Working Paper Series
ISSN 1170-487X

Computer improvisation of blues melodies

by Mark Andrew Hall

Working Paper 92/2

February, 1992

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Computer Improvisation of Blues Melodies

A thesis
submitted in partial fulfilment
of the requirements for the degree
of
Bachelor of Computing and Mathematical Sciences
at the
University of Waikato
by

Mark Andrew Hall

University of Waikato

1991

Abstract

A computer program has been written which composes blues melodies to fit a given backing chord sequence. The program is comprised of an analysis stage followed by a synthesis stage. The analysis stage takes blues tunes and produces zero, first and second order Markov transition tables covering both pitches and rhythms. In order to capture the relationship between harmony and melody, a set of transition tables is produced for each chord in the analysed songs. The synthesis stage uses the output tables from analysis to generate new melodies; second order tables are used as much as possible, with fall back procedures, to first and zero order tables, to deal with zero frequency problems. Some constraints are encoded in the form of rules to control the placement of rhythmic patterns within measures, pitch values for long duration notes and pitch values for the start of new phrases. A listening experiment was conducted to determine how well the program captures the structure of blues melodies. Results showed that listeners were unable to reliably distinguish human from computer composed melodies.

Acknowledgements

I wish to thank Lloyd Smith for his supervision and valuable contribution to this project.

Contents

List of Figures

Figure

List of Tables

Table

1. Introduction

Musical improvisation is governed by principles that must be in the musician's mind and must be adequate for the generation of music in real time. When a musician improvises there is no opportunity to go back and improve or revise, and therefore the musician cannot afford to make mistakes (choose notes that do not make up a satisfactory melody of the appropriate variety).

The greatest improvisers today are found in modern jazz and blues. Common to most forms of improvisation is a reliance on two distinct components. Firstly a long term memory for a set of basic structures (in the case of jazz, a chord sequence and related musical scales), and secondly a set of principles which underlie improvisatory skill. The chords and scales are accessible to the conscious mind - they can be written down, taught and described. The improvisatory principles, however, are less tangible and are inaccessible to conscious thought. Some musicians are aware of a few aspects of them; no one, however, has introspective access to them all. Musicians learn to improvise by imitation and by experimenting. They learn to improvise by improvising; the process takes years to master.

Musicians often improvise melodies to fit a large variety of different chord sequences. The chord sequences are usually known by heart, and the same basic sequence is used throughout the piece. Modern jazz and, to a certain extent, blues may call for melodies to be generated at an extremely rapid rate. The computational problem, therefore, is to produce notes using as little memory as possible for intermediate results and thus mimic the improvisatory process that the musician uses.

You might ask 'well what's the point? Who wants a computer that can improvise blues?' This is a good question. It's unlikely that a computer would ever surpass a human musician, and even if it did that would still not guarantee acceptance. Who would want to idolise a machine rather than a human musician? Nevertheless, investigating computer improvisation may help improve the understanding of the structure of music or the process of human improvisation. At the very least it may provide some interesting ideas if not usable phrases for human improvisers.

1.1 Background

Composers have been using the computer as an aid to writing music since the mid 1950's. Composition with the computer actually predates the use of the computer to synthesise sound.

The problem, as far as it can be described as a problem, has come about mainly through human curiosity, as in 'can we get a computer to improvise or compose music?' Through investigating the problem and devising solutions to it light may be shed on musical structure and on the methods of human composers and improvisers.

Although there are a number of different approaches to computer composition, most of the work has fallen into two broad categories: 'stochastic' music, in which events are generated according to the statistical characterisation of a random process, and music which has been derived by generative grammars.

Markov chains and transition tables for musical analysis have been investigated by Richard C. Pinkerton [17] and by Harry F. Olsen [16], the inventor of the RCA Sound Synthesiser in the late fifties. Pinkerton [17] investigated how Markov transition tables could be used to capture the entropy (information content) of music. He analysed nursery tunes and created a 'banal' tune generator which used a first order transition table. He found that the banal tune generator had a redundancy of about 63%, ie. it produced very monotonous tunes. However, many of them were not as monotonous as some actual nursery tunes. A certain amount of repetition is necessary in order to have tuneful melodies.

Olsen [16] analysed successions of melodic tones in eleven songs by Steven Foster, a nineteenth century American composer. He tabulated the relative frequency of scale step occurrences using zero, first and second order transition tables. He found that melodies generated using zero order tables showed little of the musical style on which the table was based. This was because the table did not take into account of any previous values. First order transition tables generated melodies somewhat closer to the original style and second order tables closer still. A good overview of the Markov process and work which has used it can be found in a paper by Charles Ames [2].

One of the earliest serious computer composition was made by Lejaren A. Hiller [9]. Using the ILIAC digital computer at the University of Illinois, he devised a program to compose counterpoint using a random number generator (white noise) to select notes and a set of counterpoint rules to screen for acceptance. The program produced counterpoint of fair quality if the rather monotonous rhythm was ignored. Hiller also devised another program which used a Markov probability table of musical intervals to generate notes for a melody. The probabilities depended on previous choices and to the opening note of the piece. This introduced a feeling of tonality.

Other attempts have been made at getting computers to generate counterpoint, most notably by William Schottsteadt [18]. The main difference between Schottsteadt's program and Hiller's is that Hiller used at most only sixteen rules to screen notes for acceptability. Scottsteadt's program, on the other hand, used all of the rules described by Fux in the 18th century - a total of 43 rules.

Since Fux stressed that the rules were guidelines and not absolutes, Schottsteadt defined the relative importance of the rules by assigning a penalty to each. The higher the penalty the worse it was to break the associated rule. The main constraint Schottsteadt found was compute time (i.e combinatorial explosion). In the end he settled for a best first search strategy. Nevertheless, for five-voice counterpoint, compute times remained high.

The use of transition tables is not favoured by all advocates of stochastic composition. Martin Gardner favours the generation of music by one-over-f noise [7]. He asserts that music based on transition tables, however closely related to modelling the small, is still random in the large. Consider the melody over four or five notes and the tones are strongly related. Compare a run of five notes with another five note run later on and you are basically back to 'white noise'. In comparison to this, one-over-f music is very self similar. Values in a sequence generated by one-over-f music correlate logarithmically with the past. For example, the average activity of the last ten values has as much influence on the current value as the average of the last one hundred or one thousand. This property means that the process has a relatively long memory. White music has no memory at all; one-over-f-squared or brownian music places such a heavy weight on the previous event that events prior to the previous few have virtually no effect on the current outcome.

