Mining Scalar Representations in a Non-Tagged Music Database

Rory A. Lewis, Wenxin Jiang, and Zbigniew W. Raś

University of North Carolina, Dept. of Comp. Science, Charlotte, NC 28223, USA

Abstract. In the continuing investigation of the relationship between music and emotions it is recognized that MPEG-7 based MIR systems are the state-of-the-art. Also, it is known that non-temporal systems are diametrically unconducive to pitch analysis, an imperative for key and scalar analysis which determine emotions in music. Furthermore, even in a temporal MIR system one can only find the key if the scale is known or vice-versa, one can only find the scale if the key is known. We introduce a new MIRAI-based decision-support system that, given a blind database of music files, can successfully search for both the scale and the key of an unknown song in a music database and accordingly link each song to its set of scales and possible emotional states.

1 Introduction

It is known in the field of psychology and neuro-endocrinology that data from neurotransmitters in laboratories prove that certain music scales evoke measurable sensory sensations and emotions [12]. On the point of emotional analysis, the science contains a varied array of papers concerning emotions in music [10] [7]. Furthermore, it is understood in the field of Music Information Retrieval (MIR) that emotions can be mined in a closed domain [2], [6]. If a machine, given a polyphonic musical wave-form, could recognize all the instruments and the correlating notes each instrument played then, if given the key, it could calculate the scale of the music or, if given the scale, it could calculate the key and subsequently the emotions of the song thereof [11], [13]. In summation, if MIR can find the scale and key of a piece of music then it can also mine emotions. The obstacles preventing MIR methods from successfully mining emotions in music are weak Blind Source Separation (BSS) of musical instruments in a polyphonic domain, imprecise instrument identification, the inability to find a scale unless given the key or vice-versa. Putting aside the polyphonic research the authors presented in BSS [3], this paper presents a system that given a blind piece of non-polyphonic music, it first correctly determines the scale and key and then secondly, the subsequent human emotions linked to said retrieved musical scale.

The process presented builds upon the authors' processes of calculating the fundamental frequency of notes (*see* [3]) and mining music scalar theory (*see* [4]) in a music database (*see* [14]) set in a non-Hornbostel hierarchical manner (*see* [5] *and* [9]). Accordingly this paper sets forth a methodology of finding fundamental frequencies in a database comprising both temporal and non-temporal attributes

upon which it clusters possible scales. Next it assigns weights to each possible root and subsequently uses a heuristic-based distance algorithm to match the correct root to the correct scale.

Finding the fundamental frequency of a sound wave enables one to determine the pitch of the sound wave. The middle A above middle C has a frequency of 440 Hz in standard tuning. The frequency is doubled to make the same note an octave higher, and halved to make the same note an octave lower. The distance of all the other frequencies in contemporary tuning is calculated using semitones. The frequencies within an octave, starting from a given note and going up in the frequency scale, can be calculated using coefficients according to the following formula:

$$f_k = f_1 \cdot 2^{k/12} \tag{1}$$

where k is the number of semitones separating f_k and f_1. However, we operate in the non-temporal domain and hence we consider the transient duration as the time to reach the quasi-steady state of fundamental frequency. We calculate fundamental frequency by first computing the local cross-correlation function of the sound object, and then computing mean time to reach its maximum within each frame, and finally choosing the most frequently appearing resultant frequency in the quasi-steady status.

Let $r(i,k)$ is the normalised cross correlation of frame i with lag k. To calculate $r(i,k)$, we look at how it reaches its maximum value with ω as the maximum fundamental period expected, ours being 60ms:

$$r(i,k) = \sum_{j=1}^{m(i)+n-1} s(j)s(j-k) \Bigg/ \sqrt{\sum_{j=m(i)}^{m(i)+n-1} s(j-k)^2 \sum_{j=m(i)}^{m(i)+n-1} s(j)^2}, \quad k \in [1, S_r \times \omega] \tag{2}$$

where s is the audio signal, $m(i) = i * n$, where $i = 0, 1, .., M-1$ is the frame index, M is a number of frames, $n = t * sr$, where $t =$ analysis window size, ours being 20ms, sr is a sampling rate, $k = 1, 2, ..., K$, where $K = lag = \omega * sr$.

