**Ryan Nikolaidis, Bruce Walker, and Gil Weinberg**
Georgia Tech Center for Music Technology
840 McMillan St
Atlanta, Georgia 30320, USA
ryannikolaidis@gmail.com
bruce.walker@psych.gatech.edu
gil.weinberg@coa.gatech.edu

# Generative Musical Tension Modeling and Its Application to Dynamic Sonification

This article presents a novel implementation of a real-time, generative model of musical tension. We contextualize this design in an application called the Accessible Aquarium Project, which aims to sonify visually dynamic experiences through generative music. As a result, our algorithm utilizes real-time manipulation of musical elements in order to continuously and dynamically represent visual information. To effectively generate music, the model combines low-level elements (such as pitch height, note density, and panning) with high-level features (such as melodic attraction) and aspects of musical tension (such as harmonic expectancy).

We begin with the goals and challenges addressed throughout the project, and continue by describing the project's contribution in, and comparison to, related work. The article then discusses how the project's generative features direct the manipulation of musical tension. We then describe our technical choices, such as the use of Fred Lerdahl's formulas for analysis of tension in music (Lerdahl 2001) as a model for generative tension control, and our implementation of these ideas. The article demonstrates the correlation between our generative engine and cognitive theory, and details the incorporation of input variables as facilitators of low- and high-level mappings of visual information. We conclude with a description of a user study, as well as self-evaluation of our work, and discuss prospective future work, including improvements to our current modeling method and developments in additional high-level percepts.

## Previous Work

After originating in the early 1950s, computer-based generative music branched into several different directions. The probabilistic generative approach we take in this project can be related to the pioneering work of Lejaren Hiller and Leonard Isaacson, who premiered their algorithmic composition *Illiac Suite*, for string quartet, in 1957 (Belzer, Holzman, and Kent 1981). One of the techniques that Hiller and Isaacson used was the Monte Carlo method, where, after randomly generating a note, an algorithm tested it against a set of compositional rules. If the note passed the test, the algorithm accepted it and began generating the next note. If the proposed note failed the test, the algorithm erased it and generated a new note that was again tested by the rules. Although this approach produced melodic and even contrapuntal examples that followed certain voice leading principles, this algorithm had no higher-level model for the structure of the piece.

Our approach is also informed by David Cope's Experiments in Musical Intelligence, which sought to capture both high- and low-level features of compositions in order to generate stylistically authentic reinventions of music. His early work in this field, in the 1980s, revolved around the concept of defining a set of heuristics for particular genres of music and developing algorithms to produce music that recreates these styles. By Cope's own account, these early experiments resulted in "vanilla" music that technically followed predetermined rules, yet lacked "musical energy" (Cope 1991). His succeeding work built on this research with two new premises: every composition had a unique set of rules, and an algorithm determined this set of rules autonomously. This was in contrast to his previous implementation, where a human realized the rule set. This work ultimately relies on pattern recognition for analysis and recombinancy for synthesis, in an effort to create new musical material from pre-existing compositions. Although this implementation produces effective reconstructions true to the form of the original composition, it does

Computer Music Journal, 36:1, pp. 55–64, Spring 2012

not have the ability to generate music in real time (Cope 1991).

Belinda Thom and François Pachet each developed software that addressed the challenges of real-time generative algorithms with authentic musicality. In 2001, Thom completed the first generation of Band-Out-of-the-Box (BoB; Thom 2001). Her work relies on two models for improvisational learning. First, with previous knowledge of the work's harmonic structure, an offline algorithm listens to solo improvisations and archives probabilistic information into histograms. Then, in real time, BoB analyzes a human player's solo improvisation for modal content. Based on this content and the information learned offline, BoB then generates its own solo improvisation. From here, in the classic jazz tradition, both human and computer *trade fours* (each taking turns individually improvising for four bars of music) for the remainder of the performance. Although BoB provides real-time improvisation and does so in a nearly human manner, the previously determined harmonic structure limits the work's versatility. Pachet's Continuator (Pachet 2002), on the other hand, builds on harmonic and melodic content from human performances to generate improvisational responses. The Continuator employs a series of Markov chains to uniquely define voice leading used throughout a segment of human improvisation. These chains, combined with the detection of the improvisation's chord content (based on discrete time segmentation of note clusters), allow the algorithm to seamlessly continue and build upon human performance.

