Linear Computer-Music through Sequences over Galois Fields

H. M. de Oliveira¹ e R. C. de Oliveira²

¹ Federal University of Pernambuco, Department of Statistics, Recife, Brazil
² Amazon State University, Department of Computer Engineering, Manaus, Brazil

hmo@de.ufpe.br, rcoliveira@uea.br

ABSTRACT

It is shown how binary sequences can be associated with automatic composition of monophonic pieces. We are concerned with the composition of e-music from finite field structures. The information at the input may be either random or information from a black-and-white, grayscale or color picture. New e-compositions and music score are made available, including a new piece from the famous Lenna picture: the score of the e-music “Between Lenna’s eyes in C major.” The corresponding stretch of music score are presented. Some particular structures, including clock arithmetic (mod 12), GF(7), GF(8), GF(13) and GF(17) are addressed. Further, multilevel block-codes are also used in a new approach of e-music composition, engendering a particular style as an “e-composer.” As an example, Pascal multilevel block codes recently introduced are handled to generate a new style of electronic music over GF(13).

0 INTRODUCTION

Many ways have been devised to compose music with aid of computers [1], [2], [3], [4], [5]. One of the most common is adopting the basic principle of mapping some (binary or multilevel) data source to musical notes [6]. Among these, a random song is quite straightforward, simply by generating random sequences at the input of the mapping [7], [8]. The input information can also come from another source such as 1/f noise [9], or fractal structures [10]. In the case of a nucleotide sequence of a genome (or a particular genome stretch) of species [11], [12], this is named DNA-music or genemusic [13], [14] (do not confuse with music generated by a genetic algorithm [15]). In the same line of reasoning, an amino acid sequence in the generation of a protein can be used as data to generate the sound: the protein-music [16], [13], [17], [18]. Another approach may use an image as a data source to create a song (for example, [19], [3]). In this investigation, we propose miscellaneous of image-to-note maps from multilevel sequences over a Galois field to music notes, without the concern of put them in categories. DNA-music can be seen as sequences over GF(4), a particular case. We can also use nibbles from bytes (from a binary files, whatever be the information) to define both note and note value associated with each byte of the file. If you want to hear some interesting computer music generated by this approach, many sites are available, • http://www.youtube.com/watch?v=qNf9nzvndlk
• http://www.toshima.ne.jp/k7Eedogiku/TextTable/WhatismGM.html
• http://www.genomamusic.com/genoma/ing/inicio.htm
• http://larrylang.net/GenomeMusic/

Indeed, many different mathematical-based descriptions are possible. There are other very interesting approaches to composing songs, including polyphonic [20], much more sophisticated and attractive, such as the one developed at Sony Computer Science Laboratories, Paris [21]. Not to mention Iamus, classical music’s composer, that conceive the first complete album composed solely by a computer [22]. Chermillier presented the Nzakara people’s music from Congo and Sudan to five strings harp [23]. Symmetries and group structure have long been exploited, as in [24], [25]. Since all these approaches deal with sequences over finite field, this can be generalized. Furthermore, coded sequences (from multilevel error-correcting codes) can be used to replace the input sequences given rise some sort of signature of sequences, defining a “style” of musical composition. This is called here an e-composer (derived from the block code used to encoder the input sequence). In particular, we consider here the new (multilevel) block codes called Pascal Codes [26]. Encoding the data from a given image, we generate a music composed by this virtual composer: in this case, Mr Pascal code (Equation 1).

1 COMPUTER MUSIC FROM UNCODED BINARY SEQUENCES

There are at least two straightforward ways to create e-music from particular input: i) music engendered from an image file ii) music derived from a random
input sequence. In both case, we are initially concerned with 8-level information sequences. For picture-music, the image may be resized to a suitable size and then converted from color to black-white/grayscale as to yield note sequences in a standard octave-repeating (Diatonic scale). For random music generation, input are merely random sequences from a uniform numbers $X \sim U(0,7)$ generator. From sequences generated over GF(7), a natural mapping is presented in Table 1. Also, taking $\alpha$ as a primitive element of GF(8), different maps can be assigned using some key signature. For instance, when adopting F major as key signature, the reading of any binary file in hexadecimal can be done in octal and each octal symbol set the note to 1.1 Variant 1