In each frame i, the fundamental frequency is calculated in this form:

$$f(i) = \frac{S_r}{K_i / n_i} \tag{3}$$

where S_r is the sample frequency, n_i is the total number of $r(i,k)$'s local valleys across zero, where $k \in [1, K_i]$ and K_i is estimated by k as the maximum fundamental period.

Finding the fundamental frequency of a series of notes determines the relationship of the musical scales. The vast majority of scales in contemporary western music consist of 5 to 7 different notes (pitches). To calculate the number of possible scales we assert that the starting and ending notes are fixed and that there are twelve notes in an octave leaving 10 notes between the starting and ending notes. Also, we consider each note by moving from the lowest to the highest note. We cannot repeat a note and this is leaving one possible order,

or scale. There are N semitones including the tonic t_1 which forms the remaining notes $t_2, ..., t_M$ in the scale which in turn are distributed over the remaining $N-1$ points. Scales can be represented using the *spiral array*, where pitch states are associated by coordinates downward along an ascending spiral [1]. Musicians represent scales in numerous forms all of which are incompatible with knowledge discovery in music. With this in mind the authors chose to represent basic score classification of music not as a music system but rather as Pawlak$'s$ (see [8]) information system $S = (Y, A, V)$, called Scale Table, where Y is a set of music scales, $A = \{J^I, J^{II}, J^{III}, J^{IV}, J^V$, *Region, Genre, Emotion, sma*$\}$ (see Table 1). Jumps between notes are represented by $J^I, J^{II}, J^{III}, J^{IV}, J^V$ which correlate to specific scales, regions and genre of music. The values $\{s, m, a\}$ of attribute *sma* should be read as *scale, mode, arpeggio.*

Table 1. Basic Score Classification Scale Table

Y	J^I	J^{II}	J^{III}	J^{IV}	J^V	Region	Genre	Emotion	sma
$PentatonicMajor$	2	2	3	2		Western	Blues	melancholy	s
$BluesMajor$	3	2	1	1	2	Western	Blues	depressive	s
$PentatonicMinor$	3	2	2	3		Western	Jazz	melancholy	s
$BluesMinor$	3	2	1	1	3	Western	Blues	dramatic	s
$Augmented$	3	1	3	1	3	Western	Jazz	feel-good	s
•									
•									
•									
$Minor9th$	2	1	4	3		neutral	neutral	not happy	a
$Major11th$	2	2	1	2	3	neutral	neutral	happy	a
$Minor11th$	2	1	2	2	3	neutral	neutral	not happy	a
$Augmented$	4	4				neutral	neutral	happy	a
$Diminished$	3	3	3			neutral	neutral	not happy	a

This table was built by our team of music experts on the basis of analyzing when composers use particular scales. Its complete version can be found in [4].

2 Experiments

To run experiments to determine both key and scale the authors focused on analyzing songs wherein a musician performed a solo. We randomly selected two songs out of a database of 422 songs, each in both *.au* and *MIDI* formats, containing solo performances, namely, Eric Clapton's "*Nobody Knows You When You're Down and Out*", and The Allman Brother's "*Blue Sky*". Upon extracting the solo sections of each song, we split the sections into phrases and bars making three sets of each song for analysis.

Using the unsegmented version of each song we submitted it to the MIRAI system (*see* [4] and [9]) for the first run of the analysis, where the pitch and

duration of each note is computed (see Table 2) and where each frame duration unit is 0.12 seconds.