Similar to all of these projects, our algorithm uses weighted probabilities to generate music. In the tradition of Pachet and Thom, we use a real-time generative algorithm. Unlike Thom's BoB and Pachet's Continuator,however, which rely heavily on live human performance to drive their real-time generation, our autonomous process uses parameters determined by dynamic visual information (specifically, the movement of fish in an aquarium) as input. In addition, our work develops previously unexplored areas of generative musical tension.

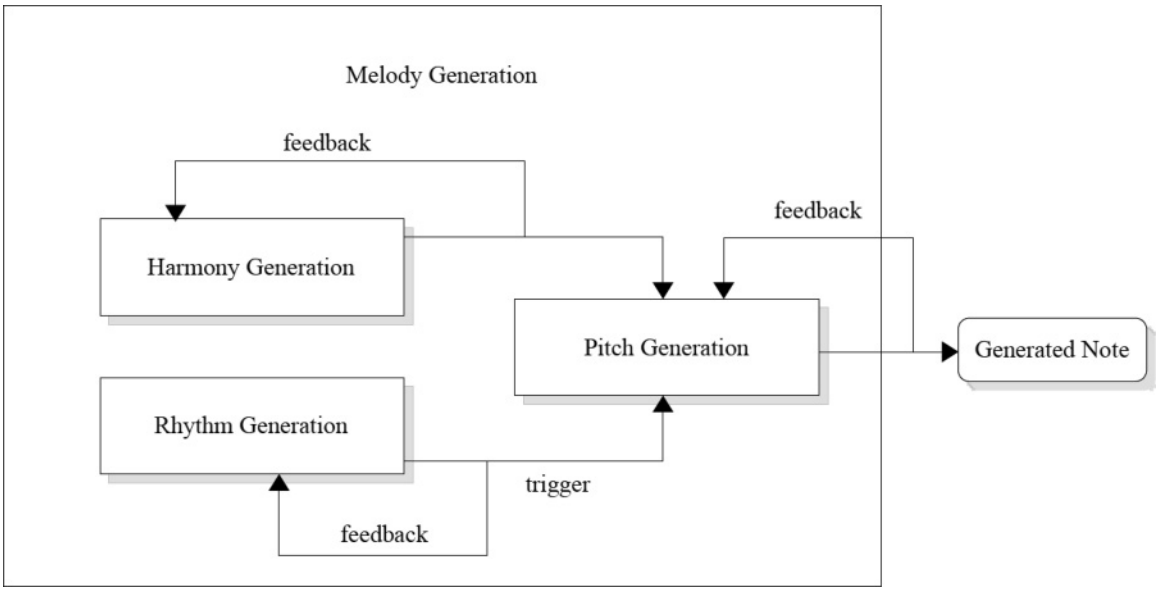
## Design

A stable groundwork of relatively independent modules, capable of continued additions and evolution, was our primary design goal. To this end, our design focused on the development of a simple yet robust algorithm. It is simple in that the design does not rely on a complex network of rules and conditions; it is robust in that the music produced by the algorithm should be capable of effectively representing a diversity of musical gestures. In order to permit our music system to operate in real time, we designed and implemented the generative components in Max/MSP. Our design consists of three major components: pitch-, rhythm-, and harmony-generation modules, as shown in Figure 1. The three modules interact to generate the notes of a single-voice melody. Rhythm generation (which determines the onset time of the notes) triggers pitch generation (which determines the pitch based on the current state of harmony generation). Output from the harmony module informs pitch selection by generating chords that contain "anchoring" tones, or tones to which pitches are attracted. This will be explained in further detail subsequently. All three modules behave as state machines, relying on feedback of the previous state to determine the next state.

In the context of applying the generative algorithm for sonification, we drive these generative modules with input from computer-vision-tracked fish. Our system uses OpenCV (Agam 2006), an open-source library of image-processing algorithms designed for computer-vision applications. With images from a single Prosillica camera, the system works with models of fish, based on their general size, shape, and color. This allows the system to effectively identify and track each fish's independent movement. Using this data, our generative music algorithm represents the experience of viewing the aquarium.

In order to map both low- and high-level visual parameters to musical parameters, we segmented the various attributes of the visual information we wanted to represent. At the lowest level, we decided to convey simple location-based information such as the position of a fish at any given time. Additionally, we wanted the sonification to depict gestural

*Figure 1. The interaction between rhythm-, pitch-, and harmony-generation modules used to generate the next note.*




information about their movements by mapping the speed of their gestures to the rhythms of the generated music. With respect to higher-level features, we decided to represent (1) the general ambiance in the aquarium by changes in harmonic expectancy, and (2) individual behavior, such as predictable or erratic swimming patterns of the fish, by relative melodic tension. The latter led to the design and application of the generative tension algorithm described in this article.