For instance, the gif file corresponding to the bmp has 107 bytes (i.e. the character “B” then the character “M” in ASCII encoding). The file size is 562 bytes from which there are 416 data bytes, 72 pixel/inch, 105 pixel (row). Indeed, the score generated may or not contain the header information: in this case we throw out the header. In-
terior position of image data is $\phi = 0$. 1.) (562 bytes from which there are 416 data bytes, the region between Lenna’s eye was cropped (Figure 1). 1.2 Variant 2

For example, the particular five bytes 00 06 1b 27 e2 result in the following sequence of notes: {rest} \{A\} \{E\} \{F B\} \{E F D\}, since 00=0, 06=6s, 1b=33s, 27=7s, e2=342s. Length: 828 musical notes.

1.3 Variant 3

Further straightforward approach should be to use the Hamming weight [27] each data byte of the file to define the musical codes according with the Table 3. The idea of cut out the code “0” becomes not only to fulfill the mapping of Table 1 (from 8-point to 9-point), but also to avoid the common long sequences of zeros, which imply long rests. Here is the result for the file (Length: 237 musical notes):

173271226127562255627737123212100000354324337437311325
33773773571400000363261222377121425427773773751020000027
11242773777252113377773734627620000003061533777377532625
37737715323730000001711012777374244151737772167377320000
140657377321333173731415737000000032537733415226377
37016273600000022012337736320321377374230735000000241377
350144127277761201360000001377603053777601360000000000
3773412031226773740036000000040177342261473773400374000
0017730022027237773073600000000017745112717737704360000
0003770266566377370434000000003750315733777200340400000
010702523737737730000000000661773773737373773773773737000000000000125213673773774000000000000021251
73737737740000000000001106273773773777777777770000000000001137
1157737773300000000361400140302441473773773274500000037
0125601212113773777377777210100000 416 bytes of data.
Table 1: Musical notes codes as a function of the 8-Gray pixel level (or GF(8)).

| signal | 0  | 1  | 2  | 3  | 4  | 5  | 6  | over GF(7) | over GF(8) |
|--------|----|----|----|----|----|----|----|------------|------------|
| 8-gray | α⁰ | α¹ | α² | α³ | α⁴ | α⁵ | α⁶ | α⁷         | α⁰         |
| C major | rest | C | D | E | F | G | A | B         |
| F major | rest | F | G | A | Bb | C | D | E        |
| G major | rest | G | A | B | C | D | E | F#       |

Table 2: Note values defined in terms of the accumulative indexes by reducing it modulus $p' = 5$.

| symbol of GF(5) | duration | note               | symbol |
|-----------------|----------|--------------------|--------|
| 0               | $1/2^0 = 1/1$ | Whole note (Semibreve) |        |
| 1               | $1/2^1 = 1/2$ | Half note (Minim)    |        |
| 2               | $1/2^2 = 1/4$ | Quarter note (Crotchet) |       |
| 3               | $1/2^3 = 1/8$ | Eighth note (Quarver) |        |
| 4               | $1/2^4 = 1/16$ | Sixteenth note (Semiquarver) |    |

Figure 2: Score of an excerpt of “Between Lenna’s eyes” in C major.

Table 3: Musical notes codes for the Hamming weight of a byte.

| level | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|---|---|
| note  | erase | rest | C | D | E | F | G | A | B |