Table 2. Step 1: Computing duration for each note of each song

Blue Sky												
Note	a	$a\sharp$	b	c	$c\sharp$	d	$d\sharp$	e	f	$f\sharp$	g	$g\sharp$
Duration	5	2	15	9	128	16	0	103	2	47	11	32

Nobody Loves You												
Note	a	$a\sharp$	b	c	$c\sharp$	d	$d\sharp$	e	f	$f\sharp$	g	$g\sharp$
Duration	168	29	50	82	41	117	20	89	40	58	54	11

The authors chose to focus on scalar emotions in this paper. Composers and musicians push and pull away and back towards the root and scale of a song to create tension and release. Music theory in essence dictates that these push and pulls are best suited when patterned in conjunction with two criteria of a composition: 1) Bars, which determine syncopation and rhythm of a song and 2) Phrases in the song that typically align to sentences whether aligned to verbal sentences of the singer of the piece of music, or whether aligned to musical sentences and phrases. One can typically see patterns of tension and release in the aforementioned bars and phrases. We have determined that in order to find the dominant key: First we segment each bar and phrase into notes and then categorize the music based on what scale the most notes have been played. Next, we weight this number by the likelihood value of each note when it is classified to this scale. For example, if all the notes in the music piece are grouped into k bars: $B_1; B_2; ...B_k$, with B_i corresponding to one of the scales in Table 1, then we compute a bar-score $\phi(x)$ for each $x \in Note$ (see Table 4) as

$$\phi(x) = \Big[\sum_{i=1}^{k} B_i(x)\Big] \Big/ \Big[\sum_{y \in Song} \sum_{i=1}^{k} B_i(y)\Big] \tag{4}$$

and if all the notes in the music piece are grouped into k phrases: $P_1; P_2; ...; P_k$, with each P_i, $1 \leq i \leq k$, corresponding to one of the scales in Table 1, then we compute a phrase-score $\psi(x)$ for each $x \in Note$ (see Table 5) as

$$\psi(x) = \Big[\sum_{i=1}^{k} P_i(x)\Big] \Big/ \Big[\sum_{y \in Song} \sum_{i=1}^{k} P_i(y)\Big] \tag{5}$$

Next, we identify the note x for which the value $[\psi(x) + \phi(x)]/2$ is maximal. In our examples *Blue Sky's* $\phi(x)$ returns note $c\sharp$ which wins with a score of

32.44% and for *Nobody Loves You*, $\psi(x)$ returns note a which wins with a score of 20.985%.

Accordingly, we set forth an algorithm and methodology to identify the scale of the song and classify push and pulls of the roots in accordance with bars and phrases.

2.1 Stage 1 of 3: Initial 100% matches

Before we present the algorithm, we introduce the term *"Root-Matching"*.

Definition
Let $seq_1 = (j_1, j_2, ..., j_k)$ and $seq_2 = (i_1, i_2, ..., i_n)$ be any two sequences. We say that seq_1 is root-matching seq_2 if the following conditions hold:
(1) if $k \leq n$, then $(\forall m \leq k)[j_m = i_m]$,
(2) if $n \leq k$, then $(\forall m \leq n)[j_m = i_m]$.

Continuing the algorithm, as seen in Table 3, we search for 100% matches where each jump sequence calculated from Note Sequence matches with each jump corresponding to the *ith* scale (see Table 1). In other words, for each tuple in Table 3 we search for a supporting object in Table 1 such that its Jump Sequence is root-matching the Jump Sequence $(J^I, J^{II}, ..., J^V)$. The resultant was that Eric Clapton played precisely a *PentatonicDominant* in the key of f in phrase 1, a *Balinese* in the key of $f\sharp$ in phrase 2, and another *PentatonicDominant* in the key of g in phrase 8. Accordingly, *PentatonicDominant* in f, *Balinese* in $f\sharp$ and *PentatonicDominant* in g are possible candidates for the key and scale of Nobody Loves You, at this point. Similarly, The Allman Brothers played precisely a $PentatonicMajor$ in the key of e in phrase 1, a *Diminished* in the key of $c\sharp$ in phrase 2, $PentatonicMinor$ in the key of $c\sharp$ in phrase 4, and $BluesMajor$ in the key of $c\sharp$ in phrase 5. Accordingly, $PentatonicMajor$ in e, *Diminished* in $c\sharp$, $PentatonicMinor$ in $c\sharp$ and $BluesMajor$ in $c\sharp$ are possible candidates for the key and scale of Blue Sky, at this point.