## Implementation

We divide the task of implementation into tracking visual tension and mapping this tracking to the generation of music.

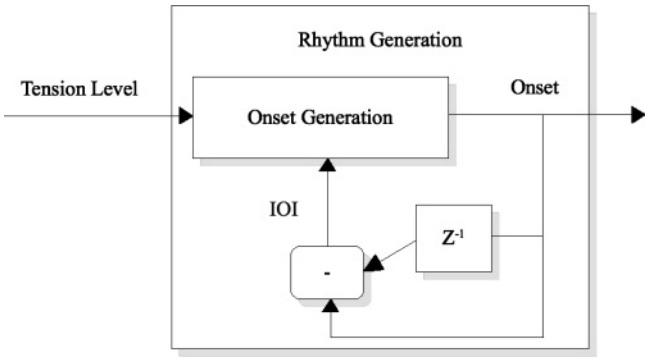
### Tracking Visual Tension

One of our sonification goals was mapping visual tension to musical tension. In order to detect visual tension, we developed a measurement of the flow of fish movement. This calculation assigns lower numbers to consistent movements and higher numbers to erratic movements. We define visual gestures involving multiple rapid changes in direction as erratic and unexpected behavior. A component was developed to detect directional information and reveal the nature of the fishes' gestures.

The first difference of each $x$ and $y$ coordinate indicates a direction vector. Comparing this direction vector to the previous one reveals whether the tracked fish has changed direction. The summation of the number of changes in the tracked fish's direction over a given period of time provides the expectancy of its movements. In particular, we use a running sum over a period of three seconds. A maximum threshold of ten changes in direction, across the running sum, is chosen to indicate a maximum visual tension level, and zero changes in direction is chosen to indicate a minimum visual tension level. These visual tension values map directly and linearly to the input tension values of the generative music tension algorithm.

The sonic tension levels (converted from the visual tension levels) influence the generation of harmonic, melodic, and rhythmic features. Thus, as the tracked fish changes from flowing movements to disjunct movements, the melody corresponding to that fish changes from less to more tense.

*Figure 2. Rhythm generation based on tension level and previous inter-onset interval.*




## Rhythm Generation

We based the rhythm generation module on a model proposed by Desain and Honing (2002) for analysis of rhythmic stability. Their work demonstrated the relationship between rhythmic stability and the bounds between contiguous inter-onset intervals (IOIs). In particular, they showed direct proportionality between the complexity of ratios between contiguous durations and relative rhythmic stability.

Extending the concept for analyzing stability into a predictive model, we implemented a method for rhythmic generation. In our predictive implementation, the algorithm refers to previous IOIs to inform the generation of future onsets, as shown in Figure 2. Specifically, provided a high or low input tension level, the algorithm accordingly gives preference to future onsets that form either complex or simple ratios, respectively, with the previous IOI.

The onset prediction relies on a lookup table in order to pseudo-randomly generate future onsets. Its lookup table includes a list of ratios arranged according to complexity, where ratios such as 1/2 and 2/1 occur low on the list, whereas 9/2 and 2/9 occur significantly higher. Influencing the pseudo-random generation, high input tension values give weight to ratios high on the list, and, vice versa, low tension values give weight to lower ratios.

In our sonification context, we continuously map the speed of the fish movements to the note density, as shown in Figure 3.

In this case, the algorithm combines the note density mapping with the rhythmic stability prediction. To do so, the algorithm first considers the influence of the speed mapping. This determines the relative note density. The onset generation then pseudo-randomly generates the next onset with a more or less complex ratio between IOIs, but also weights the lookup table probabilities based on distance from the relative note density. As such, a fish's speed maps directly to the density of notes, and the visual tension maps to the input tension value of rhythmic stability (as described earlier).

## Harmony Generation

Harmony refers to the pitch relationships between groups of notes that are simultaneous or close together in time, and it typically governs the choice of pitches in simultaneous, independent melodies (polyphony). The harmonies generated by our algorithm influence the movement of each melody. As we will explain in the Melody Generation section, the notes rely on attraction to harmonic anchoring tones. In stable conditions, melodies move towards the harmonic tones.

As listeners, we have expectations about the movement from one harmony to the next. For years, researchers have studied these expectations. Through subjective and physiological response studies, many have found a correlation between harmonic expectations and chords related by the circle of fifths (see Figure 4), a theoretical model that orders pitches according to a regular interval shift of seven semitones (or five diatonic scale steps) (Justus and Bharucha 2001; Steinbeis, Koelsch, Sloboda 2006).