2 LINEAR COMPUTER MUSIC FROM SEQUENCES OF MULTILEVEL BLOCK CODES

A possible way of choosing the note is to take into account each symbol of the GF(13)-valued codeword assuming a look-up table such as shown in Table 4 to deal with the chromatic scale. Indeed, clock arithmetic (mod 12) can also be used, simply by neglecting rest. Again, the note values can be computed according to Table 2. Let us now pick a clef. Here we also offer the use of multilevel error correcting codes over GF(13) to introduce some redundancy to the input data, i.e. by doing a block encoding on the sequence over GF(13). In order to play multilevel block codewords over GF(13), and starting new linear electronic music, an $(N,K)$ code over GF(13) is chosen. Input: i) random generation of codewords as $k$ numbers with distribution $\sim U(0,12)$, or ii) an image for the generation of codewords from the binary file of the picture and use as the information $\sum_{i=1}^{12} i$. A Pascal (13,8) blockcode over GF(13) has generator matrix $G$ given by (meet Mr
Table 4: One-to-one mapping between notes of the temperate scale and symbols of GF(13).

|    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|---|---|---|---|---|---|---|---|---|---|----|----|----|
| rest | C | C# | D | Eb | E | F | F# | G | G# | A | Bb | B |

Pascal):

\[
\begin{array}{cccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 2 & 12 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 10 & 3 & 12 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 12 & 4 & 7 & 4 & 12 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & 11 & 12 & 4 & 10 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 12 & 8 & 12 & 6 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 12 & 4 & 1 & 2 & 12 \\
\end{array}
\]

(1)

The Number of Distinct Excerpts is $13^K = 815,730,721$. Coding the bmp image “between Lenna’s eyes” with the generating matrix of Equation 1, we have the associated music generated by composer Mr Pascal. For instance, the simple repetition code \(2, 1\) over GF(13) with random information symbols such as \(11, 2, 5, 0\)... generate the sequence of repeated notes: \(11 11 | 2 2 | 5 5 | 0 0\) ... i.e. \(Bb 1/4 Bb 1/2 | C# 1/4 C# 1/2 | E 1/4 E 1/1 | rest 1/1 rest 1/1\) ... For a block code of length say \(N = 10\), a particular codeword \((11 2 5 3 12 1 4 8)\) engender the following accumulated (mod 3) sequence: \(11 13 18 18 21 24 36 37 41 49 \equiv 2 1 0 0 0 0 1 2 1\). In this particular case, since \(q=3\), only Whole note, Half note, and Quarter note are considered. The note sequence would thus be: \(Bb 1/4 C# 1/2 E (1/1) rest (1/1) D (1/1) D (1/1) B (1/1) C (1/12) Eb (1/4) G (1/2)\)

Now by assuming \(q=5\) for the same codeword, the corresponding sequence is: \(11 13 18 18 21 24 36 37 41 49 \equiv 1 3 3 3 1 4 1 2 1 4\)

The note sequence this time would be: \(Bb 1/2 C# 1/8 E 1/8 rest 1/8 D 1/2 D 1/16 B 1/2 C 1/4 Eb 1/2 G 1/16\). Of course, many variants are possible. For example, to incorporate different lengths for rests, one can choose to use the structure on GF(17), using an (arbitrary) mapping of the type described in the Table 5.

3 CONCLUSIONS

A new approach to generating electronic music is presented which consists of the use of structures on finite fields, including multilevel block codes. Musical compositions employing block codes with different input symbols can be viewed as songs of the same composer (characterized by block code). Different variants are presented for both diatonic scale and chromatic scale. Some score of new musical pieces are available, including “Between Lenna’s eyes.”

REFERENCES

[1] Marvin Minsky, "Music, Mind, and Meaning". In: Roads, Curtis (ed), The Music Machine: selected readings from the music journal, The MIT Press, Cambridge, Mass., 1989.

[2] Eduardo Miranda, Composing music with computers, CRC Press, 2001.

[3] Curtis Roads, The computer music tutorial, MIT press, 1996.

[4] Wilhelm Fucks, “Mathematical analysis of formal structure of music,” IRE Transactions on Information Theory, vol. 8, no. 5, pp. 225–228, 1962.

[5] Bill Manaris, Dallas Vaughan, Christopher Wagner, Juan Romero, and Robert B Davis, “Evolutionary music and the zipf-mandelbrot law: Developing fitness functions for pleasant music,” in Workshops on Applications of Evolutionary Computation. Springer, 2003, pp. 522–534.

