OPTIMIZATION OF MUSIC DIGITIZATION

Emil Kelevedjiev Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria •How to convert generally accepted (traditional) musical notation into digitized formats ?



- MIDI (Musical Instrument Digital Interface)
- Computer friendly representation

- Most of the music we hear in based on a system called *12 tone equal temperament* (12TET), where the octave is divided into 12 equal parts.
- Microtonal music is generally defined as any music that is not 12TET. Some folks base their music on the harmonic series, some divide the octave into 19 or 31 equal parts, some divide the octave into 43 unequal parts,

	,,		
	76		
	70		
	/5		
	74		
500.05	/3	05	
523.25	/2	C5	C
	71	B4	h'
	70	A#/Bb4	ais/b'
440	69	A4	a'
	68	G#/Ab4	gis/as'
	67	G4	g'
	66	F#/Gb4	fis/ges'
	65	F4	f'
	64	E4	e'
	63	D#/Eb4	dis/es'
	62	D4	d'
	61	C#/Db4	cis/des'
261.6	60	C4	c'
	59		
	58		
	57		
	56		
	55		
	54		
	53		

• **12TET** is the standard tuning for **Western music** since the mid 19th-century.

• There are a variety of artistic, theoretical, historical and philosophical reasons through which musicians may be drawn to **'other intervals**'.

• As a result there are several disciplines, each of them studying '**microtonality**'.

- simply adding pitches to 12-note equal temperament (most often through microtonal equal temperaments such as 24-note ('quartertones'), 36-note ('sixth-tones'), 48-note, 72note, 96-note, etc.);
- contemporary pure tuning methods such as the various modern forms of just intonation, Pythagorean and mean-tone tunings;
- historically accurate tunings of Medieval, Renaissance, Baroque and Classical pieces.

 Much non-Western music is microtonal: classical music from India, Turkey, Arabia and Persia, gamelan music from Indonesia, xylophone music from Africa, Byzantine liturgical music, folk music from the Caucasus, Middle- and Eastern Europe, "Trio Bulgarka -The Mystery Of Bulgarian Voices", etc

MIDI Tuning Standard

• If *f* is a frequency in hertz, then we may compute a corresponding logarithmic frequency value by

$d = 69 + 12 \log_2 (f / 440)$

$f = 440 \times 2^{(d-69)/12}$

• The pitch values *d* are presented in a form of three-digit integers of base 128 (each digit in hexadecimal notation by 00 through 7F, i.e., from 0 to 127 in base 10)

• The first byte digit represents the MIDI note, or integer notation, value.

• The next two digits allow the semitone to be divided into $128^2 = 2^{14} = 16384$ parts, which means the octave is divided into 196608 equal parts.

• This is far below the threshold of human pitch perception and which therefore allows a very accurate representation of pitch.

Towards optimization of MIDI-like computer files

Mathematical and Computational Approach

• Simultaneous Diophantine approximation.

• Given *n* positive real numbers, $r_1, r_2, ..., r_n$, find nonnegative integers, $p_1, p_2, ..., p_n$, and *q*, such that the value of $|p_i/q-r_i|$ is "possibly small" for all i = 1, 2,..., *n*, and *q* is "not so great":

$$\min_{i} \max_{i} \left| \frac{p_i}{q} - r_i \right|$$

- Linear Programming. Well known and widely used in many applications, the method of solving problems of (integer) linear programming is well equipped now with sophisticated software.
- For an example, we may use the freely distributed program lp_solve, which is originated at the Technical University in Eindhoven, The Nederland, by Michel Berkelaar. Under GNU Lesser General Public License that software is maintained now in the group
- http://groups.yahoo.com/group/lp_solve

The solving of a problem of Simultaneous Diophantine approximation can be transformed into the integer linear programming task:

Minimize *w* under conditions:

$$0 < p_i < q,$$

$$-w \le q r_i - p_i \le w,$$

$$w > 0,$$

where

$$q > 0$$
 is fixed,
 $r_i > 0$ are given $(i = 1, 2, ..., n)$,
 p_i are integers for $i = 1, 2, ..., n$

- **Pythagorean tuning.** The classical problem leading to the contemporary twelve semitones equal tempered scale is how to approximate Pythagorean fractions:
- 2/1, 3/2, 4/3, 5/4, 6/5, 7/6, 8/7, 9/8, ..., or fractions in the form (*i* + 1)/*i*, *i* = 1, 2,....
- These fractions are supposed to be "pleasantly sounded"; when two physical sounds are reproduced with frequencies that related each other as a Pythagorean fraction, we hear them in harmony.

• The frequency of a sound that is produced by a vibrating string is determined by the string length



We may try to find an approximation of the form:

$$\frac{i+1}{i} \approx 2^{\frac{p_i}{q}}$$

or

$$\log_2 \frac{i+1}{i} \approx \frac{p_i}{q}$$

Let us denote

$$r_i = \log_2 \frac{i+1}{i}, \quad i = 1, 2, \dots, 8$$

and solve the corresponding linear programming problem, probing for different values of *q*.
Numerical experiments. For values *q*=12 and *q*=19 the following results are obtained:

i	<i>i</i> + 1	(<i>i</i> + 1)/ <i>i</i>	p	q	2 ^{p/q}	p	q	2 ^{p/q}
1	2	2.0000	12	12	2.0000	19	19	2.0000
2	3	1.5000	7	12	1.4983	11	19	1.4938
3	4	1.3333	5	12	1.3348	8	19	1.3389
4	5	1.2500	4	12	1.2599	6	19	1.2447
5	6	1.2000	3	12	1.1892	5	19	1.2001
6	7	1.1667	3	12	1.1892	4	19	1.1571
7	8	1.1429	2	12	1.1225	4	19	1.1571
8	9	1.1250	2	12	1.1225	3	19	1.1157

The case q = 12 corresponds to the twelve semitones equal tempered scale and the values of computed approximations are shown on the Figure, where the decimal fractions are approximate values of 2^{2/12}, 2^{3/12}, 2^{4/12}, 2^{5/12} and 2^{7/12}, respectively.





- **Conclusion.** The proposed approach of finding approximations may serve to improve frequency coding in some digitizing standard (e.g. MIDI) for musical pieces.
- The contemporary equal tempered musical notation is a standard, but it is not enough adequate to record some modern genres written in microtonal notation (e.g. rock music) as well as many ancient and/or non-European compositions.
- The Byzantine music is based on 72-tone equal temperament and the above described approach may serve to transform with a minimal deflection to the 12-tone equal temperament scale or to find some intermediate scale.

Thank you for your attention!