Similar to the rhythm-generation module, harmony generation depends on a lookup table to generate the next harmony. We wanted to limit the scope of the harmonic possibilities in order to rely on a simple model of harmonic expectation. In doing so, we limited the lookup table to diatonic triads of a major scale. We ordered the table according to expectation. Based on the last harmony generated, we calculate expectation from movement on the circle of fifths. Low values on the table—values that are more expected—correspond to small movements on the circle of fifths. Higher values, relating to

*Figure 3. Mapping speed to note density.*

*Figure 4. The circle of fifths, a theoretical model of harmonic relationships.*

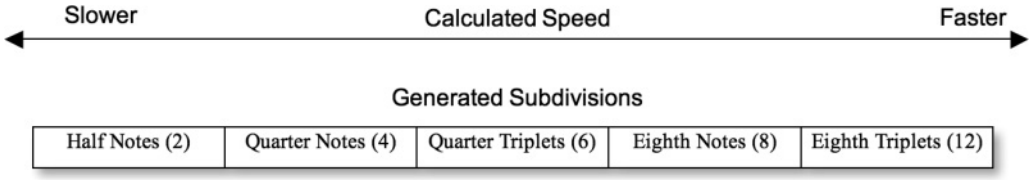



*Figure 3.*

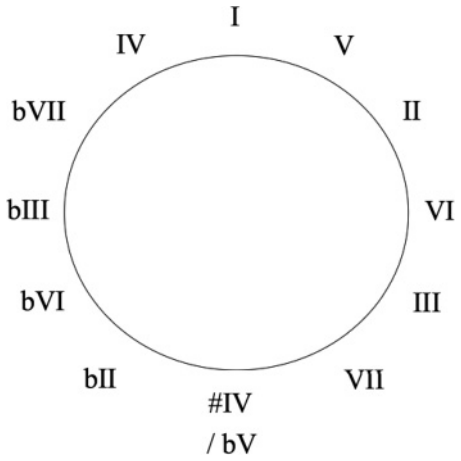



*Figure 4.*

more unexpected and therefore tense harmonic shifts, correspond to large movements on the circle.

A harmonic tension value influences the generation of the next harmony. Again, as with rhythm generation, higher tension values weight the probability of generating a more unexpected harmony. Conversely, low tension values increase the chance of the algorithm generating a low table value, an expected harmony.

Returning to our sonification example, we drive the harmonic tension value with a global visual tension value. As discussed in the Tracking Visual Tension section, the algorithm derives local tension values from the movements of each tracked fish. By summing all of these local values, the system generates a global visual tension value, which essentially describes the overall activity in the aquarium. As harmony generation globally affects all of the individual local melodies corresponding to each fish, we map the global visual tension to the harmonic tension value.

**Melody Generation**

We developed a method for pitch generation that could controllably change melodic stability and tension in real time. We based our method of melody generation on Fred Lerdahl's theories of tonal pitch space (Lerdahl 2001). Compared to similar work in the same field (Narmour 1992; Margulis 2005), Lerdahl's research in cognitive theory addresses in detail the concepts of stability and tension. Although Lerdahl originally intended this work as a theoretical means of deciphering relative stability, Nattiez (1997) described these formulas as unproven and bearing limited usability as an analytical tool. It has been shown more recently, however, that these formulas can be used effectively in a generative and interactive manner (Farbood 2006; Lerdahl and Krumhansl 2007).

Our implementation is based on Lerdahl's analysis of voice leading, which depends on two major components: anchoring strength and relative note distance. The concept of anchoring strength maintains that, given a certain pitch space value, there remain areas of greater and lesser attraction.

Our algorithm uses the input harmony to determine the anchoring-strength pitch space values. The 0 value in Table 1 represents the root of any harmony, 11 represents its leading tone, and values 1 through 10 correspond to the ten notes in between. The values 0, 4, and 7 have the strongest anchoring strength, and these pitch classes correspond to the tones of a major triad. The anchoring strength of each pitch class directly affects its probability of being chosen as the next pitch.

