Hambone: A Bio-Acoustic Gesture Interface

Travis Deyle[†], Szabolcs Palinko[‡], Erika Shehan Poole[‡], and Thad Starner[‡]
[†]School of Electrical and Computer Engineering and [‡]School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332
{tdeyle, palinko, erika, thad}@gatech.edu

Abstract

Mobile input technologies can be bulky, obtrusive, or difficult to use while performing other tasks. In this paper, we present Hambone, a lightweight, unobtrusive system that affords quick access, subtlety, and multitasking capabilities for gesture-based mobile device interaction. Hambone uses two small piezoelectric sensors placed on either the wrist or ankle. When a user moves his hands or feet, the sounds generated by the movement travel to Hambone via bone conduction. Hambone then transmits the signals digitally to a mobile device or computer. The signals are recognized using hidden Markov models (HMMs) and are mapped to a set of commands controlling an application. In this paper, we present the hardware and software implementation of Hambone, a preliminary evaluation, and a discussion of future opportunities in bio-acoustic gesture-based interfaces.

1. Introduction

The mobile computing community envisions computation available anywhere and anytime. Despite the widespread availability of mobile devices, interacting with them can be challenging. For example, devices are often stored in pockets or bags, which causes an interaction delay. From a usability perspective, this delay between the desire to interact with a device and the actual interaction can be prohibitive [10].

To exacerbate the issue, buttons on mobile devices are small and their user interfaces are frequently impoverished. For example, small keypads require users to devote cognitive and visual attention to the device itself. The amount of required attention for the interface can limit the user's ability to perform other simultaneous tasks, such as driving an automobile.

Voice controlled commands are an alternative, but may not be the best option in noisy or otherwise inappropriate situations (e.g. when a radio is playing). Mobile input technologies external to the device itself can be bulky or obtrusive. Many presently-available data gloves and chording keyboards can be larger than the devices they are controlling or may evoke stares from bystanders. Data gloves are expensive, too fragile for everyday use, and may prevent users from using their hands for other purposes.

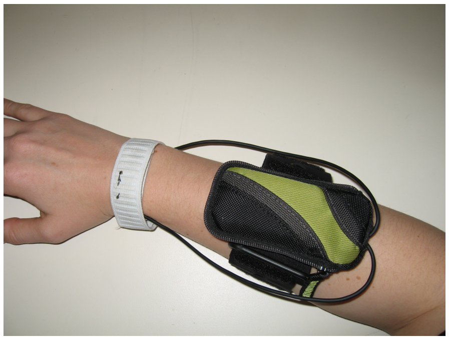

Figure 1. Hambone Prototype

In response to these issues, we present Hambone, which is a lightweight, unobtrusive system that provides gesture-based input to mobile devices. It affords quick access and the capability to interact with a mobile device while engaged in other physical activities. To use Hambone, a user places two small piezoelectric sensors on either the wrist or ankle. When the user moves his hands or feet, the sounds generated by the movement travel to the sensors via bone conduction. Hambone then transmits the signals wirelessly to a mobile device or computer. The gesture-based signals are classified using hidden Markov models (HMMs), and the recognized gestures are then mapped to a set of commands to control an application. The Hambone prototype hardware, shown in Figure 1, fits in an iPod Nano case; however, we envision that future versions of the hardware would be embedded into a wristwatch or piece of jewelry.

In the following sections, we discuss related work, our approach, hardware and software implementations of Hambone, a preliminary evaluation, and future opportunities in bio-acoustic gesture-based interfaces.

2. Related Work

Amento et al pioneered the Hambone approach, in that they used the acoustics of hand gestures captured by piezoelectric sensors to conceive a gesture-based input device [1]. However, their system had a number of shortcomings: including an extremely limited gesture set in the sample application (tap and double-tap), analyzed one sensor location, and made no mention of real-time operation. We extend the work of Amento et al by addressing these shortcomings. Specifically, we provide quantitative results for a number of complex gestures, for both one sensor (as in Amento) and for two sensors, showing that the latter leads to better gesture recognition accuracies. We also perform a qualitative evaluation wherein several users use Hambone in real-time to control realistic applications. Finally, the device functions wirelessly in a small form-factor, lending itself to true ubiquitous, wearable operation.