J. Ulrich [19], D Levitt [14] and P.N Johnson-Laird [10,11,12] have ideas in the specific area of jazz improvisation by computer. All of them use a grammatical approach to a greater or lesser degree. Both Ulrich and Levitt believe that, when jazz musicians improvise, they weave together little pieces of existing melodies that they have stored in their memories during the course of their musical experience. The problem, as they see it, is the modification of these 'motifs' to fit a new harmonic background. Ulrich [19] concentrated his efforts in the area of analysing and assigning harmonic meaning or function to the background chords of a given piece of jazz. This is important because the 'meaning' of a given chord within the context of others governs the choice of notes or scales the musician can use to improvise over it. Levitt took the idea one step further and actually implemented an extra part to the program to fit motifs to the analysed background chords.

Johnson-Laird [11] claims it is easier (and takes less computational power) in the long run to make up new melodies than to remember a vast array of motifs and to modify them to fit the chord sequence. He advocates that a grammar be used to generate the melodies as well as analyse the harmony, and that some such device is necessary even on Ulrich's or Levitt's account, since motifs must be invented by someone. He has developed a program which uses a grammar to generate the 'contours' of a bass line for a jazz bassist. The principle is that after a series of small steps in pitch, a step of a rather larger interval, and vice versa, make for a pleasing melody. The program functions as a finite state device, which is psychologically plausible, according to Johnson-Laird, because it would enable a musician to improvise a melody as quickly as possible with barely any need to compute intermediate representations.

Kevin Jones has suggested the use of stochastic grammars to generate music [13]. A stochastic grammar includes a probability assignment over the ordered set of production rules. What he calls a 'space grammar' can operate across many dimensions, so that when such a grammar is applied in a musical context, the parameters specifying simultaneous occurring events are related to one another as well as to their temporal neighbours.

2. Methodology

2.1 Computer Composition Program

For modelling blues music the stochastic approach using transition tables was chosen. This method was chosen because it was hoped that through analysis the underlying structure of blues would be captured by the transition tables.

2.1.1 Program Evolution, an Overview

The initial version of the program analysed pitches without regard to harmony. In order to capture some of the relationship between harmony and melody, the first modification was to consider pitch with respect to backing chords. At this point rhythm was not taken into account, so, while the resulting tunes sounded pleasant enough, they didn't sound much like blues. The next step, then, was to look at rhythms of blues tunes.

Initial analysis considered only durations of individual notes. The rhythms resulting from this process were disappointing. They sounded odd and did not conform to the rules of music. A new way of representing rhythm using short rhythmic patterns was developed, along with rules to ensure that the placement of these patterns conformed to the rules of music. The rhythms produced by this process were far more pleasant and sounded much like the tunes they were based on.

The development of rhythm highlighted a problem that could occur with pitch. Since there was no relationship between the pitches and the rhythms being produced, passing tones occasionally fell on long duration notes. This does not occur in blues tunes (and sounds terrible besides). The next stage then, was to try to capture some of the relationship between rhythm and pitch.

To this end, pitch was considered with respect to both rhythm and the backing chord. While this eliminated the long duration problem, the data was fragmented too much by this process. The stochastic element was lost and the model was becoming deterministic. In an attempt to reduce the determinism, the transition tables that resulted from the consideration of pitch with backing chord were 'collapsed', although pitch was still considered with respect to rhythm. Unfortunately, due to time constraints and complications with coding, this version of the program was abandoned, although some preliminary results looked promising.

At this point it was decided to move back to the version of the program that had no connection between rhythm and pitch. To this version rules were added to screen long duration notes for acceptable pitches based on what was seen in actual blues tunes. This appeared to work well. The final improvements to the program included the addition of a conditional probability to control ties, modification to produce an anacrusis, addition of a rule to produce termination on a whole note, and the addition of a special transition table to handle acceptable pitches for the start of new phrases.

2.1.2 Program Design

Analysis/Synthesis

The system has two distinct stages. These are actually implemented as separate programs. The first stage, analysis, constructs transition tables capturing the structure of blues melodies. The second stage, synthesis, uses these transition tables to generate new tunes.

Analysis

Transcriptions of tunes by some of the greats of American blues were used by this stage of the program.

Dimensions: Rhythm and Pitch

Analysis considers 2 dimensions: rhythm and pitch. With a few exceptions these 2 dimensions are analysed independently.

The program's pitch range is 3 octaves. This is more than adequate for the blues tunes that were used for analysis, although the range could easily be expanded. In order to analyse tunes in all keys, the program normalises pitch by translating all notes to their relative distance (in half steps) from the root of the key.

In order to analyse rhythm a number of short rhythmic patterns were identified. These patterns, when used in conjunction with each other, can reproduce the complete rhythms of all tunes analysed. The patterns form a complete set, down to the eighth note level, for blues tunes in common time. Figure 1 shows the set of rhythm patterns that was used. The use of these patterns introduces a moderately variable time scale as these patterns are of differing lengths.

