The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface

Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner
College of Computing and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
{jszzang, jiazhe, kent, thad}@cc.gatech.edu

Abstract

We introduce the Gesture Watch, a mobile wireless device worn on a user's wrist that allows hand gesture control of other devices. The Gesture Watch utilizes an array of infrared proximity sensors to sense hand gestures made over the device and interprets the gestures using hidden Markov models. The Gesture Watch maps intuitive gross hand gestures to control signals such as the play and pause commands commonly found on mobile media players. We present our evaluation of the Gesture Watch designed to determine its the accuracy and usability. In our study, 10 participants used the Gesture Watch in both mobile and stationary conditions as well as indoors and outdoors. Overall, we attained a recognition accuracy of 95.5% and found that the Gesture Watch worked well in both indoor and outdoor environments and while mobile.

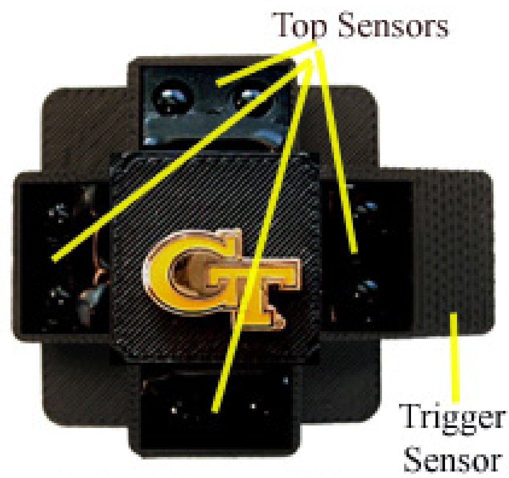

Figure 1. The Gesture Watch prototype with four sensors on the top of the device and a fifth placed horizontally to act the trigger for gesture recognition.

1 Introduction

With advances in electronics, mobile devices such as cell phones and MP3 players are growing smaller and more lightweight while also increasing in computational power. Unfortunately, this drastic reduction in form factor has a limit because the user interface components of these mobile devices cannot similarly be reduced in size without becoming unusable. The device needs to be large enough so that a user can press physical buttons or move a finger around a touch sensitive surface. For example, today's MP3 players, such as the newest generation of Apple's iPod Shuffle, are quickly nearing the point where the physical interface of the device determines its size.

One potential solution to this problem is to decouple the area in which the user interacts with the system from the physical device. In particular, in this work we explore using the space above the device as an interaction area. The Gesture Watch is a prototype system that employs five infrared proximity sensors to detect gestures made in the volume of space above it (Figure 1). The proximity sensors could be made very small allowing for the continued decrease in device form factor, while still maintaining a large area for user interaction.

We designed the Gesture Watch to be worn on the wrist similar to a normal wristwatch. We are interested the watch form-factor because the wrist location has several interesting properties. First, given the history of wrist watches, it is a common and socially accepted place to wear technology. The wrist also has the potential to enable interfaces that are glanceable and enable fast access since the device takes minimal effort to retrieve (unlike other mobile technology such as mobile phones). However, the wrist watch also presents usability challenges. In particular, any input system must contend with the limited size of the device and thus provides a good opportunity to explore the use of the space above the watch for interaction.

The Gesture Watch is composed of several components. First, four proximity sensors are arranged on the top of the device and are used to sense hand gestures. The sensor data is then sent over a wireless connection to another mobile device such as a mobile phone or wearable computer for processing and control. The computer interprets the data as a gesture and triggers the appropriate command. For example, the Gesture Watch may be wirelessly connected to a user's smartphone and the gestures could be used to control the phone's digital music player.

By using multiple proximity sensors, the Gesture Watch can detect the direction from which the user's hand approaches and leaves the sensed area. This feature enabled us to explore several different types of gestures while developing the Gesture Watch. Figures 2 through 10 show nine example gestures that we developed. The more simple gestures include a single hand movement in an up, down, left, or right direction (Figures 2 and 5). In contrast, more complicated gestures may involve multiple simple gestures (Figures 3 and 6), circular motions (Figures 8 and 9), or changes in direction (Figures 7 and 10).

