THE AUDIO ABACUS: REPRESENTING A WIDE RANGE OF VALUES WITH ACCURACY AND PRECISION

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
bruce.walker@psych.gatech.edu

ABSTRACT

Point estimation is a relatively unexplored facet of sonfication. 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, boding well for this type of display. Further investigations are planned.

1. 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.

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.

1.1. Point Estimation in Daily Life

While the sonification of scientific data is an important venue for point estimation tasks, and has accordingly received the majority of research attention, there are certainly other situations where sound could be appropriate and desirable for communicating exact values. Household devices ranging from ovens to tape measures to thermostats produce numerical data that are typically displayed visually. For the visually impaired, or for workers who cannot look due to attentional demands (e.g., while cutting wood with a saw) or cannot see due to environmental conditions (e.g., firefighters in a smoky building), an auditory representation of a specific value can be very useful.

In many cases, a simple text-to-speech approach may work well. However, that is definitely not the only approach, and in some cases it may even be undesirable for privacy reasons (e.g., banking situations) or in consideration of human information processing limits. For example, in a situation where there is already spoken communication (e.g., firefighters), using non-speech audio is considered more effective since it uses a different information processing channel {Wickens, 1988, #21}. A novel and creative approach to non-speech auditory point estimation is certainly worth investigating.

1.2. Improving Point Estimation

It is important briefly to consider ways to improve existing approaches to auditory point estimation, in the event they can be as effective as one would need, or else provide insight into how any new approaches can be more effective. Smith and Walker {, 2002 #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. For example, adding a click track to help in parsing the time axis in a sonification has been used in several past sonification designs usually because the added sounds "just seemed to help," and not for any theoretical reason. Regardless, we can determine from this that some timing assistance may be useful. There are also many guidelines that come from the visual design world that might be relevant to auditory point estimation. Recommendations for conspicuity, detectability, and discriminability are well known in visual graph design [16]. Unfortunately 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 [17]. However, there are limits as to the performance that can be obtained in any of these examples. Upson [18] 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 [19] 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. One of the challenges pointed out by Peres and Lane [19] 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, with non-speech audio, and we present one example of a different approach to this thorny problem.

2. 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 (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 (a scoop of grain, horse, etc.), the next wire represents tens of units, then hundreds, thousands, and so on. To start (i.e., to zero the abacus), all beads are slid to one side of the abacus (often the side closest to the user). 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 wire second from the right 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 set of sounds to represent the different units, with the pitch of each sound representing a value from 0 to 9. In this way, only 10 distinct pitches need to be used to represent value within a unit, and with just four separate sounds in a set, one can represent the numerical range of 0-9999. With this concept, we have created the Audio Abacus.

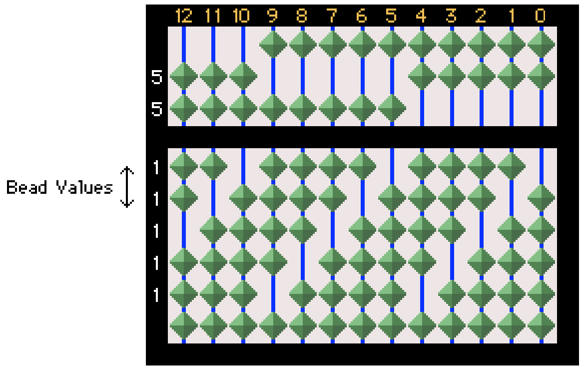

Figure 1. *Sample abacus, showing each wire with different beads. In this sophisticated version, counting is aided by the use of the beads in the upper register to indicate groups of 5. Simpler versions just have 9 beds on a single wire for each unit or digit.*

3. THE AUDIO ABACUS

The Audio Abacus (see Figure 2) 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 set 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 sequence composed of MIDI notes 65, 68, and 62. The first note in the sequence obviously represents the hundreds digit, the second note in the sequence represents the tens, and the third note represents the ones digit.

