Processing, Feature Extraction, Motion to Music Feature Association, Implementation Merging and Composer. After this we present our evaluation were we talk about how we evaluate our developed system and the results we obtained. Finally We conclude this paper on the last section called Conclussions.

RELATED WORK

A program able to act in a way considered creative is a task that has been keeping many researchers busy for a long time. Since we aim to build a system that can be considered creative, and to make a translation of human body motion to music composition we see three different areas of study worth gaining some knowledge upon: Computational Creativity, Human Body Motion Tracking and Recognition and Computational Music Composition. So, before we can start to develop a solution to our problem, we first need to see what has been made so far in those fields.

Creativity

The word "creativity" comes from the Latin word "creare" which mean "to create". However creativity was only defined later in history. R Keith Sawer in [12] say that in the ancient Greece to be creative meant to have been possessed by a demon, as a divine gift granted by the gods only given to certain selected individuals.

Margaret Boden defines creativity as "the ability to generate novel, and valuable, ideas." [1] This is one of the most accepted definitions by the scientific community. In this case, a valuable idea can be seen as one that is interesting, useful, beautiful or even extremely complex. Basically everything that has some sort of purpose in a field can be seen as valuable. From paintings to algorithms, from complex sculptures to a simple photograph, any idea, physical or not, can be considered to be creative as long as that idea is new. Boden divides novelty into two categories:

Methods

Methods of computer vision to capture human body motion were studied and later inspired our work as we developed our system.

All the work made in this area was then introduced in tensorflow posenet model. This model does automatic human capture in images.

Regarding automatic music composition we found that it would be better to explain what exists since our main focus is music production.

Jose D Fernández and Francisco Vico in [6], summarize many of the possible artificial intelligence approaches to music composition. They say that algorithmic composition can be grouped in four categories:

- **Symbolic Artificial Intelligence**
  Under this group we have rule based systems which have been proven quite effective since they can learn and reason over a set of rules given by experts in the field. An example of music composition with this kind of approach is given in 2008 by Georg Boenn et al. in [2]. Using the melodic composing rules of music, the authors were able to create a system capable of music composition called ANTON. Although this system can produce melodic pieces, it cannot do entire pieces of music.

- **Machine learning (Markov Chains and artificial neural networks)**
  Most machine learning algorithms tend to imitate the input or categorize it based on what type of data the model is trained.

  Music can be viewed as a very complex and sophisticated probability distribution over a sequence of sounds. With this premise, many researchers adapt Markov models and artificial neural networks to be able to learn this probabilistic sequences in order to make new sequences of notes, that later could be called music.

  In 1957 Lejaren Hiller and Leonard Isaacson composed the first known computer made piece of music. This was a string quartet composition made with pseudo random Markov chains [9]. The notes the code generated were later tested with a certain number of rules and they only kept the ones that agreed with all the rules they implemented. This is a basic probabilistic approach to the music composition problem. With the evolution of AI we see another approach to this problem, we see many people trying to make automated music composition using machine learning algorithms.

  In 1992 Hermannn Hild et al. used Artificial Neural Networks in [8] to learn Bach composition rules to make compositions similar to his style.

  Also in 1992 we see another proposition with Artificial Neural Networks [10]. Here the authors propose that recurrent neural networks, a specific type of Artificial Neural Networks, can learn music structures, though be able to compose music with higher quality. So Florian Colombo et al. in [4] used recurrent neural network to learn music structures of Irish melodies. With the structure learning capabilities of this algorithms, the music composed by them has better and more similar structures to when using normal Artificial Neural Networks.

  Long Short-Term Memory, a different type of Artificial Neural networks, is also very used to music composition in [3, 5]. On these papers, the authors claim that although Recurrent Neural Networks can in fact learn music structures, Long Short-Term Memory algorithms have better result in timing and context of music structures.

- **Optimization techniques (evolutionary algorithms)**
  Darwin told us that we evolve as species through generations as well as all other species on the planet. Based on that theory, this type of algorithms were created. Given a population as input, this algorithms creates new generations with the good features of the older generations. These good features are define by a fitness function given by the programmer.

  In music composition we can use these types of algorithms are used to combine a group of musics to make new ones. With a good adjustment of the fitness function we can see good, never seen or listened results, from this approach.