As shown in these figures, the user makes gestures using the arm not wearing the Gesture Watch directly above the sensors. To prevent the system from interpreting data from normal activities when the user is not performing a gesture, a fifth infrared proximity sensor is used as a trigger (Figure 1). Unlike the other four sensors, this proximity sensor faces in the direction of the hand and detects whether the hand is raised or lowered. When the wrist of the arm wearing the Gesture Watch is raised by a significant amount, the fifth trigger sensor is activated. This process informs the system to start collecting data, and the system collects data while the user's hand is raised. The user lowers the wrist when the gesture is complete, and the system interprets the collected data as a gesture.

2 Related Work

The predecessor and inspiration for the Gesture Watch is the Gesture Pendant created by Starner *et al.* [6]. The Gesture Pendant is a device worn as a necklace and consisted of a ring of infrared light-emitting diodes (LEDs) and a central black and white camera. The Gesture Pendant detected various hand motions made in front of it by sensing the infrared light reflected off of the user's hand. These hand motions were interpreted as gestures and were in turn used to control the operation of various household devices such as opening and closing automatic window blinds, or controlling audio/video equipment. While this system allowed for numerous types of gestures to be recognized, one major drawback of the system was the camera's sensitivity to ambient infrared light. In particular, the Gesture Pendant did not work in sunlight and thus had limited applications. A similar camera based gesture system is Hamette and Tröster's FingerMouse. Using stereo cameras this system can track the user's finger as it moves in front of the device.

Metzger *et al.* [4] overcame the sensor limitations of the Gesture Pendant in the FreeDigiter system. This work utilizes a single infrared proximity sensor and could detect and interpret very simple hand gestures gestures that passed in front of it. A prototype mobile phone headset was developed that was controlled by FreeDigiter. The system sensed the occlusions of the proximity sensor as the user's hand passed. By counting this signal, the system could perform simple functions such as dialing the mobile phone.

Wristwatch based platforms have also been explored in the literature. IBM's Linux wristwatch is a fully function Linux computer in the form factor of a wristwatch [5]. In an exploration of interaction techniques for such a device, Blaskó and Feiner have created different touch-based input systems. In particular, they have investigated the use of the bezel/frame of a wristwatch along with tactile landmarks and bidirectional segmented strokes for input [1]. They have also explored a touch screen based system to allow for numeric data entry using haptic feedback [2].

3 Gesture Watch Implementation

The Gesture Watch uses an array of four SHARP GP2Y0D340K proximity sensors that are arranged facing up in a cross shape to allow the watch to detect a variety of gestures. The fifth proximity sensor faces toward a hand and parallel to the arm and is used to segment the data.

The prototype is separated into two parts. The top part shown in Figure 1 measures 58x33x40mm and houses the sensors which each measure 15x9.6x8.85mm. The bottom part of the Gesture Watch (not shown) is 58x29x40mm and worn on the bottom of the wrist. This component includes a Bluetooth module and a battery and the two parts of the Gesture Watch are connected through a watch strap.

All five of these sensors are rated to detect objects in the range of 10-60 cm, however in practice we have found that they are most effective between 5 and 20 cm. One key feature of the sensors is that they can be used in a variety of environments. In particular they are robust to extreme lighting conditions and work in direct sunlight as well as complete darkness.

Each proximity sensor outputs a digital low when it detects an object and otherwise stays high thus allowing the detection of the proximity of and object to the sensor. Together, our four proximity sensors provide a set of time series data consisting of a sequence of four binary values that indicate which of the four proximity sensors are obscured during a gesture.

The fifth proximity sensor is used to segment the data from the other four sensors. When the user raises her wrist, the other sensors' data are collected by the system for recognition. Once the wrist is lowered, recognition is triggered. The use of the fifth sensor in this way helps prevent the in-

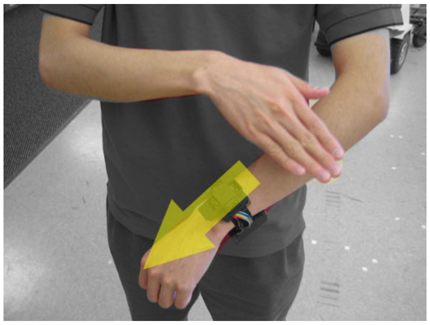

Figure 2. One time forward: move hand from elbow towards fingertips once.

