

## Kutztown University Research Commons at Kutztown University

---

Computer Science and Information Technology  
Faculty

Computer Science and Information Technology  
Department

---

4-2-2016

# Timbral Data Sonification from Parallel Attribute Graphs

Dale E. Parson

*Kutztown University*, [parson@kutztown.edu](mailto:parson@kutztown.edu)

Danielle Emily Hoch

*Kutztown University of Pennsylvania*, [dhoch984@live.kutztown.edu](mailto:dhoch984@live.kutztown.edu)

Hallie Langley

*Kutztown University of Pennsylvania*, [hlang151@live.kutztown.edu](mailto:hlang151@live.kutztown.edu)

Follow this and additional works at: <https://research.library.kutztown.edu/cisfaculty>

 Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Proceedings of the 31st Annual Spring Conference of the Pennsylvania Computer and Information Science Educators (PACISE)  
Kutztown University of PA, Kutztown, PA, April 1-2, 2016. This won the Best Faculty Paper award.

This Conference Proceeding is brought to you for free and open access by the Computer Science and Information Technology Department at Research Commons at Kutztown University. It has been accepted for inclusion in Computer Science and Information Technology Faculty by an authorized administrator of Research Commons at Kutztown University. For more information, please contact [czerny@kutztown.edu](mailto:czerny@kutztown.edu).

Proceedings of the  
31<sup>st</sup> Annual Conference of  
The Pennsylvania Association of Computer and  
Information Science Educators  
April 01 - 02, 2016  
Hosted by Kutztown University of PA

**TIMBRAL DATA SONIFICATION FROM PARALLEL ATTRIBUTE GRAPHS**

Dale E. Parson, Danielle Emily Hoch, Hallie Langley  
Department of Computer Science and Information Technology, Kutztown University of PA  
parson@kutztown.edu

**Best paper, faculty:**

Dale E. Parson, Danielle Emily Hoch, and Hallie Langley  
Department of Computer Science and Information Technology, Kutztown University of Pennsylvania  
Timbral Data Sonification From Parallel Attribute Graphs

# TIMBRAL DATA SONIFICATION FROM PARALLEL ATTRIBUTE GRAPHS

Dale E. Parson, Danielle Emily Hoch, Hallie Langley  
Department of Computer Science and Information Technology, Kutztown University of PA  
parson@kutztown.edu

## ABSTRACT

Parallel coordinate plotting is an established data visualization technique that provides means for graphing and exploring multidimensional relational datasets on a two-dimensional display. Each vertical axis represents the range of values for one attribute, and each data tuple appears as a connected path traveling left-to-right across the plot, connecting attribute values for that tuple on the vertical axes. Parallel coordinate plots look like time-domain audio signal waveforms, and they can be translated into audio signals through straightforward mapping algorithms. This study looks at three data sonification algorithms, sonification being the mapping of data into sounds for perceptual exploration, similar to uses of data visualization. Sound-response survey results and subsequent analyses reveal that the most direct method for mapping parallel coordinates of data tuples to audio waveforms is the most accurate for generating sounds that listeners can use to classify data. Future work has begun on improving the accuracy of this audio waveform-based, timbral approach to classifying data.

## KEY WORDS

data analytics, parallel coordinates, sonification, timbre, visualization, waveform

## 1. Introduction

This report presents an investigation into the relative effectiveness of three alternative techniques for the perception-based classification of individual data records (a.k.a. instances or tuples) in a relational dataset by using sonification of attribute values for each attribute in a record. **Sonification** is the process of mapping data attribute values to properties of sound [1]. Sonification is the aural counterpart to **visualization**, which is the process of mapping data attribute values to visual structures, for example in computer graphical displays. Sonification and visualization play roles in at least two stages of data analysis [2]. They serve the mechanisms of perceptual pattern recognition, helping the respective auditory and visual cognitive systems of an analyst to detect patterns in data as a guide to subsequent formal analysis. They can also serve to illustrate relationships found through formal analysis, coming after formal

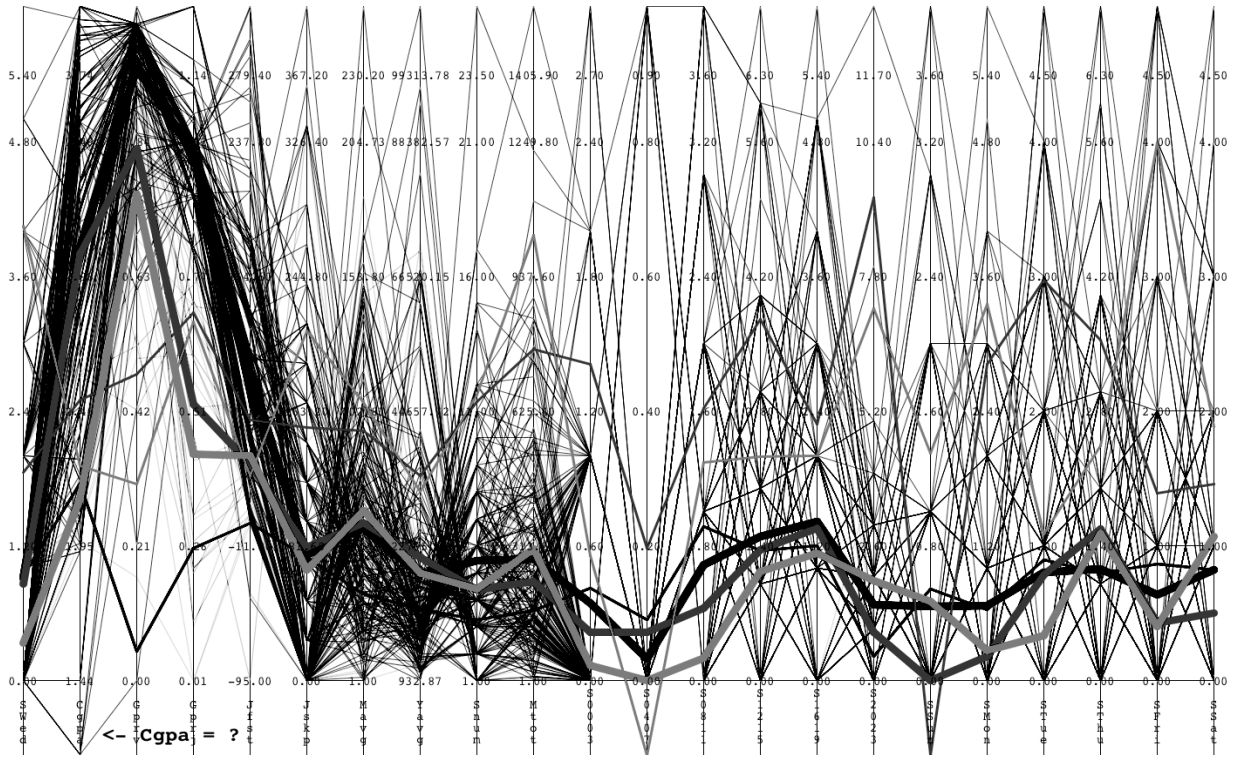
analysis. The current investigation relates to the former role, the use of perceptual pattern matching in exploring and classifying data instances. This paper reports the first stage of research into using synchronized sonification and visualization for exploring datasets.

