Searching for Beauty in Music
—Applications of Zipf's Law in MIDI-Encoded Music

Bill Manaris, Valerie Sessions, and James Wilkinson
Computer Science Department
College of Charleston

Technical Report CoC/CS TR#2001-7-1
August 17, 2001

Where shall you seek beauty, and how shall you find her unless she herself be your way and your guide?
And how shall you speak of her except she be the weaver of your speech?

—Kahlil Gibran, *The Prophet*, p. 74

Abstract

This project applies Zipf's law on musical pieces encoded in MIDI. Our hypothesis is that this will allow us to identify musical pieces that humans find "pleasing, beautiful, harmonious." Specifically, we have identified an initial set of attributes (metrics) of music pieces on which to apply Zipf's law. These metrics include pitch of musical events, duration of musical events, the combination of pitch and duration of musical events, and several others. Our preliminary results—derived mostly by tedious manual processing—are encouraging. We are working on automating our metrics so that we can test our hypothesis on a wide variety of music genres (baroque, classical, 20th century, blues, jazz, etc.). If this project is successful, we plan to investigate how such metrics may be used to generate computer music that sounds "pleasing, beautiful, harmonious."

Keywords: Zipf's Law, Computer Music, Music Analysis, Music Generation, MIDI

1. Introduction

Computers have been used extensively in music to aid humans in analysis, composition, and performance [3, 4]. This is facilitated by the use of MIDI (Musical Instrument Digital Interface)—a coding scheme used to encode music data, such as pitch, duration and timbre [6, 7, 8].

In this project we plan to use MIDI renderings of various music genres (e.g., baroque, classical, 20th century, jazz, blues, rock) to investigate the applicability of Zipf's law in analysis, composition, and performance of music via computer.

1.1. Zipf's Law

Zipf's law, named after Harvard University's Linguistics professor George Kingsley Zipf (1902-1950), is the observation that phenomena generated by self-adapting organisms (e.g., humans) follow *the principle of least effort* [10, 17, 18]. In essence, this principle states that, in any environment containing self-adapting agents able to interact with their surroundings, such agents tend to minimize their overall effort associated with this interaction. That is, a system of such interacting agents tends to find a global optimum that minimizes overall effort [16]. This interaction involves some sort of exchange (e.g., information, energy, etc.). This agent interaction can be viewed as a phenomenon and every agent exchange as an event of this phenomenon.

Zipf discovered that if we plot the logarithm of the frequencies of all events in such a phenomenon against the logarithm of the rank of these events, we get a straight line with a slope of approximately –1 (see figure 1.).

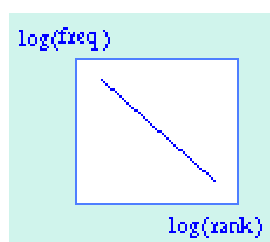

Fig. 1. Zipf's law [10].

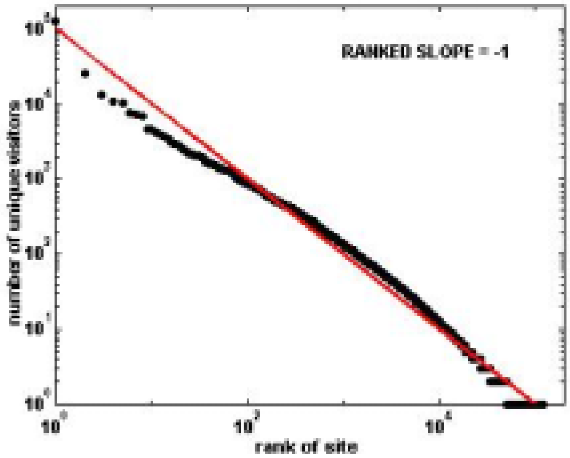

Fig. 2. Distribution of users among sites on the World-Wide-Web [1].

These results can be extended to include more than two interacting agents—producers, consumers, or both producers and consumers. Zipf initially demonstrated his idea using phenomena from natural language (e.g., words in a book). He successfully applied this principle to many other phenomena [17].

Incidentally, a similar theory was independently developed in the field of economics by Pareto (at the end of the 19th century) [2, 13]. Zipf's (and Pareto's) work has inspired and/or contributed to other fields studying the complexity of nature. Such fields include Fractals and Chaos Theory [11, 14]. Zipf's law has been extended to include similar distributions—distributions that produce a straight line, albeit with a slope other than –1. Such distributions are called *power-law* distributions. Power-law distributions have been encountered in many man-made and naturally occurring phenomena including city sizes, incomes, word frequencies, earthquake magnitudes, thickness of sediment depositions, extinctions of species, traffic jams, and visits of websites (see figure 2) [1, 2, 9].