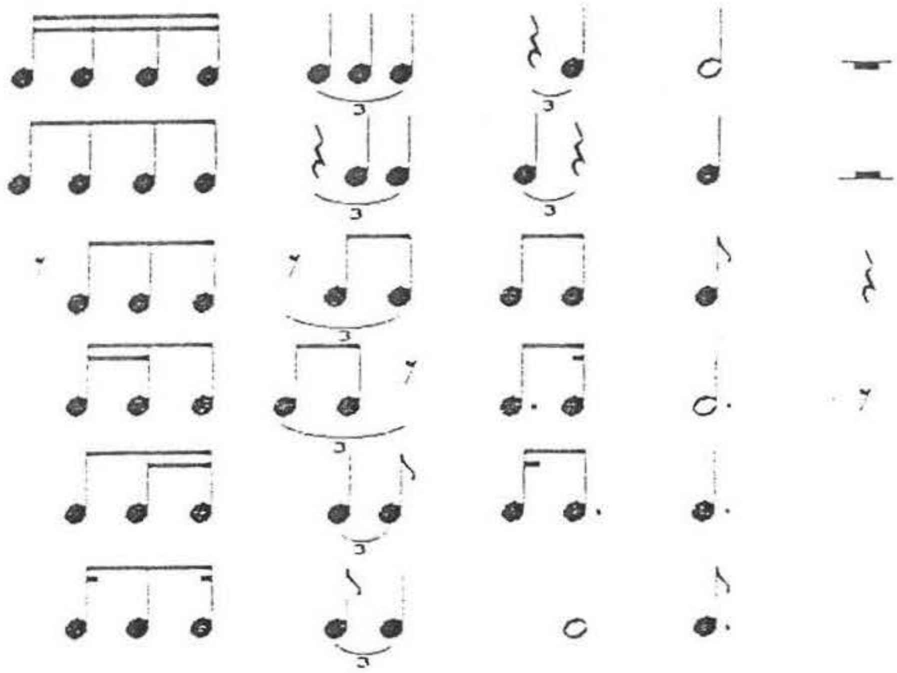


Figure 1. Rhythmic Patterns

Music and Program Representation

One of the first problems was to develop a representation of standard music notation that could be used by the program. Figure 2 shows an example of standard musical notation and the representation used for the program. The first entry in the program representation indicates the key of the piece (in this case C). The next entry indicates the first backing chord, C. The program will analyse all pitches and rhythms with respect to the chord C, until it reads a new backing chord. The entries after this first chord provide rhythm and pitch information. For each entry the number outside parentheses is the code for the rhythmic pattern. Entries inside parentheses are pitches. The digit indicates the octave of the pitch. An equal sign before an entry indicates that the first pitch in that entry is tied to the last pitch of the previous entry.

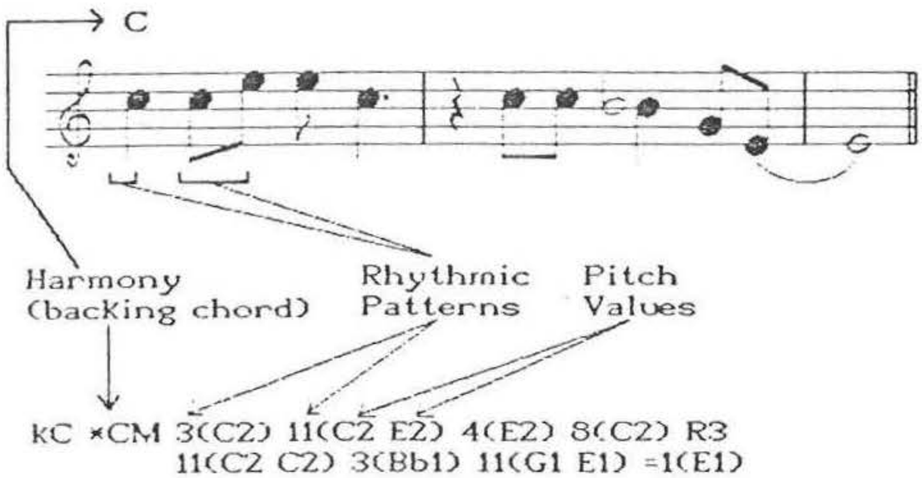


Figure 2. Standard music and program representation

Transition tables

The major result of analysis is the generation of a set of transition tables which describe sequences of events in the 2 dimensions of rhythm and pitch, with zero, first and second order tables created for each.

Zero order tables hold probabilities for occurrences of single events. First order tables give the likelihood of an event given the most recent event. Second order tables give the likelihood of an event given the two most recent events.

Since pitch is analysed with respect to harmony, first and second order tables are created for each unique backing chord seen in the analysed tunes.

There is also a special zero order table created for the start of new phrases. It contains the likelihoods of different pitches being used to start new phrases, where a the start of a new phrase is defined as the beginning of a tune or the first note after a rest of half a measure or more.

Figure 3 shows an example of partial transition tables for pitch under the tonic chord. These are from analysis over 48 tunes. For the first and second order table previous events (pitches) are shown in the far left column and the next event along the the top of the table. Probabilities are displayed as percentages. The tables shown in this figure are but a small fraction of their true size.

0 Order Pitch

Pitch	Probability
C1	4.9
D1	0.7
Eb1	1.7
E1	2.8
F1	2.1
Gb1	0.2
G1	9.7
A1	4.1
Bb1	3.5
B1	0.6
C2	23.6

1st Order Pitch

	Next Pitch										
	C1	D1	Eb1	E1	F1	Gb1	G1	A1	Bb1	B1	C2
C1	60.3		1.5	13.2			7.4	1.5	4.4		11.8
D1	60.0			40.0							
Eb1	35.0		55.0								
E1	28.9	2.6	2.6	28.9	2.6		23.6		2.6		2.6
F1			16.7	38.9	16.7	5.6	22.2				
Gb1							100				
G1	8.7		2.9	10.7	10.7	8.9	36.9	10.7	3.9		6.8
A1							46.7	20.0	6.7		24.4
Bb1	11.1						24.4	6.7	31.1		22.2
B1											100
C2	1.7			0.3			5.2	5.2	4.2	0.7	56.3

2nd Order Pitch

	Next Pitch										
	C1	D1	Eb1	E1	F1	Gb1	G1	A1	Bb1	B1	C2
C1 C1	42.9		3.6	14.3			10.7	3.6	10.7		14.3
C1 D1				100							
C1 Eb1			100								
C1 E1	22.2			22.2			44.1		11.1		
C1 G1							100				
C1 A1											100
C1 Bb1									100		
C1 C2											100

Figure 3. Partial transition tables for pitch

Rules For Pitch and Rhythm

In addition to transition tables, analysis establishes some high order rules for pitch. As was explained above, transition tables created for pitch have no relationship with those created for rhythm. So during generation of new tunes it is possible that a passing tone, generated by pitch tables, might fall on a long duration note, generated by rhythm tables - where in fact passing tones should only occur fleetingly between more consonant tones. Rules for long duration notes solve this problem. A long duration note is defined, for this purpose, as a note which lasts for half a measure or more. During analysis the program records all pitch values that occur for half notes, dotted half notes and whole notes. These pitch values become the criteria for screening notes of long duration.