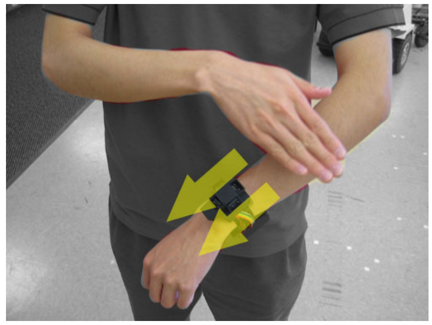

Figure 3. Two times forward: move hand from elbow towards fingertips twice.

Figure 4. Cover all sensors: hold hand over sensors.

Figure 5. One time back: move hand from fingertips to elbow once.

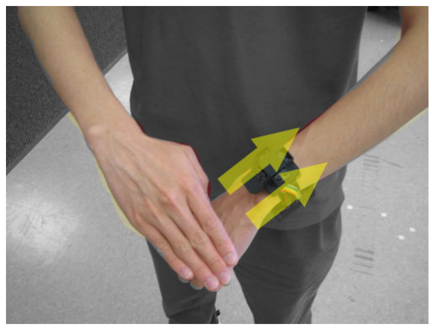

Figure 6. Two times back: move hand from fingertips to elbow twice.

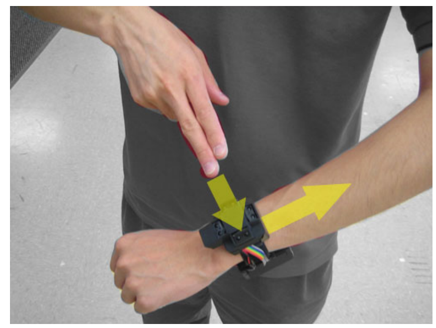

Figure 7. Out–left: move hand from body to sensor and then towards elbow.

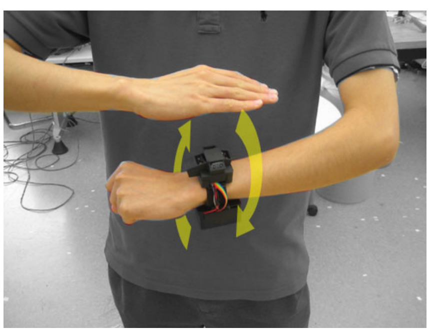

Figure 8. Around: move hand from body, over top and then around the wrist.

Figure 9. Clockwise: move hand clockwise over top of sensors.

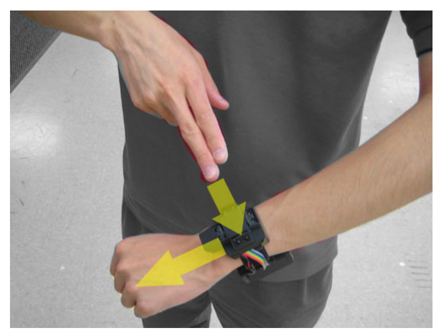

Figure 10. Out–right: move hand from body to sensor and then towards fingertips.

advertent recognition of the data as well as facilitates the gesture recognition by providing segmented data. We have also added timing information to the system to help detect false triggers. In particular, if the fifth sensor's activation period is shorter than half a second or longer than three seconds, the software assumes that system was triggered accidentally and the gesture data is ignored.

Figure 11 shows the data traces from the four proximity sensors for five different gestures. Figure 11a corresponds to the cover all sensors (CAS) gesture shown in Figure 4. Figure 11b shows the trace for one time forward (OTF) and Figure 11c is one time back (OTB). Figures 11d and 11e are for two times forward (TTF) and two times back (TTB) respectively.