Our system depends on generating the probability for any possible next note provided the previous note. It also derives the probability for any given note to sound an octave above or below the previous note. Given a certain harmony, we wanted a unique

**Table 1. Anchoring-Strength Table for Computing the Attraction Between Pitches**

| *Strength* | *Basic Pitch Space (0 = tonic, 11 = leading tone. . . )* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 0 | | | | | | | | | | | |
| 3 | 0 | | | | 4 | | | 7 | | | | |
| 2 | 0 | | 2 | | 4 | 5 | | 7 | | 9 | | 11 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

**Table 2. Relative Note Distance**

| *G♭* | *G* | *A♭* | *A* | *B♭* | *B* | *C* | *C♯* | *D* | *D♯* | *E* | *F* | *F♯* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

anchoring-strength set within two octaves and, as such, we extended Lerdah's single octave anchoring-strength set, Table 1, to 24 columns. We extended it by adding columns left of 0, therefore providing an anchoring set one octave below any tone. This adjustment extended the opportunity for more precise manipulation of the equations.

The other major component of Lerdahl's voice leading equation relies on relative note distance. In terms of our generative algorithm, this measures the distance between the most recent pitch value and all prospective pitch values. The center of Table 2 represents the previous pitch—in this example, C. The relative note distance grows as notes move farther away from C. This distance inversely affects the probability of selection as a following note. (C to C has a distance of 1 to avoid division by 0.) Accordingly, there is a generative preference towards smaller melodic intervals.

In Lerdahl's stability equation for voice leading (Equation 1), the effect of the next note's stability is inversely proportional to the previous note's anchoring strength:

$$S = \left(\frac{a_2}{a_1}\right)\left(\frac{1}{n^2}\right), \tag{1}$$

where $a_1$ and $a_2$ represent the previous and next note's anchoring strength, respectively, and $n$ represents the relative step-size from the previous pitch to the next pitch. Equation 2 is an altered form of Equation 1, specialized for generative purposes:

$$L(P) = \left(\frac{a_2}{a_1}\right)^z \left(\frac{1}{n^y}\right) + x, \tag{2}$$

where $L(p)$ represents the likelihood that a given pitch will occur next, and where the variables' values lie in these ranges: $a_{1,2}$: 15–1; $z$: 2–0; $n$: 0–12; $y$: 1–0.1; $x$: 10–100; and input tension parameter $T$ (not shown in equation): 0–1. Responding to critics (e.g., Nattiez 1997) of Lerdahl's work, and in an effort to reach our own subjectively satisfactory musical results, we decided to experiment with and manipulate some of the parameters in the formula. As shown in Equation 2, we added variables $x$, $y$, and $z$. We mapped these variables to a single input, $T$ (for tension), to these variables, controlling whether stable or unstable pitches are more likely to be generated. The larger this input parameter, the more likely it is for an unstable pitch to be played. Changing $z$ controls the influence of anchoring strength in determining the next pitch. As tension $T$ increases, $z$ decreases, reducing the likelihood that strong anchoring pitches will be generated. Similarly, $y$ affects the impact of the relative step size. As discussed earlier, theorists have shown that smaller steps between pitches increase the perception of stability. As the tension input value approaches zero, a small pitch step size becomes more likely, and therefore the output becomes more stable. Variable $x$ effectively adds noise to the equation. By raising $x$, anchoring strength and step size become relatively less significant in generating

the next note. This makes unstable pitches more likely.

We empirically derived the mapping from input tension $T$ to variables $x$, $y$, and $z$. Through trial, error, and tweaking all three parameters, we gradually found a range for each value that intuitively corresponded to the input tension values. We consider this extension of Lerdahl's formula to be a primary contribution of the present research.

## User Study

In an effort to evaluate the effectiveness of the algorithm in representing various degrees of tension in real time, we conducted a user study designed to assess the relationship between algorithmically generated tension and perceived tension. The user group included 100 volunteer students pooled from our university. We presented to each subject 100 four-second excerpts of audio. To account for the relative effects imposed by the order of the excerpts, each trial employed a randomized sequence.

To evaluate the influence of these parameters on perceived tension, we manipulated the register, density, and instrumentation of the musical excerpts generated by the algorithm. Knowing how these other features affect the perception of tension will allow us, in future revisions of the algorithm, to normalize across features. Pitch material was classified as either high- or low-register, as excerpts contained notes that are exclusively either higher or lower than C4. Note density was categorized using average IOI, as either longer or shorter than 750 milliseconds. We subcategorized instrumentation by sustain and brightness levels. Two of the instruments were sine-tone generated, one with long sustain and the other with short sustain. Three other sampled instruments offered differences in sustain and brightness, classified as either bright or dark in timbre. For all combinations of these categories we generated excerpts at five different tension levels, with level 5 representing high tension and level 1 representing low tension.