2. Zipf's Law and Music

Voss and Clark have successfully applied Zipf's distribution in music [15]. Specifically, they measured several fluctuating physical variables, including output voltage of an audio amplifier, loudness fluctuations of music, and pitch fluctuations of music. Their samples included music from classical, jazz, blues, and rock radio stations. Their results show that pitch and loudness fluctuations in music follow Zipf's distribution. However they were unable to show this for note fluctuations.[1] This work was carried out at the level of frequencies in an electrical signal. Finally, Voss and Clark reversed the process so they could compose music through a computer. Their computer program used a Zipf's distribution (1/f power spectrum) generator to generate individual events. The results were remarkable:

> *The music obtained by this method was judged by most listeners to be much more pleasing than that obtained using either a white noise source (which produced music that was 'too random') or a 1/f^2 noise source (which produced music that was 'too correlated'). Indeed the sophistication of this '1/f music' (which was 'just right') extends far beyond what one might expect from such a simple algorithm, suggesting that a '1/f noise' (perhaps that in nerve membranes?) may have an essential role in the creative process.* [15, p. 258]

Nagai extends this line of reasoning as follows:

[1] Elliot and Atwell [5] were also report that the did not find Zipf's distribution in note fluctuations.

> *The science of complexity locates life at the edge of chaos. Life is neither dead disorder nor dead order, but a critical point of phase-transition from chaos to order. In other words the life principle is neither white noise nor monotone, but a 1/f fluctuation.*
>
> *We are often "beside ourselves" with ecstasy over beautiful music or scenery, while we feel an aversion to high entropy such as a din or filth and weary of low entropy such as monotonous sound or figure, that is to say, we keep away from the two extremes, pure low entropy and utter high entropy. When you analyze the power spectrum of sound frequency, you will find that beautiful music shows 1/f frequency. You can also find 1/f fluctuations in curves rich in changes characteristic of beautiful natural scenery* [12, p. 2].

3. Project Hypothesis

Our project focuses on the level of music notation, as opposed to power spectra. We are applying Zipf's law on music pieces encoded in MIDI. Our hypothesis is that Zipf's law will allow us to work at the level of music notation to identify musical pieces that humans find "pleasing, beautiful, harmonious." Specifically, we have identified an initial set of characteristics (metrics) of music pieces on which to apply Zipf's law. These metrics include pitch of musical events, duration of musical events, the combination of pitch and duration of musical events, and several others. Our preliminary results—derived mostly by tedious manual processing—are encouraging.

We are in the process of automating the application of our metrics so that we can test our hypothesis on a wide variety of music genres (baroque, classical, 20th century, blues, jazz, etc.). Assuming that our hypothesis proves correct, then a future direction of this project would be to investigate the reversal of the process to automatically (or semi-automatically) generate computer music that sounds "pleasing, beautiful, harmonious."

We further envision a collection of tools that could be used in music composition. Such tools, when applied properly, could enhance/facilitate artistic expression. Such tools may be analytical or generative in nature. For instance, an analytical tool could be easily developed to measure the "entropy" of music compositions along a variety of music-theoretic dimensions. This could provide quantitative feedback to a musician's creative process. Examples of such feedback include whether a piece is too monotone or too chaotic, and whether it is similar to, say, Bach's Toccata and Fugue along certain dimensions. In terms of music generation, tools could be developed for allowing a composer to supply a music phrase for automatic generation of variations.[2] This is analogous to the visual effects (transformations) available in graphic packages for manipulation/enhancement of images.

The following is a list of planned activities:

- Literature review on Zipf's law, chaos theory, fractals, and MIDI.
- Development of software to filter MIDI events found in MIDI files; for this we will reuse existing MIDI software resources [7].
- Research to locate MIDI files of interest (e.g., classical music, 20th century music, etc.). See [8] for examples.
- Application of Zipf's law on attributes of musical pieces encoded in MIDI. Such attributes include pitch, intervals, duration, chords, and others.
- Preliminary investigation of how above results could be used to algorithmically (help) compose music that sounds "pleasing, beautiful, harmonious."
- Report research results at professional conferences and/or in journals.