To understand Hambone's place in the mobile input space, we classify such devices into a few "typical" categories:

Accelerometer-Based: Use telemetry data from accelerometers to classify gestures [5]

Glove-Based: Use embedded strain gauges to measure finger orientation [11]

Buttons and Keypads: Including mini-qwerty keypads common on mobile phones [3], as well as mobile chording keypads, which are one-hand portable keyboards that provide fairly high data rates [7]. Devices with a few buttons (eg some GPS receivers) can also be classified in this category, though their input rates are typically lower.

Touch and Stylus: Use a touchscreen to recognize penstroke gestures [14]

Vision: Use a camera to process signals from the environment. Examples would include the WristCam [9, 12]

Voice: Use audio speech recognition to provide input to the system [6, 8]

Buxton provides three criteria for partitioning mobile input devices: physical, cognitive, and social [2]. Here physical refers to easy sensory coupling. A "good" physical device is easy to use for prolonged periods of time. Most input devices meet this criteria, though glove-based systems are not suited to repetitive, prolonged use. Cognitive refers to the amount of active attention and learning required to use the device. Keypads, stylus, and touch-based devices require additional sensor modes (vision) in order to operate and thus have a high cognitive load.

Finally, social refers to how others perceive an individual's use of the device. Because of common public use, many mobile input devices are socially acceptable, though a few are not. Vision-based systems require cameras, which may raise privacy concerns. Voice-based systems depend on circumstances; while voice may be acceptable while walking through the park, it may not be while sitting in classes or during movie screenings. Using Buxton's criteria, we conservatively grouped the mobile input device categories as shown in Figure 2.

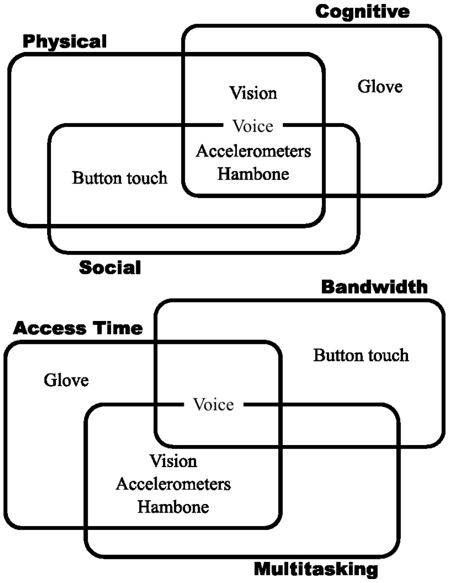

Figure 2. How mobile devices fit into Buxton's (top) and Lee's (bottom) frameworks

Lee provides a complementary structure that looks at input bandwidth (how much information can be conveyed per second), access time (the time to retrieve the device and begin interaction), and multitasking (ability to perform other activities while providing input) [4]. Keypads and voice-based devices tend to provide high relative input bandwidths. Gesture-based devices, however, tend to have lower bandwidths. Access time is quite large for touch-based devices, whereas gesture devices are usually persistent, making their access times low to non-existent. Multitasking is largely related to cognitive and physical requirements. Touch-based devices tend to require more cognitive and physical resources, making them weaker in this respect. Voice is good for multitasking in cross-modal situations (walking and talking), but may not be good in common-mode situations (talking to a friend while trying to dictate). Again, we conservatively group the categories in Figure 2.

For both partitioning criteria, we have included Hambone in the diagrams. Hambone under Buxton's three mirrors is social (subtle), cognitive (does not divert much attention), and physical (uses easy-to-do movements, does not have buttons that are tiny and difficult to press). Based on Lee's criteria, Hambone is best suited for applications requiring low bandwidth, low access time, and multitasking capabilities.