This is obviously not the only possible way to map the numbers onto sound attributes. Different timbres could be used, tempo, and timing, among others, could be employed. We decided to start with the simple approach described above for the core data representation. Even still, there are a huge number of additional attribute settings that one can use in the display, especially when more complicated examples are considered (e.g., larger numbers, numbers with decimals). In order to experiment with this form of number presentation, and see how successful it might be in any of its various incarnations, we designed the Audio Abacus to be quite flexible. The Abacus software has a multitude of settings that provide for a wide range of possible manipulations. We are studying many of these settings to determine their effect on users' ability to interpret sonified data.

3.1. Pitch.

In the mapping approach described above, the values within each unit or digit are mapped onto sequential MIDI notes. Using musical notes has the benefit that the sounds are largely equated in terms of perceptual separation (i.e., 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,

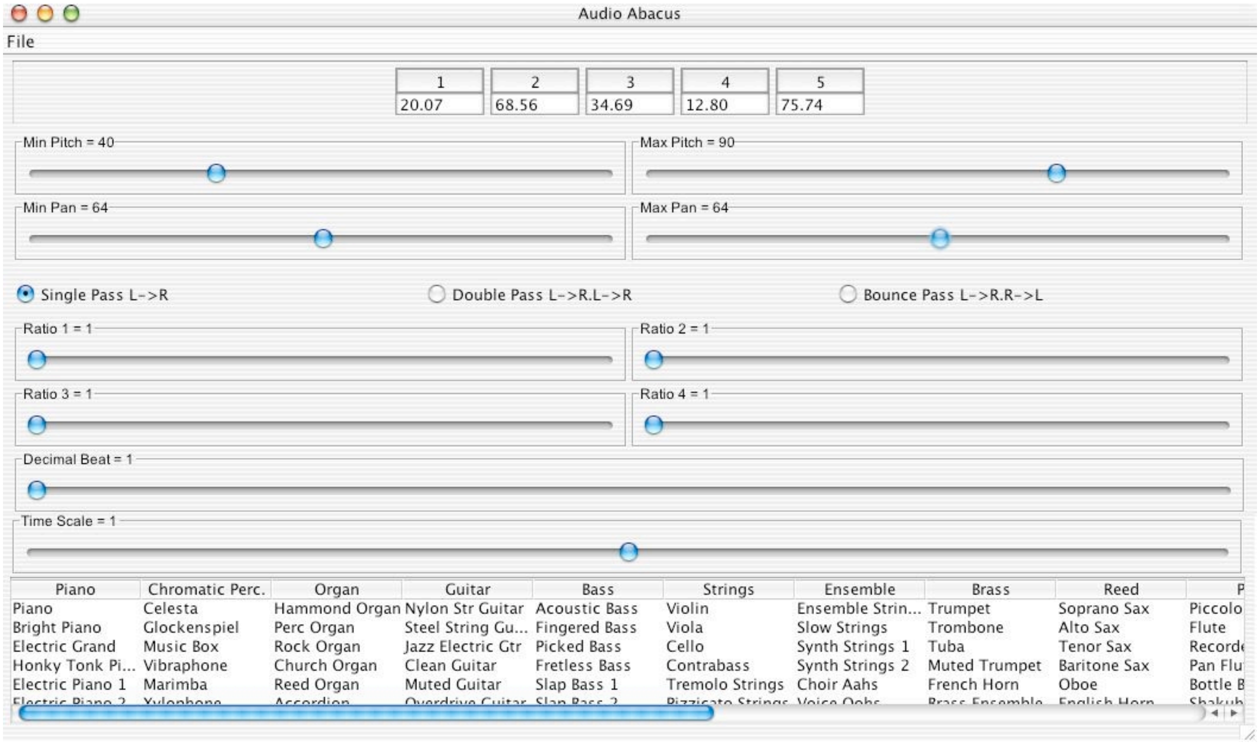

Figure 2. *Screen capture of the Audio Abacus. Generates sounds from data values. See text for description.*

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 (see Figure 2). 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.

3.2. 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 hundreds could be to the left, the tens in the center, and the ones 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 degrees) to all the way to the right (+90 degrees) (see Figure 2).