2.2 Stage 2 of 3: Reducing the Search Space of Distance Algorithm

It is too expensive to search every possible close scale of every note according to bars and phrases. For example, in our small database of 422 songs with about 200 notes per solo, about 628 scales, 20 bars, and 10 phrases, it would require millions of calculations. To eliminate this problem, we developed a classification system that, to coin a new term, makes *musical cuts* according to weights for the purpose of invoking *Levenshtein* distance algorithm (see Section 2.3) to only search a knowledge base of relevant keys and scales that are most likely to be top candidates.

Bar Weights: We calculate the weights of fundamental frequencies in terms of bars. This is because, as mentioned above, the root of a song is often located in the first and/or last note of a bar. We store all the scores, but as a reference,

Table 3. Step 2: Find all possible scales and cut at 100% matches

Song	P (Phrase)	Note Seq.	Jump Sequence	100% Match	Scale
Blue Sky	1	$bc\sharp def\sharp g\sharp$	21222	$*$	$null$
	1	$c\sharp def\sharp g\sharp b$	12223	$*$	$null$
	1	$def\sharp g\sharp bc\sharp$	22232	$*$	$null$
	1	$ef\sharp g\sharp bc\sharp d$	22321	2232	$PentatonicMajor$
	1	$f\sharp g\sharp bc\sharp de$	23212	$*$	$null$
	1	$g\sharp bc\sharp def\sharp$	32122	$*$	$null$
	2	$c\sharp eg$	33	333	$Diminished$
	2	$egc\sharp$	36	$*$	$null$
	2	$gc\sharp e$	63	$*$	$null$
	3	$cc\sharp ef\sharp g\sharp$	1322	$*$	$null$
	3	$c\sharp ef\sharp g\sharp c$	3224	$*$	$null$
	3	$ef\sharp g\sharp cc\sharp$	2241	$*$	$null$
	3	$\sharp g\sharp cc\sharp e$	2413	$*$	$null$
	3	$g\sharp cc\sharp ef\sharp$	4132	$*$	$null$
	4	$bcc\sharp ef\sharp g\sharp$	11322	$*$	$null$
	4	$cc\sharp ef\sharp g\sharp b$	13223	$*$	$null$
	4	$c\sharp ef\sharp g\sharp bc$	32231	3223	$PentatonicMinor$
	4	$ef\sharp g\sharp bcc\sharp$	22311	$*$	$null$
	4	$f\sharp g\sharp bcc\sharp e$	23113	$*$	$null$
	4	$g\sharp bcc\sharp ef\sharp$	31132	$*$	$null$
	5	$c\sharp ef\sharp g$	321	32112	$BluesMajor$
	5	$ef\sharp gc\sharp$	216	$*$	$null$
	5	$f\sharp gc\sharp e$	163	$*$	$null$
	5	$gc\sharp ef\sharp$	632	$*$	$null$
	6	$cc\sharp def\sharp g\sharp$	11222	$*$	$null$
	6	$c\sharp def\sharp g\sharp c$	12224	$*$	$null$
	6	$cc\sharp def\sharp g\sharp$	22241	$*$	$null$
	6	$def\sharp g\sharp cc\sharp$	22411	$*$	$null$
	6	$ef\sharp g\sharp cc\sharp d$	24112	$*$	$null$
	6	$g\sharp cc\sharp def\sharp$	41122	$*$	$null$
Nobody	1	$acd\sharp efg$	33112	$*$	$null$
	1	$cd\sharp efga$	31122	$*$	$null$
	1	$d\sharp efgac$	11223	$*$	$null$
	1	$efgacd\sharp$	12233	$*$	$null$
	1	$fgacd\sharp e$	22331	2233	$PentatonicDominant$
	1	$gacd\sharp ef$	23311	$*$	$null$
	2	$ac\sharp def\sharp g$	41221	$*$	$null$
	2	$c\sharp def\sharp ga$	12212	$*$	$null$
	2	$def\sharp gac\sharp$	22124	$*$	$null$
	2	$ef\sharp gac\sharp d$	21241	$*$	$null$
	2	$f\sharp gac\sharp de$	12412	1241	$Balinese$
	2	$gac\sharp def\sharp$	24122	$*$	$null$
	3	$bcc\sharp def$	11121	$*$	$null$
	3	$cc\sharp defb$	11216	$*$	$null$
	3	$c\sharp defbc$	12161	$*$	$null$
	3	$defbcc\sharp$	21611	$*$	$null$
	3	$efbcc\sharp d$	16111	$*$	$null$
	3	$fbcc\sharp de$	61112	$*$	$null$
	4	$abcc\sharp dg$	21115	$*$	$null$
	4	$bcc\sharp dga$	11152	$*$	$null$
	4	$cc\sharp dgab$	11522	$*$	$null$
	4	$c\sharp dgabc$	15221	$*$	$null$
	4	$dgabcc\sharp$	52211	$*$	$null$
	4	$gabcc\sharp d$	22111	$*$	$null$
	5	$acc\sharp dd\sharp e$	31111	$*$	$null$
	5	$cc\sharp dd\sharp ea$	11115	$*$	$null$
	5	$c\sharp dd\sharp eac$	11153	$*$	$null$
	..	.....	.....	..	....
	..	.....	.....	..	....
	7	$ef\sharp aa\sharp cd$	23122	$*$	$null$
	7	$f\sharp aa\sharp cde$	31222	$*$	$null$
	8	$abdff\sharp g$	23311	$*$	$null$
	8	$bdff\sharp ga$	33112	$*$	$null$
	8	$dff\sharp gab$	31122	$*$	$null$
	8	$ff\sharp gabd$	11223	$*$	$null$
	8	$f\sharp gabdf$	12233	$*$	$null$
	8	$gabdff\sharp$	22331	2233	$PentatonicDominant$

