The Audio Abacus: Representing Numerical Values with Nonspeech Sound for the Visually Impaired

Bruce N. Walker, Jeffrey Lindsay, and Justin Godfrey
Sonification Lab, School of Psychology
Georgia Institute of Technology,
654 Cherry Street
Atlanta, GA, USA 30332-0170
404 894 8265
bruce.walker@psych.gatech.edu, gte457e@mail.gatech.edu, gtg470s@mail.gatech.edu

ABSTRACT
Point estimation is a relatively unexplored facet of sonification. We present a new computer application, the Audio Abacus, designed to transform numbers into tones following the analogy of an abacus. As this is an entirely novel approach to sonifying exact data values, we have begun a systematic line of investigation into the application settings that work most effectively. Results are presented for an initial study. Users were able to perform relatively well with very little practice or training, bonding well for this type of display. Further investigations are planned. This could prove to be very useful for visually impaired individuals given the common nature of numerical data in everyday settings.

Keywords
Sonification, visually impaired, point estimation, auditory display

INTRODUCTION
In the broadest sense, many auditory displays have been designed to present information about numbers, and the relations between those numbers, to a listener. Such sonifications [1] have been used in cases where the primary task of the listener is to extract meaning from the auditory display, and also when the display is used in conjunction with a visual or other type of task. In either case (unimodal or multimodal), there are at least two distinct and basic information extraction tasks that the listener can perform with the auditory display: trend analysis and point estimation.

Trend analysis is the task of determining patterns in the data set over the course of several data points, such as determining if the price of a stock is rising or falling. Point estimation is the task of determining as nearly as possible the exact value of a data stream at a specific point in time or space. For example, a broker might want to know the specific price of a stock on a particular date. Both tasks are important, though they may be differentially important in particular task settings. An auditory display designer must assess the needs of the user, and, following such a task analysis, create a display that supports the correct goals.

Trend analysis and point estimation are both tasks that occur frequently during everyday activities. For sighted individuals often a graphical or numeric display is used to convey trend information or specific values. However, for those with visual impairments such displays can be difficult to use, if they are usable at all. For example, when in the checkout line at the grocery store, many cash registers have some type of tone to signal a successful scan/entry of an item to be purchased. If a sighted individual wishes to check the price of an item as it is scanned, all they need do is glance at the screen on the register. For a visually impaired individual, however, there is no way to verify the price of an item short of a verbal inquiry. What if instead of a simple tone each time an item is scanned, there was a sound that indicated the price of the item? This would allow the visually impaired shopper to hear the prices of items as they are scanned and also allow the cashier to have a nonvisual verification that items have been priced correctly. Most sonifications are not created with the goal of supporting point estimation tasks. For this reason it is vital that approaches that facilitate this task, such as the Audio Abacus, be investigated.

Many sonifications use changing data values to drive changes in pitch, loudness, tempo, or other sound qualities [e.g., 2, 3-6]. Some sonifications have used a granular approach, wherein a display is built up from small elements, ranging from brief tones to sampled words to frog calls [e.g., 7, 8, 9]. There has, understandably, been a growing interest in determining how best to design such displays to support a range of analysis tasks. For example, Flowers and Hauer [10] found that important characteristics of data, such as slope, shape, and level, were perceptually salient when sonified, and that people could interpret the trends or shapes of line graphs where each data point was represented with a musical note. Flowers, Buhman, and Turnage [11] found
that auditory scatter plots are effective at conveying magnitude and sign of correlations. Individuals have also been able to match auditory graphs with two data series to visual graphs [12]. Brown and Brewster [13] suggest that individuals can interpret and draw at least two data series from an auditory graph. Barrass, [14], Walker, [15], and others have looked at the appropriate choice of mapping, scaling, and polarity, to further enhance the effectiveness of auditory displays. Nevertheless, most of the displays have been designed in a fundamentally similar manner: a time-series of sounds represents a time series of data. While this basic approach has been shown to be very effective for a great variety of trend analysis tasks, we contend that it is not ideal for point estimation.