Note that accelerometer-based systems share Hambone's partitioning. While accelerometers are well-suited to gross motion gestures, Hambone is designed to work with fine, subtle gestures via mechanical vibrations.

3. Apparatus

Hambone's hardware includes two piezoelectric sensors, a custom printed circuit board (PCB) for analog signal conditioning, a PCB containing a microcontroller with Bluetooth transceiver, and a lithium-ion mobile phone battery. The current Hambone prototype fits in an iPod Nano carrying case, and the two sensors are coupled to the skin using surgical tape and an elastic band. We envision, however, that future iterations of Hambone will rely on hardware embedded in a wristwatch or piece of jewelry.

To "listen" to the sound of gestures being made, we used two piezoelectric sensors (SDSystems PU-2). These sensors are extremely sensitive to mechanical vibrations while simultaneously disregarding external audio signals. This characteristic allows the sensors to measure vibrational bone-conducted signals while being immune from external noise.

Due to high output impedance, piezoelectric sensors require amplification prior to digitization. To amplify the signals, we constructed a custom analog signal conditioning PCB with a Texas Instruments INA155 instrumentation amplifier for each sensor. Figures 3 and 4 show the schematic and PCB layout.

To transfer data from the Hambone hardware to the device being controlled, we use Bluesense[1], a microcontroller with a Bluetooth transceiver. Bluesense retrieves 10-bit samples at 100 Hz from five analog channels. Three channels interface to accelerometers mounted orthogonally in an X–Y–Z configuration and two channels contain the amplified piezoelectric sensor signals from the other PCB. We do not use the accelerometers for gesture detection, but they could be used in future versions of Hambone to detect gross body movements in addition to the subtle gestures detected via bone conduction.

The power consumption of the entire device is 246 milli-Watts (63 mA at 3.9V). Most of the power consumption is due to the Bluetooth transceiver and unused accelerometers, as only 14 milli-Watts (3.5 mA at 3.9V) is due to the sensors and amplifiers. Almost all of the sensor and amplifier power is attributed to the quiescent current draw of the amplifiers, which is greater than 3.4 mA. Since the sensors are piezoelectric, they are essentially self-powering. This is an advantage since they can be incorporated into future devices with minimal power budget.

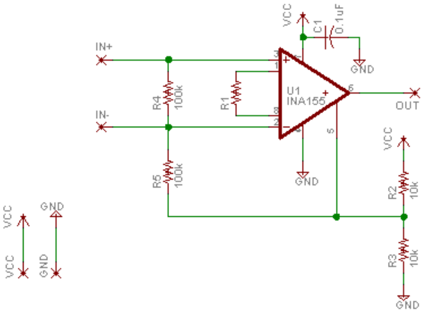

Figure 3. Hardware Schematic

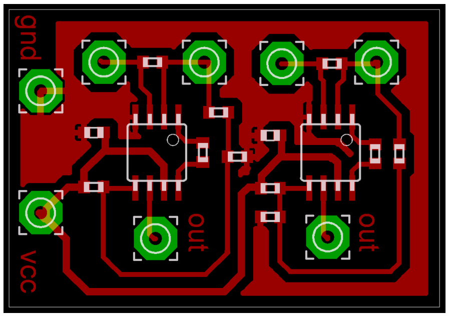

Figure 4. PCB Layout

4. Data Collection & Model Training

We developed an application in Python to support gesture data collection and hidden Markov model training. For each gesture, the application prompts the user to perform it. The system waits until energy in the input signal rises above an arbitrary threshold and then records the sensor signals for a two second interval. Once recording completes, the system asks the user whether it should retain the particular sample. After the application collects twenty samples for a particular gesture, it repeats the same data collection process with the next gesture. After collecting data for all of the gestures, the application generates a hidden Markov gesture model via the Georgia Tech Gesture Toolkit (GT^{2}k) [13].