To further illustrate the combination of pitch and pan, consider the number "145" being sonified with the mapping described at the beginning of Section 3. The listener would hear the note 61 (representing the value "100") played on the left side of their sound image, the note 64 (representing "40") in the center, and the note 65 (representing "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.

3.3. Decimal pass.

Representing the numbers 145 and 1.45 (e.g., $1.45) is conceptually the same in the Audio Abacus—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. For example, three note sequences would represent each of the numbers 145, 34.8, and 6.47. the system must include a method for disambiguating that series from 1.45, 34.8, and 647. In the Audio Abacus, a decimal point is represented by the sound of a high hat cymbal crash (percussive, largely non-pitched). So in this simplest case the number 1.45 would be represented by [note, cymbal, note, note], with the actual notes depending on the pitch scaling in use.

There are other approaches to indicating a decimal, and it remains to be seen which is most effective, so the Abacus has at least three methods available. The "Decimal Pass" radio buttons control these three methods (see Figure 2), which only differ in the order of notes being played, and their spatial location (when panning is active).

The first Decimal Pass setting, "Single Pass L -> R", simply plays the series of notes in one complete sequence (or *pass*), and pans them from left to right. Therefore, if the number 145.34 is played, the note representing the "1" (actually 100) is played all the way to the listener's left, then the "4" (40) is played a little to the right of that (at 45 degrees), followed by the "5" directly in front. Then, to indicate the transition from whole numbers to decimals, the cymbal is played (it is always played in the center of the sound space), and in this example would be followed by the "3" (or 0.3), played 45 degrees to the right, and then the "4" (0.04) played all the way to the right. Thus, there is one continuous left-to-right pass, with the decimal (cymbal) being played to demarcate the start of the decimal digits.

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 decimal digits 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), and played in sequence. 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, so in this case, "3" played to the left and "4" played to the right.

The final setting, "Bounce Pass L -> R . R -> L" plays whole numbers from left to right, similar to the Double Pass setting. However, after the cymbal crash signifying the decimal, the decimal digits are played going from *right to left* across the sound image. That is, the note sequence starts at the left, moves over to the right with each successive note, and "bounces" back towards the left when playing the decimals. 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, in contrast to the patterns described for the Double Pass, above.

It bears explaining why these different approaches are available for indicating the presence and location of a decimal. In practice with the system, if numbers like 123.4 are included in the same series as 12.34 then the left-most (and first-played) note indicates 100 in one case, and only 10 in another case. Further, the two data values have different numbers of decimal digits, so there needs to be a way to sort out what a given sound in the sequence actually represents (10, 1, 0.1, 0.01, etc.). The Single Pass approach is the simplest, but it makes sorting out the value of a sound in any given spatial location difficult. The Double Pass method "aligns" the numbers so that the right-most tones in the first pass are always ones, the second-from-right tones are always tens, and so on. However, the decimal digits then can have the same issues of alignment. Thus, the Bounce Pass is a way to ensure that regardless of the number of digits in a given data point, the location of a sound is a consistent indicator of that sound's value (i.e., the place it holds: tens, hundreds, etc.). Thus, each approach is intended to solve some issue related to this novel data representation; but each, in turn presents its own issues, so all of them will need to be investigated and studied for utility and usability.

Of course, many other possible settings could be implemented, and it all may seem overly complicated. However, it is really no different from the complications that arise when using exponential (i.e., engineering) notation (1.34567E+10) and standard notation (9874.234) at the same time. It is just an issue of sorting out what works in a given situation, and developing a standard way of displaying the numbers. For the current study, only the first setting, "Single Pass L -> R" was investigated.

3.4. Ratio.

The amount of time that each sound in a sequence plays can also be adjusted on a per-digit basis. For example, the hundreds unit sound could be 3 seconds long, with the tens unit sound lasting 2 seconds, and the ones unit sound 1 second long. This sound duration ratio of 3:2:1 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 (hundreds), which might be more relevant in some cases. For example, if the Abacus system were implemented in a grocery checkout line, with each food item's price being sonified when it is scanned, it would be mostly relevant for the listener to pay attention to the major digits, to ensure that the lettuce was not scanned in a $100, rather than $1.00. However, if all the data values are changing between 194 and 199, then the largest unit is probably less interesting, so the system need not play the hundreds digit 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.