**Improving Point Estimation**

Smith and Walker [16] have pointed out some techniques that can be used to improve point estimation. Adding additional context sounds that perform like axes and tickmarks in an auditory graph can improve performance; this approach has been used in sonification designs usually because the added sounds seemed to help, and not for any theoretical reason. While Tufte [17] has developed several guidelines for displaying points on a visual graph, yet it is still unclear which, if any, of these principles apply to sonification concepts. Training listeners how to do the complex sub-steps (e.g., interpolation) required in a point estimation task is another way to improve performance [16]. However, there are limits as to the performance that can be obtained in any of these examples. According to Quinones & Ehrenstein [18], training is defined as attempting to change individuals in a manner that is consistent with task requirements and as a way of applying principles of human learning and skill acquisition. Training to improve performance on sonification-based tasks is a relatively new area of research. Upson [19] saw small improvements in students’ mathematical abilities after using sonification exercises for educational purposes, after having participants go through a short tutorial on his SoundGrid program. Peres and Lane [13] conducted a study in which participants were asked to identify a box plot from a multiple choice selection after hearing a sonified representation of the visual graphs. However, performance on the task did not increase with practice despite a 15 minute training session and 50 experimental trials. Was this lack of performance increase due to a real limitation on human abilities to perform a point estimation task? One of the challenges pointed out by Peres and Lane [13] and others is that getting good numerical resolution is hard because of the relatively large just noticeable differences (JNDs; and perhaps more importantly, the reliably noticeable differences) in the auditory system. That is, if one uses, for example, MIDI notes to represent changes in a value, there is a limit to the actual numbers one can represent, as well as a limit to the pitch recognition abilities of the listener. To represent 100 separate values, 100 perceptually distinct pitches need to be available. In practice this is very untenable. We suggest that there must be other, categorically different ways to represent exact data values, and present one example of a different approach to this thorny problem.

**The Abacus**

In order to overcome the need to have a great number of sounds to represent a large range of data values, we took inspiration from the concept behind the abacus. The abacus (see Figure 1) is an ancient counting device (though still used today in many places!) that has a frame that holds wires, on which beads can move freely. In the simplest form, each wire holds a set number of beads (e.g., 9), and the beads on each wire represent different units. For example, the beads on the right-most wire each represent one item (scoop of grain, horse, etc.), the next wire represents tens of units, then 100s, 1000s, and so on. To start (i.e., to zero the abacus), all beads are slid to one side of the abacus. That is, the absence of any beads on the “counting” side of the abacus indicates zero in that unit. To record two units, two beads from the right-most wire are moved to the opposite side of the frame. Moving two of the beads on the second wire across would indicate the addition of 20, for a total of 22. In actual practice, most “modern” abaci use a more sophisticated counting scheme, but the simpler version described here is sufficient for our purposes. Building on this approach, we decided to use a series of sounds to represent the different units, with the pitch of each sound representing a value from 0 to 9. In this way, only nine distinct pitches need to be used to represent value within a unit, and with just four separate sounds one can represent the range of 0-9999. With this concept, we have created the Audio Abacus.

![Abacus diagram](image_url)
The Audio Abacus

The Audio Abacus is a Java program designed to allow the sonification of discrete data points, for example “582”. The basic idea is that in order to represent an exact numerical value, a series of brief sounds (in this example, three sounds) are played in succession. Each of the sounds can, itself, be one of ten notes produced by a MIDI instrument. For example, if zero is mapped to MIDI note 60, then “1” could be mapped to note 61, “2” to note 62, and so on up to note 69. To “play” the number 582, the program would play a three-note series composed of MIDI notes 65, 68, and 62. The first sound obviously represents the 100s digit, the second sound represents the 10s, and the third sound represents the 1s. Given this basic scheme, there are a huge number of additional attribute settings that one can make, in theory. In order to experiment with this form of number presentation, and see how successful it might be in any of its various incarnations, we created the Audio Abacus. The Abacus software has a multitude of settings that provide for a wide range of possible manipulations. Many of these settings will be studied and manipulated in the future to determine their effect on users’ ability to interpret sonified data.

Pitch

In the example above, the values within each unit or digit were mapped onto sequential MIDI notes. Using musical notes has the benefit that the sounds are largely equated in terms of perceptual separation (notes 60 and 61 are the same distance apart in perceptual space as notes 63 and 64, so that the numbers “0” and “1” are heard as having the same separation as the numbers “3” and “4”). However, there is no reason that the numbers need to be exactly one note apart. They could be two or five notes apart, in order to make the tones (and therefore, the numbers) more distinguishable. In order to support this flexibly, a pair of sliders allows the minimum and maximum note numbers to range from MIDI note 0 to 127. The tones representing the digits 0 – 9 are equally spaced between the minimum and maximum pitch value. For the current study (described below), minimum pitch was set at note 40 and maximum at note 90.

Pan

In the basic three-note sequence described at the outset, all the notes could be played through one central speaker (or over headphones with pan set to the center). Alternatively, each sound could be played in different spatial locations. For example, the 100s could be to the left, the 10s in the center, and the 1s off to the right. In that way, the unit or digit would be doubly mapped to the order in the series (largest played first) and to the spatial location it appears to come from (largest to the left). To support this panning possibility, we again use two sliders to set the minimum and maximum “panning position”. For example, a minimum pan value of 0 and a maximum of 127 would cause the digits of the number to be equally spaced through the listener’s sound image, ranging from all the way to the left (-90°) to all the way to the right (+90°).