Table 4. Bar Weights Calculating weights to calculate the musical cuts.

Song	Note	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	ϕ
Blue Sky	a			1	1									8%
	$a\sharp$						1							4%
	b							1						4%
	c											2		8%
	$c\sharp$	1	2		1	1	1						2	33%
	d										1			4%
	$d\sharp$													0%
	e	1		1		1		1	1	1				25%
	f													0%
	$f\sharp$													0%
	g										1			4%
	$g\sharp$								1	1				8%
Nobody	a		1		1		1							19%
	$a\sharp$	1												1%
	b								1					1%
	c													
	$c\sharp$													
	d					1	1	1						19%
	$d\sharp$													
	e			1										1%
	f	1												1%
	$f\sharp$					1		1						12.5%
	g		1		1				1					19%
	$g\sharp$			1										1%

Table 5. Phrase Weights Duration is summed for each note in each phrase in each song.

Song	Note	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	ψ
Blue Sky	a				2		3			1%
	$a\sharp$						2			1%
	b	8			7					4%
	c	2		2	2		3			2%
	$c\sharp$	14	7	3	20	7	77			35%
	d	2					14			4%
	$d\sharp$									0%
	e	22	8	10	26	13	24			28%
	f						2			1%
	$f\sharp$	8		3	8	4	24			13%
	g		6			3	2			3%
	$g\sharp$	5		4	6		17			9%
Nobody	a	16	27		9	15	25	59	17	22%
	$a\sharp$				3	4	7	11	4	4%
	b	2		3	6	7	11	2	19	7%
	c	5	4	3	5	26	18	14	7	11%
	$c\sharp$		5	2	5	13	8	8		5%
	d		5	25	20	20	13	20	14	15%
	$d\sharp$	2				7	5	6		3%
	e	11	5	14	2	14	7	25	11	12%
	f	2		9	2			4	23	5%
	$f\sharp$		5			2	8	30	13	8%
	g	10	5		5		7	10	17	7%
	$g\sharp$		2				3	3	3	1%

one can see that by looking at Table 4 $c\sharp$ receives a score 33% and e receives a score 25% for Blue Sky.