After listening to each clip, listeners indicated tension using magnitude estimation (Stevens 1975), where any number may be assigned to represent perceived tension. Magnitude estimation provided a solution to two major concerns. First, in an assignment system constrained by maximum and minimum values, the subject limits the range with the first assignment of either boundary. For instance, if the maximum permitted value was 10 and the subject indicated 10 for the previous excerpt yet found the next excerpt even more tense, they would have no additional range for expressing this relativity. In order to resolve this problem, the procedure could have first provided maximum and minimum examples of tension. This would impose designer-interpreted conditions on the subjects, however. On the other hand, magnitude estimation, and in particular "modulus-free" magnitude estimation, is used to address these issues. In order to account for earlier inconsistencies due to initial ambiguity in the perceived range and resolution, the first five values of each trial were discarded.

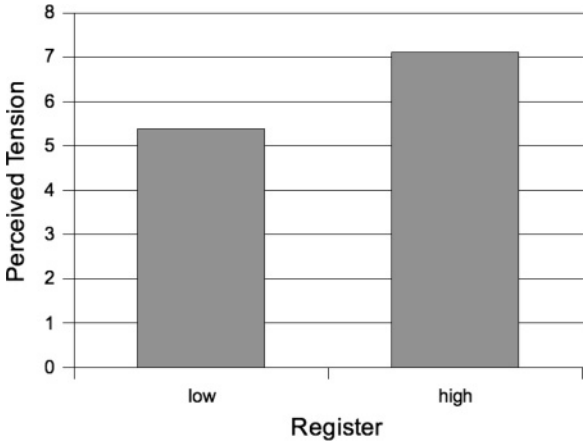
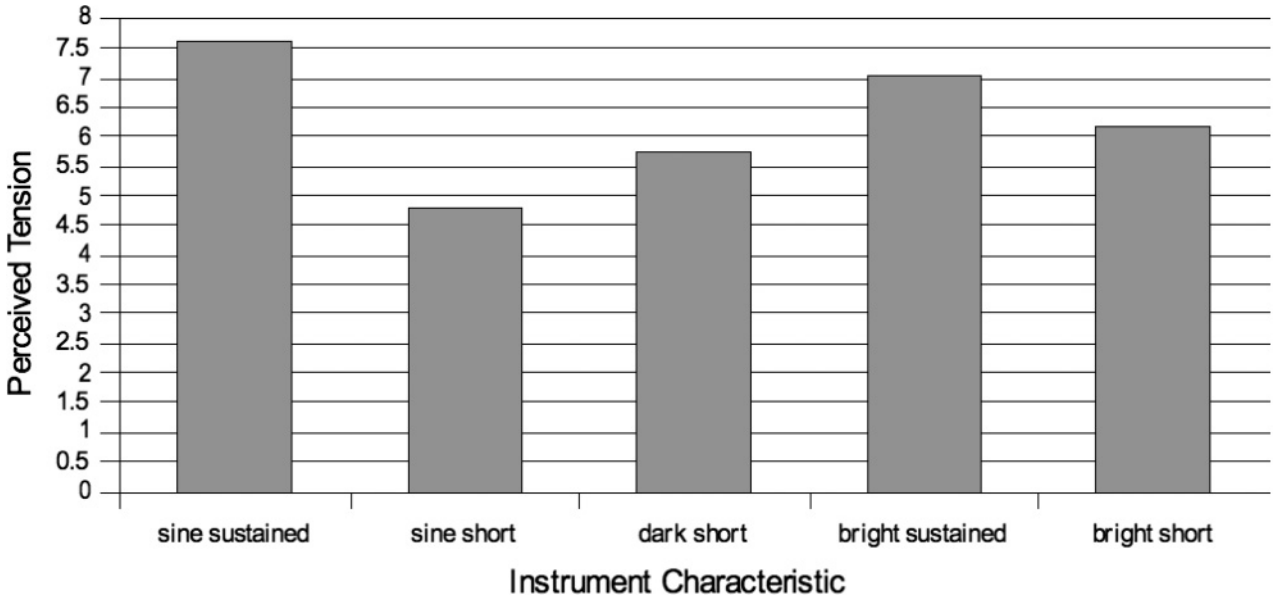



*Figure 5. Geometric mean response in perceived tension as compared to change in register.*

Working with data from magnitude estimation that has no consistency in range and boundary across subjects, we used geometric means, rather than normal arithmetic means, to represent all of the available data within an equivalent context across categories and between subjects. Although the IOI compared to perceived tension showed only slight correlation, registration and instrumentation proved significantly influential towards affecting perceived tension. Post hoc Tukey-Kramer correction ($\alpha = .05$) was used to evaluate and verify significance across all of the results. As shown in Figure 5, music

*Figure 6. Geometric mean response in perceived tension as compared to changes in instrumentation.*




generated with the same parameters but in a higher register proved, on average, 24% more tense than when compared to music in a lower register.