We performed training on a PC, but the application can be ported to a number of mobile devices. Note that if the mobile device cannot, for some reason, support the training application, the user can complete gesture training using a PC and then transfer the generated model to the mobile device.

5. Real-Time Recognition

The continuous live recognition software component is intended for installation on the device being controlled, though for prototyping we used Bluetooth to transmit sensor data to a PC for recognition. The software is imple-

[1] Additional information about Bluesense is available at http://www.gvu.gatech.edu/ccg/resources/btacc

mented in Python and leverages the recognition capabilities of HTK[2]. During the live recognition, the software continuously gathers sensor data from the hardware. When energy in the input signal rises above a threshold, the live recognition software records the sensor signals for a two second interval. The threshold filters out low-level noise and triggers gesture recognition.

Once the two second sample is recorded, it is sent to HTK for recognition. Using the gesture model produced by the training application and the new gesture sample, HTK performs Viterbi decoding and returns the best-fitting gesture according to the trained models. The recognition software then issues a command corresponding to the gesture to the mobile device.

Note that the two second recording interval results in an approximate two second lag between gesture initiation and command issuance. This recording lag is inherent in the HTK recognition suite, as the entire gesture must be received before classification. For this implementation, two seconds was chosen since all of our gestures fit within this time interval. While the recording duration is variable, it is ultimately dependent on the gesture durations. As is shown in Section 10, the two second lag is acceptable for certain applications.

6. Sensor Placement

During body movement, sounds are generated inside the body. When using Hambone, the sounds travel via bone conduction from the point of interest (for example, the fingertips) to the location of the piezoelectric sensors. The vibration signals are coupled from the bone, through the skin, and to the piezoelectric material. Since bone is stiff, the audio signals travel with less attenuation through bone than soft tissues. For this reason, the piezoelectric sensors should be placed in locations that minimize soft tissues between the sensors and the bones. In addition, there should be little skin movement at the sensor mounting location (e.g. the jaw has skin close to bone but the skin doesn't stay with the bone when it moves).

We experimented with a variety of sensor placements, including on the shoulder, jaw, elbow, knee, and forearm. Two locations seem particularly well suited for Hambone operation. The first location is on the ulnar and radial styloids (protruding wrist ”bumps”), which is the same location that Amento used [1]. The second location is on the lateral and medial malleoli (protruding ankle ”bumps”). Both of these locations have bones close to the skin, and afford gestures useful for device control. Note that unlike the bone-conduction interface created by Amento [1], we have two sensors (e.g. there is one sensor for each protruding “bump”) for better monitoring.

When deciding on a sensor placement, the number of gestures required for a particular device/application should also be considered. For example, if Hambone is placed on the elbow, its gesture space is limited to extending or contracting the forearm. If Hambone is placed on the wrist, however, the number of available gestures expands dramatically, because this placement allows for detection of subtle finger gestures.

7. Gesture Selection

We defined two sets of gestures for our sample applications and preliminary evaluation. To select the gestures, we observed and compared analog waveforms from a number of body movements on an oscilloscope. Although the potential gesture space is large, we found through experimentation that some gestures look too similar to be distinguished accurately. Through this process of testing, we selected a set of three wrist gestures and four foot gestures, shown in Figure 5. Rather than choosing the most comfortable or subtle gestures, we aimed to have gestures that involved a variety of movements including moving fingers in the air and rubbing or tapping fingers together. The foot gestures included movements such as toe flicks to the left and right, as well as toe and heel raises.

In the course of our system development and evaluation, we found that user gesture preferences are highly subjective. An uncomfortable or physically difficult gesture for one person may be a favorite for another person. For example, one participant in the evaluation (described in sections 9 and 10) had previously broken his left ankle and experienced physical discomfort when performing the “rotate ankle right” gesture.

Additionally, we (the authors) found variations in our own abilities to complete particular gestures, as well as variations in which gestures we preferred. For example, some people may have difficulty tapping fingers on a table top one-by-one from pinky to index finger and vice versa. This gesture, however, is preferred by one of the authors. Given differences in the preferences and physical abilities of individuals, it may be best to have users select their own gestures. A challenge of using this approach, though, is that users may choose a set of gestures that are either difficult to distinguish from other control gestures or are too similar to “common” movements.