---

[2] Musical phrases that retain aspects of the original phrase but have been altered for artistic effect.

4. Preliminary Results

So far we have performed a preliminary literature review on Zipf's law, power laws, fractals, theory of chaos, and MIDI (see reference section).

In order to test our hypothesis that Zipf's law will allow us to identify musical pieces that humans find "pleasing, beautiful, harmonious", we have begun identifying a series of metrics that could be applied to musical pieces to discover if they incorporate instances of Zipf's distribution. We are also in the process of automating some of these metrics using Visual Basic and CAL (Cakewalk Application Language)—a LISP-and-C-based programming language specifically geared towards manipulating MIDI renderings of musical pieces. By automating these metrics, we should be able to quickly test our hypothesis on hundreds of musical pieces readily available in MIDI rendering.

4.1. Measurable Music Attributes

We have identified several attributes of music that could be used in deriving metrics for Zipf's distribution. We suspect that some pieces may exhibit Zipf's distribution in one or more dimensions. These dimensions include the following as well as combinations of the following: pitch, rests, duration, melodic intervals (horizontal), harmonic intervals (vertical), chords, movements, volume, timbre, tempo, dynamics. Some of these can be used independently, e.g., pitch; others can be used only in combinations, e.g., duration. Some lend themselves easily to the task, such as melodic intervals, whereas others not so easily, such as timbre. Of course, the fact that some dimensions may not seem good candidates for Zipf's distribution may be only because they have not traditionally been used as means of musical artistic expression, e.g., timbre.

4.2. Initial Metrics

Given our background as music listeners, we have identified several metrics (based on the above music attributes) to search for instances of Zipf's distribution. These were selected because they have traditionally been used to express musical artistic expression and creativity—they are studied extensively in music theory and composition. Obviously, this list is not complete. This is because there is probably no limit in the ways that sound could be used for artistic expression.

Derived metrics were manually applied to J.S. Bach's 2-Part Invention #13 in A minor – first 5 measures and 2 and 1/16 beats; this particular excerpt will be denoted as JSB#1 herein (see figure 3). This piece was selected because it seems "pleasing, beautiful, harmonious" to the authors.

Fig. 3. Excerpt JSB#1: J.S. Bach's 2-Part Invention #13 in A minor (first 5 measures and 2 and 1/16 beats).

The following is a summary of the derived metrics and examples of their manual application to JSB#1.

Simple Notes (pitch): This metric is derived by counting the instances of each of the 12 tones in the piece, e.g., number of A's, number of F#'s, etc. Figure 3 shows the result when applied to sample JSB#1.

Simple Notes (duration): This metric is derived by counting the instances of each possible note duration in the piece, e.g., number of whole notes, number of half notes, number of ¼ notes, etc. Figure 4 shows the result when applied to sample JSB#1.

Simple Notes (pitch & duration): This metric is derived by counting the instances of each possible combination of notes and corresponding durations in the piece, e.g., number of A whole notes, number of F# half notes, number of C ¼ notes, etc.[3] Figure 5 shows the result when applied to sample JSB#1.

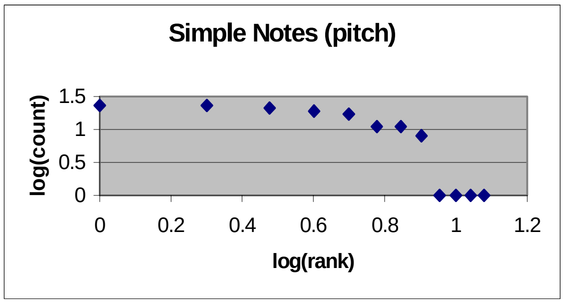

Fig. 3: Pitch of notes in JSB#1.

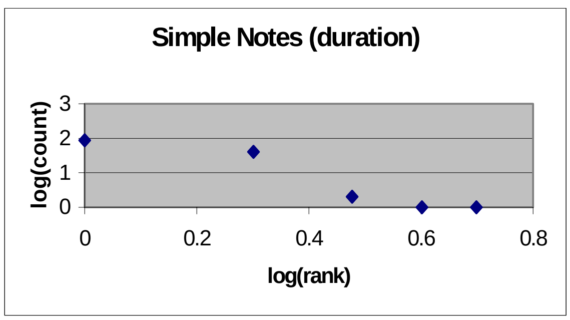

Fig. 4: Duration of notes in JSB#1.