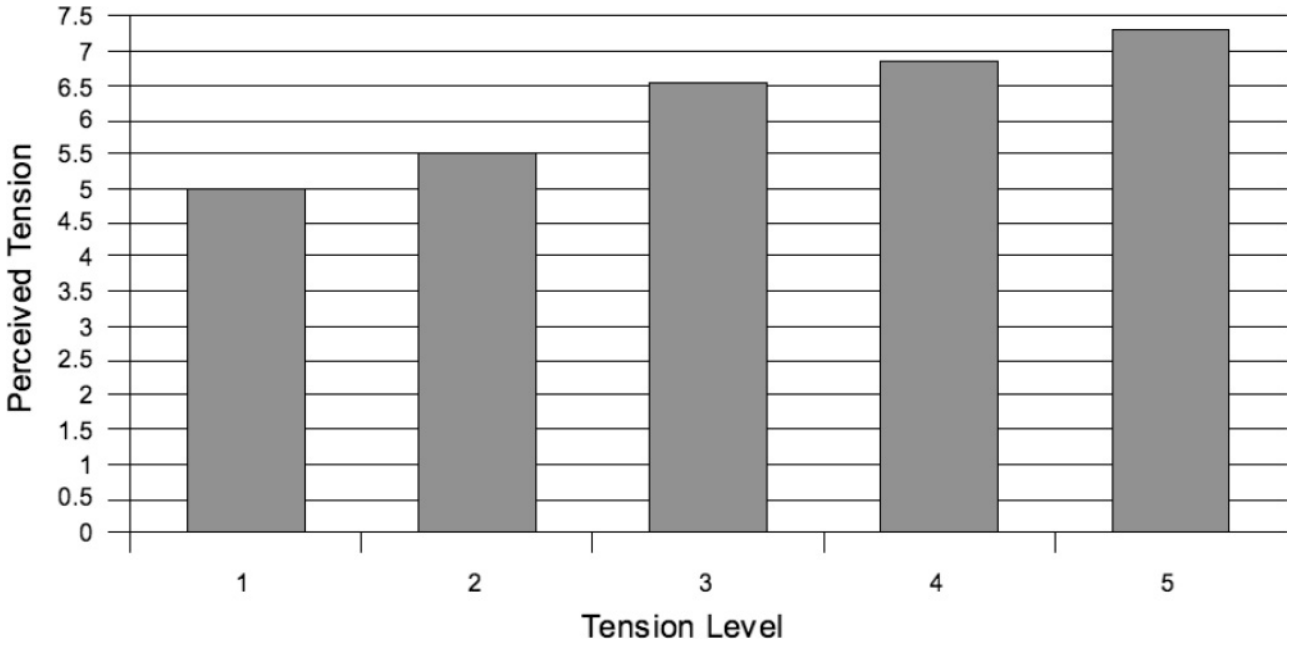
Comparing sine-tone instruments, we found, as expected, that sustaining notes are perceived as sounding more tense than shorter, resonating notes. We hypothesize that as the sustained notes overlap succeeding notes, they may cause beating, and therefore a more distinct sensory dissonance. Additionally, we found that brighter instruments, as shown in Figure 6 (right), appeared more tense than darker instruments. This finding is supported by existing research in sensory dissonance, with brighter sounds having more/stronger high-frequency harmonics beating against each other (Helmholtz 1954 [1885]; Plomp 1964; Hutchinson and Knopoff 1978; Vassilakis and Fitz 2007). In our sonification application, each fish species maps to a different musical instrument. For instance, we represent Yellow Tang with rich string sounds and the smaller Blue Chromis with a bright glockenspiel sounds. We aim to consistently model tension across instrumentation. In order to normalize across these different instrumentations, we must model the impact of each instrument on the perceived tension, as shown in Figure 6.

In evaluation of the tension control of the algorithm, we compared perceived tension to the tension input level across all manipulated conditions. Figure 7 shows the results of this analysis, with a direct, linearly proportionate correlation ($r = 0.98$) between the input tension level and subjectively perceived tension. This correlation demonstrates a 1:1 relationship between the tension control of our generative system and the perceived tension. It also supports the melodic tension percepts laid out by Lerdahl (2001), and the effectiveness of our modifications of Lerdahl's formulas.