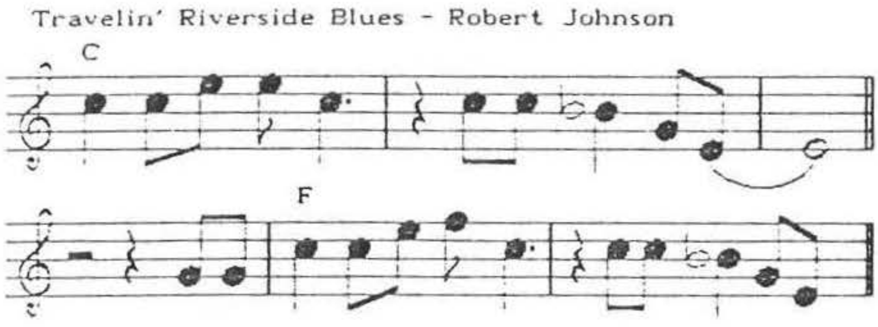
A set of rules are used to ensure that during synthesis, generated rhythmic patterns can be legally placed at the current position in a measure. These rules are not determined through analysis as they hold globally for all types of music.

Since most blues tunes end on a sustained whole note which is the tonic note of the tune, a rule was implemented in the synthesis stage to make this happen.

Example of Analysis

Figure 4 shows the first eight measures of Travelin' Riverside Blues, by Robert Johnson [8]. When the program analyses this tune it first sees the key, in this case C. This gives the program a tonality to refer the pitches to. The next thing it encounters is the first backing chord, C major. At this point the program starts assembling tables for the tonic chord. The next thing encountered is the first rhythmic group, a quarter note (code number 3). This goes into the zero order table for rhythm with a tally of 1. Next the program sees the pitch value for this note, which is C2. This pitch value is placed into the phrase start table with a tally of 1 (the start of the song is also the start of a new phrase), and into the zero order pitch table with a tally of 1. At this point all the information for the first entry has been processed so the program moves on to the next entry. The next entry's rhythmic group is 2 eighth notes (code number 11). This goes into the zero order rhythm table with a tally of 1. Now that there is an event in the past, the transition of a quarter note to 2 eighth notes can be recorded in the first order rhythm table with a tally of 1. Next the second pitch of the song is analysed, a C2. Since it is not the start of a song/phrase this value is not recorded in the phrase start table. It is recorded in the zero order table for pitch, increasing the tally for C2 to 2, and the transition C2 to C2 is recorded in the first order pitch table. At this point there is still one more pitch left in the second rhythmic group, E2. This pitch is recorded in the zero order pitch table. The transition C2 to E2 is recorded in the first order pitch table, and now the transition C2 C2 to E2 can be recorded in the second order pitch table. Analysis continues in this manner until all songs have been analysed.

There are a couple of points of interest in this example. The first is the whole note in measure 3. This is a long duration note, so its pitch value is placed in the rule table for whole notes. The second point of interest comes in measure 4, where there is a rest of three beats. This signifies the start of a new phrase, therefore the first pitch value after the rest goes into the phrase start table.




```
KC *CM 3(C2) 11(C2 E2) 4(E2) 8(C2) R3 11(C2 C2)
       3(Bb1) 11(G1 E1) =1(E1) R2 R3 11(G1 G1)
       3(C2) 11(C2 E2) 4(F2) 8(C2) R3 11(C2 C2)
       3(Bb1) 11(G1 E1)
```

Figure 4. Example tune: standard and program representation

Figure 5 shows the tables resulting from analysis of this short example.

PITCH

Long Duration

E1

Phrase Start

Pitch	Probability
C2	50.0
G1	50.0

0 Order

Pitch	Probability
E1	15.4
G1	23.1
Bb1	7.7
C2	38.5
E2	15.4

1st Order

	E1	G1	Bb1	C2	E2
E1	50	50			
G1	50	50			
Bb1		100			
C2			20	60	20
E2				50	50

2nd Order

		E1	G1	Bb1	C2	E2
E1	E1		100			
E1	G1		100			
G1	E1	100				
Bb1	G1	100				
C2	Bb1		100			
C2	C2			33.3	33.3	33.3
C2	E2					100
E2	C2				100	
E2	E2				100	

RHYTHM

0 Order

R. Patern	Probability
1	8.3
3	16.7
4	8.3
8	8.3
11	33.3
41	8.3
42	16.7

1st Order

	1	3	4	8	11	41	42
1						100	
3					100		
4				100			
8							100
11	33.3	33.3	33.3				
41							100
42					100		

2nd Order

		1	3	4	8	11	41	42
1	41							100
3	11	50		50				
4	8							100
8	42					100		
11	1						100	
11	3					100		
11	4				100			
41	42					100		
42	11		100					

Figure 5. Tables for Travelin' Riverside Blues

Synthesis

Synthesis is very similar to analysis, but the process is reversed.

Backing Chord File

The user supplies the generation program with a file containing the backing chord sequence for the tune. This chord sequence uses a representation similar to the one described above in Music and Program Representation. The only difference is that, instead of pitch information being supplied between parentheses, the duration of the chord in *beats* is given. The chord progression for an average blues tune would look something like:

kC *CM(16) *FM(8) *CM(8) *GM(4) *FM(4) *CM(8)

This sequence represents 4 measures (16 beats) of C, 2 measures of F, 2 measures of C and so forth. The program then generates a melody to fit this sequence.

Use of Tables

Since the rhythmic patterns determine the number of pitches needed, these are generated first. For the most part, second order tables are used for generating the rhythm and pitch - the exceptions being the start of a tune or phrase (zero order phrase start), the final note of a tune (rule) and zero frequency occurrences (zero and first order tables; see below). Although the initial pitch value for a long duration note may come from a second or perhaps first order table, it will be screened, and perhaps modified, by the rules governing acceptable pitches for long duration notes. This screening process is as follows. If the initial pitch of the long duration note is in the long duration pitch table then no change is made. If, however, the pitch does not appear in the table, then the initial pitch is changed to the closest pitch which is in the table.