- **Self-similarity**
  The authors say that these techniques are not a form of artificial intelligence. This method consists in using similarities or musical patterns and repeat them to compose. Usually the music composed by these systems are very rough yet they are certainly novel. This is used mostly by composers to create raw material for them to compose on.

In 2017 Joana Teixeira made a system capable of producing music inspired by images. [13] Features were retrieved from an image, and, with a visual to music features association made by the author, the program was able to directly translate what it sees to music.

FEATURE EXTRACTION

Now with the processed video, we then calculate all the features we previously mentioned, using the keyjoints captured on the previous module that have a confidence percentage value over than 0.5:

Velocity (v), can easily calculated with the previous keyjoints coordinates using the following formula for every 2 sequential frames:

$$v = \frac{\sum_{i=1}^{k} \sqrt{(x_{i2} - x_{i1})^2 - (y_{i2} - y_{i1})^2}}{k}$$

Being k the number of keyjoints, xiN is the x coordinate of keyjoint i on the nth frame and yin is the x coordinate of keyjoint i on the nth frame.

Acceleration (a) can be calculated from velocity by the expression:

$$a = v_2 - v_1$$

Being vN, the velocity on the frame N. Frame 1 and 2 are sequential.

Fluidity (f) is, so, calculated from acceleration with a similar formula:

$$f = |a_2 - a_1|$$

Being aN, the acceleration on the frame N. frame 1 and 2 have to be sequential.

Here we assume that a fluid movement is one that has a low acceleration change rate.

The contraction index (ci) is obtained from the area of the bounding box. We use the code of the algorithm 1 to get the silhouette bounding box per frame. An example can be seen in figure 1.

Algorithm 1 Calculate Bounding Box limits

```
maxX ← -inf
maxY ← -inf
minX ← inf
minY ← inf
for all keyjoint i: keyjoints do
    if i[x] > maxX then
        maxX ← i[x]
    end if
    if i[x] < minX then
        minX ← i[x]
    end if
    if i[x] > maxY then
        maxY ← i[y]
    end if
    if i[y] < minY then
        minY ← i[y]
    end if
end for
```

And then we apply the following formula to get the contraction index value:

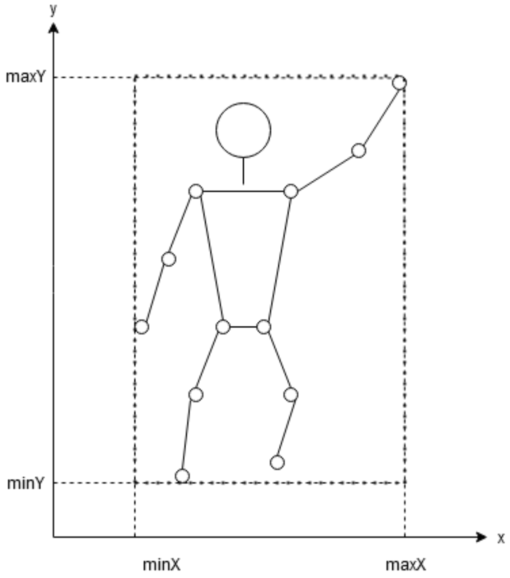


Figure 1. bounding box calculation example

$$ci = (maxX - minX) * (maxY - minY)$$

Quantity of movement (qom) is the next feature we extracted. This one was calculated based on the variation of the contraction index value. It was calculated using the following formula:

$$qom = ci_2 - ci_1$$

Being ciN the contraction index value on the frame N. Also with frame 1 and 2 sequential.

We found useful also to calculate the variation of quantity of motion ($2ndqom$). So we calculate it based on the previous obtained quantity of motion value with the following formula:

$$2ndqom = qom_2 - qom_1$$

Being $qomN$ the contraction index value on the frame N. As always with frame 1 and 2 sequential.

From these features we create an array of the type [v,a,f,ci,qom,2qom] per frame starting of the third frame. This happens since acceleration is calculated with the values of velocity of the current and the previous frame. If N is the total number of frames we have, then we can only get N - 1 acceleration values. And since fluidity is calculated with the same formula but with acceleration values, we only have N - 2 fluidity values.

