Transferring Music Pieces into Different Scales based on Nyquist Language

Jixin Li *

Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, the US

* Corresponding author email: jixinl2@illinois.edu

Abstract. As the increasingly need for art in the society and the current boost in technology, computer music is known as an increasingly important field in our life. While for music scales, which determine a music piece's genre and style, is also important to any music pieces. The idea of this study is to combine the concepts of music scales and computer music, realizing the ability for music scales transformation. This article will use Nyquist language as an example to provide a clearer perspective of the way that computer algorithms are able to transfer music pieces into different music scales. This study combined the advantages programming within Nyquist, which will be the music synchronize functions and the presence of the unique data type "score" with our music ideas. Thus, it was easy for users to compose the notes in a clear and ordered manner. After composing the scales as functions in Nyquist, one can modify music pieces into different styles according to the scales. In addition, based on the auto backup function in Nyquist, it is easy to retrieve a .wav format video file after processing the program on our terminals. These results shed light on guiding further exploration of music pieces transferring based on computer techniques.

Keywords: Transferring Music; Music Scales; Nyquist.

1. Introduction

The official definition for computer music can be abstractly described as the practical application of the combination of computing techniques and music composing methods to help individual composers to create new pieces of music based on computing synthesis algorithms [1-3]. Nowadays, computer music becoming more and more popular and more recent work has been down on this topic. Based on the past papers related, many have been proposed their ideas. For instance, Zhang and Hou stated that the demand for computer music is increasing in the current years [4].

Moreover, the computer music industry will even be considered as the focus of the new media in the near future. This indicates that the research based on computer music can be described as future promising. In addition, there are already a lot of computer music based applications emerged in our society. To be specific, there are also articles related with this topic. According to Ref. [5], these applications can be divided into many different sections, e.g., the ability to use synthesizing instrument to play notes to mimic the process of the playing of real instruments while in a manner that human musicians cannot achieve (e.g., extreme fast speed) [6].

This study will focus on a specific topic in music, which will be the music scales and the transformation between different music scales. There have been articles focusing on the research of scales, e.g., described music scale related information [7]. First of all, it provides a definition for what is music scale, it described music scales as a series of notes defined by musicians as appropriate for a music piece. Subsequently, it also provided ideas such as music scale modeling, this model basically stored the pitches of music notes into an array containing the notes pitch value after mod by 12.

This is convenient to be clear about the music note difference between the notes and can also be easy for us to do further operations accordingly. These materials can be very useful when one is actually converting scales.

To achive the goal, this study is going to use the Nyquist language, a strong computer music composing and synthesizing language to realize transferring scales using computer programming [8]. This language is particularly strong since it is a function-based programming language, and it also

provides direct access to audio files which will be very easy for musicians to manipulate the notes within. The nested database also provides functions which can increase the availability and efficiency of computer music programming. Therefore, this language is chosen to help the author to carry out the research.

In this article, using the materials related and the knowledge of programming and corresponding language grammar in Nyquist, an application will be demonstrated to help transfer the scale into corresponding scale notes and rewrite music pieces. The article will be mainly focused on the hexatonic blues scale and transfer the octave scale to blues scale using the techniques above. After transformation, the article will analyze the flaws as well as the criteria for success and get a further conclusion accordingly.

2. Data & Method

To understand the way to transfer a piece of music in octave scale into hexatonic scale, one needs to first understand the property of the two scales accordingly. Octave scale can be easy to identify, it contains eight notes in its scale composition, which is often used in music pieces. On this extent, the octave scale-based music pieces is known to be representative and in some extent, normal. While on the contrary, the blues scale is not common in music pieces. Blues scale, in this case we chose one of the most commonly regarded, contains 6 notes instead of 8 notes in one of its scales. The reason for changing music pieces is because music pieces written in blues scale can have some applications associated. In fact, there is some innovative use of the blues scale, or its associated music genre, jazz [9].

The scale can make music pieces more recognizable and more focused in steps. In addition, it clearly demonstrates music flows and make the music sounds more beautifully. In other words, hexatonic blues scale is often known to be a unparalleled music scale. Another issue will be that the major and minor notes of music scales are not often the same. Based on the previsou findings, major scale and minor scales can be transferred and the value for this two scales can be separately implemented when transferring music pieces [10].