Because of familiarity, the authors sonified attributes of a dataset that is part of an ongoing study into the correlation of temporal work habits of computer programming students to their success in projects as measured by project grades [3,4]. The authors anticipated the fact that familiarity with the dataset would make it easier to detect sonification opportunities and mistakes. However, the approaches explained in this report are domain neutral. They can apply to any relational dataset in which some attributes co-vary according to instance membership in disjoint sets of instances to be classified.

## 2. Related work

The primary influence on the attribute sonification techniques of this report is the visualization technique of using **parallel coordinates** [5]. Figure 1 is a typically dense parallel coordinates plot of 22 of the 106 attributes found in the student work habit dataset of 282 student-project records [3,4]. Each vertical line represents the overall range of one attribute, with the minimum value at the numeric label near the bottom, and the maximum value at the top; an unknown value is at the very bottom. Figure 1's second vertical axis from the left, for example, bears the label "Cgpa" for "computer science grade point average", ranging from 1.44 near the bottom to 4.0 at the top. Going left to right, each thin multi-segment **path** represents one record in the dataset, intersecting a vertical axis at the point of the value for that record's attribute. The reduction of 106 attributes to the 22 in Figure 1 was part of a reduction of the scope of exploration of this dataset determined by the previous studies [3,4].

The homegrown software tool used to create Figure 1 uses partial transparency (a.k.a. low alpha) to plot the 282 instances in this dataset so that their paths do not obscure each other. Overlapping path segments increase opacity, which appears as brightness on a color computer screen. Figure 1 also shows three thick paths and three mid-thickness paths for the mean and population standard deviation, respectively, of three sets of instances. The

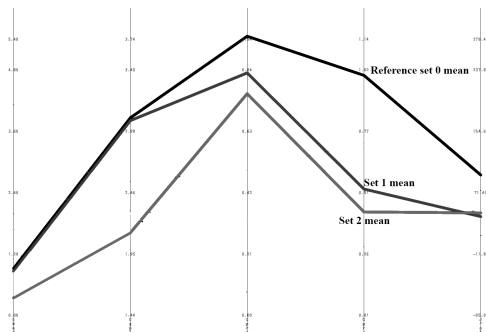


**Figure 1: Parallel Coordinates Plot of 22 of the 106 attributes in the Student Work Pattern -> Grade Dataset**  
**The thick paths show mean values for three distinct sets of records. 282 individual records appear as thin lines.**

thick black path shows the mean of all attributes for records with a mean project grade (Gprj, the fourth attribute from the left) that is  $\geq 80\%$ ; we refer to this set of records as **Reference Set 0**. The thick medium-gray path shows the mean of all attributes for records (**Set 1**) with a mean project grade  $< 80\%$  and a computer science grade point average (Cgpa)  $\geq 2.5$ . The thick light-gray path shows the mean of all attributes for records (**Set 2**) with a mean project grade  $< 80\%$  and a computer science grade point average  $< 2.5$ . Projects in this course serve more as learning exercises than as tests, so a grade that is  $< 80\%$  is a poor grade. The mean project grade for all records is 92.3% with a standard deviation of 21%. The maximum project grade for one project per semester is

125% because of bonus points, giving the top of the Gprj range in Figure 1.

A less cluttered parallel coordinates plot containing the mean values for the leftmost 5 attributes of Figure 1 appears in Figure 2. Informally, Reference Set 0 is the set of all student-projects with high grades, Set 1 is the set with low grades and high Cgpa values, and Set 2 is the set with low grades and low Cgpa values. Sets 1 and 2 represent at-risk students who are at risk for potentially different reasons. Figure 2 includes set identifier tags. Table 1 gives the mean and standard deviation values for each set and each attribute of Figure 2.



**Figure 2: Parallel Coordinates of Mean Values for SWed, Cgpa, Gprv, Gprj & Jfst Attributes by Set**

An early observation by the author who works with sound and audio signal processing in other contexts (Parson) is that the multi-segment paths of Figures 1 and 2 look a lot like audio time-domain waveform plots. Figure 3 shows plots for the basic triangle and sawtooth audio waveforms, where the X axis is Time and the Y axis is the Voltage of an electrical audio signal in a normalized range  $[-1.0, 1.0]$ , corresponding to sound pressure level (SPL) in the air. A *triangle wave* can be modeled as the sum of a fundamental sinusoidal waveform and a series of its positively weighted odd harmonics [6], where a harmonic is a frequency multiple of the fundamental, and frequency is  $1.0 / \text{Time}$ . A *sawtooth wave* can be modeled as the sum of a fundamental sinusoidal waveform, its positively weighted odd harmonics, and its negatively