We also have a second array of the type [vup, aup, fup, vdown, adown, fdown]. We created this as a second type of visual feature extraction.

From the formula, velocity is calculated as the average velocity from a certain number of keyjoints. For the first array we use every eligible keyjoint (confidence value over 0.5) for our velocity calculation.

On the second array we have 6 "new" features. These features come from the analyses of the keyjoints from above the waist and from the waist and below. So vup, aup, fup are velocity, acceleration and fluidity calculated with the keyjoints above the waist. vdown, adown,fdown are so, the velocity, acceleration and fluidity captured from the keyjoints from the waist and below.

As we saw the videos we adjust two threshold values of contraction index and quantity of motion for the program to be able to perceive the emotion transmitted with the captures movement. These values were generated based on our personal perception of emotion transmitted from those movements.

With these values we created a 2 dimension chart from which our program is able to identify emotions per frame. We then count the instances of these emotions and pass to the next module the most common emotion value associated with the video body movement.

After this we also extracted the main colour of the silhouette and the background.

To do so we first needed to see as the first frame per instance what was background and what was foreground, in our project the foreground is the person silhouette.

For the background identification we used the median per pixel to see the most common colour it appears on that pixel. This works since the silhouette is always moving through the whole video so it does not appear on the median color for every pixel.

Now we have the background we remove it from the first frame to obtain the silhouette. To do this we go through every pixel of the background and every corresponding pixel on the first frame and if the RGB values difference is lower than 20 we replace that pixel with the color black.

With these two images we go through every pixel and label it with one of nine colour, seven rainbow colors plus black and white. Then we count those colours and we get the most common colour on each image.

Now for the last visual feature we also wanted to see how much of the background can pop into a user eyes as visual stimulation.

To do this we use the background image we obtained previously and apply a Gaussian filter to blur it, since our peripheral vision also does it.

Now in the blurred background image, colour contrasts may call our eye to it so we apply now a Meijering neuriteness filter to transform our blurred background into a black and white image. The white pixels represent pixels identified as ridges, or in our case a visual stimulation since it represents a contrast of the background.

By counting the number of white pixels and divide them with the total number of pixels, we get a percentage value of how much of that picture is attractive to our eye and can disturb our focus on the main attraction.

We use that value as our final visual feature, we called it background attraction value.

MOTION TO MUSIC FEATURE ASSOCIATION

Now we have these features:

- Array of overall motion [v, a, f, ci, qom, 2qom] per frame
- Array of specific parts motion [vup, aup, fup, vdown, adown, fdown] per frame
- Overall emotion
- Background main colour
- Silhouette main colour
- Background attraction value

An algorithm was created in order to pick three continuous features and one fixed value and, from that, create an array of notes for the composer module to generate midi files.

First of all we decided that every music has two tracks, so we should run the algorithm twice to get our two tracks, each with different visual features.

For each track we assigned a musical instrument. These instruments come from the fourth and fifth features given to us by our previous module.

Based on a personal choice, we associated a colour to an instrument:

- White - Electric Guitar
- Black - Sax
- Blue - Piano
- Green blue - Cello
- Green - Violin
- Yellow - Harp
- Orange - Flute
- Red - Celesta
- Purple - Guitar

We used the silhouette main colour to get the first music track instrument. The background main colour is used then to get the second track instrument.

Now for the main algorithm we start to generate a random number from 0 to 11. This number represents the main note of the musical scale the generated music composes on.

The emotion value defines the type of scale our system composes on:

- Joy - Major scale
- Pleasure - Penthatonic Major scale

- Anger - Chromatic scale

- Sadness - minor scale

Then we generate the scale based on the main note with the type previously defined.

This algorithm starts with the first possible note on the scale and adds the note intervals specific of the scale we want.

Now with the array of notes that belong to the musical scale we can start to generate the music.

Firstly we needed to find a way to discover the rhythm of the music we wanted to generate. In other words, what is the value of beats per minute our music has.

To find the rhythm of the video we observed the fluidity values as a wave. A wave have a periodicity. So the number of frames associated with the periodicity of our wave is the number of frames associated with a beat.

So we decided to apply a Fast Fourier Transform on the derivative of acceleration, in our case the fluidity value, to obtain the Discrete Fourier Transform of our wave.