3.5. Time Scale.

The Time Scale slider (see Figure 2) changes the duration of one beat for the series of tones. Thus, if the hundreds unit were set to have a ratio value of 3 beats, and each beat is set to one second, the hundreds digit 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 one second. Note that by having this sophisticated 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.

3.6. Decimal Beat.

The actual length of the cymbal crash that indicates the decimal place (as described at length, above) can also be adjusted. This allows the user to increase or decrease the temporal separation between the end of the tones representing the whole number digits and the start of the tones representing the decimal digits. For the current study this setting was left at unit of time.

3.7. 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 (see Figure 2) shows the choices, and lets the user easily pick an instrument.

3.8. Java Coded Interface

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.

The Abacus imports data from text files and most commonly comma-delimited values (CSV) files. The data values are displayed in the text cells at the top of the interface. Alternatively, users can type numbers into the cells provided. Clicking on a particular cell will play the sound sequence representing 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 with a pause between sets of sounds.

4. EVALUATION

In order to determine the viability, effectiveness, and potential limits of this 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 and examined systematically. We present results of the first experiment where we asked participants to try to determine the prices of individual stocks from a fictitious data set, with a specific data-to-sound mapping in place (described below).

4.1. Method

4.1.1. Participants

Fifteen undergraduate participants (12 females and 3 males) took part in the study for course credit. The average age of participants was 19.7 years (SD = 1.5), and 11 had played a musical instrument regularly. Of those who had played a musical instrument, they had started playing at an average age of 9.4 years (SD = 2.5) and had been playing an average of 6.2 years (SD = 4.3), including an average of 4.6 years (SD = 2.8) of formal instruction. Some of the instruments participants had played or were currently playing included piano, oboe, violin, and flute. All participants were right handed.

4.1.2. 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.

4.1.3. Procedure

Participants were informed that they were taking part in an experiment designed to assess how well individuals could interpret simple sounds intended to represent 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 clarification 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 recalibrate 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 of trials would have either one, two, three, or four digits in front of the decimal point (e.g., 1.34, 23.45, 345,67, and 4567.89, respectively), with all trials in a block having the same number of digits. However, the fifth block would contain numbers with a mixture of one, two, three, or four digits in front of the decimal point. All numbers had two decimal digits.

The experimenter played each multi-tone sequence only once, after which the participant wrote down the number that they felt the sound sequence 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.

4.2. 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 and all participants.

The response 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 on each digit, summed across all digits. 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). This error value is a measure of how close the listener's estimate was to the intended value of each digit, and does not consider the actual numerical value of the response.

Plotting the summed 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 (see Figure 3). This data supports the fact that participants are able to determine the digit presented by the abacus with reasonable accuracy. That is, the basic idea of mapping the numbers from 0 to 9 onto pitches is viable, even with very limited training or practice. This bodes well for the utility of the system, since the mapping in use in this study was the simplest, even impoverished mapping, devoid of other cues that the Abacus does support, such as the spatial separation of the digit sounds.