The order of table use for generating a tune might look something like the following, depending on what rhythms are generated. The number of pitches needed, as determined by the rhythm, is given in parenthesis.

(1) Initial rhythmic pattern (2 pitches)	: 0 order rhythm tbl.
(2) First pitch	: Phrase start tbl.
(3) Second pitch	: first order pitch tbl.
(4) Second rhythmic pattern (1 pitch)	: first order rhythm tbl
(5) Third pitch	: second order pitch tbl.
(6) Third rhythmic pattern (rest)	: second order rhythm tbl

etc.

Zero Frequency and "Fallback"

A zero frequency problem occurs when the program generates a sequence of events that it hasn't seen during analysis. This can happen for both rhythm and pitch; however the two are dealt with in slightly different ways.

For rhythm, if a sequence of 2 rhythmic patterns is generated which was not seen during analysis, then the program will be unable to use the second order rhythmic table to generate the next pattern. It then has to "fall back" to the first order table and use the most recent rhythmic pattern to generate the next one. If, in the same way, the first order table can't be used then the program falls back and uses the zero order table. Fallback for rhythm can be forced by the rules which govern the placement of rhythmic patterns within measures. For example, if all the possible next patterns from the second order table given the previous 2 patterns are unacceptable for placement at the current location in the measure, then fallback to the first order table is forced. Fallback to the zero order table could be forced in the same way.

For pitch, fallback from second order to first order is the same as for rhythm. Since pitch is ordinal, zero frequency occurences in the first order table can be dealt with using the closest pitch (to the generated one) which is represented in the table to predict the next pitch For example, if the generated pitch is C# and the first order table contains the pitch C, and the next highest pitch is E, then the C is used to predict the next note as C# is closer to C than to E. This approach is used by the program.

Fallback to zero order pitch was necessary in the version of the program in which pitch was analysed by rhythmic pattern as well as by chord. This comes as a direct consequence of the forced fallback which can occur in the rhythm. If fallback to zero order rhythm is forced then a rhythm pattern might be generated which has never been seen under the current chord. Since pitch is analysed by rhythmic pattern this means that there would be no corresponding first or second order pitch table from which to generate the next pitch event. So in this case the program would fall back to zero order pitch.

2.2 Listening Test

A listening test was carried out in order to determine how well the program captures the structure of blues music.

Test Format

A cassette tape containing 10 pairs of tunes was made. In each pair of tunes one was computer composed and the other human composed. Listeners were asked to indicate, for each pair, which tune they thought was computer composed and which was human composed.

There were 58 tunes in the database. Ten tunes were randomly selected by drawing numbers from a hat. These 10 tunes, removed from the database, were the 10 human tunes used in the test. Twenty tunes were generated based on analysis of the remaining 48 tunes. Ten of these 20 computer generated tunes were then selected by the same method as above. Finally, the 10 human and 10 computer tunes were assembled into pairs by drawing 1 human and then 1 computer tune from different hats.

All tunes on the tape were played by a music program, and all were played against a standard harmonic background This was done for uniformity and to eliminate any bias that might result from human performance of the music.

Appendix C is the questionnaire used in the listening test. Participants indicated their choice by circling H or C for each tune in each pair, and then gave their decision a confidence rating.

Hypothesis

The listening test can be viewed as a series of Bernoulli trials [6]. If the program successfully captures the structure of blues melodies, then listeners should have trouble distinguishing between human composed tunes and computer tunes. In order to provide a distribution to test against, the following hypothesis was stated:

> Listeners can correctly distinguish human composed from computer tunes 90% of the time.

If the hypothesis is correct, results from the listening test should follow a binomial distribution with $p = 0.9$ and $q = 0.1$, where p is the number of 'successes,' or correct responses, and q is the number of 'failures,' or incorrect responses.

The hypothesis was tested, at the 0.01 level of statistical significance, by a Chi-square test for goodness of fit.

Subjects

A total of 198 people took part in the listening test. They came from a first year music course, a first year computer science course and a third year computer science course.

3. Results

Listeners made an average of 4.3 errors on the listening test, with a standard deviation of 1.8 errors. Figure 6 shows the distribution of listeners by number of errors.

Figure 7 shows the number of listeners who chose incorrectly on each pair. The human generated tune in pair 9 was the well known 'Kansas City', by Jerry Lieber and Mike Stroller [3]. While this tune was consciously recognised by only one listener, Figure 7 shows that people did better on that pair than on any other.