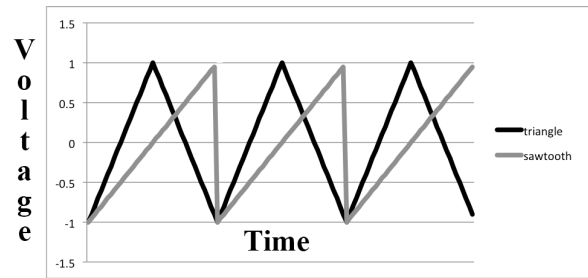
Attribute	Class		
	set2 (0.07)	set1 (0.06)	set0 (0.86)
=====			
<b>SWed (Wednesday work sessions)</b>			
mean	0.3333	0.7692	0.9158
std. dev.	0.4714	1.1867	1.224
<b>Cgpa (computer science GPA)</b>			
mean	2.1298	3.081	3.1254
std. dev.	0.1284	0.4257	0.6478
<b>Gprv (previous project grade)</b>			
mean	0.7558	0.809	0.9519
std. dev.	0.215	0.2081	0.1327
<b>Gprj (current project grade)</b>			
mean	0.4326	0.5064	0.9889
std. dev.	0.2508	0.2339	0.0761
<b>Jfst (start hours before deadline)</b>			
mean	43.7237	33.4414	94.6208
std. dev.	31.2262	62.7025	70.1115

**Table 1: Mean & standard deviation values for Figure 2**

weighted even harmonics. Triangle waves sound consonant (“sweet”), and sawtooth waves sound dissonant (“raspy”). These particular sounds become significant later in this discussion. The main point for now is the similarity in the shapes of multi-segment record paths in the parallel coordinates plots of Figure 1 and 2 on the one hand, and the time-oriented sound waveform plots of Figure 3 on the other. That similarity, and the potential isomorphism between parallel coordinate plots and audio waveforms that yield distinct timbres (classes of sounds, e.g. sweet, raspy, tinny, etc.), provide the inspiration and basis for this research.

Neuhoff presents a taxonomy of applying *pitch* (a perceptual function of frequency, with frequency = 1.0 / Time, appearing on the X axis in Figure 3), *loudness* (a function of amplitude, the Y axis of Figure 3, and to a lesser degree the frequency), and *timbre* (a function of the sum of a sinusoidal fundamental frequency and its weighted harmonics, the waveform shape of Figure 3) as the primary approaches with which to sonify data [7]. Duration of sound, spatial location, and sequences of distinct sounds are additional approaches.

Recent work that has inspired the present study involves the sonification of material x-ray scattering data by mapping two-dimensional arrays of x-ray intensity values directly to two-dimensional arrays of sound frequency components that define an audio waveform (timbre, or informally, “instrument sound” or “voice sound”) [8]. That approach is similar to the approach of the present study in mapping domain data directly to waveforms (timbre) while avoiding any kind of musical or other aesthetic interpretation of the data that might introduce arbitrary sonic artifacts. In contrast to that work, the hallmark of the present study is the use of parallel



**Figure 3: Triangle and Sawtooth Audio Amplitude Time-Domain Waveforms**

coordinates plots of domain data as the source of mappings to sound.

### 3. Classification through Sonification

The present study uses sound for instance classification. The modus operandi of our study is to investigate competing approaches for sonifying the dataset summarized in Figures 1 and 2. Our approach generates a reference sound for each of the three sets of means of Figure 2, and it generates a sound for each data record. A listener classifies a data record’s sound as being closest to Set 0’s reference sound, or Set 1’s or Set 2’s sound, thereby classifying the record as belonging to Set 0 or 1 or 2. The experiments reported include three competing sonification methods for turning the mean reference records and the individual data records into sounds.

#### 3.1 Reducing the number of attributes to sonify

There are a total of 282 student project records in the dataset, where each record shows the work patterns (primarily temporal patterns) and performance of one student completing one programming project. With 106 attributes per record, there is a total of 282 x 106 = 29,892 data points. Some attributes are partially redundant, for example project grade in letter and numeric form. We eliminated redundant attributes, keeping the most precise form (e.g., numeric project grade). The year-of-study attribute of a student as a sophomore, junior, or senior turned out to be significant, with sophomores performing better on average than juniors or seniors. High motivation among early takers of this major elective course is the most likely cause. However, we eliminated such discrete attributes from the present sonification study because creation of sounds that differ across a few discrete steps can be misleading. There are high-performing juniors and seniors, and with only 3 values in the range for the year-of-study attribute, we would have to include too many additional attributes to overcome the dominating effect of this one. Inclusion of a large number of attributes becomes noisy and aurally confusing. In addition, we wanted to stay with numeric attributes in the first round of this sonification study in order to simplify mapping, so we discarded the discrete year-of-study attribute.

After eliminating redundant attributes, non-continuous attributes, and non-useful attributes as determined by the previous studies [3,4], the 22 attributes of Figure 1 remain. Semi-automated classification tools within the Weka data analysis toolset [9,10] provide one means for reducing the 22 attributes of Figure 1 down to the 5 attributes for sonification of Figure 2. The CfsSubsetEval attribute evaluator of Weka evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them in predicting the value of a target attribute such as project grade. This attribute evaluator indicates that three of the first five attributes appearing in Figure 1, namely computer science grade point average coming into the course (Cgpa), the grade on the previous project (Gprv), and the number of hours a student started a project before its deadline MINUS 24 hours for each day the student did not work on the project (Jfst) are the three best indicators for a record's project grade (Gprj).

The ordering of attributes in Figure 2 is SWed, Cgpa, Gprv, Gprj, and Jfst. The number of work sessions on a Wednesday (SWed, where a session is one or more continuous work steps separated by fewer than 60 minutes), comes out of Weka extraction of a linear regression formula for estimating project grade as a function of the remaining 4 attributes:

$$\text{Gprj} = 0.0006 * \text{Jfst} + 0.0729 * \text{Cgpa} + 0.0228 * \text{SWed} + 0.4049 * \text{Gprv} + 0.2512$$

Correlation coefficient= 0.5068

Mean absolute error = 0.1209

Our conjecture is that SWed is significant because it is proof that the student is not procrastinating to the limit in starting or completing work. Projects in this course are typically due on Friday or Saturday evening, so working on a Wednesday is proof of working at least 2 days in advance of a deadline.