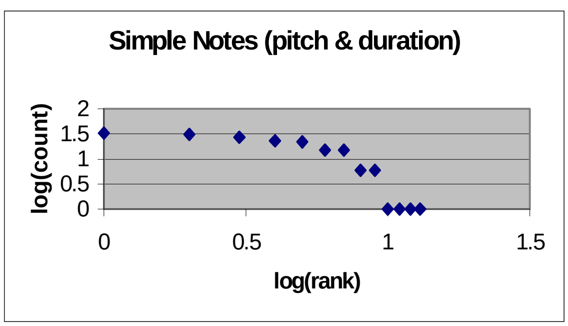

Fig. 5: Duration of tones in JSB#1.

[3] Another way to formulate this is to count the number of beats in which a particular pitch is heard.

Chords (pitch): This metric is derived by counting the instances of each type of chord in a piece, e.g., number of *i*, *ii*, *iv*, and *v* chords. We have not yet applied this metric on sample JSB#1.

Chords (duration): This metric is derived by counting the instances of each possible chord duration in a piece, e.g., number of chords extending one beat, number of chords extending two beats, number of chords extending three measures, etc. We have not yet applied this metric on sample JSB#1.

Chords (pitch & duration): This metric is derived by counting the instances of each possible combination of chords and corresponding durations in a piece, e.g., number of i chords extending one beat, number of i chords extending two beats, number of iv chords extending three measures, etc. We have not yet applied this metric on sample JSB#1.

Rests (duration): This metric is derived by counting the instances of each possible rest duration in a piece, e.g., number of whole note rests, number of half note rests, number of ¼ note rests, etc. Figure 6 shows the result when applied to sample JSB#1.

Melodic Intervals (interval only): This metric captures the change between consecutive pitches. It is derived by counting the instances of each possible melodic interval between consecutive notes within a voice, e.g., number of minor 2nds, number of major 2nds (M2's), number of perfect 5ths (P5's), etc.[4] Figure 7 shows the result when applied to sample JSB#1.

Melodic Intervals (interval & duration): This metric differs from the previous one in that it is an attempt to capture the effect of each interval in the overall melody, i.e., how pronounced an interval is in a melodic line. Specifically, some intervals are only passing, while traveling between neighboring tones, whereas others are more sustained. Since intervals themselves lack duration, to capture this we look at the durations of the notes producing the interval. A better way would be to count the time between the

Fig. 6: Duration of rests in JSB#1.

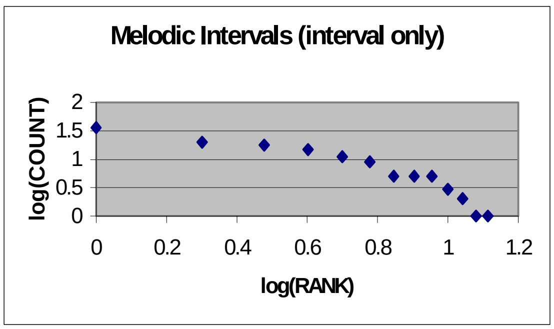

Fig. 7: Melodic intervals in JSB#1.

[4] Another way to formulate this is to count half tone steps between consecutive notes.

start of the first note to the end of the second note – this would account for *staccato* notes. To simplify measurement, we add the two note durations and use this to characterize the durational quality of the interval. For example, M3's spanning a whole note (generated by two half notes, or a quarter and a dotted half notes, etc.), M3's spanning ¼ note, M3's spanning 1/16 note, etc. We have not yet applied this metric on sample JSB#1.

Harmonic Intervals (interval only): This metric is derived by counting the instances of each possible melodic interval between notes that sound together across all voices, e.g., number of minor 2nds (m2's), number of major 2nds (M2's), number of perfect 5ths (P5's), etc.[5] For example, given a C major chord, the notes C, E, and G are heard together; these produce the following intervals: M3, m3, and P5. Figure 8 shows the result when applied to sample JSB#1.

Harmonic Intervals (interval & duration): This metric differs from the previous one in that, for each possible interval (e.g., m3, M3, P5), it also captures the variability in duration, e.g., M3's spanning a whole note, M3's spanning a ¼ note, M3's spanning a 1/16 note, etc. We have not yet applied this metric on sample JSB#1.