Extracting the maximum of our Discrete Fourier Transform gives us the most likely periodicity of our wave in number of frames.

To get the beats per minute value (*bpm*) we used the following formula:

$$bpm = \frac{frameRate * 60}{max(dtf)}$$

Being *dtf* the values of our Discrete Fourier Transform.

Now it did not make sense to see our visual data in terms of which frame x happened. So we reduced our feature array by grouping them in groups of the number of frames presented in a beat, then we average all the values presented in that group to get all the features we previously had but now in beats instead of frames.

As mentioned before a note has four features: pitch, duration and intensity and beat.

In our project we defined intensity as a constant value. We used the maximum value possible in midi, 100, because as our time was limited we did not thought of how or which visual features could be "translated" into this intensity value.

Beat is defined as the time the features we analyse occurred.

Pitch is defined as one of the notes presented in our pre-generated scale, and as duration, both are chosen by the algorithms 2, 3 and 4:

These algorithms use three features per beat, and has two threshold arrays.

First it generates a random note in our scale as a first node for our algorithm to start generating on.

Algorithm 2 Note generator

```
notes ← []
noteIdx ← int(random() × size(scale))
timeBeat ← 0
while timeBeat < size(featuresByBeat) do
  beatFeatures ← featuresByBeat[timeBeat]
  transaction ← getTransaction( beatFeatures[1], beatFeatures[2] )

  noteDuration ← getnoteDuration ( beatFeatures[0] )
  if noteDuration < 1 then
    nBeats ← 1
  else
    nBeasts ← noteDuration
  end if
  i ← 0
  while i < nBeats do
    noteIdx ← noteIdx + transaction
    if noteIdx < 0 or > size(scale) then
      noteIdx ← noteIdx - (2 × transaction)
    end if
    note ← scale[noteIdx]
    notes.append(note, timeBeat + ( noteDuration ×
     i ), noteDuration, 100)
    i ← i + 1
  end while
  timeBeat ← timeBeat + nBeats
end while
```

Algorithm 3 getTransaction

```
f ← beatFeatures[1]
a ← beatFeatures[2]
transaction ← 0
rand ← random()
while f > threshold2[transaction] do
  transaction ← transaction + 1
end while
if rand <= 0.382 then
  transaction ← transaction + 0
else if rand <= 0.532 then
  transaction ← transaction + 1
else if rand <= 0.682 then
  transaction ← transaction - 1
else if rand <= 0.774 then
  transaction ← transaction + 2
else if rand <= 0.866 then
  transaction ← transaction - 2
else if rand <= 0.910 then
  transaction ← transaction + 3
else
  transaction ← transaction - 3
end if
if a < 0 then
  transaction ← transaction × (-1)
end if
```

Algorithm 4 getNoteDuration

```
$vel \leftarrow beatFeatures[0]$
if $vel < threshold1[0]$ then
    $noteDuration \leftarrow 0.25$
else if $vel < threshold1[1]$ then
    $noteDuration \leftarrow 0.5$
else if $vel < threshold1[2]$ then
    $noteDuration \leftarrow 1$
else if $vel < threshold1[3]$ then
    $noteDuration \leftarrow 2$
else
    $noteDuration \leftarrow 4$
end if
```

Now, per beat, it gets the duration the note is going to have associated with that beat, to do that it sees in which values of our first threshold array the feature is and then returns the duration of the note to be generated.

It also gets the transaction value, the distance between our previous note and our next note. To get this value our algorithm uses the next two features. First, we get a transaction value the same way we got the note duration previously, now with different threshold values, these are associated with the new feature it is analysing. Then we generate a random number and use a normal distribution to variate the transaction a little so we do not get the same music every time we run our program.

To see if we go up or down the scale we use the last feature, if this feature is positive we go up the scale, if negative we go down. Every time we can not go up or down the scale we go the other way around.

The threshold values were generated by us as we tested the algorithm and saw which values could better describe the movement in our opinions.

With this algorithm we could generate a large number of musics using three features as we wanted.