More complex Weka classifiers come up with the same set of attributes.

The other, interactive, visual means for reducing the 22 attributes of Figure 1 down to the 5 attributes of Figure 2 comes from using our homegrown software tool for interacting with parallel coordinates data displays to find attributes with diverging means. Recall from the previous section that Reference Set 0 of Figures 1 and 2 consists of student-project records with a project grade (Gprj) that is  $\geq 80\%$ . Set 1 consists of student-project records with a project grade (Gprj) that is  $< 80\%$  and a computer science grade point average (Cgpa) that is  $\geq 2.5$ . Set 2 consists of student-project records with a project grade (Gprj) that is  $< 80\%$  and a computer science grade point average (Cgpa) that is  $< 2.5$ . Diverging mean values for multiple attributes in Figures 1 and 2 provide a basis for attribute-distance-based sonification. All three of our experimental

approaches sonify the distance between a given record's attribute value and the Set\_0\_mean for that attribute, as a function of the Set\_0\_standard deviation for that attribute. An attribute value within 1 Set 0 standard deviation of the Set 0 mean generates a sound property that is relatively sweet; an attribute value within 2 standard deviations generates a sound that is a mix of sweet and sour; and an attribute value greater than 2 standard deviations generates a sound that is all sour. Upcoming discussion quantifies "sweet" and "sour", and explains their application in generating sounds.

A final concern is distinguishing Set 1 instances from Set 2 instances. All attributes in Figure 2 except Jfst (lead time MINUS 24 hours for unworked days) have distinct mean values for Sets 1 and 2. The Cgpa attribute is the defining difference between these two sets. Therefore, a good sonification algorithm should have sufficient data for generating sounds that distinguish Set 1 from Set 2 instances. We limited the number of attributes to 5, based on our Weka analysis and inspection of the parallel coordinates, to limit the complexity of the sounds. Too few attributes do not distinguish set membership of individual data records adequately, while too many generate complicated and confusing sounds.

It is noteworthy that Weka's J48 decision tree classifier [10] is extremely accurate at classifying the data records summarized in Figure 2 into sets when the Set number is included as a sixth, discrete attribute. The following decision tree classifies student records from the dataset with 99% correct classifications. This classification parallels our sonic classification of a record's sonified attributes to one of three reference set sounds.

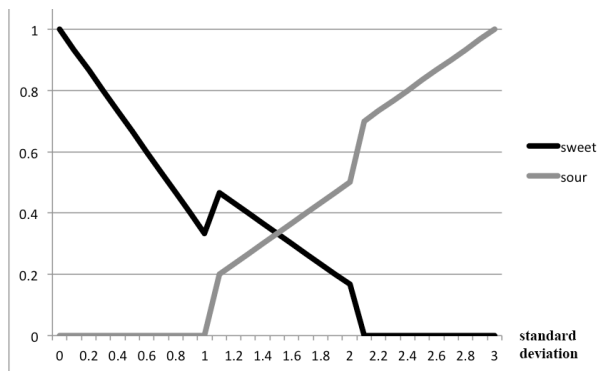
```
Gprj <= 0.79
| Cgpa <= 2.48: set2
| Cgpa > 2.48: set1
Gprj > 0.79: set0
```

Several of Weka's other classification and clustering algorithms do almost as well. We did not expect to hit anything near 99% accuracy in this first stage of sonification. Our goal is to find out which of three competing sonification techniques works the best, and then enhance it in a second round of upcoming research.

### 3.2 Sweet and Sour Sonification

All of the sonification algorithms in this study use a helper algorithm that we call **sweetAndSour** to extract two numeric values for each attribute in either the Set of mean values or in an individual data record. This algorithm first computes the difference between an attribute being sonified and that attribute value for Reference Set 0, which is the mean of the reference set of records as defined above. If the attribute being sonified lies within 1 population standard deviation of Reference Set 0 for that attribute, it receives a **sweetWeight** in the

range [0.33, 1.0] – the 1.0 end of that continuum is for a value that equals the Reference Set 0 value, and the 0.33 is for one standard deviation away – and it receives a **sourWeight** of 0.0. If the attribute being sonified lies within the range (1, 2] standard deviations of Reference Set 0 for that attribute, it receives a **sweetWeight** in the range [.167, .5) and a **sourWeight** in the range (.167, .5], with the maximum standard deviation of 2.0 giving a **sweetWeight** of .167 and a **sourWeight** of .5. The further the distance from the Reference attribute mean, the less sweet and more sour the **sweetAndSour** numbers. Finally, the algorithm clamps the standard deviation at 3.0. If the attribute being sonified lies within the clamped range (2, 3] standard deviations of Reference Set 0 for that attribute, it receives a **sweetWeight** of 0.0 and a **sourWeight** in the range (.667, 1.0]. Figure 4 shows the curves for these functions. The temporary change in direction of the sweet curve at the standard deviation of 1.0 was an unintentional bug, but by the time it was discovered, experimental classification response collection had already begun. The mix of sweet and sour differs to the immediate left versus right of the  $\text{stddev} = 1.0$  point, and furthermore, the overall non-linear relationship of the curves was intentional. Each attribute contributes some amount of sweet versus sour, each in the range [0.0, 1.0], and within a standard deviation band, these parameters vary linearly. The basis for this approach is to cause abrupt changes in sound when crossing a discrete boundary. That basis is experimental.



**Figure 4: Sweet and Sour values as a function of attribute standard deviation**

### 3.3 Three Competing Sonification Algorithms

The three competing sonification algorithms are named **harmonic**, **melodic**, and **waveform**. This section discusses them in the order of their creation. All three generate sounds in some relation to a baseline frequency, for which we selected 220 Hz (Hertz, a.k.a. cycles per second). We picked this baseline frequency because it is one octave below the reference frequency for the note “A” in Western music of 440 Hz, and because it is low enough in the hearing range to allow generation of higher audible frequencies in a data-dependent way.