8. Avoiding False Positives

During real-time operation, Hambone classifies a gesture every time the signal energy exceeds a given threshold, thus triggering recognition.

[2] Due to speed concerns, we avoid using GT^{2}k for gesture recognition and make direct, low-level calls to HTK, which is the hidden Markov model toolkit underlying GT^{2}k.

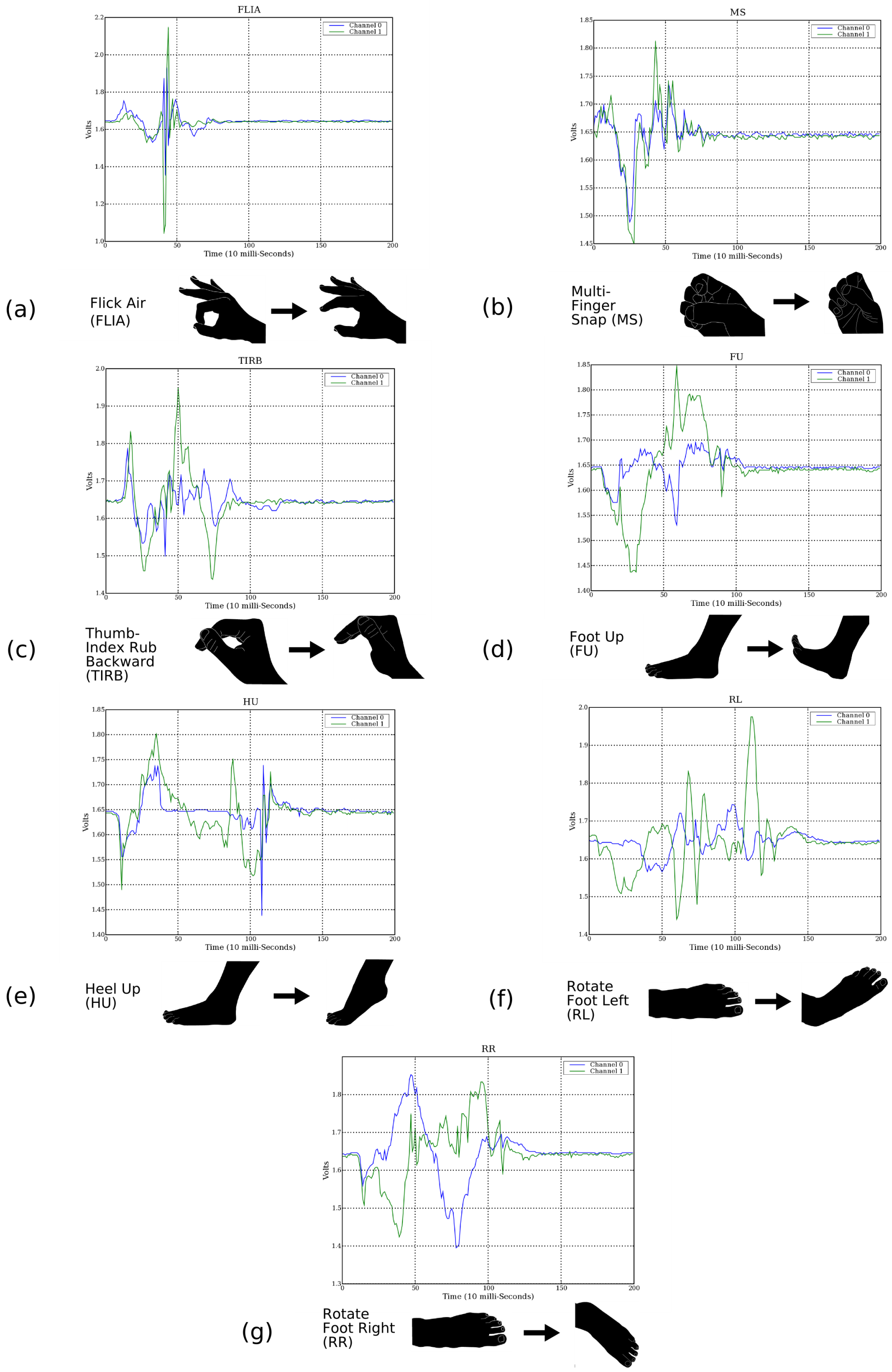

Figure 5. Gesture descriptions, start and end positions, and sample waveforms

Table 1. Averaged Wrist Confusion Matrix With One Sensor

	MS	FLIA	TIRB	% correct
MS	14	1	3	78%
FLIA	4	13	1	72%
TIRB	1	0	17	94%

Overall correctness: 81%

This means extraneous "noise" signals from non-gesture movements will be incorrectly interpreted by the HMM as a known gesture, thus producing a false positive.

We first attempted to prevent unintentional triggering by using the built-in accelerometers of the Bluesense board. With the accelerometer-based method, the HMM would only be triggered when the energy exceeded a given threshold *and* the accelerometers were in a pre-defined position. However, this technique restricted the range of motion during gesture performance.

We found that instead of trying to prevent unintentional triggering, a better approach was to add a dummy "noise" gesture for the HMM. To provide noise data, users trained the noise gesture by performing a set of random movements. During real-time operation, non-gesture movements were classified as "noise," thus greatly reducing the number of false positives while allowing full range of motion.

9. Evaluation

We performed a *preliminary* evaluation of the wrist and ankle based interface with three adults. This evaluation examined system accuracy and user independence. The participants wore Hambone with two sensors, first on the right wrist (with no wristwatches or adornments) and then on the left ankle (with socks and shoes covering the sensors). Each participant provided twenty samples for each of three wrist and four ankle gestures (shown in Figure 5), as well as sixty "noise" gestures for real-time operation. These samples were used for off-line training and evaluation. Two users went on to perform a qualitative evaluation, controlling two realistic applications.