All metrics applied to sample JSB#1 produce graphs that appear to exhibit a Zipf-like distribution, albeit some better than others. The most promising ones are figures 7 and 8. This is because interval related metrics tend to be independent of/unaffected by the quality of the performance or MIDI rendering of a piece.[6] It should be noted that our sample JSB#1 exhibits much variability in the types of events captured by these metrics. On the other hand, the least interesting result is associated with a metric that encountered only four types of events in JSB#1 (see figure 6).

Higher-Order Metrics: The interval-based metrics mentioned above capture change in pitch – they can be thought of as derivative functions. Following this analogy, we plan to investigate higher-order interval metrics – metrics which capture the change of intervals, or even the change of the change of intervals. Such metrics, although computationally intensive, may prove invaluable in capturing higher aspects of music structure/content. Perhaps, they could be used simultaneously to model musical pieces similarly to the multi-level models used in natural language processing (e.g., lexical, syntactic, semantic, and pragmatic models).

For instance, using only the melodic interval metric of, say, JSB#1 to generate a "similar" piece, one might produce a piece which begins with a cluster of all A notes, followed by all C notes, etc. Using the next-order interval metric, which captures changes of intervals, would alleviate this mindless clustering of notes; however, it could introduce clusters of intervals, e.g., all M3 intervals together, followed by all m3 intervals, etc. Nevertheless, such a piece would be more "interesting" than the earlier one. Obviously, this process of introducing higher-order interval metrics (derivatives) will soon degenerate to metrics

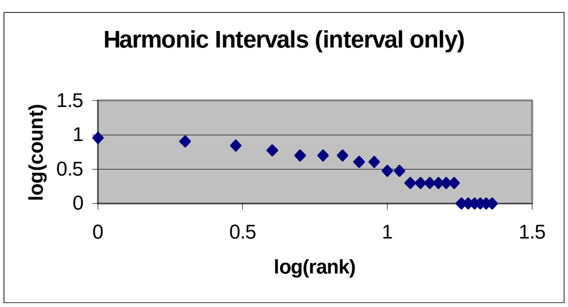

Fig. 8: Harmonic intervals in JSB#1.

[5] Another way to formulate this is to count half tone steps between concurrent notes.

[6] We suspect they capture aspects of music used by human listeners to identify a particular piece. For instance, human listeners use mostly melodic intervals to identify a melody.

capturing near-zero change – similarly to higher-order derivatives. But, perhaps, the information captured by all these metrics together would be sufficient to model significant aspects of music style or mood and thus be useful in music analysis and generation.

4.3. Preliminary Automated Results

In addition to the manually-derived examples above, we have produced a few more examples using a prototype implementation. This prototype automates three metrics (pitch, duration, pitch & duration). It was implemented using CAL. Four pieces were chosen to be evaluated using these metrics: Bach's 2-part Invention #13 in A minor, Beethoven's Für Elise, Chopen's Impromptu, and Schönberg's Mondestrunken. While the results from both the pitch metric and duration metric were not promising, the graphs obtained for each piece using the pitch & duration metric resemble the results we were hoping to achieve (see figures 9, 10, 11, and 12).

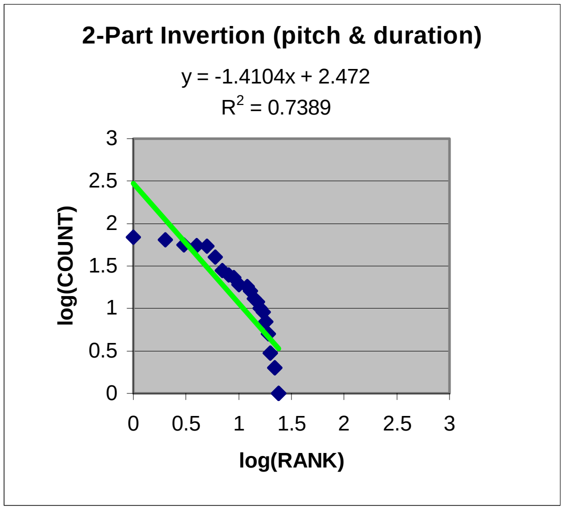

Fig. 9: Bach's 2-Part Invention #13 in A minor (slope = –1.4104).