The **harmonic** sonification algorithm generates simultaneous notes in a chord (technically, a set of harmonic intervals) for the 5 attributes of Figure 2. All of the sonification algorithms are open-ended with respect to number of attributes, but for the current discussion there are always the 5 attributes of Figure 2. Harmonic sonification maintains two tables of frequencies, **sweetNotes** and **sourNotes**, that are in-scale and out-of-scale respectively for the baseline “A” note of 220 Hz. The non-musician reader can think of the **sweetNotes** as the white keys and the **sourNotes** as the black keys on a piano, although the actual white keys on a piano are rooted in a “C” note rather than a 220 Hz “A” note when playing a major scale. The point is that the **sweet** parameter of the previous section plays its positional **sweetNote** in the “A” major scale with the intensity of the **sweet** parameter, and the **sour** parameter plays its positional **sourNote** outside of the “A” scale with the intensity of the **sour** parameter. An attribute value within 1 standard deviation of the mean Reference Set 0 value for that attribute is all sweet, with its amplitude growing linearly with its proximity to the Reference Set attribute value. Figure 4 illustrates the mix of in-scale (**sweet**) and out-of-scale (**sour**) notes played simultaneously for a given attribute. All attributes play their sequential regions of the scale simultaneously, so there can be up to 10 simultaneous, distinct notes playing when all 5 attributes are a mix of non-zero **sweet** and non-zero **sour** parameter values. Data records in Sets 1 and 2 tend to be discordant for one or more (attribute : note) positions.

In addition to using frequency perceived as pitch, the harmonic algorithm takes a very brute-force waveform (timbral) approach. It uses sine wave generators for the **sweet** notes and sawtooth wave generators for the **sour** notes. Sine waves are even sweeter and less attention grabbing than the relatively sweet triangle waves of Figure 3, while the sawtooth waves are very raspy. The intent is to reinforce the consonant-versus-dissonant properties of frequency sonification of the last paragraph with consonant-versus-dissonant properties of timbre sonification. A harmonic chord lasts 2 seconds.

Sound generation takes the form of generating one short **ChucK** program for each Set record of mean attribute values and one **ChucK** program for each data record. **ChucK** is an audio programming language that provides waveform generators including sine, triangle, and sawtooth oscillators with programmable frequency and amplitude [11]. Harmonic sonification generates a parallel set of sine and sawtooth **ChucK** generators with amplitude (a.k.a. gain) given by their respective **sweet** and **sour** parameters, and with frequency given by the position of each attribute in its **sweetNotes** and **sourNotes** scales. The non-musician reader should envision banging out a chord of simultaneous notes on a piano for a data record, in which **sweet** notes are consonant to the listener, **sour** notes are dissonant, the intensity of a note is proportional to its

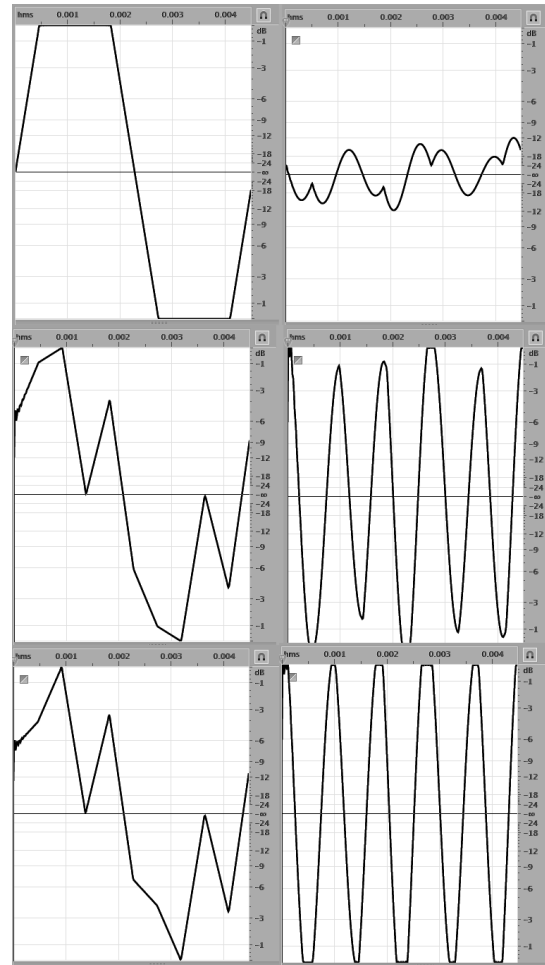
sweet or sour value, and the timbre of the sour notes is raspy.

The **melodic** sonification algorithm (technically, a set of melodic intervals) generates note frequencies, amplitudes, and waveforms that are identical to the harmonic approach, but it generates them to play only one attribute’s sweet and sour notes at a time, sequencing them across a 2 second duration. The only other difference is that the harmonic method scales down the amplitude of the mix in order to avoid excessively loud sounds, since it is playing many notes together. The melodic approach plays each note at higher amplitude because it plays at most one sweet and one sour note at a time, in a temporal sequence. The idea behind generating harmonic and melodic sonifications with identical notes is to determine whether sequencing the notes over time assists or detracts from the abilities of listeners to classify otherwise identically sonified data.

The **waveform** sonification algorithm is substantially different from the other two. Its intent goes back to the observation that the parallel coordinate plots of Figures 1 and 2 look like waveforms. In fact, the first 5 attributes of Figure 1, which are the attributes of Figure 2, are sorted to approximate a triangular waveform for the Reference Set 0 mean values. The original idea was, to the degree that Set 1 and 2 waveforms deviate from the Set 0 waveform, a listener would distinguish timbral differences and classify into the three sets on the basis of those.

Mapping the parallel coordinates plots directly to audio waveforms created sounds that were hard to distinguish, so we went back to using sweet and sour parameters to introduce discontinuities. For each attribute of a record, after determining the sweet and sour parameters, the waveform generator tests which is greater in magnitude, sweet or sour. For attributes where sweet dominates, it saves the attribute value in its position. For attributes where sour dominates, it saves the additive inverse (the “negative”) of the attribute value in its position. The intent is to create more “kinks” in a sour attribute’s position in a waveform, increasing possibly dissonant overtone frequencies (a.k.a. partials). After traversing all attributes of a record, the waveform generator normalizes the range of values to the range [0.0, 1.0] by scaling. It then generates a ChuckK program that plays this waveform for 2 seconds. The sounded waveform is actually the original waveform in the [0.0, 1.0] range, and then its mirror image in the [-1.0, 0.0] range, in order to preserve symmetry and avoid introducing additional overtones.