## Future Work

Although the current model successfully addressed our intended goals, this work only lays a foundation for future work. We want to extend the concept of musical roles—varying degrees of leading and supportive roles—to our generative system. Finally, we want to adapt the algorithm to compensate for relative changes in tension based on information gathered from our study.

*Figure 7. Geometric mean response in perceived tension, as compared to change in the input tension parameter.*




Through combinatorial processing of control parameters, we also hope to further explore the full range of the system's possible generative outputs. From this study we will define distinct characteristics of the music output that result from certain input parameters. We can classify these characteristics as certain musical roles. For instance, parameters limiting movement only to leaps between chord tones would most likely yield a supportive role, whereas increasing the likelihood of stepwise movement and non-harmonic tones may result in a more melodic and prominent lead role. Extending this to sonification, we may orchestrate the musical output, as salient moving objects (brightly colored fish) will be assigned melodic lead roles, and less prominent objects (less noticeable fish) are assigned background roles of harmonic support.

In our user study we found a positive correlation between register and perceived tension. We also found that the choice of sounds (what we have called instrumentation) affected perceived tension. Specifically, brightness of timbre correlated with perceived tension, as did duration. Based on these data, we can adjust our current model to compensate for variations in instrumentation and register. This will provide a controlled method for manipulating musical tension across varying features.

Examples of study stimuli and aquarium sonification videos can be found online at gtcmt.coa.gatech.edu/tension_examples.

## References

Agam, Gady. 2006. "Introduction to Programming with OpenCV." Available online at www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro. Accessed 23 October 2011.

Belzer, J., A. Holzman, and A. Kent. 1981. "The Computer and Composition." *Encyclopedia of Computer Science and Technology*, vol. 11. New York: Facts on File, pp. 80–81.

Cope, D. 1991. "Computer Simulations of Musical Style." In *Conference on Computers in Music Research*, pp. 15–17.

Desain, P., and H. Honing. 2002. "Rhythmic Stability as Explanation of Category Size." Paper presented at the International Conference on Music Perception and Cognition, 17–21 July, University of New South Wales, Sydney.

Farbood, M. 2006. "A Quantitative, Parametric Model of Musical Tension." PhD Dissertation, Media Lab, Massachusetts Institute of Technology. Available online at web.media.mit.edu/~mary. Accessed 9 June 2009.
Helmholtz, H. 1954. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, trans. A. J. Ellis. 2nd ed. New York: Dover.
Hutchinson, W., and L. Knopoff. 1978. "The Acoustic Component of Western Consonance." *Interface* 7(1): 1–29.
Justus, T., and J. Bharucha. 2001. "Modularity in Musical Processing: The Automaticity of Harmonic Priming." *Journal of Experimental Psychology: Human Perception and Performance*. 27(4):1000–1011.
Lerdahl, F. 2001. *Tonal Pitch Space*. New York: Oxford University Press.
Lerdahl, F., and C. Krumhansl. 2007. "Modeling Tonal Tension." *Music Perception* 24(4): 329–366.
Margulis, E. 2005. "A Model of Melodic Expectation." *Music Perception* 22(4):663–714.
Narmour, E. 1992. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. Chicago: University of Chicago Press.
Nattiez, Jean-Jacques. 1997. "What is the Pertinence of the Lerdahl-Jackendoff Theory?" In I. Deliège and J. A. Sloboda, eds. *Perception and Cognition of Music*. Hove, UK: Psychology Press, pp. 383–388.
Pachet, F. 2002. "Playing with Virtual Musicians: The Continuator in Practice." *IEEE Multimedia* 9(3):77–82.
Plomp, R. 1964."The Ear as a Frequency Analyzer." *Journal of the Acoustical Society of Americal* 36(9):1628–1636.
Steinbeis, N., S. Koelsch, and J. Sloboda. 2006. "The Role of Harmonic Expectancy Violations in Musical Emotions: Evidence from Subjective, Physiological, and Neural Responses." *Journal of Cognitive Neuroscience* 18(8):1380–1393.
Stevens, S. S. 1975. *Psychophysics: Introduction to its Perceptual, Neural, and Social Prospects*. New York: Wiley.
Thom, B. 2001. "BoB: An Improvisational Music Companion." PhD Dissertation, School of Computer Science, Carnegie-Mellon University.
Vassilakis, P., and K. Fitz. 2007. "SRA: A Web-based Research Tool for Spectral and Roughness Analysis of Sound Signals." Available online at www.acousticslab.org/roughness. Accessed 7 February 2012.