[6] Thomas Rossler, Richard Moore, and Paul Wheeler, The Science of Sound, Addison Wesley, 3rd ed edition, 2002.

[7] Harald Niederreiter, Random number generation and quasi-Monte Carlo methods, SIAM, 1992.

[8] Harry F Olson and Herbert Belar, “Aid to music composition employing a random probability system,” The Journal of the Acoustical Society of America, vol. 33, no. 9, pp. 1163–1170, 1961.

[9] Richard F Voss and John Clarke, “‘1/f noise’ in music: Music from 1/f noise,” The Journal of the Acoustical Society of America, vol. 63, no. 1, pp. 258–263, 1978.

[10] Zhi-Yuan Su and Tzuyin Wu, “Music walk, fractal geometry in music,” Physica A: Statistical Mechanics and its Applications, vol. 380, pp. 418–428, 2007.

[11] Peter Gena and Charles Strom, “Musical synthesis of dna sequences,” in XI Colloquio di Informatica Musicale, 1995, pp. 203–204.

[12] Serge Morand and François Gasser, Langage des gênes - musique des hommes? Nodulations - la musique des gênes, INRA Editions, Versailles, 1998.

[13] Nobuo Munakata and Kenshi Hayashi, Gene Music; Tonal assignments of Bases and Amino Acidsin: Visualizing Biological Information, World Scientific, Singapore, 1995, ISBN 9810214278.

[14] Nobuo Munakata, Musical Representation of Gene Sequences, in: In Arts Medicine, Saint Louis, 1997, 0-918812-92-5.
Table 5: Mapping $\text{GF}(17)$ for the chromatic scale allowing different rest lengths (until Semiquaver-length rest).

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | C# | D | Eb | E | F | F# | G | G# | A | Bb | B |

[15] Dragan Matić, “A genetic algorithm for composing music,” Yugoslav Journal of Operations Research, vol. 20, no. 1, pp. 157–177, 2010.

[16] John Dunn and Mary Anne Clark, “Life music: the sonification of proteins,” Leonardo Music Journal, vol. 32, no. 1, pp. 25–32, 1999.

[17] Ross D King and Colin G Angus, “Pm—protein music,” Computer applications in the biosciences: CABIOS, vol. 12, no. 3, pp. 251–252, 1996.

[18] Rie Takahashi and Jeffrey H Miller, “Conversion of amino-acid sequence in proteins to classical music: search for auditory patterns,” Genome biology, vol. 8, no. 5, pp. 405, 2007.

[19] Thomas Oberst, “Blind graduate student ‘reads’ maps using cu software that converts color into sound,” Cornell Chronicle, vol. 36, pp. 5, 2005.

[20] Victor Lavrenko and Jeremy Pickens, “Polyphonic music modeling with random fields,” in Proceedings of the eleventh ACM international conference on Multimedia. ACM, 2003, pp. 120–129.

[21] Gaëtan Hadjeres and François Pachet, “Deepbach: a steerable model for bach chorales generation,” arXiv preprint arXiv:1612.01010, 2016.

[22] Gustavo Diaz-Jerez, “Composing with melomics: Delving into the computational world for musical inspiration,” Leonardo Music Journal, vol. 21, pp. 13–14, 2011.

[23] Marc Chemillier, “Mathématiques et musiques de tradition orale,” Mathématiques et sciences humaines, no. 178, pp. 11–40, 2007.

[24] Kathryn Bailey, “Symmetry as nemesis: Webern and the first movement of the concerto, opus 24,” Journal of Music Theory, vol. 40 10.2307/843890, no. 2, pp. 245–310, 1996.

[25] Daniel Muzzulini, “Musical modulation by symmetries,” Journal of Music Theory, vol. 39, no. 2, pp. 311–327, 1995, 00222909.

[26] Arquimedes J A Paschoal, Hélio M de Oliveira, and Ricardo M Campello de Souza, “A transformada numérica de pascal,” in Anais do XXXIII Simpósio Brasileiro de Telecomunicações, Juiz de Fora, 2015, SBrT.

[27] Richard W Hamming, Coding and Theory, Prentice-Hall, 1980.