The left side of Figure 5 shows the original waveforms for Reference Set 0, Set 1, and Set 2, starting at the top. Reference Set 0 is somewhat problematic because all of its attributes are 100% sweet and 0% sour, giving a flat line. The waveform sonification algorithm treats the lowest attribute value as a minimum and scales the others



**Figure 5: Unfiltered (left) and filtered waveforms for Reference Set 0, Set 1, and Set 2 mean values**

according to their range, winding up with a near square wave for Reference Set 0 at the top left of Figure 5.

Set 1 and 2 original waveforms appear below Reference Set 0. Careful inspection shows 5 vertices in the positive, initial side of the Set 1 waveform, corresponding to the five attributes SWed, Cgpa, Gprv, Gprj and Jfst. There is the initial point (SWed), a slight bend to a lesser slope near .45 ms. (milliseconds) (Cgpa), a peak (Gprv), a trough at the 0 center line (Gprj), and a final peak (Jfst) before going to the negative mirror half of the waveform. The overall waveform occupies 4.5 ms., which is 1.0 / 200 Hz baseline frequency. The trough for the second-last attribute (Gprj) in the Set 1 and 2 waveforms corresponds to the distances between their means and the Reference Set 0 mean in Figure 2. Gprj is the only parameter for which the deviation of Sets 1 and 2 are so great that they generate a negative-going trough, which sonically acts as an overtone in the timbre. Cgpa, the attribute for which Sets 1 and 2 differ from each other most significantly, gives a reduction in slope for Set 1, and an increase in slope for Set 2, in going from Cgpa to Gprv.



The original sounds from the left side of Figure 5 are so dominated by 200 Hz and other low-frequency harmonics that it was hard to distinguish them when listening. Since the intent was to generate high-frequency artifacts in the non-reference Sets 1 and 2, we added a high-pass filter operating at 4.5 X the baseline frequency of 220 Hz, with a filter Factor that makes it moderately selective (filter “Q” = 10). It passes frequencies above 990 Hz with relatively little attenuation, and it allows some frequencies below that threshold to pass with gentle but increasing attenuation as frequency goes down. We picked these values for the filter through listening. The waveforms on the right side of Figure 5 are the results of high-pass filtering. Reference Set 0 at the top has its amplitude diminished considerably because it consists mostly of low-frequency components that manage to make it past the filter. Sets 1 and 2 show more remaining amplitude because of their sour-parameter-generated overtones, with Set 2 saturating at the -1.0 and 1.0 limits in more places than the Set 1 waveform. The waveforms on the right side of Figure 5 are the ones actually used in the surveys of the next section.

### 3.4 Conducting and Analyzing the Sonic Surveys

In addition to generating sounds, running the Chuck programs generates uncompressed WAV (Waveform Audio File Format) audio files that store those sounds. For our sonic surveys we created a Java survey application that loads the mean reference set and individual data record WAV files and presents them to listeners via a GUI and desk monitors (speakers), adjusted by one of the authors to safe levels. The sonic survey allows the three Set reference tones for harmonic sonification to be sounded any time while manually sequencing through 39 pseudo-randomly selected data record sounds, 13 belonging to each of Set 0, 1, and 2. The listener selects the Set 0, 1 or 2 that they feel is closest in sound to current data record sound, and then goes on to the next record. The numbers of sounds are a function of the numbers of records in the least-populated set of data. After making responses to each of the 39 harmonic sounds, a listener listens to three melodic reference set sounds and then responds to 39 instances, and then listens to three waveform reference set sounds and then responds to 39 instances. There were 29 volunteer participants, mostly computer science students at Kutztown University. There was no data collection about student familiarity with music or computer audio, and no discussion or revelation about the data or sonification techniques used in the sonic survey, beyond the names “harmonic”, “melodic”, and “waveform” for the sound sets triggered by interaction with the GUI. There were 117 selection mouse clicks (39 sounds for each of 3 sonification approaches) X 29 listeners = 3393 data points, 1131 per sonification technique.

The survey results appearing in Table 2 were a surprise to the project leader (Parson), to say the least. In comparing

overall classification accuracy without looking at individual sets, waveform at 61.4% is clearly superior to harmonic at 55.8% or melodic at 55.4%. The project leader had assumed we would discard waveform after the first semester of research because of its anticipated lack of accuracy for classification. Instead, it has the greatest overall accuracy. Furthermore, the project leader’s survey responses accord with those of Table 2. Apparently, conscious perception of sound can diverge from unconscious perception that drives reaction in the survey, at least for the project leader. After taking the survey, some students reported clearly perceived distinctions in the waveform approach.

Sonification	Category	Correct responses
Harmonic	All 3 sets	55.8%
Harmonic	Reference Set 0	65.5%
Harmonic	Set 1	41.4%
Harmonic	Set 2	60.5%
Melodic	All 3 sets	55.4%
Melodic	Reference Set 0	47.5%
Melodic	Set 1	50.9%
Melodic	Set 2	67.9%
Waveform	All 3 sets	61.4%
Waveform	Reference Set 0	74.8%
Waveform	Set 1	57.8%
Waveform	Set 2	51.5%