10. Results

For quantitative results, we performed user-dependent cross-validation for all three users. Tables 1 and 2 show confusion matrices for the wrist and ankle data when only *one* sensor's data was utilized, as in Amento et al [1]. Tables 3 and 4 show confusion matrices, again for the wrist and ankle data, but this time with *two* sensors enabled. The two-sensor case shows marked improvement. For the averaged confusion matrices, separately for each participant, two-thirds of the data trains the HMM. The remaining one-third from each participant cross-validates the model. The cross-validation data from each separate user was then averaged into the confusion matrices shown. In the confusion matrix, each row represents a gesture. The corresponding column represents how the previously withheld data (the one-third not used for training) was classified by the HMM. The matrix also shows accuracy for each gesture.

Table 2. Averaged Ankle Confusion Matrix With One Sensor

	HU	FU	RL	RR	% correct
HU	15	3	0	0	83%
FU	1	17	0	0	94%
RL	3	1	14	0	78%
RR	0	3	2	13	72%

Overall correctness: 82%

Table 3. Averaged Wrist Confusion Matrix With Two Sensors

	MS	FLIA	TIRB	% correct
MS	18	0	0	100%
FLIA	0	18	0	100%
TIRB	0	0	18	100%

Overall correctness: 100%

The confusion matrices in Tables 5 and 6 show the cross-validation results of training across users for both the wrist and ankle. In this case, two-thirds of all the participant data trains the HMM model. The remaining one-third of the data cross-validates the model.

For an initial qualitative analysis, one participant used Hambone with two sensors on the right wrist to control a PowerPoint presentation in a realistic setting (that is, standing, speaking, using expressive gestures, and moving). The gestures were mapped as follows: flick to blank the screen, multi-finger snap to advance to the next slide, and thumb index rub backward to go to the previous slide. In this particular instance, accuracy was approximately 90%. Of the inaccurate classifications, some were classified as noise, flick was confused with other abrupt movements, and there was also some minimal confusion among the valid gestures.