Phrase Weights: We calculate the weights of fundamental frequencies in terms of duration in phrases. This is because the root of a song is often located in patterns found in context to phrases. We store all the scores, but as a reference, one can see that by looking at Table 5 $c\sharp$ receives a score 35% and e receives a score 28% for Blue Sky.

Total Weights: Summation of the weights $[\phi(x) + \psi(x)]/2$ provides us with musical cuts used to reduce the search space of a distance algorithm. Here we find that for *Blue Sky*, $c\sharp$ wins with a score of 32.44% and for *Nobody Loves You*, A wins with a score of 20.985%. This means, we can be assured that by reducing the search space to all Note Sequences (attribute *Note Seq* in Table 3) beginning with $c\sharp$, the algorithm will produce the correct ith scale for *Blue Sky* and similarly, by reducing the search space to all Note Sequences beginning with note a, it will also produce the correct ith scale for *Nobody Loves You*

Table 6. Mining All Possible scales and Cuts in "Nobody"

Song	P (Phrase)	Note Seq.	Jump Sequence	100% Match	Scale
Nobody	1	$acd\sharp efg$	33112	$*$	$null$
	1	$cd\sharp efga$	31122	$*$	$null$
	1	$d\sharp efgac$	11223	$*$	$null$
	1	$efgacd\sharp$	12233	$*$	$null$
	1	$fgacd\sharp e$	22331	2233	$PentatonicDominant$
	1	$gacd\sharp ef$	23311	$*$	$null$
	2	$ac\sharp def\sharp g$	41221	$*$	$null$
	2	$c\sharp def\sharp ga$	12212	$*$	$null$
	2	$def\sharp gac\sharp$	22124	$*$	$null$
	2	$ef\sharp gac\sharp d$	21241	$*$	$null$
	2	$f\sharp gac\sharp de$	12412	1241	$Balinese$
	2	$gac\sharp def\sharp$	24122	$*$	$null$
	3	$bcc\sharp def$	11121	$*$	$null$
	..	.....	.....	...	...
	..	.....	.....	...	...
	8	$dff\sharp gab$	31122	$*$	$null$
	8	$ff\sharp gabd$	11223	$*$	$null$
	8	$f\sharp gabdf$	12233	$*$	$null$
	8	$gabdff\sharp$	22331	2233	$PentatonicDominant$

2.3 Stage 3 of 3: Calculating the Distance Between Jump Sequences

We use the *Levenshtein* Distance to calculate the distance between retrieved jump sequences which are not 100% matched with our scale table. The distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. For example, the Levenshtein distance between "21232" and "22222" is 2, since these two edits change one into the other, and there is no way to do it with fewer than two edits.

During the process of key note searching, we use two measures, ψ and ϕ, to evaluate the importance of each note. First one is the position of the note in each bar of the song. The points are calculated by adding 1 point to one note when it occurs in the first frame or last frame in one bar. Second measure is the duration of each note in each phrase of the song. Then we get ultimate weights for each note by adding up these two measures (ratio). The key note is the one with the highest weights. And we only search the candidate key notes from the list of first notes of all the retrieved note sequences for phrases since we only consider the match among these sequences by matching the first note of each note sequence with the key note. During the matching process, we get the list of accepted candidate scales identified by key notes, among which the scales with the shortest Levenshtein distance are chosen scale patterns together with 100 percent matched patterns. Note that unlike Table 2.2 that shows cuts, Table 5 illustrates the process before the cuts divide each phrase.

In *Blue Sky* a $c\sharp$ Pentatonic Major has a score of 8, making it the most likely scale and key. This is correct. In *Nobody Loves You* a a Balinese has a score of 8, making it the most likely scale and key. This is correct based on the data but the input data was polluted because the input system could not correctly assimilate polyphonic notes, which are in abundance in this piece of music. The correct scales, to humans or future MIR systems that can assimilate polyphonic sounds would be the mixture of c Spanish 8-Tone scale and c Major scale.