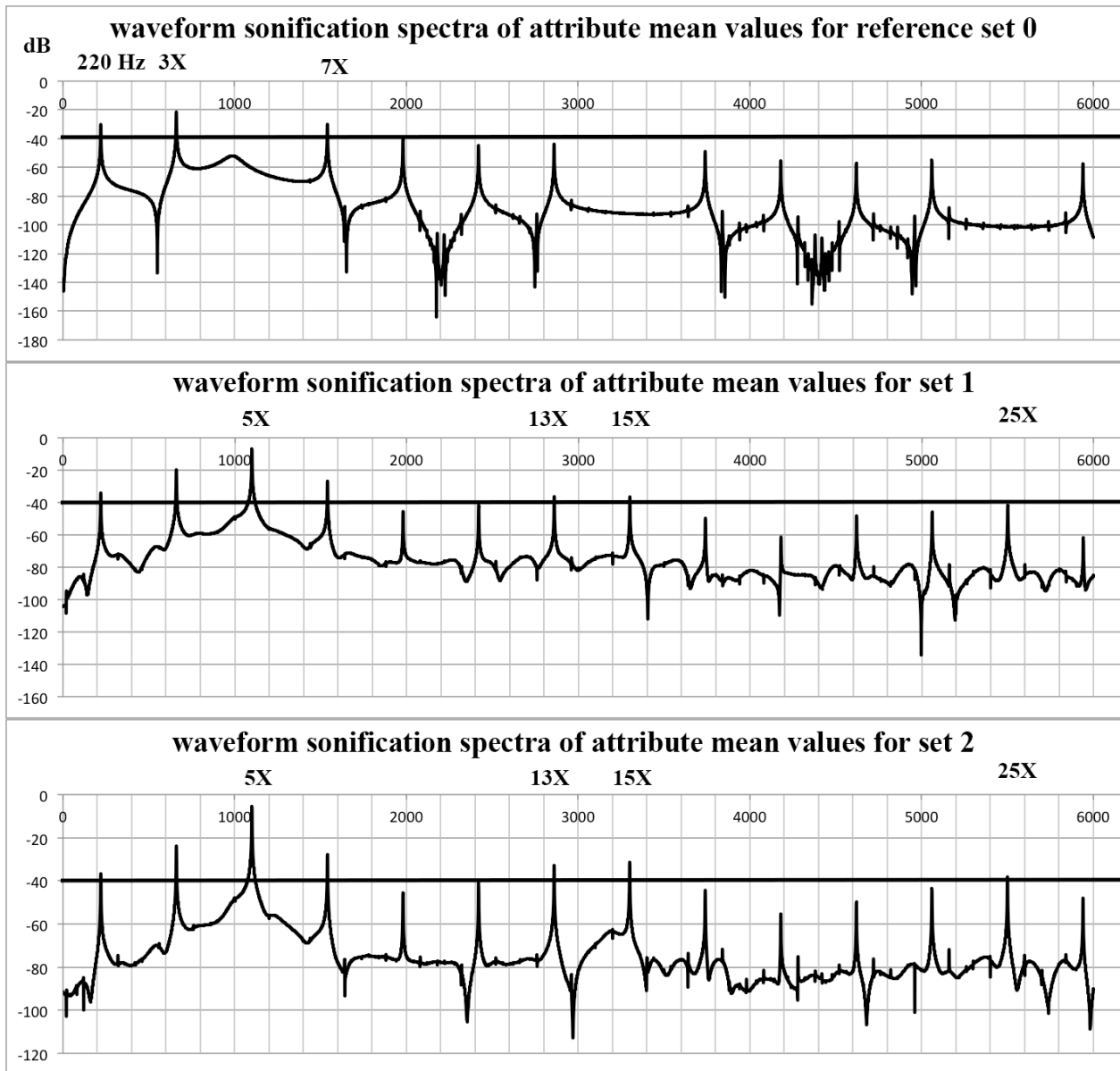
**Table 2: Correct Classifications from Sonic Survey**

Looking at Table 2 on a per-set basis, waveform at 74.8% is the best for classifying Reference Set 0 instances correctly, and at 57.8% is the best for classifying Set 1 instances. Melodic, which is generally poor, performs the best for Set 2 at 67.9%. Presumably, melodic does well for Set 2 because there is one note that distinguishes Set 2 from Set 1 (probably Cgpa), and melodic isolates that note in time. Harmonic plays that same note, albeit at the same time as all other notes, and it scores only 60.5% for Set 2. Set 2 classification appears to be the sole advantage of sequencing sounds across time in the melodic approach. Waveform comes in at 51.5% for Set 2, slightly better than half correct.

### 3.5 Analysis of Results and the Next Round

In looking at the results of Table 2 in conjunction with the right-side waveforms of Figure 5, the problem at this point is one of coming up with a variation of waveform that distinguishes Set 1 and 2 instances from each other, while improving accuracy of Set 0 classification further.

Figure 6 gives the frequency-domain spectra plots that are counterparts to the time-domain waveforms on the right side of Figure 5. They are the waveforms used in the survey. The decibel (dB) vertical scale is a log<sub>10</sub> scale for frequency strength [12], noting that humans hear sound



**Figure 6: Filtered waveform frequency-domain spectra for Reference Set 0, Set 1, and Set 2 mean values**

loudness in a logarithmic way that emphasizes faint sounds and deemphasizes very loud ones. In Figure 6, 0 dB is maximum frequency strength, and -40 dB is roughly the threshold of audibility.

Note the first peak at the left side of the Figure 6 spectra for 220 Hz, the baseline frequency of the waveforms. The dominant frequencies of the overtones are odd multiples of 200 Hz, caused by the fact that the waveforms on the right side of Figure 5 approximate triangular waves.

Note for Sets 1 and 2 in Figure 6 there are significant peaks at 5X, 13X, 15X and 25X harmonics that do not exceed the -40 dB level for Reference Set 0. These are the harmonics that distinguish Set 0 from the other two sets.

However, they support distinguishing Set 1 and 2 from each other only where their strength is substantially different. Differences in these harmonic strengths appear in Figure 6, but they are not pronounced. In fact, if a listener could distinguish waveform sonifications of Reference Set 0 instances correctly 100% of the time, then the 57.8% and 51.5% correct responses for Sets 1 and 2 would be marginally better than guesswork. With Set 0 out of the way, listeners have to distinguish remaining sounds only between Set 1 and Set 2, a 50/50 probability for pure guesswork. The actual algorithms do somewhat better than 50%, and substantially less than 100% for Set 0.