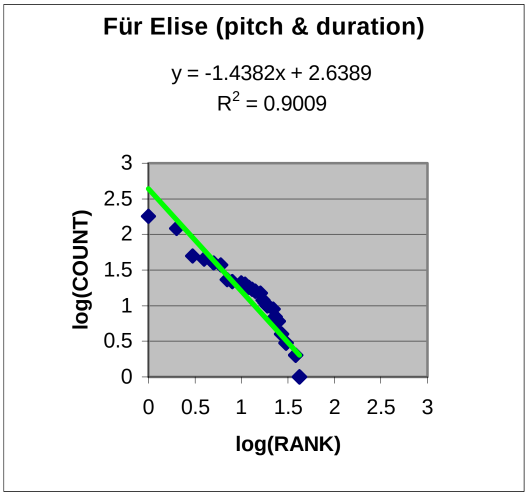

Fig. 10: Beethoven's Für Elise (slope = –1.4382).

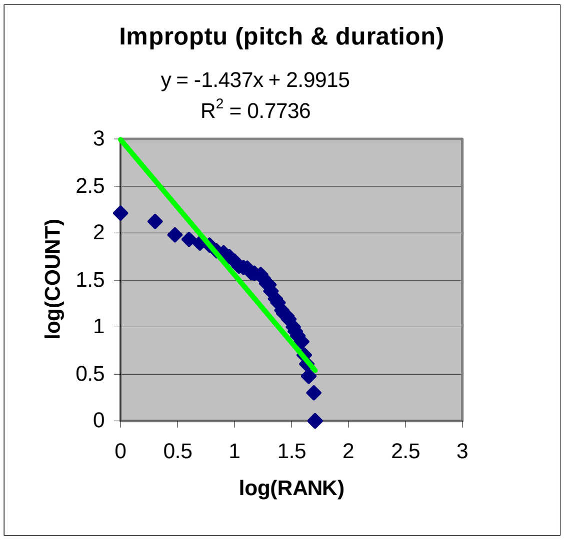

Fig. 11: Chopen's Impromptu (slope = –1.437).

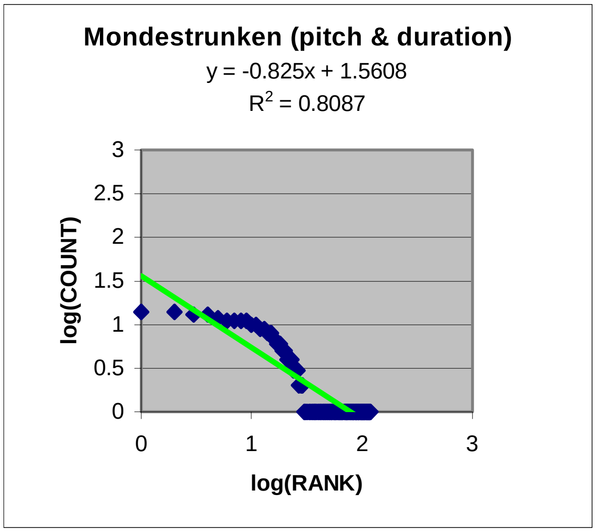

Fig. 12: Schönberg's Mondestrunken (slope = –0.825).

Obviously, these results are preliminary. However, they demonstrate that our hypothesis has promise. It is clear that, given our pitch & duration metric, each of the pieces produces a straight line with a negative slope near –1. We also see that pre-20th-century pieces produce graphs with similar slopes tending towards order (lower entropy), while Schönberg's piece produce a graph with a slope tending towards randomness (higher entropy). It is not clear if the pattern observed in these results will generalize, i.e., whether it holds for all music, or it is coincidental. Obviously, applying this metric to a wide variety of pieces will provide an empirical answer to this question.

Since CAL is not a complete high-level programming language (for instance it lacks file I/O), our newest prototype is being implemented in Visual Basic.

5. Conclusion

This is a preliminary report on a project that applies Zipf's law on musical pieces encoded in MIDI. Earlier research has demonstrated that the power spectrum of sound frequency in beautiful music approximates a Zipf-like distribution. Not-so-beautiful music and non-musical sound phenomena (e.g., radio talk shows) do not approximate a Zipf-like distribution [15].

In this project we are searching for different ways to apply Zipf's law in order to identify musical pieces that humans find "pleasing, beautiful, harmonious." Preliminary results are encouraging. We have started to automate our metrics using CAL and Visual Basic. This will allow us to test our hypothesis on a wide variety of composers and music genres (e.g., baroque, classical, 20th century, blues, and jazz).