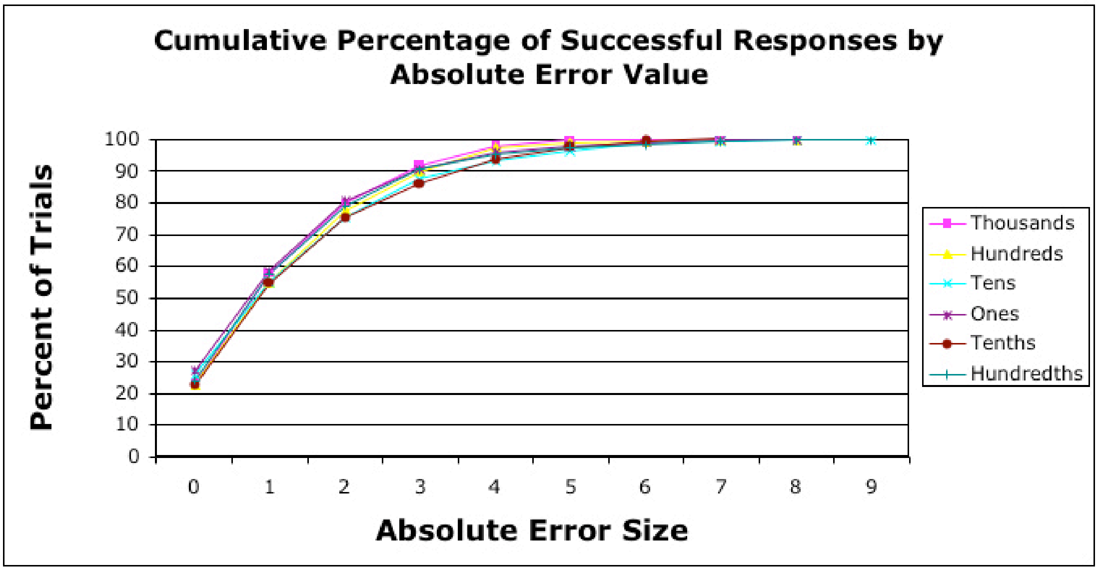

Figure 3. *Percentage of successful responses for each digit. Each line represents a digit place. Note that approximately 80 percent of trials are within 2 numbers of the actual value.*

5. GENERAL DISCUSSION

There are several key points these 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. The simple mapping of numerical values onto pitches, and the assembly of from three to six pitches into a sequence representing a complete number seems to be basically effective. The pitch mapping used here allowed listeners to come close to the intended values right from the start. Further, there are few differences between the accuracy on the digits. That is, the hundreds are about as accurate as the tens or the ones. This is important for the scalability of the system.

Given the multitude of display options provided by the program, the data presented here barely scratch the surface of the possible interface configurations that the Abacus supports. It is exciting to look forward to a whole line of investigation aimed at uncovering what works best with this new interaction method. It is likely that the addition of the spatial separation and different temporal ratios will make the system even more effective. It will also be important to consider the effect of both unaided practice and intentional training, as well as the effect of real world contexts on Abacus users' performance.

The work presented here involves unitary point estimation, in which a single set or sequence of tones is played to represent a single data point such as 234.67. However, as already mentioned, the Auditory Abacus also supports the sequential playing of multiple data values, such as the stream of data that would come from a stock market feed, or a file containing the daily high temperatures for the past month or year. This functionality is important, since the combination of exact data point communication for each data value and the handling a continuous series of data values would make for a very effective auditory graph. Such usage would presumably allow for both trend analysis and point estimation. In the Abacus, once the listener has learned the pitch and spatial mappings well, the overall speed of the playback can be increased without altering pitch or duration ratios of the individual tones in a sequence. This is accomplished by shortening the "beat" unit, described previously, which will shorten the whole sequence accordingly. As is the case with screen reader software for the visually impaired, practice should lead to extremely fast presentation speeds still being comprehended by the (experienced) listeners. A complete four or six digit number can be played in around a second or two, depending on the speed and ratio settings. This is comparable to the length of time that many "traditional" sonifications play a single pitch to represent a number. Thus, a lot more information can be conveyed in the same amount of time. It remains to be seen how effectively such a presentation can communicate both the trends and the details of a data stream, but it seems from our experience that that both can be tracked with the Abacus-style presentation.

Presenting a series of rapid sequences to display a data set is a situation where test-to-speech output (i.e., saying the numbers) is completely unhelpful. That is, the length of time that it takes to say the numbers, and the fact that the acoustic properties of the words "eight", "three", and so on, have no relation to the value of the number they represent, make speech ineffective for a rapidly presented auditory graph. All things considered, the Auditory Abacus brings a fresh approach that can overcome limits of previous sonification and auditory display methods.

Another broader point that is important to make is the overall need for more research into sonification in general, 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 (discussed above). 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. The Audio Abacus is jut one step on this path, and it seems to be an interesting application that will serve as a catalyst to turn up a wealth of new approaches.