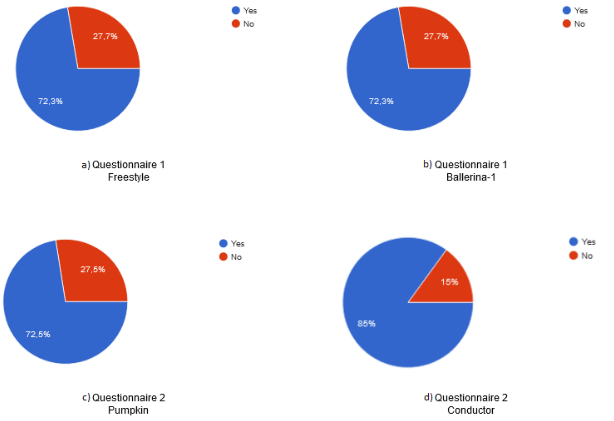
We decided to generate two musics, each with two tracks.

In our first music, the general movement implementation, uses velocity as feature which determines the note duration, fluidity gives us the transaction and acceleration as weather the transaction is positive or negative in our first track. And for the second track we used contraction index as note duration feature determiner, quantity of movement to generate the transaction and its derivative for the positive or negative transaction value.

The second movement, the specific movement implementation, uses the approach used on the first track of the first music on the upper body features to generate the first track and the same approach for the lower body features for the second track.

IMPLEMENTATION MERGER

We added all of the notes of both musics on a big array. Now we produced a fitness function to choose which of the notes appear on our third music piece.



a) Questionnaire 1 Freestyle

b) Questionnaire 1 Ballerina-1

c) Questionnaire 2 Pumpkin

d) Questionnaire 2 Conductor

Figure 2. Do you consider what you heard to be music?

This fitness function basically eliminates the duplicate notes and choose upon two different notes played at the same beat the one that is closer to the previous note if our fluidity value is under our threshold, otherwise it chooses the note that is further way from the previous one. If there is no previous note, it means our function has to chose the initial note of our generated music. In this case the function chooses at random out of the two initial notes.

After this we finally used our background attraction value in our approach. We use it as a percentage, and at random we choose that percentage of notes and change them to be out of scale by adding a random value at the pitch value.

COMPOSER

To compose our musics we use midiutil.midiFile python package with what we calculated before. As go through we add the notes to the specific track thus obtaining our output: .midi files with our composed music.

EVALUATION

To evaluate our system we chose to use questionnaires.

Our goals are to produce a system capable of produce music based on the human motion present in the video. Our second objective is to see if the music generated match the movement and can be seen as "inspired by" it. Our final objective has to do with the field we are hoping to contribute to, Computational Creativity. So we aim that this system produce creative objects.

Questionnaires are a good way to see if our system matches our goals, since creativity and music are said to be subjective fields.

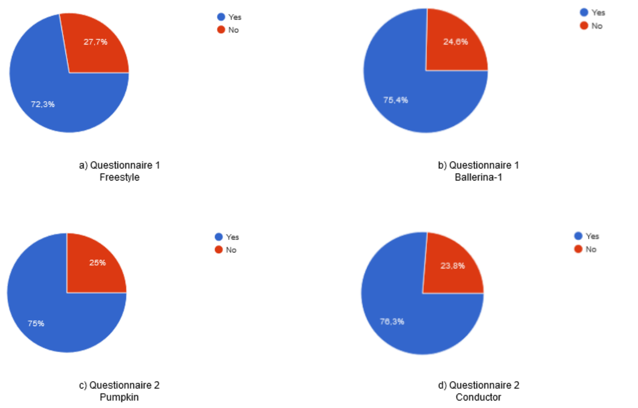
Results

We queried which version was preferred among the three produced, and from the answers we can say that there is not a specific version as seen by the values in table 1.

When asked if our generated sound is music over 70% of our answers were positive considering what they heard to be music. Figure 2

Table 1. Which version did you prefer?

Video	Mean	Median	Mode	Standard deviation
Video-1	1.82	2	1	0.61
Video-2	2.02	2	2 & 3	0.65
Video-3	1.89	2	2	0.57
Video-4	2.24	2.5	3	0.71



a) Questionnaire 1 Freestyle

b) Questionnaire 1 Ballerina-1

c) Questionnaire 2 Pumpkin

d) Questionnaire 2 Conductor

Figure 3. Do you consider what you heard to be creative?