To further illustrate the combination of pitch and pan, consider the number “145” being sonified. The listener would hear the note 61 (representing “100”) played on the left side of their sound image, the note 64 (“40”) in the center, and the note 65 (“5”) to their right. The user would be required to add up the digits (100 + 40 + 5) to
comprehend the overall value of 145. Note, however, that for the current study, the pan function was not used.

**Decimal pass**

Representing the numbers 145 and 1.45 (e.g., $1.45$) is conceptually the same—each sound represents a unit or digit, but in the case of decimal digits the units are tenths and hundredths. What does change is communicating to the user that there is a decimal, and then indicating where it is in the sequence. First of all, a decimal point is represented by the sound of a high hat cymbal crash (percussive, largely non-pitched). The way in which a decimal point is represented within the number sequence can be changed using three radio buttons in the Audio Abacus. The radio buttons determine the relationship of the other tones to the cymbal crash. This is especially relevant when the spatial panning is being used.

The first setting, “Single Pass L -> R”, simply plays the series of tones from left to right. Therefore, if the number 145.34 is played, five digits are divided into the sound image, with a cymbal crash between sounds 3 and 4. If the sound is spatialized, the decimal occupies a spatial location in keeping with its order in the series (so in this case, there are actually six sounds, in six successive spatial locations).

The second setting, “Double Pass L -> R . L -> R”, plays all of the whole units in series, left to right, plays the decimal sound, and then returns to the beginning (left) of the sound image before the digits after the decimal point are played. For example, when the number 145.34 is played on this setting the digits “1”, “4”, and “5” will be divided into three equal sections of the listener’s sound image (also influenced by the Pan setting described above). After the decimal cymbal crashes, the sound image is “reset”, and the digits “3” and “4” will be divided into two equal sections of the user’s sound image.

The final setting, “Bounce Pass L -> R . R -> L” plays numbers to the left of the decimal point from left to right, similar to the first setting. However, the numbers to the right of the decimal point are played going from right to left across the sound image. Consider, again, the number 145.34. For the decimal digits “3” and “4”, the “3” would be played on the right of the sound image and “4” would be played on the left.

Of course, many other possible settings could be implemented. These are just three that we thought were conceptually distinct. For the current study, only the first setting, “Single Pass L -> R” was used.

**Ratio**

The amount of time that each sound in a series plays can also be adjusted on a per-digit basis. For example, the 100s unit sound could be 3 seconds long, with the 10s sound 2 seconds, and the 1s sound 1 second long. This might assist in distinguishing the different units (as the spatial separation does), and it might also allow for more processing time for the larger units (100s), which might be more relevant. However, if all the data values are changing between 194 and 199, then the largest unit is probably less interesting, so it need not play for so long, in relation to the other digits. In that case a sound length ratio of 1:1:1 or even 1:1:3 might be most appropriate. To support this flexibility, sliders allow the duration of each note in front of the decimal to be set relative to one another. For example, if the ratios were set to 3, 2, and 1, then in the number “145” the digit “1” would have the longest duration, “4” slightly less, and “1” would have the shortest duration. The precise length of one unit of time is determined by yet another slider, namely the Time Scale setting, described below. For the current study, all of the ratios were set at 1 until further experiments call for manipulation of these values.

**Time Scale**

This slider changes the duration of one beat for the series of tones. Thus, if the 100s unit were set to have a ratio value of 3 beats, and each beat is set to 1 second, the 100s would play for 3 seconds. It ranges in value from 1/5 second to 5 seconds. For the current study, this slider was set at 1 second. Note that by having this seemingly complicated ratio setup, the relative lengths of the sounds can be maintained, but the whole series can be sped up or slowed down. This helps with early training trials being played slowly, and then later trials can be played quickly, without changing the relative sound lengths.

**Decimal Beat**

Allows a separation between the end of the tones representing the digits to the right of the decimal point and the start of the tones representing the digits to the left of the decimal point. For the current study this setting was left at 1.

**Instrument panel**

The default MIDI instrument used to play the notes is the piano, but the Abacus allows the user to choose any MIDI instrument for representing the digits. The instrument panel in the application shows the choices, and lets the user easily pick an instrument.

Because the program is written in Java, it can operate on any system with Java installed. The code itself resides in a Java Archive (JAR) file for simply portability and execution. Abacus imports numbers in from text files, most commonly comma-delimited. Alternatively, users can also type numbers into the cells provided to have them sonified. Clicking on a particular cell will play just that number, whereas clicking the “Play” button plays the whole data set, with each entry in the data set being turned into separate series of sounds. The series are then all played back to back.