Table 7. Final Results

Song	Jump Matched	Scale	Root	Match Count	Similarity
Blue Sky	2232	Pentatonic Major	$c\sharp$	8	100%
	32112	Blues Major	$c\sharp$	2	60%
	3223	Pentatonic Minor	$c\sharp$	4	100%
Nobody	1241	Balinese	a	8	100%
	2233	Pentatonic Dominant	a	2	60%
	43	Major	a	4	100%

In *Blue Sky* a $c\sharp$ Pentatonic Major has a score of 8 making it the most likely scale and key. This is correct. In *Nobody Loves You* a a Balinese has a score of 8 making it the most likely scale and key. This is correct based on the data but the input data was polluted because the input system could not correctly assimilate polyphonic notes, which are in abundance in this piece of music. The correct scales, to humans, or future machines that can assimilate polyphonic sounds would be mixture c Spanish 8-Tone scale and c Major scale.

3 Conclusion

The algorithm worked 100% correctly on data it received by finding the correct musical cuts and then correctly reducing the search space of the distance algorithm and focusing it on the cut roots to find the scales. We randomly selected

these two songs. *Blue Sky* was completely correct because in the sol there were no polyphonic notes. However, because *Nobody Loves You* had polyphonic notes, the data was skewed and thus the scale was off. But, according to the input data, albeit wrong, it received, it did correctly calculate the correct key and scale. Our future work, clearly is to start figuring out how to assimilate polyphonic notes.

4 Acknowledgements

This research is supported by NSF under grant IIS-0414815.

References

1. Chew, E. (2002) Music information processing: a new application for operations researchers, in *Bulletin of AIROnews*, Vol. 7, No. 3, 9-14
2. Hevner, K. (1936) Experimental studies of the elements of expression in music, in *American Journal of Psychology*, Vol. 48, 246-268
3. Lewis R., Zhang X., Raś Z. (2007) Knowledge discovery based identification of musical pitches and instruments in polyphonic sounds, in *International Journal of Engineering Applications of Artificial Intelligence*, Vol. 20, No. 5, Elsevier, 637-645
4. Lewis R., Raś Z. (2007) Rules for processing and manipulating scalar music theory, in *Proceedings of MUE 2007, IEEE Conference*, in Seoul, Korea, 26-28
5. Lewis, R., Wieczorkowska, A (2007) Categorization of musical instrument sounds based on numerical parameters, in *Proceedings of RSEISP*, LNAI, Vol. 4585, Springer, 784-792
6. Li, T., Ogihara, M. (2003) Detecting emotion in music, in *ISMIR 2003 Procced.*, Available at http://ismir2003.ismir.net/papers/Li.PDF
7. McClellan, R. (1966) The healing forces of music, in *Element Inc.*, Rockport, MA
8. Pawlak, Z., (1991) Information systems - theoretical foundations, in *Information Systems Journal*, Vol. 6, 205-218
9. Raś Z., Zhang X., Lewis R. (2007) MIRAI: Multi-hierarchical, FS-tree based music information retrieval system, in *RSEISP Proceed.*, LNAI, Vol. 4585, Springer, 28-30
10. Sevgen, A. (1983) The science of musical sound, in *Scientific American Books, Inc.*, New York, NY
11. Sloboda, J. A., ONeill, S. A. (2001) Emotions in everyday listening to music, in *Music and Emotion: Theory and Research*, Juslin, P. N, Sloboda, J. A. (Eds.), Oxford Univ. Press, 415-430
12. Valentinuzzi, M.E., Arias, N.E. (1999) Human psychophysiological perception of musical scales and nontraditional music, in *IEEE/Eng Medicine and Biol. Mag.*, Vol. 18, No. 2, 54-60
13. Vink, A. (2001) Music and Emotion, living apart together: a relationship between music psychology and music therapy, in *Nordic Journal of Music Therapy*, Vol. 10, No. 2, 144-158
14. Wieczorkowska, A., Synak P., Lewis, R., Raś, Z. (2005) Creating reliable database for experiments on extracting emotions from music, in *IIPWM 2005 Proceedings*, Advances in Soft Computing, Springer, 395-402