The data from the proximity sensors is read by a PIC16LF873 microcontroller which converts the data into

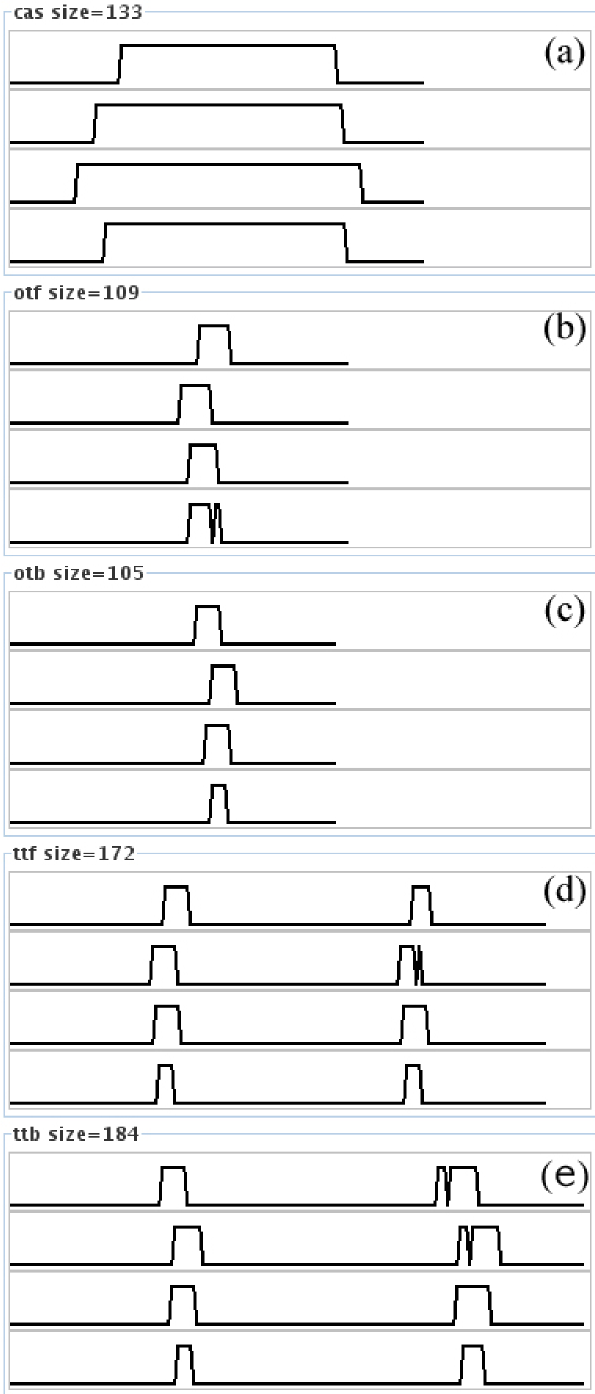

Figure 11. Representation of the signals for five gestures. (a) Cover all sensors. (b) One time forward. (c) One time backward. (d) Two times forward. (e) Two times backward.

packets and sends it over a Taiyo Yuden EYMF2CAMM Bluetooth radio module (Figure 12). A remote computer connects to the sensors wirelessly and processes the data to recognize gestures.

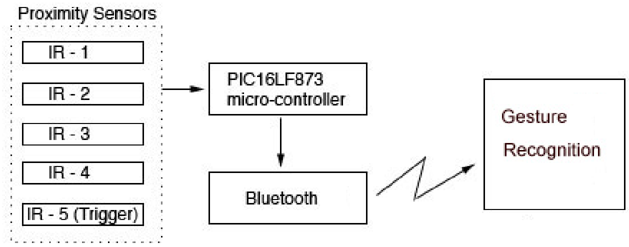

Figure 12. The Gesture Watch is composed of a PIC microcontroller which reads data from 5 IR proximity sensors. The PIC sends the sensor data wirelessly using a Bluetooth module to a remote computer which interprets the data as gestures.

In terms of power, both the PIC and the Bluetooth module utilize 3.6V, while the proximity sensors require approximately 4.5V. Due to the need for an easily accessible power supply, our prototypes uses a simple 9V battery and regulated the voltage with an LM340TS(7805). With each proximity sensor drawing a typical current of 25mA, the entire prototype draws a total current of 180mA, resulting in a power consumption of 1.6W while transmitting.

Once the data from the four proximity sensors has been received by the computer, it must be processed and recognized as a gesture. For recognition we are using the Gesture and Activity Recognition Toolkit (GART) [3]. GART is a toolkit built to facilitate the development and exploration of gesture recognition applications. GART utilizes the machine learning algorithms from Cambridge University's Hidden Markov Model (HMM) Toolkit for training and recognition [10]. We chose to utilize HMMs given past success in previous work using them to model complex time series data [7] [8] [9]. For our model, we are using an eight state left-to-right topology.