Another participant used Hambone with two sensors on the left ankle to control a puzzle video game called Professor Fizzwizzle[3]. In this game, players navigate a character either up, down, left, or right. We selected four foot gestures with the following mappings: foot up to move up, heel up to move down, rotate right to move right, and rotate left to move left. After completing the training, the user controlled the video game with foot gestures while sitting in a chair.

[3] Professor Fizzwizzle is a video game developed by Grubby Games.

Table 4. Averaged Ankle Confusion Matrix With Two Sensors

	HU	FU	RL	RR	% correct
HU	18	0	0	0	100%
FU	0	18	0	0	100%
RL	0	0	16	2	89%
RR	0	0	1	17	94%

Overall correctness: 96%

Table 5. Wrist Training Across Users

	MS	FLIA	TIRB	% correct
MS	7	10	1	39%
FLIA	0	18	0	100%
TIRB	9	0	9	50%

Overall correctness: 63%

During game play, this person intentionally performed several random foot and leg movements. For this application, approximately 5% of the gestures were misclassified. Most of the errors were a result of classifying a valid gesture as noise, while valid gestures were rarely misclassified.

11. Discussion

The recognition accuracy showed substantial improvements when two sensors were used instead of just one; the improvement was on the order of 15–to–20%. With two sensors, both the initial qualitative analysis and the user-dependent analysis showed promising results. In the qualitative analysis, accuracy ranged from 90-95%. In the user-dependent analysis, the accuracy was about 95% for both the ankle and the wrist. Training across users, however, produced mixed results. While FLIA and RR gestures (see Figure 5) were easily recognized, the others had difficulty. These results suggest that Hambone is a user-dependent system.

12. Possible Applications

Hambone seems best suited for applications requiring low bandwidth, low access time, and multitasking capabilities. Hambone is appropriate for applications requiring 2–4 way navigation (e.g. up, down, forward, back), such as MP3 players, DVD players, or games in which an avatar is moved across a screen.

It could also be useful for applications that require either a subtle gesture or multitasking capabilities. For example, a person could use Hambone to advance PowerPoint slides. By using Hambone, this person would be able to advance through slides subtly—without interrupting the flow of the presentation.

Table 6. Ankle Training Across Users

	HU	FU	RL	RR	% correct
HU	14	2	0	2	78%
FU	6	11	0	1	61%
RL	6	5	7	0	39%
RR	0	0	0	18	100%

Overall correctness: 69%

Hambone could also be used for interfacing to devices with impoverished user interfaces. For example, some mobile devices have small buttons. The access time to get to these buttons is slow, and people need fine motor control to be able to push them. Another example is that tiny (or even implantable) devices may not have a direct user interface of any variety and would need alternatives to traditional user interfaces to receive input.

Finally, because Hambone appears to be user dependent, authentication applications may be possible.

13. Future Work

When a user performs a gesture with the current implementation of Hambone, there is approximately two seconds of lag from the initiation of the gesture until the final command is issued. HTK requires a full gesture waveform (not a gesture in progress) to be submitted for recognition, so the command cannot be issued until the two second gesture recording window closes. In order to reduce the lag time, we plan to revise the software to allow variable length recording windows based on energy thresholds. By reducing lag time, Hambone can be used for a number of new applications such as an “any-surface” chording keyboard or electronic musical instrument. We plan to prototype a number of applications based on this expanded functionality. Additionally, we are interested in creating an extension to Hambone that would allow for two-handed input.

Further, since Hambone relies on vibrations, it may be useful for “hands-full” situations, such as holding a box or other large item with both hands. Finally, since Hambone’s sensors can be placed on a variety of areas of the body, it may have applications for people who are disabled as an alternative interface for interacting with off-the-shelf mobile devices.