One of the computer-aided approaches to music composition and changing music styles is to directly transpose the pitch vector. A pitch set is a sequence of MIDI numbers in increasing order which corresponds to one octave of a well-used scale in the intended music style. For example, {60 62 64 67 69 72} would be the pitch set corresponding to the pentatonic scale [11]. Moreover, numeric notation for pitches (MIDI notation) has definite advantages over the conventional names in dealing with computer manipulation of pitch data, especially because conventional names for pitches vary across the world. The root note of the music scale (Major or Minor) of a melody is estimated by fitting the notes to a music scale model. The estimated root note can then be used as the key in mode changing [12-15].

Afterwards, this study is going to talk about the success criterion for this project, this project is more art-based. So, the music content generating will be hard to evaluate, while one also comes up with potential ways to analyze if the project is considered to be successful. The ultimate aim of the project is to create a program that can switch between various music modes and styles easily, without creating a new function for every new music mode or scale added. The project tries to combine the advantages of programming within Nyquist by implementing music synchronization functions and using the unique data type “score” as much as possible, ultimately changing music modes in a clear, ordered and elegant manner [16, 17].

This framework, or in other words the logic will be fulfilled using Nyquist language in the first place. After the construction of the framework, the blues hexatonic scale will be implemented into the program by an array using the method stated above. Then, the array will be processed using a count-based loop, then for every single note, the result shall be processed and changed and be stored in a new array, the new array will then be played note-by-note by the nested function inserted in the Nyquist language. Using the nested function in Nyquist we can then be able to read any files, as soon

as the input is received, a hexatonic blues scale version of the audio file will be generated. This function will then be able to covert any music pieces in .mid file and octave scale into blues scale version and can be played accordingly. The typical sketch of the Nyquist is shown in Fig. 1.

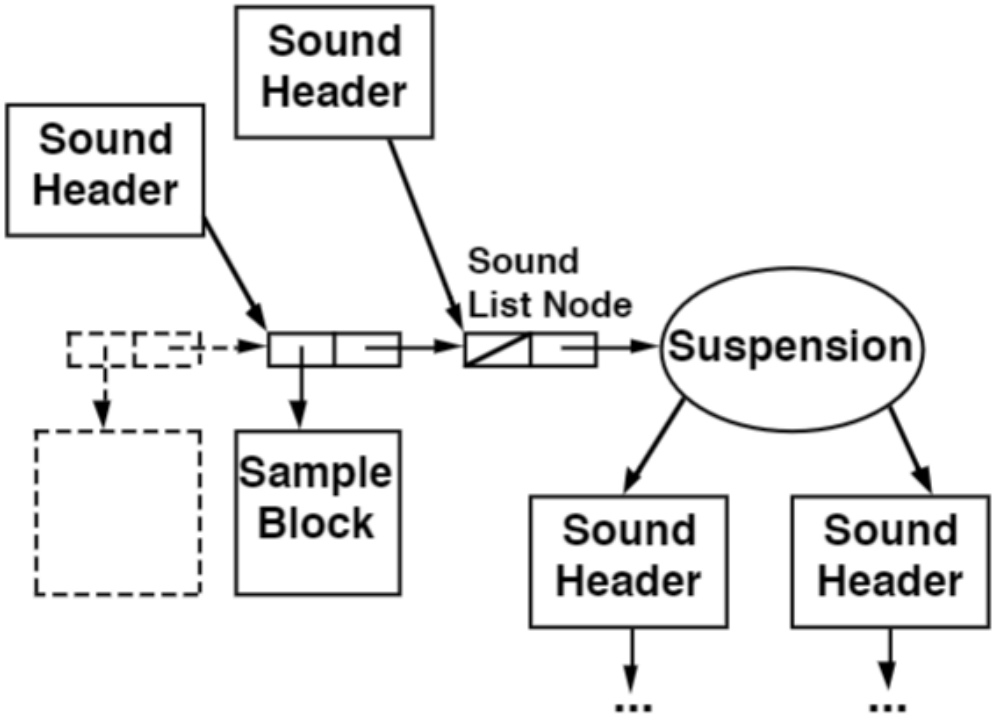


Fig 1. A sketch of the Nyquist sound dealing modules.