4 Evaluation

We conducted an evaluation to determine the effectiveness of the Gesture Watch. In particular, we wanted to examine if participants could successfully employ our proximity sensor technique which utilizes the space above the mobile device instead of interacting on its surface. Next, we were interested in the potential impact a user's mobility may have on the gesture system. Given our motivation of enabling very quick interactions, we did not want to require the user to stop to input a command. For example, if this device were to control a digital music player, the input technique would need to work while a user is jogging and listening to music. From a technical perspective, we wanted to evaluate how effective the infrared sensors worked under various lighting conditions for gesture recognition. More specifically, we wanted to determine if the sensors could still effectively be used while outdoors in sunlight. Finally, we wanted to examine the gesture recognition accuracy of the Gesture Watch.

Given these requirements, we designed our experiment as a 2x2 within subject design. Our first factor is mobility (standing or walking) and the second factor is location (indoors or outdoors). The standing condition gives us a baseline assessment of how well the Gesture Watch recognizes a user's gestures. The walking condition allows us to look for differences in performance induced by the motion of the user and see the effects of having the participant engaged in a task with larger attentional demands. The location factor provides data about the effectiveness of the Gesture Watch in artificial lighting conditions (an office) as well as in direct and indirect sunlight.

For this study we evaluated the performance of five of the gestures we created for use with the Gesture Watch. These five gesture are "cover all sensors," "one time forward," "one time backward," "two times forward," and "two times backward" and are depicted in Figures 2 through 6 and Figure 11. These gestures were chosen because we felt that they were the most memorable and easiest to perform, yet still provided enough functionally to control a simple mobile device such as an MP3 player.

Participants performed each of the five gestures four times resulting in a total of twenty gestures per user per condition. The order of conditions and the order of gestures within a condition were both determined randomly.

4.1 Participants

We recruited 12 undergraduates and graduate students from the Georgia Tech student body. Of those, we successfully obtained full data sets for ten participants (one did not fully understand the directions and another experienced a software failure and therefore we exclude their data from analysis). Of those ten participants, one was female. One participant was left handed but all wore the Gesture Watch on their left hand.

4.2 Recognition Model

To train the system we gathered 30 examples of each of the five gestures and labeled them accordingly for a training set. This data was collected indoors and while stationary by the first two authors with each contributing a total of 15 samples per gesture. Using GART, the HMM was trained using two-third cross validation and we obtained a recognition accuracy of 97.78%. These trained models were used for all recognition by the system for the experiment.

4.3 Procedure

The experiment began with a brief training period for each participant. The researchers described and demonstrated each of the five gestures. Next, participants attached the Gesture Watch to their wrist and were allowed to practice each gesture as much as they desired. To give the participants feedback during this training period, we showed them our testing interface. This program visualized which of the proximity sensors were covered and provided the results of gesture recognition. We proceeded to the next step only after the user said that they were comfortable with performing each gesture.

Following initial training, we tested the participants to ensure that they had fully learned each gesture. The testing was accomplished by having the participants progress through a list of ten randomly selected gestures. After each gesture, the recognition results were verified to ensure that the participant performed a gesture that was correctly identified by the system. If the participant did not perform a gesture correctly, they were asked to repeat it until they did.

Once the users had successfully shown that they could perform all of the gestures, we provided them with a set of headphones connected to a Linux computer. This computer ran a custom Java application that prompted the participants to perform specific gestures by playing short audio clips.

At this point the trials for each of the four conditions began. During the trials, the participants were followed by two researchers approximately three feet behind them. One researcher held the laptop computer running the recognition system and the other recorded observations.

For the indoor location, we selected an area inside the lobby of our research building. When asked to walk, participants progressed along a moderately–trafficked path around our laboratory. For the outdoor conditions participants stood in a sunny area outside our building. When asked to walk, the participants again followed a moderately-trafficked path, only this time through the courtyard in front of our building. When selecting the path, we ensured that participants would be exposed to direct sunlight as they walked. For both of the walking conditions, we offered the users a general path to follow, but we did not specify any boundaries that they had to stay within. Thus, they were allowed the freedom of normal movement.