Now that we established that most of our answers say that the sound is music, in table 2 we see the mean, median, mode and standard deviation values of how much our evaluators liked our generated music. Looking at the mean and median the values are all around the number 3 which is a neutral, the users neither hated nor loved our music. The mode values variate between 3 and 4 which tells us that there are a lot of neutral and positive answers but since our average is around or bellow three it means that there are some people that put the number 1, hated the music this means.

The users found the music to be creative. We can see more or less the same answers given to weather the user found the sound to be considered music. Over 70% of our answers were positive, which means we can consider our system to be considered as creative. Figure 3

The evaluator was asked to describe both the movement and sound of their favourite version from a list of adjectives given by us. From these questions we can see if the sound generated is related with the movement seen.

We have over 60% of the total distinct answers given to both movement and sound description. So it is safe to say that our sound can be considered related with the movement it was generated on.

when filtering the results by gender, age range and musical knowledge we could not find interesting or significant results to make new assumptions. This means that our question do not have a age, gender or musical knowledge bias.

CONCLUSIONS

In this work we developed and tested a system that perceives human body motion present in videos and generates music based on that perception. Our main goals are to create sound pieces that could be seen as movement inspired, for those sound pieces to be considered music inspired by the movement and finally for that music to be considered a creative piece in order to our system to be considered creative as well.

Regarding our implementation we use different techniques of computer vision, artificial intelligence, mathematics to be able to develop our system. In the end we were able to produce three music midi files each with a different movement to sound association.

For the evaluation of our work, we used questionnaires as a way to evaluate it based on our goals. We can say that our goals were reached with the development of this system.

As said in the title, our system is a computational creativity approach to the movement to sound association. There can be an almost infinite different ways to make a system capable of what our system is capable of.

REFERENCES

[1] Margaret A Boden. 2009. Computer models of creativity. *AI Magazine* 30, 3 (2009), 23–34.

[2] Georg Boenn, Martin Brain, Marina De Vos, and others. 2008. Automatic composition of melodic and harmonic music by answer set programming. In *Proceedings. International Conference on Logic Programming*. 160–174.

[3] Keunwoo Choi, George Fazekas, and Mark Sandler. 2016. Text-based LSTM networks for automatic music composition. *arXiv preprint arXiv:1604.05358* (2016).

[4] Florian Colombo, Samuel P Muscinelli, Alexander Seeholzer, Johanni Brea, and Wulfram Gerstner. 2016. Algorithmic composition of melodies with deep recurrent neural networks. *arXiv preprint arXiv:1606.07251* (2016).

[5] Douglas Eck and Juergen Schmidhuber. 2002. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. 747–756.

[6] Jose D Fernández and Francisco Vico. 2013. AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research* 48 (2013), 513–582.

[7] Edward H Hagen and Gregory A Bryant. 2003. Music and dance as a coalition signaling system. *Human nature* 14 (2003), 21–51.

Table 2. How much did you like the sound? (being 1 hated it and 5 loved it)

Video	Mean	Median	Mode	Standard deviation
Freestyle	2.86	3	3	1.01
Ballerina-1	3.05	3	4	1.05
Pumpkin	2.74	3	4	1.24
Conductor	2.96	3	3	0.91

[8] Hermann Hild, Johannes Feulner, and Wolfram Menzel. 1992. HARMONET: A neural net for harmonizing chorales in the style of JS Bach. In *Proceedings. Advances in neural information processing systems*. 267–274.

[9] Lejaren A Hiller Jr and Leonard M Isaacson. 1958. Musical composition with a high-speed digital computer. *Journal of the Audio Engineering Society* 6, 3 (1958), 154–160.

[10] Michael C Mozer. 1992. Induction of multiscale temporal structure. In *Proceedings. Advances in neural information processing systems*. 275–282.

[11] Gerard J Puccio. 2017. From the dawn of humanity to the 21st century: creativity as an enduring survival skill. *The Journal of Creative Behavior* 51 (2017), 330–334.

[12] R Keith Sawyer. 2011. *Explaining creativity: The Science of Human Innovation*. Oxford University Press.

[13] Joana Teixeira. 2017. *Cross-Domain Analogy from Image to Music*. Master's thesis. Instituto Superior Técnico, Universidade de Lisboa.