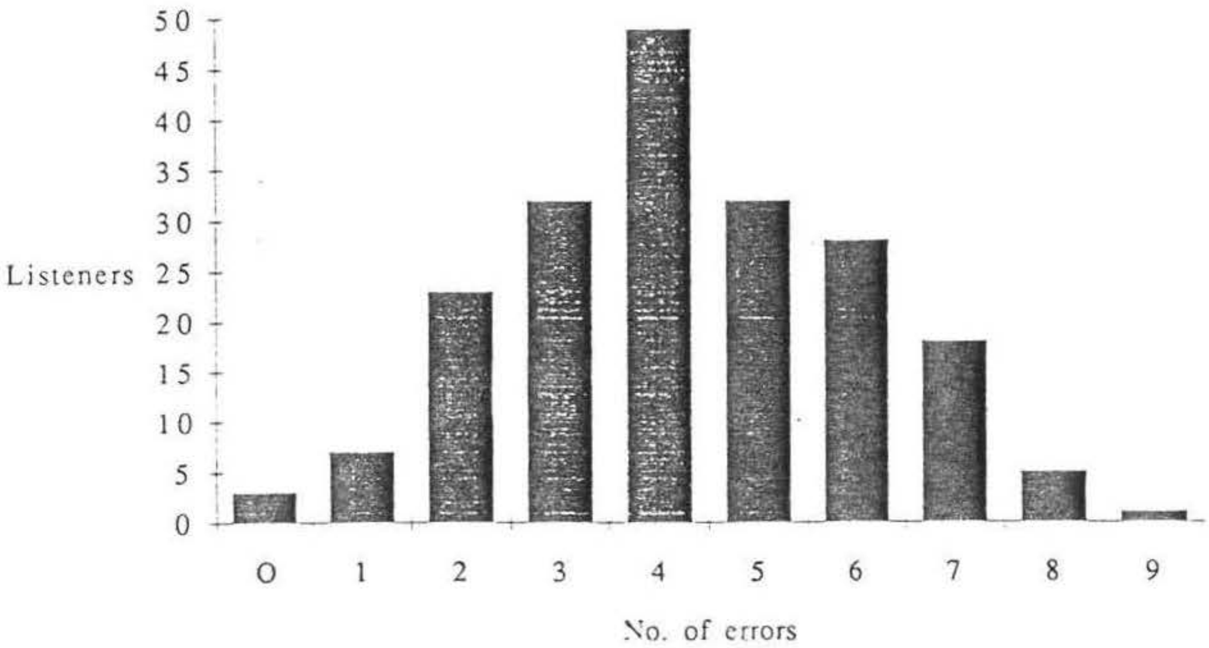


Figure 6. Distribution of listeners by number of errors

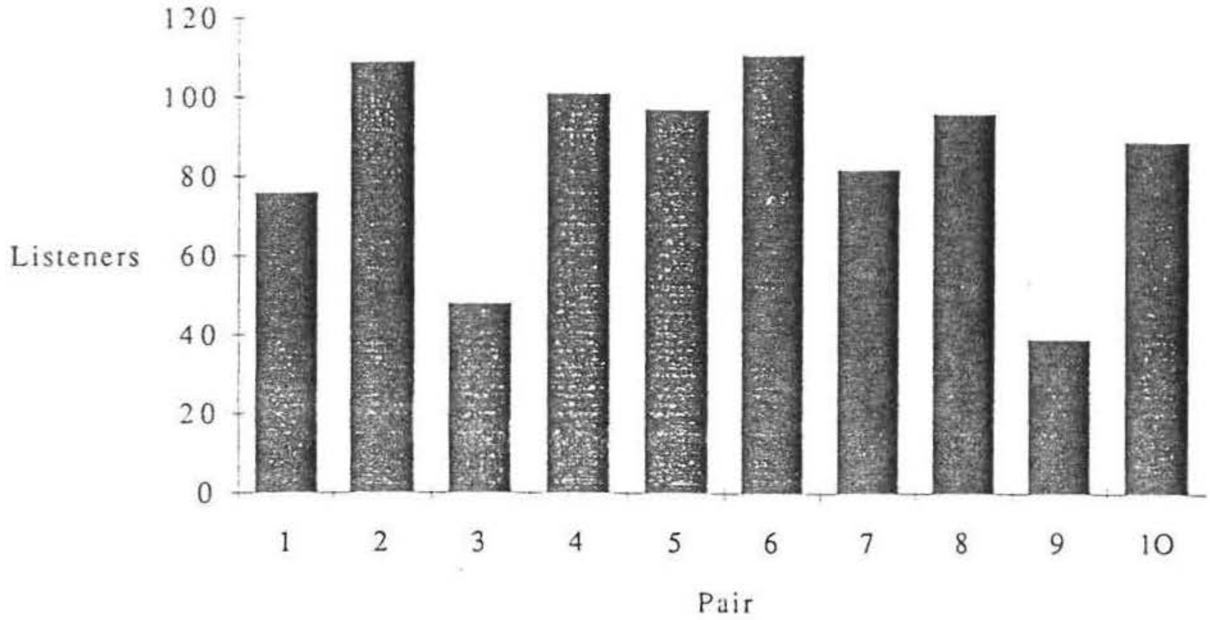


Figure 7. Number of listeners who chose incorrectly on each pair of tunes

Table 1 illustrates the Chi-square calculation, showing expected and observed numbers of listeners making less than two errors and two or more errors.

no. of errors	expected	observed	$(f_o - f_e)^2 / f_e$
0 or 1	145.74	10.00	126.43
> 1	52.26	188.00	352.57
	=========	=========	=========
	198.00	198.00	479.00

$X^2 = 479.00$

Table 1. Expected and observed numbers of listeners making errors

At the 0.01 level of significance, the Chi-square value falls well outside the acceptance region. Therefore the hypothesis stated above is rejected and it is concluded that people are unable to reliably distinguish between human and computer composed tunes.

Confidence

There was an average confidence rating of 2.8 and standard deviation of 1.1 for the pairs of tunes for which listeners identified incorrectly. For the pairs of tunes that listeners identified correctly, the average confidence rating and standard deviation were 3.0 and 1.2 respectively. There is no significant difference in listeners confidence between correct and incorrect answers at the 0.01 level.

Table Use

Table 2 shows, for each computer tune in the listening test, the total number of rhythm patterns, the total number of pitches and how frequently each table was used in generating the tunes.

	Tune									
	1	2	3	4	5	6	7	8	9	10
Rhythm										
Total patterns	38	31	36	41	36	31	32	37	44	39
Zero (%)	2.6	3.2	2.8	2.4	2.9	6.5	3.1	2.7	2.3	2.6
First (%)	2.6	6.5	11.1	4.9	5.6	6.5	9.4	13.5	15.9	7.7
Second (%)	94.7	90.3	86.1	92.7	91.7	87.1	97.5	83.8	81.8	89.7
Pitch										
Total pitches	38	44	44	44	42	40	39	49	52	47
Phrase (%)	5.3	2.3	2.3	4.6	4.7	2.5	7.7	4.1	1.9	4.3
First (%)	18.4	9.1	4.5	9.1	9.4	5.0	7.7	6.1	3.9	6.4
Second (%)	76.3	88.6	93.2	86.4	85.8	92.5	84.6	89.8	94.2	89.4

Table 2. Use of transition tables in computer generated tunes.

In general, second order tables produced 85 to 90% of the pitches and rhythms in the listening test tunes. Because of pitch substitution (described above), no pitches were generated by the zero order pitch table.

4. Conclusion

A program was written which composes blues tunes to fit a given chord structure. Listening tests showed that people were unable to reliably distinguish between human and computer composed tunes, indicating that, in some sense, the program captures the structure of blues melodies.