At the end of the four conditions we asked the participants for qualitative feedback about the Gesture Watch. We asked questions about how they felt about the physical form factor of the watch, which gestures they were most and least comfortable with and how socially acceptable they felt the Gesture Watch and the gestures were.

4.4 Equipment and Measures

For this experiment, the Bluetooth Gesture Watch was paired with an laptop computer running Linux carried by a researcher. The computer ran a custom Java application that prompted the participants with which gesture to perform. The prompts consisted of audio clips that were played over headphones worn by the participant. The application also recognized the sensed gesture data using the GART toolkit and the model trained on the data provided by the first two authors. For this experiment, we decided to provide very little feedback from the system. The software would play a short beep whenever a gesture was recognized, however we did not inform the user if this gesture was correct or not. We took this approach to minimize any learning and adaptation effects that might occur over the duration of the experiment.

The application automatically logged several measures. First it recorded both the classified results and actual command that was given to the participants. The application also logged all of the gesture trigger events generated by the fifth proximity sensor, regardless of whether or not the participant was instructed to perform a gesture. Finally, we recorded the reaction and gesture times. The reaction time is the period from when the audio prompt finished playing to the time when the user started a gesture with the trigger. The gesture time is the time taken to make a complete gesture.

5 Results

For this experiment our 10 participants provided a total of 800 recognized gestures. For our first result, we examined the accuracy of the gesture recognition system. The overall recognition accuracy across all four conditions is 95.5% (Table 1). Using an ANOVA we found a statistically significant main effect for location on the accuracy results ($p<0.05$). The accuracy of the indoor condition is 97.8% (SD=4.13%) while the accuracy dropped to 93.3% (SD=7.99%) while outdoors. There was no significant main effect for mobility ($p=0.13$) nor an interaction ($p=0.21$) with this sample size.

	Stand	Walk	Mean
Indoors	98.0 (3.50)	97.5 (4.86)	97.8 (4.13)
Outdoors	96.0 (6.15)	90.5 (8.96)	93.3 (7.99)
Mean	97.0 (4.97)	94.0 (7.88)	95.5 (6.78)

Table 1. Recognition accuracy and (standard deviation) percentages for the four conditions.

Table 2 shows the confusion matrixes for each gesture across the four conditions. For all ten users, we obtained 40 samples for each gesture in each condition. The rows show the gesture the user was instructed to perform and columns show the classified results. The not available (n/a) column indicates the number of gestures where the system was not triggered. This occurred when the user was prompted with a gesture, but failed to do so, or when the trigger sensor was activated for less than half a second or longer than three seconds.

As can be seen in Table 2, the largest number of the recognition errors occurred while participants were walking outdoors. In particular, one time forward was recognized as two times forward 6 of the 40 times. Some of these errors may result from the difficulty of some users' ability to perform the gestures. Our qualitative data indicates that when asked about the gestures, most of the participants felt that all five of the gestures were comfortable to perform. However two participants specifically mentioned the "two times forward" and "two times backward" gestures as awkward. Another participant mentioned that the backwards gestures were harder to perform than the forwards ones. Finally, two participants noted that the "cover all sensors" gesture required knowing exactly where the watch arm was in order to be able to determine if the hand was actually covering all the sensors.

Next, we examine the number of false system triggers made by the participants (Table 3). Here we found a marginal main effect for mobility ($p=0.063$). Overall, users were more likely to falsely trigger a recognition while walking (M=2.95, SD=6.14) than while standing (M=0.30, SD=0.80).

This data is further substantiated by the qualitative feedback we obtained. From our own experiences with the Gesture Watch, we suspected that the triggering based on the fifth sensor and wrist movement might prove to be problematic. Indeed, six participants specifically mentioned that they did not like the trigger mechanism because it was either uncomfortable or because it triggered at unwanted times while they were moving.

	Stand	Walk	Mean
Indoors	0.20 (0.63)	1.50 (1.58)	0.85 (1.35)
Outdoors	0.40 (0.97)	4.40 (8.51)	2.40 (6.24)
Mean	0.30 (0.80)	2.95 (6.14)	1.63 (4.53)

Table 3. The number of falsely triggered gestures for the four conditions.