3. Results & Discussion

The result turned out to be expected and fluent. From the top, this paper will first evaluate the application section. The application can be mostly divided into three sections, the input section, the processing section and the output section. We will first discuss the first section, the input section as summarized in Table. 1

Table 1. Three Scheme comparing

Steps	codes
Step 1	set my-seq=seq-create()
Step 2	print type-of(my-seq)
Step 3	load"midi/use=midi-directory"
Step 4	set midi-file=open-binary("Minuet in G.midi")
Step 5	exec seq-read-smf(my-seq, midi-file)
Step 6	exec close(midi-file)
Step 7	set score=score-from-seq(my-seq)

The block of code is used to read a file located in the “bin” file of the Nyquist folder, in this case, this study chooses the “Minuet in G.midi”. This block reads the file and divide the file to notes and store the notes into a sequence. Aftewards, it can transfer the sequence into a score to be further processed following the procedure listed in Table. 2.

Thus, one will be looking into the processing section, also the most important section among the three sections. This section this study begins by introducing a function called “add-accents”. This is a nested function in build in Nyquist, it takes three parameters, time, dur, expr. These three parameters are set to be a particular value by the system.

The expression parameter, however, can be changed by using logical operations. In this case, we need to modify the pitch of the notes, so the first thing one needs to do is to get the expr attribute and set the attribute to the pitch value to a range between 0-12.

The values can then be estimated by the array. The loop downward is used to search for the right note. After one obtains the difference of the note's pitches. The value will be stored into the variable called "difference". This study will then use the difference value to compare with every pitch value, if the difference is calculated to be the minimum, then the note will be switched. On the above extent, one will be able to switch every two notes and it will be completely transferred based on the hexatonic blues scale. This operation will then be applied to the expr parameter in the function. After determined the expr parameter, one will be able to restore every of the pitches into the list. Then, the list will be returned, where the core codes are given in Table. 3.

Table 2. The Further proposed codes as a function

codes

define function add-accents(time, dur, expr)
begin
set I=expr-get-attr(expr, keyword(pitch))
set note=(I-7) % 12
set BluesScale=list(0, 3, 5, 6, 7, 10)
set difference=999
set minimum=999
loop
for i from 0 to 5
set difference=note-nth(i, BluesScale)
if s-abs(difference)<s-abs(minimum) then
begin
set minimum=difference
set expr=expr-set-attr(expr, keyword(pitch), I+minimum)
end
return list(time, dur, expr)
end
end

Table 3. List return codes

Steps	codes
Step 1	s2=score-apply(score, quote(add-accents))
Step 2	exec score-play(s2)

The output part will require simple functions to transfer. The score-apply function will transfer the original score with the new add-accents function, the score will be able to be modelled to s2. The score will then be played using the score-play function.

The research successfully transposes the music score into a wide variety of different scales without creating any extra dissonance. The implementation of the method is flexible as the programmer does not need to create a new function or add extra lines to the source code. Most related work focused on using computer programs to extracting information from music data or helping the composer with the music scales, but none attempt to change the music style or mode of an already existing music piece using a computer program only.

In this case, this paper uses the Minuet in G as a major example, this example is a G major key and is based on octave scale. This keynote of this music piece will be "do" in the music scale, thus we can do any music scale shift based on this note and the final result is available. The music sounds promising, and the overall status of the music piece changed. The status changed from a very soft and

delight mood into a more focused and independent place. The music piece sounds differently which provides me with a promise on the criteria for success. This piece of music is only a representative example of what this application is able to achieve, while every music pieces will be available for this function to be transfer.

In addition, this is only a part of the scale transferring, we also decided to change the music mode to Chinese pentatonic mode and the Dorian mode used in Celtic music, these two scales contains 5 and 8 notes in one scale respectively. Beisdes, the results of transferring music pieces into these two scales are also promising. These results indicating that the logic is also valid in logic. At the same time, it also denotes that music scale transferring is also valid in Nyquist language and computer music.

4. Limitations & Future Outlooks

However, the main defect to changing a music style this way is that the program only changes the music mode in technical terms, i.e., the program merely makes variations to the original scale of music. In reality, there are many more ways to truly change musical style and genre which are difficult to implement (e.g., changing the rhythm or cadence) due to the flexibility of the standard of a music genre or style. Musicians or composers are able to make additional changes based on the output of the program and save a vast amount of time.