5.1. Future Directions

We expect that our metrics, regardless of whether or not they support our hypothesis, may provide a "signature" mechanism—a way to classify musical pieces in terms of composer or genre. Such classification may be implemented through connectionist means. A neural network could be easily trained on arithmetic results (e.g., slope of trend line in graph) from applying our metrics to a wide variety of musical pieces. Such a neural network could perhaps be successful for music identification and/or classification tasks.

Assuming that our metrics prove successful, they could easily be incorporated into a collection of tools for computer-aided music composition. Such tools may be analytical or generative in nature.

In terms of analysis, such tools could be used by music composers for formative evaluation—that is to measure the entropy/balance (beauty?) of a musical piece under development. As mentioned earlier, examples of such feedback include whether a piece is too monotone/random, and whether it is similar to, say, Bach's Toccata and Fugue along certain music-theoretic dimensions. Obviously such metrics should be used with caution, similarly to readability metrics in natural language composition. Such metrics are, at best, crude abstractions which, if followed blindly, could limit a composer's creativity and range of expression. On the other hand, if applied properly, they could enhance/facilitate artistic expression.

In terms of music generation, we would like to investigate how such metrics could be used in computer-aided music composition. Specifically, using Artificial Intelligence techniques (e.g., genetic algorithms), one could perhaps reverse the process incorporated in the above metrics and generate computer music that sounds "pleasing, beautiful, harmonious." A composer could perhaps contribute an initial seed (a melodic outline) and have the system expand it automatically. Alternatively, tools could be developed for allowing a composer to supply a music phrase for automatic generation of variations. This would be similar to various visual effects (transformations) available in all mainstream graphic packages for manipulation/enhancement of photographs.

References

1. Adamic, L.A., and Huberman, B.A. (2000), "The Nature of Markets in the World Wide Web", *Quarterly Journal of Electronic Commerce*, **1(1)**:5-12.
2. Adamic, L.A., (1999), "Zipf, Power-laws, and Pareto - a Ranking Tutorial", www.parc.xerox.com/istl/groups/iea/papers/ranking/
3. Dobrian, C. (1988), "Music Programming – An Introductory Essay", www.arts.uci.edu/dobrian/CD.MusicProgramming.htm
4. Dobrian, C. (1992), "Music and Artificial Intelligence", www.arts.uci.edu/dobrian/CD.music.ai.htm
5. Elliot, J. and Atwell, E. (2000), "Is Anybody Out There? The Detection of Intelligent and Generic Language-Like Features", *Journal of the British Interplanetary Society 53(1/2)*, pp. 13-22, www.comp.leeds.ac.uk/eric/jbisjournal2000.ps
6. Glatt, J., "Tutorial for MIDI Users", www.borg.com/~jglatt/tutr/miditutr.htm
7. Glatt, J., "MIDI Utilities", www.borg.com/~jglatt/progs/software.htm
8. Glatt, J., "MIDI Files", www.borg.com/~jglatt/files/midifile.htm
9. Knuth, K. (1997), "Power Laws and Hierarchical Organization in Complex Systems—From Sandpiles and Monetary Systems to Brains, Language, and Music", *CUNY Cognitive Science Symposium*. http://bulky.aecom.yu.edu/users/kknuth/complex/powerlaws.html .
10. Li, W. (2000), "Zipf's Law" http://linkage.rockefeller.edu/wli/zipf/
11. Mandelbrot, B.B. (1977), *The Fractal Geometry of Nature*, W.H. Freeman and Company.
12. Nagai, T. (2000), "The Rhythm of Life", www.themestream.com/articles/133106.html
13. Pareto, V. (1897), *Cours d'Economie Politique*, Rouge (Lausanne et Paris). Cited in [10].
14. Schroeder, M. (1991), *Fractals, Chaos, Power Laws*, W.H. Freeman.
15. Voss, R.F., and Clarke, J. (1975), "1/f Noise in Music and Speech", *Nature* 258, pp. 317-318.
16. Wegner, P. (1997), "Why interaction is more powerful than algorithms", Communications of the ACM 40(5), 80-91, www.acm.org/pubs/articles/journals/cacm/1997-40-5/p80-wegner/p80-wegner.pdf .
17. Zipf, G.K. (1949), *Human Behavior and the Principle of Least Effort*, Addison-Wesley.
18. Zipf, G.K. (1965), *Psycho-Biology of Languages*, MIT Press (original publication Houghton-Mifflin, 1935).