Finally, we examine the timing for reacting to and performing the gestures. As described above, the reaction time is the time from when the system finished prompting the user to when the gesture was started (Table 4). Here we found a significant main effect for mobility ($p<0.05$) with the reaction time being faster for standing (M=0.61s, SD=0.16s) than for walking (M=1.16s, SD=0.88s). There was no main effect for location, nor an interaction. Examining the time taken to perform the gestures reveals no statistically significant effects with gestures taking on average 1.73s (SD=0.38s) to perform.

	Stand	Walk	Mean
Indoors	0.63 (0.16)	1.23 (1.08)	0.93 (0.81)
Outdoors	0.59 (0.15)	1.10 (0.68)	0.85 (0.55)
Mean	0.61 (0.16)	1.16 (0.88)	0.89 (0.68)

Table 4. The reaction time in seconds for the four conditions.

		Stand						Walk					
		cas	otf	otb	ttf	ttb	n/a	cas	otf	otb	ttf	ttb	n/a
Indoor	cas	40	0	0	0	0	0	36	0	0	3	0	1
	otf	0	38	0	1	1	0	0	39	0	1	0	0
	otb	0	0	39	0	1	0	0	0	40	0	0	0
	ttf	0	0	0	40	0	0	0	0	0	40	0	0
	ttb	0	0	0	1	39	0	0	0	0	0	40	0
Outdoor	cas	36	0	0	1	2	1	34	0	0	2	3	1
	otf	0	36	1	2	1	0	0	33	1	6	0	0
	otb	0	0	40	0	0	0	0	0	38	0	2	0
	ttf	0	0	0	40	0	0	0	0	0	39	0	1
	ttb	0	0	0	0	40	0	0	0	0	3	37	0

Table 2. Confusion matrixes for the five gestures (cover all sensors, one time forward, one time back, two times forward and two times back) across all four conditions.

6 Discussion

The analyses of our data are quite promising. Returning to the motivation of our experiment, our data has provided several useful results. First and most importantly, the evaluation has shown that participants are able to successfully utilize our interaction technique and interact with the device using the area above the Gesture Watch. This result is even more interesting given the very minimal training needed. Furthermore, the Gesture Watch performed well while the user was walking; there was no statistically significant drop in accuracy, which is encouraging. There were however more falsely triggered gestures and users did take longer to respond to our prompt and start a gesture. It is possible these last two effects are a result of the more complex situation that the user must manage while mobile, but further investigation would be required to confirm this hypothesis.

Another potential issue our evaluation raises is with our use of the fifth proximity sensor for gesture segmentation. As revealed by comments from the participants, some people had some difficultly using the tilt of the wrist to trigger a gesture especially while mobile. Given these data we are interested in exploring better mechanisms in future work.

From a technical perspective, the sensors were able to function in both indoor and outdoor environments. There was a reduction in recognition accuracy for the outdoor conditions, but it is not clear if this drop in performance is a function of the sensor or of the user. It is possible that adding training data collected while outdoors may lead to better results.

Examining recognition accuracy reveals that overall participants could successfully use the system to perform our various gestures. This result shows that our arrangement of proximity sensors enabled the system to detect and differentiate between gestures that approached from different directions. This functionally is something past work that utilized a single proximity sensor was unable to accomplish [4]. However, there is still room for improvement. The confusion matrix reveals that several gestures are being recognized less than optimally. In all but the standing inside condition, several of the cover all sensor gestures were recognized incorrectly. These misclassifications could be a result of the user not knowing his exact hand placement or potentially not keeping all of the fingers of his hand together. For other gestures, the data indicate that we may want to explore different gesture design. In particular, the system had some difficulty differentiating between one time forward and two times forward and users also indicated similar issues. It is possible that the problems with these gestures are because the user must ensure not to pass over the sensors between gesture stokes. A slightly reconfigured gesture where the user passes her hand back and forth over the Gesture Watch may prove easier to perform and therefore result in better recognition results.