Quality

The listening test and its results do not reflect the quality of the music. Quality of music, especially blues, is strongly tied to its performance and is highly subjective. All the songs in the listening test were played by a computer, which certainly left a lot to be desired in the performance category. While it may have been preferable to have the tunes performed by musicians, it then becomes difficult to separate quality of music from quality of performance. Many people would say that the two are inseparable, that what makes the blues the blues is how it's performed. To a certain extent this is true, but, since a knowledgeable musician can create a stylistic performance from a written score, the score must capture something of the structure of the music. It is this structure, divorced from performance, that is captured and generated by the program described here. It should be noted that none of the participants of the listening test were told that the tunes they were about to hear were blues tunes. At any rate, the claim here is not that the program produces good blues - only that it produces adequate blues.

Psychological Plausibility & Future Work

Is this model psychologically plausible? Probably not as a complete model of human improvisatory process. There are many aspects such as phrasing, form and mood (to name a few) that play a role in the improvisatory process which the model does not take into account. However, within the program's viewpoint (short pitch and rhythmic sequences in a harmonic context) the model is psychologically plausible. Human short term memory has a capacity of 7 plus or minus 2 items [15]. The second order process is well below this limit; it is possible that a musician does not utilise the full capacity of his or her short term memory, given the speed at which music is often improvised.

The program produces melodies without having to store vast numbers of motifs for later use. This is desirable because it seems unlikely that musicians produce new improvisations based entirely on previously heard motifs. It can be argued that the transition tables capture the same information a human musician does when learning from existing musical examples within a particular genre. Since nothing is truly 'right' or 'wrong' in music the probabilities in the tables capture the degrees of 'rightness' or 'wrongness' and all the shades in between. In fact, extending the boundaries of a particular genre may include introducing something that was previously considered wrong, but in a new context.

However, the viewpoint of this program is a fairly narrow one. Higher level, broader viewpoints surely exist [5]. One such viewpoint might exist on the phrase level. As it stands the program has a very primitive view of what constitutes a musical phrase. A phrasal viewpoint containing information about the structure, length, and placement of phrases is needed. This viewpoint might exist under an even higher one, the viewpoint of form perhaps. Musical forms (eg. AB, ABA) govern the structure of pieces of music on the whole. This might be more applicable to composition than improvisation. Improvisation, by definition, has more freedom. Nevertheless, some aspects of these viewpoints would be applicable. Another area in which the program is lacking is in use of repeated motifs. Although it is unlikely that a human musician carries around thousands of previously heard motifs in his or her head, it is plausible that an existing motif of a tune may be modified and used later in the tune, or that a new one created during improvisation may be used later. These areas and the discovery of the true link between rhythm and pitch constitute possible areas of further research and improvement.

Appendix A
Principles of Blues Improvisation

The most common form of blues is the 12 bar blues, so called because it is made up of a repeating pattern of 12 bars of music. Although there are many variations of 12 bar blues, two of the most common are as follows:

Ex. 1

| I | I | I | I | IV | IV | I | I | V | IV |
| I IV | I V |

Ex. 2

| I | IV | I | I | IV | IV | I | I | V | IV |
| I IV | I V |

The Roman numeral between each of the bar lines indicate a chord and its relationship to the key. For example: If the key is A then a 'I' chord would be A, 'IV' would be D and 'V' would be E. The IV chord in the second bar of Ex.2 is called a quick change and adds a little more interest to the progression. The last two bars of the 12 bar blues are called the 'turnaround' since they prepare us for the repeat back to the beginning of the progression. There are many variations on the turnaround. Another common one is:

| I | I bV I V |

Other variations on the 12 bar blues include the use of 7th and diminished chords as passing chords to connect the I, IV and V chords.

The blues Scale

Every blues improviser uses the blues scale to a greater or lesser extent in his or her improvisations. The blues scale is simply a minor pentatonic scale with one additional note, the flatted fifth. Figure 8 shows the scale.

Figure 8. The blues scale in A

Since all the notes tend to agree with the chords in a 12 bar blues progression, it's almost impossible to hit a 'wrong' note. The b3 and b5 are examples of 'blue' notes or notes that are deliberately wrong when played against major chords. The tension created by these blue notes are part of what gives the blues it's distinctive sound.

The Major Pentatonic Scale

Unlike the blues scale, which because of the flatted 3rd and 5th has a hard edge, the major pentatonic scale is somewhat more consonant. Some musicians such as B. B King use the major pentatonic sound often; others, like Albert King, use it hardly at all. Some players use both the blues and major pentatonic scales. "Red House" by Jimi Hendrix is an example of this, although the blues scale is leaned on more heavily than the major pentatonic. Figure 9 shows this scale.

Figure 9. The major pentatonic in A

Appendix B
Transition Tables & Rules

Tables

Zero order pitch	probabilities of single pitches.
First order pitch	probabilities of sequences of 2 pitches; used to generate the next pitch given a previous one.
Second order pitch	probabilities of sequences of 3 pitches; used to generate the next pitch given 2 previous pitches.
Phrase start	A zero order pitch table. It is used to generate the initial pitch of a new song or phrase.
Zero order rhythm	probabilities of single rhythm patterns.
First order rhythm	probabilities of sequences of 2 rhythm patterns; used to generate the next rhythm pattern given a previous one.
Second order rhythm	probabilities of sequences of 3 rhythm patterns; used to generate the next rhythm pattern given 2 previous rhythm patterns.
Long duration	acceptable pitches for notes of half a measure or more in duration.

Rules

Long duration	screens pitches generated from zero, first or second order tables for notes of half a measure or more in duration..
Rhythmic placement	ensures the placement of rhythm patterns within measures obey the rules of music.
End of song whole note	ensures that the last note generated is a on the tonic closest to the previous pitch.

Appendix C
Listening Test Form

Please indicate (where applicable) your- theory grade level:
instrument grade level:
years studied instrument:

Instructions:

A tape of 20 tunes will be played. The tunes on the tape are arranged in pairs (TuneA & TuneB). While all of the tunes have been **played** by a computer, one of the tunes in each pair has been **composed** by a human and the other by the computer program (only the melody was composed by the program, the chord structure was provided). Prior to the start of each pair of tunes its number will be announced to help you keep track. For each pair of tunes do the following:

Step 1. Listen to the melody. On the left hand side circle either 'H' if you believe it was composed by a human or 'C' if you believe it was composed by the program.