**EVALUATION**

In order to determine if this is even a viable approach to sonifying exact data values, we have begun a series of systematic studies in which various parameters of the application are considered. Clearly this is an ongoing process, as a number of options can be modified. We present initial results of the first experiment where we asked participants to try to determine the prices of individual stocks from a fictitious data set.
Method

Participants
Thirty undergraduate participants (21 females and 3 males) took part in the study for course credit. The average age of participants was 19.6 years (SD = 1.3), and 20 had played a musical instrument regularly. Of those who had played a musical instrument, they had started playing at an average age of 9.6 years (SD = 2.7) and had been playing an average of 5.9 years (SD = 3.8), to include formal instruction (M = 4.9 years, SD = 3.0). Some of the instruments participants had played or were currently playing included piano, oboe, violin, french horn, and flute. All participants were right handed.

Materials / Apparatus
Participants were tested using a Dell Inspiron 8500 laptop computer, with a 2.20 GHz processor, 512 MB of RAM, and running Windows XP Professional with Java v1.4.2 installed. Stereo headphones (Sony MDR-7506) were used to play the tones.

Procedure
Participants were informed that they were taking part in an experiment designed to assess how well individuals could interpret simple sounds to deduce information about specific prices in the stock market. Subjects were first asked to calibrate the volume of the headphones, to ensure they could hear all the tones at a safe and comfortable level.

Next, each participant was taught how the sounds would represent information and allowed practice trials to ensure they understood how the study would operate. Subjects were told to first listen to 10 tones being played, representing the digits 0 to 9, with the added caveat that as pitch increased, so did the value of the digit the tone represented. Once the participant was confident with this concept, simple numbers such as 10, 30, 100, and 1000 were played. After the simple numbers were understood, the participant was taught how decimal points were represented, and then listened to a couple of numbers containing decimals. After completing the learning phase, each participant was allowed five practice trials, during which time they attempted to write down the number they thought the practice tones represented. Once being told the correct answer, a subject was allowed to listen the tones again to attempt to recalculate their listening.

Finally, the participant moved into performing the task of interest. Five blocks of ten numbers each had been randomly generated prior to the experiment, and each subject would listen to the same 50 numbers. Each participant was instructed that the first four blocks would have either one, two, three, or four digits in front of the decimal point, with all trials in a block having the same number of digits. However, the fifth block would contain numbers with one, two, three, or four digits in front of the decimal point. The experimenter played each tone only once, after which the participant wrote down the number that they felt the sounds represented. When ready to move on, the participant signaled the experimenter to move on to the next trial.

Once participants had completed all five blocks, any questions they had were addressed, the experimenter asked them what they thought of the study, and thanked them before they left.

Results
Initially all participants’ responses were coded as either successful or unsuccessful based on whether their answer contained the same number of digits as the number presented by the abacus. The unsuccessful responses were then excluded. It is important to note that unsuccessful responses accounted for only around 10 percent of all responses across all blocks.

The data were then analyzed by comparing the actual number presented by the Abacus to participants’ responses. The primary result of interest was the absolute error across all values. This error was calculated by subtracting each digit in the actual number from the corresponding digit in the participants’ response and then taking the absolute value of this difference (e.g. participant’s response=4, actual response=3; 3-4=1; absolute error=1).

Plotting absolute error reveals that 80 percent of participants’ responses were within two of the actual value, and 90 percent of all participants’ responses were within three (Figure 3). This data supports the fact that participants are able to determine the digit presented by the abacus with reasonable certainty.

GENERAL DISCUSSION
There are two key points the pilot data described in this data make evident. The first is that users are able to relatively successfully use an interface such as the Audio Abacus to perform a point estimation task, with only very limited training or introduction. This bodes well for the use of such an alternative display method in cases where exact data values are needed. Given the multitude of display options provided by the program, the data presented here barely scratches the surface of the myriad of possible interface configurations that the Abacus is capable of. It is not possible at this time to predict with
the pilot data what will be the critical aspects for this type of interface to be most useful to potential listeners, but it is exciting to look forward to a whole line of investigation aimed at uncovering what works best with this new interaction method. Other important avenues of investigation include the effect of practice and training and the effect of real world contexts on Abacus users’ performance.

The second important point to be made is the need for more research into sonification as it relates to point estimation tasks. There exists a large body of research that clearly demonstrates the efficacy of the use of sound to convey information about trends [e.g., 10, 11]. However, this preliminary data indicates that new approaches, “thinking outside the box” so to speak, have the potential to expand the range of effective sonification to include more nontraditional tasks. By no means a definitive statement, the data presented here are merely the tip of the iceberg.

REFERENCES