Space precludes showing time-domain and frequency-domain plots for harmonic and melodic sonification. We have extracted them, and the problem appears to be that there is too much going on. Sawtooth waveforms contribute many overtones, so the frequency-domain spectra of harmonic Sets 1 and 2 are extremely busy, with few frequencies to distinguish the two. Harmonic classifies Set 0 somewhat better as seen in Table 2, because it generates only low-frequency sine waves with few overtones, but it does not come up to the accuracy of waveform for Set 0. Melodic sequences the sounds of harmonic sonification across time, but the addition of temporal spread to sawtooth noise does not help the task of classification as evidenced by the results of Table 2.

Our plan going forward into spring 2016 is to discard the harmonic and melodic approaches, to create three new variations of the waveform sonification, and then re-run the survey. We hope to make Set 1 and 2 instances more distinguishable from each other while improving accuracy for Set 0. The three new approaches, already coded and packaged for surveys at the time of this writing, generate and mix two waveforms using the same approach as the fall 2015 single-waveform approach discussed above. The two waveforms are identical, but they start at different multiples of the 220 Hz baseline frequency. The waveformDouble approach generates and mixes identical waveforms with 220 Hz and 440 Hz baseline frequencies. This approach mixes waveform with its octave double. The waveformFourThirds approach generates and mixes identical waveforms with 220 Hz and 293.3 Hz baseline frequencies. The latter frequency is  $\frac{4}{3} X$  the former, perceived by the ear as a consonant interval. The waveformOnePt95 approach generates and mixes identical waveforms with 220 Hz and 429 Hz ( $220 X 1.95$ ) baseline frequencies, where 429 is a dissonant interval. The next round of surveys will distinguish whether a consonant octave, a consonant just fourth [12], or a dissonant second waveform generates the best distinguishing features for classification. Based on extracted waveform spectra, there are frequency-domain spikes that promise to make classification among the three sets more accurate than the approaches reported here.

### 3. Summary and Future Work

The waveform approach of mapping parallel coordinates plots that look like time-domain audio waveforms into audio waveforms works. It beats the 33.3% accuracy of random guessing among three sets, at more than double that accuracy for Reference Set 0. It is not as good at distinguishing Sets 1 and 2 in this study. They share some attribute value ranges, making it necessary to emphasize differential attributes that distinguish them.

The frequency-domain spectra for the three waveform variations of the next stage of this research include unique frequency spikes for each set type that promise to make classification more accurate. The approach generates and

mixes two identically shaped waveforms with different baseline frequencies. The relationship of the two baseline frequencies differs across the three waveform sonification variants. We will resume conducting sonic surveys in spring 2016.

There is nothing to indicate that this sonification approach is in any way geared towards our dataset or domain of data analysis. The waveform sonification technique relies solely on mapping variations in mean attribute values as a function of distance from a reference mean in terms of standard deviation to audio waveforms. The attribute sequence of Figure 2 is ordered to give an approximation of a triangle wave, with the peak in the center and the low points at the edges. It is a low-to-high-to-low amplitude sort. Attributes in other reference sets or in data records that vary significantly from the Reference Set 0 mean will generate overtones that a listener can use to classify them. A generic, domain-neutral way to map parallel coordinate plots to sounds is the result. The next round of surveys and analyses promise to increase accuracy of this approach to data sonification.

### Acknowledgements

The authors would like to thank Dr. Margaret Schedel [8] of Stony Brook University for initiating us into the world of timbral sonification at a computer music seminar at Kutztown University in March of 2014. We would also like to thank Dr. Alfred Inselberg [5] of Tel Aviv University for initiating us into using parallel coordinates for data visualization and exploration at the 2014 Worldcomp Conference in Las Vegas in July 2014, and for providing us with a generous educational discount on his Parallax visualization software for using parallel coordinates to analyze multidimensional datasets.

The authors would also like to express our appreciation for work time and software grants from the Kutztown University Research Committee and the PA State System of Higher Education's Faculty Professional Development Council program in the Joint Faculty-Student Research Grant category.

### References

- [1] T. Hermann, A. Hunt, and J. Neuhoff (editors), *The Sonification Handbook*, Logos Verlag, Berlin, Germany, 2011.
- [2] J. Han, M. Kamber, and J. Pei, *Data Mining – Concepts and Techniques*, Third Edition, Morgan Kaufmann, 2012.
- [3] D. Parson and A. Seidel, "Mining Student Time Management Patterns in Programming Projects," *Proceedings of FECS'14: 2014 Intl. Conf. on Frontiers in CS & CE Education*, Las Vegas, NV, July 21 - 24, 2012. <http://faculty.kutztown.edu/parson/pubs/FEC2189ParsonSeidel2014.pdf>

[4] D. Parson, L. Bogumil and A. Seidel, “Data Mining Temporal Work Patterns of Programming Student Populations,” *Proceedings of the 30th Annual Spring Conference of the Pennsylvania Computer and Information Science Educators (PACISE)* Edinboro University of PA, Edinboro, PA, April 10-11, 2015.  
<http://faculty.kutztown.edu/parson/pubs/StudentTimePacise2015Kutztown.pdf>

[5] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*, 2009 Edition, Springer, 2009.

[6] G. Loy, *Musimathics*, Volume 2, The MIT Press, 2007.

[7] J. Neuhoff, “Perception, Cognition and Action in Auditory Displays,” *The Sonification Handbook*, T. Hermann, A. Hunt, and J. Neuhoff (editors), Logos Verlag, Berlin, Germany, 2011, p. 63-85.

[8] M. Schedel and K. Yager, “Hearing Nano-Structures: A Case Study in Timbral Sonification,” *Proceedings of the 18th International Conference on Auditory Display (ICAD2012)*, Atlanta, Georgia, June 18 – 22, 2012.

[9] Machine Learning Group at the University of Waikato, “Weka 3: Data Mining Software in Java”, <http://www.cs.waikato.ac.nz/ml/weka/>, January 2016.

[10] Witten, Frank and Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Third Edition, Morgan Kaufmann, 2011.

[11] Kapur, Cook, Salazar and Wang, *Programming for Musicians and Digital Artists: Creating music with ChucK*, Manning Publications, 2015.

[12] G. Loy, *Musimathics*, Volume 1, The MIT Press, 2006.