Step 2. Now give the decision you made in Step 1 a confidence rating by circling a number on the right hand side of the paper.

Pair#	TuneA	TuneB	Confidence(circle one): unsure - very sure				
1.	H C	H C	1	2	3	4	5
2.	H C	H C	1	2	3	4	5
3.	H C	H C	1	2	3	4	5
4.	H C	H C	1	2	3	4	5
5.	H C	H C	1	2	3	4	5
6.	H C	H C	1	2	3	4	5
7.	H C	H C	1	2	3	4	5
8.	H C	H C	1	2	3	4	5
9.	H C	H C	1	2	3	4	5
10.	H C	H C	1	2	3	4	5

Appendix D
Composed Tunes

<u>Tunes Used In Listening Test</u>

Tunes for the listening test and for analysis were taken from All American Blues [8], Honkin' Blues [1] and Blues Method [3].

anonymous	After Hours
	High Price Blues
	Kansas City Blues
	Long Handeled Shovel
	Rabbit Foot Blues
	Step It Up and Go
Willie Dixon	The Seventh Son(first section, 12 bars)
Lightnin' Sam Hopkins	Ticket Agent
Robert Johnson	If I Had Possession Over My Judgement Day
Jerry Lieber & Mike Stroller	Kansas City

Tunes Used For Analysis

anonymous	Bad Luck Blues
	Broke and Hungry
	Chilly Winds
	Come Back Baby
	Dust My Broom
	Evil Hearted Man
	Frankie and Johnie
	Good Mornin' Blues
	I'm A Stranger Here
	Long Tall Daddy
	Lucky Number Blues
	New Stranger Blues
	Sportin' Life Blues
	Take This Hammer
	Wet Weather Blues
	You Don't Know My Mind
Big Bill Broonzy & Chas Segar	Key to the Highway

Plumber Davis & Jules Taub	Worry, Worry, Worry
Willie Dixon	The Seventh Son (Second section, 12 bars)
Lightnin' Sam Hopkins	Appetite Blues Breakfast Time House Upon the Hill My Suggestion Talkin' Some Sense
Lightnin' Sam Hopkins & Stan Lewis	Back Door Friend
Lightnin' Sam Hopkins & Jules Taub	Bad Luck and Trouble
Elmore James	Where Can My Baby Be
Robert Johnson	Crossroads. Hellhound On My Trail Kindhearted Woman Blues Last Fair Deal Gone Down

	Me and the Devil Blues
	Ramblin On My Mind
	Stones In My Passway
	Terraplane Blues
	Walkin' Blues
	When You Got A Good Friend
	Travelin' Riverside Blues
	.32-20 Blues
Richard M. Jones	Trouble In Mind
B.B King	Rock Me Baby
B.B King & J. Josea	Sweet Sixteen
B.B King & Jules Taub	Woke Up This Mornin'
James Moore	I'm A King Bee
	I'm So Sorry

James Moore &	
Jerry West	Buzz Me Baby
	Rainin' In My Heart
Sonny Boy Williamson	Mighty Long Time

Appendix E
Computer Generated Melodies

Tune1

Tune 2

Tune 3

Tune 4


Tune 5
C
F
C
G7
F7
C
1
4
7
10


Tune 6
C
F
C
F7
C
1
4
7
10


Tune 7
C
F
C
F
3
C
G7
F7
C
1
4
7
10


Tune 8
C
3
F
C
G7
3
F7
C
1
4
7
10


Tune 9
C
F
C
F
G7
F7
C
1
4
7
10


Tune 10
C
F
C
F
C
G7
F7
C
1
4
7
10

References

[1] *Honkin Blues,* Creative Concepts, Ojai, CA.

[2] Ames, C. 1989. "The Markov Process as a Compositional Model: A Survey and Tutorial." *Leonardo,* Vol 22, (2): 175-178.

[3] Clement, R. *Hal Leonard Guitar Blues Method,* Hal Leonard, Milwaukee, WI.

[4] Coker, J. *Improvising Jazz,* Prentice-Hall, Englewood Cliffs, N.J.

[5] Conklin, D. 1990. "Prediction and Entropy of Music". M.sc thesis, Department of Computer Science, University of Calgary.

[6] Feller, F. G. 1968. *An Introduction to Probability Theory and Its Applications,* Vol 1. 3rd ed. John Wiley & Sons, Inc., New York.

[7] Gardner, M. 1978. "White and brown music, fractal curves and one-over-f fluctuations." *Scientific American* April (4): 16-31.

[8] Haag, J. L., ed. *All American Blues,* Creative Concepts, Ojai, CA.

[9] Hiller, L. A. 1959. "Computer Music." *Scientific American* Dec (6): 109-121.

[10] Johnson-Laird, P.N. 1987. "Freedom and constraint in creativity." In *Creativity,* ed. R.J. Sternberg. Cambridge: Cambridge University Press.

[11] Johnson-Laird, P.N. 1987. "Reasoning, imagining, and creating." *Bulletin of the British Psychological Society* (40): 121-129.

[12] Johnson-Laird, P.N. 1988. "A case history in music: Musical Improvisation." *In The Computer and the Mind,* ed. F. Kermode. London: Fontana.

[13] Jones, K. 1981. "Compositional Applications of Stochastic Processes." *Computer Music Journal* , Vol 5, (2).

[14] Levitt, D. A. 1981. "A melody description for jazz improvisation." M.Sc thesis, Department of Electrical Engineering and Computer Science, MIT.

[15] Miller, G. A. 1956. "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, Vol 63, (1): 145.

[16] Olsen, H. F. 1967. *Music, Physics, and Engineering* (2nd ed.). New York: Dover, 430-434.

[17] Pinkerton, R. C. 1956. "Information theory and melody," *Scientific American,* Vol 194, (2): 77-86.

[18] Schottsteadt, W. "Automatic Counterpoint." In *Current Directions in Computer Music Research,* ed. Max V. Mathews and John R. Pierce.

[19] Ulrich, J. W. 1977. "The Anlysis and Synthesis of Jazz by Computer." from *5th International Joint Conference on Artificial Intelligence,* 865-873.

[20] Wyatt, K. 1990. "Blues Primer." *Guitar World,* Sept: 128-135.