6. REFERENCES

[1] G. Kramer, B. N. Walker, T. Bonebright, P. Cook, J. Flowers, N. Miner, J. Neuhoff, R. Bargar, S. Barrass, J. Berger, G. Evreinov, W. T. Fitch, M. Gröhn, S. Handel, H. Kaper, H. Levkowitz, S. Lodha, B. Shinn-Cunningham, M. Simoni, and S. Tipei, "The Sonification Report: Status of the Field and Research Agenda. Report prepared for the National Science Foundation by members of the International Community for Auditory Display," International Community for Auditory Display (ICAD), Santa Fe, NM 1999.

[2] B. N. Walker and G. Kramer, "Mappings and metaphors in auditory displays: an experimental assessment," Proceedings of 3rd International Conference on Auditory Display (ICAD96), Palo Alto, CA, 1996.

[3] J. A. Gardner, R. Lundquist, and S. Sahyun, "TRIANGLE: A practical application of non-speech audio for imparting information," Proceedings of International Conference on Auditory Display, Palo Alto, CA, 1996.

[4] T. T. Hermann, P. Meinicke, and H. Ritter, "Principle curve sonification," Proceedings of International Conference on Auditory Display, Atlanta, GA, 2000.

[5] B. Sturm, "Sonification of particle systems via de Broglie's hypothesis," Proceedings of International Conference on Auditory Display, Atlanta, GA, 2000.

[6] J. H. Flowers, L. E. Whitwer, D. C. Grafel, and C. A. Kotan, "Sonification of daily weather records: Issues of perception, attention, and memory in design choices," Proceedings of International Conference on Auditory Display, Espoo, Finland, 2001.

[7] K. V. Nesbitt and S. Barrass, "Evaluation of a multimodal sonification and visualization of depth of market stock data.," Proceedings of International Conference on Auditory Display, Kyoto, Japan, 2002.

[8] N. E. Miner and T. P. Caudell, "Using wavelets to synthesize stochastic-based sounds for immersive virtual environments," Proceedings of International Conference on Auditory Display, Palo Alto, CA, 1997.

[9] H. Petrie and S. Morley, "The use of non-speech sounds in non-visual interfaces to the MS-Windows GUI for blind computer users," Proceedings of International Conference on Auditory Display, Glasgow, UK, 1998.

[10] J. H. Flowers and T. A. Hauer, "Musical versus visual graphs: Cross-modal equivalence in perception of time series data," *Human Factors*, vol. 37, pp. 553 - 569, 1995.

[11] J. H. Flowers, D. C. Burham, and K. D. Turnage, "Cross-modal equivalance of visual and auditory scatterplots for exploring bivariate data samples," *Human Factors*, vol. 39, pp. 341 - 351, 1997.

[12] T. L. Bonebright, M. A. Nees, T. T. Connerley, and G. R. McGain, "Testing the effectiveness of sonified graphs for education: A programmatic research project," Proceedings of International Conference on Auditory Display, Espoo, Finland, 2001.

[13] L. M. Brown and S. A. Brewster, "Drawing by ear: Interpreting sonified line graphs," Proceedings of 9th International Conference on Auditory Display (ICAD2003), Boston, MA, 2003.

[14] S. Barrass, "Auditory information design," Australian National University, 1997.

[15] B. N. Walker, "Magnitude estimation of conceptual data dimensions for use in sonification," in *Psychology Department*. Houston, TX: Rice University, 2000.

[16] E. Tufte, R., *Envisioning Information*. Cheshire, CT: Graphics Press, 2001.

[17] D. R. Smith and B. N. Walker, "Tick-marks, axes, and label: The effects of adding context to auditory graphs," Proceedings of International Conference on Auditory Display, Kyoto, Japan, 2002.

[18] R. Upson, "Educational sonification exercises: Pathways for mathematics and musical achievement," Proceedings of International Conference on Auditory Display, Kyoto, Japan, 2002.

[19] C. P. Peres and D. M. Lane, "Sonification of statistical graphs," Proceedings of International Conference on Auditory Display, Boston, MA, 2003.