Techniques for selecting appropriate gesture sets require additional study. For example, to find a set of gestures that do not conflict with common movements, a user may wear Hambone in a data gather mode for a period of time to obtain a set of baseline data. The data could then be used to help users choose a gesture set that does not conflict with common movements.

When training the system to recognize more than 3–4 gestures, the HMM training sessions become extremely tedious. One opportunity for future work is examining ways to make gesture training systems more tolerable (e.g. through games).

We anecdotally found that small variations in sensor placement caused large variations in signal characteristics. Thus, repeatable sensor placement is critical for effective inter-session training and recognition. This challenge may be addressed through careful interface design. For example, if Hambone were implemented as a watch, it could be designed with forcing functions ensuring that it only fits in an appropriate position. If these constraints were not possible to design into the watch, users could instead be advised to always wear the watch so that some physical marker (e.g. the watch face) is always in the same position. This challenge will be crucial for our future work on Hambone.

There are also opportunities to study issues such as physical comfort, both for the wearable and for the gestures due to physical differences in individuals. In our work, we did not control for variations in bone structure, body fat distribution, dexterity, or any other physically differentiating characteristic. Future research into bone conduction-based interfaces should investigate whether these variations are significant.

14. Conclusions

In this paper, we have demonstrated that "listening" to the sound of body movements is a viable approach to mobile device interaction. Our system based on this technique, Hambone, provides the capability to quickly access and interact with mobile devices. Hambone allows for real-time gesture recognition, allows for a diverse set of gestures, and includes a noise gesture to reduce false positives. Our preliminary evaluation suggests that the system is most likely user dependent.

15. Acknowledgments

We thank Jay Summet for his feedback and Peter Presti for providing a Bluesense board. This work was conducted while one of the authors was supported by a US Department of Homeland Security Fellowship and another of the authors was supported by a Naumann-Etienne Foundation Fellowship. The opinions expressed in this paper belong solely to the authors and do not necessarily represent the views or policies of the funding agencies supporting this work.

References

[1] B. Amento, W. Hill, and L. Terveen. The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 724–725, New York, NY, USA, 2002. ACM Press.

[2] W. Buxton. Human skills in interface design. *Interacting with Virtual Environments*, pages 1–12, 1994.

[3] E. Clarkson, J. Clawson, K. Lyons, and T. Starner. An empirical study of typing rates on mini-qwerty keyboards. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1288–1291, New York, NY, USA, 2005. ACM Press.

[4] S. Lee. Private correspondence. *Georgia Institute of Technology*, 2007.

[5] J. Lementec and P. Bajcsy. Recognition of arm gestures using multiple orientation sensors: gesture classification. *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pages 965–970, 2004.

[6] K. Lyons, C. Skeels, T. Starner, C. M. Snoeck, B. A. Wong, and D. Ashbrook. Augmenting conversations using dual-purpose speech. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 237–246, New York, NY, USA, 2004. ACM Press.

[7] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 671–678, New York, NY, USA, 2004. ACM Press.

[8] B. Shneiderman. The limits of speech recognition. *Commun. ACM*, 43(9):63–65, 2000.

[9] T. Starner, A. Pentland, and J. Weaver. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(12):1371–1375, 1998.

[10] T. E. Starner, C. M. Snoeck, B. A. Wong, and R. M. McGuire. Use of mobile appointment scheduling devices. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1501–1504, New York, NY, USA, 2004. ACM Press.

[11] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Comput. Graph. Appl.*, 14(1):30–39, 1994.

[12] A. Vardy, J. Robinson, and L.-T. Cheng. The wristcam as input device. In *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 199, Washington, DC, USA, 1999. IEEE Computer Society.

[13] T. Westeyn, H. Brashear, A. Atrash, and T. Starner. Georgia tech gesture toolkit: supporting experiments in gesture recognition. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 85–92, New York, NY, USA, 2003. ACM Press.

[14] J. O. Wobbrock, B. A. Myers, and J. A. Kembel. Edgewrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 61–70, New York, NY, USA, 2003. ACM Press.