7 Future Work

We are interested in extending the Gesture Watch in several ways. First, we would like to overcome the issues revealed in our evaluation about our gesture trigger mechanism. One simple solution would be to only start gesture recognition once the trigger sensor is activated in conjunction with one of the four other sensors. This approach would likely reduce the number of falsely trigger gestures caused by the wrist movements of a walking user. Alternatively, we could augment the Gesture Watch with additional sensors. An accelerometer could measure the position of the device relative to gravity thus providing orientation. This data might be used to determine if the gesture watch hand is in a position to be operated, or is just moving during the course of the user's everyday activities. Finally, we could use proximity sensors that provide range data. The ranging information could be used to sense if the hand performing the gesture is in a valid distance range from the Gesture Watch. Alternatively, ranging sensors could be used to create three dimensional gestures.

Figure 13. The Sharp IS471F IR sensors and how they may be integrated into a wristwatch.

We are also interested in improving usability and mobility of the watch. Our current version of the watch sends the data to the computer, and users should carry the computer with the watch which takes up extra power and space. If we can remove the computer, the watch would be more usable and easy to carry for everyday life. One possible solution is to process and to interpret the raw data on the PIC microcontroller.

Next, we are interested in expanding the accuracy of the gestures we are recognizing, increasing the number of gesture we can recognize and determining how robust our gestures are. To improve the accuracy, we may explore utilizing different features or models that may be better suited for representing our data. In expanding our gesture set, we are interested in creating gestures that are both easy and intuitive for the users that are simultaneously relatively easy for the system to recognize. Here, an iterative development cycle will be valuable as will a specific target application that can provide constraints on which gesture make the most sense to the user.

Finally, now that we have demonstrated the feasibility of performing gesture recognition using our arrangement of infrared proximity sensors, we are interested in creating a device with a better form factor. While our current system is rather bulky, a more refined Gesture Watch would shrink in physical size by using smaller sensors. For example the Sharp IS471F IR sensor measures only 4.8x4.1x2.3mm. Figure 13 shows this sensor and indicates how they may be incorporated into a wristwatch.

8 Conclusions

The Gesture Watch is a new gesture based contact-free wristwatch interface. By using a set of five infrared proximity sensors we are able to detect a variety of interesting hand gestures using Hidden Markov Models. Our evaluation with 10 participants showed the Gesture Watch was effective in various lighting conditions including direct sunlight, and that it could be used while the user was both stationary and on the go. Overall we achieved recognition accuracy of 95.5% across all of our conditions. Given our results, we believe that this type of interaction technique, which utilizes the space above the device, is a successful method for overcoming the user interface problems that result from a continuing decrease in mobile device form factors.

References

[1] G. Blaskó and S. Feiner. An interaction system for watch computers using tactile guidance and bidirectional segmented strokes. In *Eighth IEEE International Symposium on Wearable Computers (ISWC'04)*, pages 102–104, 2004.

[2] G. Blaskó and S. Feiner. Evaluation of an eyes-free cursorless numeric entry system for wearable computers. In *Tenth IEEE International Symposium on Wearable Computers (ISWC'06)*, pages 21–28, 20046.

[3] K. Lyons, H. Brashear, T. Westeyn, J. S. Kim, and T. Starner. Gart: The gesture and activity recognition toolkit. In *Proceedings of HCI International*, 2007.

[4] C. Metzger, M. Anderson, and T. Starner. Freedigiter: A contact-free device for gesture control. In *Eighth IEEE International Symposium on Wearable Computers (ISWC'04)*, pages 18–21, 2004.

[5] C. Narayanaswami, N. Kamijoh, M. Raghunath, T. Inoue, T. Cipolla, J. Sanford, E. Schlig, S. Venkiteswaran, D. Guniguntala, V. Kulkarni, and K. Yamazaki. Ibm's linux watch, the challenge of miniaturization. *IEEE Computer*, 35(1):33–41, Jan. 2002.

[6] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *IEEE Intl. Symp. on Wearable Computers*, Atlanta, GA, 2000.

[7] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 20(12), December 1998.

[8] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *ICCV*, Bombay, 1998.

[9] T. Westeyn, K. Vadas, X. Bian, T. Starner, and G. D. Abowd. Recognizing mimicked autistic self-stimulatory behaviors using hmms. In *Ninth IEEE*

International Symposium on Wearable Computers (ISWC 2005), pages 164–169. IEEE Computer Society, October 2005.

[10] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.3)*. Cambridge University Engineering Department, 2005.