In addition, it is hard to come up with a visualized success criteria since the research is mostly based on human acoustic system and whether the music pieces are successfully transformed or not is also based on human common knowledge. This idea made me think about the future of computer music and this study did further reading accordingly. According to the analysis, one sees the future of computer music with different fields of studies, such as the ecology mentioned in Refs. [12-15]. The artefact ecology framework is also appealing. There are also music scales related work emerged in the near future. Other scholars also provided many logics based transferring method, including the idea of using vectors to modeling the transformation process and mathematical-based algorithms to evaluate the results. All in all, about the future of internet using and computer music and traditional music, there is also published articles for future evaluation. Other researcher illustrates the idea and the comparison of the two genre of music and its advantages and drawbacks associated [11]. It clearly provide readers with a more thorough perspective. This topic is even associated with latest technology. For example, the VR technology is combined with computer music. Previous study also illustrates the correlation between the VR and other current technology with echoes and audio-based concepts. This certainly provide the industry with new blood [12]. To sum yp, the computer music industry is clearly prospective and any research in this filed will be claimed and carried out as necessary as possible.

5. Conclusion

In conclusion, this paper investigate music pieces transforming based on Nyquist. To be speicfic, different from the ordinary method of changing music styles through each note's duration and pitch, the project has successfully created a way of transferring between music styles using Nyquist functions with a midi input. Although a hand-edited style-change might have more character and style to it, a program using Nyquist increases the efficiency and flexibility of style transfer. In the future, it is witnessed that more advanced and state-of-art applications will be achieved. Overall, these results offer a guideline fo pieces transforming based on Nyquist.

References

[1] Mathews M V, Miller J E, Moore F R, et al. The technology of computer music. Cambridge, MA: MIT press, 1969.

[2] Collins N. Introduction to computer music. John Wiley & Sons, 2010.

[3] Manning P. Electronic and computer music. Oxford University Press, 2013.

[4] Zhang M, Hou K. Research on the application of computer music making technology in new media environment. Journal of Physics: Conference Series. IOP Publishing, 2021, 1871(1): 012142.

[5] Siphocly N N, Salem A B M, El-Horabty E S M. Applications of computational intelligence in computer music composition. International Journal of Intelligent Computing and Information Sciences, 2021, 21(1): 59-67.

[6] Scurto H, Chemla–Romeu-Santos A. Machine learning for computer music multidisciplinary research: A practical case study. International Symposium on Computer Music Multidisciplinary Research. Springer, Cham, 2019: 665-680.

[7] Zhu Y, Kankanhalli M. Music scale modeling for melody matching. Proceedings of the eleventh ACM international conference on Multimedia. 2003: 359-362.

[8] Dannenberg R B. Machine tongues XIX: Nyquist, a language for composition and sound synthesis. Computer Music Journal, 1997, 21(3): 50-60.

[9] Li C. Innovative Use of Blues as the Core Element of Symphonic Jazz: Taking Rhapsody in Blue as an example. 2022 7th International Conference on Social Sciences and Economic Development (ICSSED 2022). Atlantis Press, 2022: 2108-2112.

[10] Siphocly N N J, El-Horbaty E S M. Intelligent Technique for Automating the Conversion between Major and Minor Melodies. Future Computing and Informatics Journal, 2019, 4(2): 2.

[11] Siphocly N N J, El-Horbaty E S M, Salem A B M. Top 10 artificial intelligence algorithms in computer music composition. International Journal of Computing and Digital Systems, 2021, 10(01): 373-394.

[12] Masu R, Bettega M, Correia N N, et al. ARCAA: a framework to analyses the artefact ecology in computer music performance. Proceedings of the 9th International Conference on Digital and Interactive Arts. 2019: 1-9.

[13] Shmulevich I, Yli-Harja O. Localized key finding: Algorithms and applications. Music Perception, 2000, 17(4): 531-544.

[14] Wang S, Yu W. Space Elements of Computer Music Production Based on VR Technology. IEEE Access, 2020.

[15] Loy G, Abbott C. Programming languages for computer music synthesis, performance, and composition. ACM Computing Surveys (CSUR), 1985, 17(2): 235-265.

[16] Dannenberg R B. The implementation of nyquist, a sound synthesis language. Computer Music Journal, 1997, 21(3): 71-82.

[17] Pekař L, Matušů R, Dostálek P, et al. The Nyquist criterion for LTI time-delay systems. Recent Researches in Automatic Control, 2011: 27-29.