

Gesture recognition as ubiquitous input for mobile phones

Gerrit Niezen Gerhard P. Hancke

University of Pretoria
Lynnwood Road, Pretoria,
0002, South Africa
{gniezen, g.hancke}@ieee.org

ABSTRACT

A ubiquitous input mechanism utilizing gesture recognition techniques on a mobile phone is presented. Possible applications using readily available hardware are suggested and the effects of a mobile gaming system on perception is discussed.

Author Keywords

ubiquitous computing, accelerometers, gesture recognition, optimization, human-computer interfaces

ACM Classification Keywords

B.4.2 Input/Output Devices, H5.m. Information interfaces and presentation

INTRODUCTION

Mobile phones are the most pervasive wearable computers currently available and have the capabilities to alter and manipulate our perceptions. They contain various sensors, such as accelerometers and microphones, as well as actuators in the form of vibro-tactile feedback. Visual feedback may be provided through mobile screens or video eye wear.

Dynamic input systems in the form of gesture recognition are proving popular with users, with Nintendo's Wii being the most prominent example of this new form of interaction, that allows users to become more engaged in video games [1]. The video game experience is now affected not only by timing and pressing buttons, but also by body movement.

To ensure a fast adoption rate of gesture recognition as an ubiquitous input mechanism, technologies already available in mobile phones should be utilized. Features like accelerometer sensing and vibro-tactile feedback are readily available in high-end mobile phones, and this should filter through to most mobile phones in the future.

Hand gestures are a powerful human-to-human communication modality [2], and the expressiveness of hand gestures also allows for the altering of perceptions in human-computer

interaction. Gesture recognition allows users to perceive their bodies as an input mechanism, without having to rely on the limited input capabilities of current mobile devices. Possible applications of gesture recognition as ubiquitous input on a mobile phone include interacting with large public displays or TVs (without requiring a separate workstation) as well as personal gaming with LCD video glasses.

The ability to recognize gestures on a mobile device allows for new ways of remote social interaction between people. A multiplayer mobile game utilizing gestures would enable players to physically interact with one another without being in the same location. Gesture recognition may be used as a mobile exertion interface [3], a type of interface that deliberately requires intensive physical effort. Exertion interfaces improve social interaction, similar to games and sports that facilitate social interaction through physical exercise. This may change the way people perceive mobile gaming, as it now improves social bonding and may improve overall well-being and quality of life.

Visual, auditory and haptic information should be combined in order to alter the user's perceptions. By utilizing video glasses as visual feedback, earphones as auditory feedback and the mobile phone's vibration mechanism as haptic feedback, a pervasive mobile system can be created to provide a ubiquitous personal gaming experience. Gesture recognition is considered as a natural way to interact with such a system.

Gesture recognition algorithms have traditionally only been implemented in cases where ample system resources are available, i.e. on desktop computers with fast processors and large amounts of memory. In the cases where a gesture recognition has been implemented on a resource-constrained device, only the simplest algorithms were considered and implemented to recognize only a small set of gestures; for example in [5], only three different gestures were recognized.

We have developed an accelerometer-based gesture recognition technique that can be implemented on a mobile phone. The gesture recognition algorithm was optimized such that it only requires a small amount of the phone's resources, in order to be used as a user interface to a larger piece of software, or a video game, that will require the majority of the system resources. Various gesture recognition algorithms currently in use were evaluated, after which the most suitable algorithm was optimized in order to implement it on a mobile phone [6]. Gesture recognition techniques studied include

Copyright is held by the author/owner(s).

UbiComp '08 Workshop W1 – Devices that Alter Perception (DAP 2008)

September 21st, 2008

This position paper is not an official publication of UbiComp '08.

hidden Markov models (HMMs), artificial neural networks and dynamic time warping. A dataset for evaluating the gesture recognition algorithms was gathered using the mobile phone's embedded accelerometer. The algorithms were evaluated based on computational efficiency, recognition accuracy and storage efficiency. The optimized algorithm was implemented in a user application on the mobile phone to test the empirical validity of the study.

CURRENT IMPLEMENTATIONS

Choi et al. [7] used accelerometer data acquired from a mobile phone's built-in accelerometer. They were able to recognize digits from 1 to 9 and five symbols written in the air. During their experimental study, they were able to achieve a 97.01% average recognition rate for a set of eleven gestures. The recognition rate was cross-validated from a data set of 3082 gestures from 100 users. This was done using a Bayesian network based approach, with gesture recognition done on a PC connected to the mobile phone.

Pylvänäinen [8] employed an accelerometer-based gesture recognition algorithm using continuous HMMs, with movements recorded using an accelerometer embedded in a mobile phone, but gesture recognition was still performed on a desktop PC. A left-to-right HMM with continuous normal output distributions was used. The performance of the recognizer was tested on a set of 10 gestures, 20 gesture samples from 7 different persons, resulting in a total of 1400 gesture samples. Every model for each of the 10 gestures had 8 states. 99.76% accuracy was obtained with user-independent testing. Pylvänäinen argued that an extensive set of gestures (i.e. more than 10) becomes impractical due to users having to learn all the different gestures.

With gesture recognition one should distinguish between postures, involving static pose and location without any movements; and gestures, involving a sequence of postures connected by continuous motions over a short time span [2]. Crampton et al. [1] developed an accelerometer-based multi-sensor network to recognize both postures and gestures. The wearable sensor network detects a user's body position as input for video game applications, providing for an immersive game experience. Mahalanobis distance is used as a nearest-neighbour means of classification. This improves on using Euclidian distance as a metric, as it takes into account the correlations of the data set and is scale-invariant. They argue that the more accelerometers are used, the more accurately gestures and poses can be differentiated. This should be taken into account when developing a gesture-based system, and is discussed further later in the paper.

Current accelerometer-based motion-sensing techniques in mobile phones are either based on tilt or orientation, allowing for simple directional movement control in games. Camera-based methods for gesture recognition are also becoming more popular. A company called GestureTek [19] enables mobile phones with built-in cameras to be used as motion-sensing devices. In the case of camera-based computer vision algorithms, the necessary image processing can be slow, which creates unacceptable latency for fast-moving video

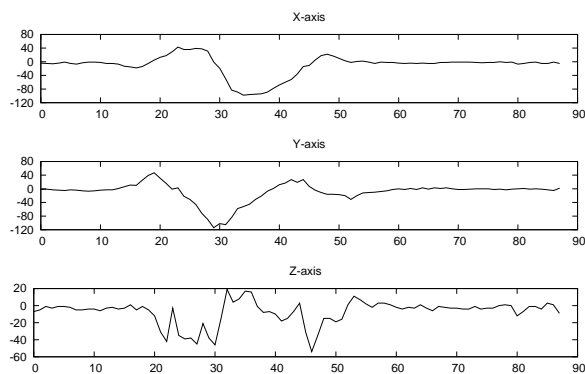


Figure 1. Raw sensor data sampled from the Nokia N95's accelerometer

games and other applications [20]. Camera-based sensors are also deemed power-hungry, which is a problem considering that the amount of power consumed during operation is of utmost importance in a mobile device.

IMPLEMENTATION AND RESULTS

In [9], we describe how various gesture recognition techniques were evaluated, after which the most suitable algorithm was optimized in order to implement it on a mobile device. We make use of the Dynamic Time Warping (DTW) algorithm, introduced by Sakoe and Chiba [10] in a seminal paper in 1978. The DTW algorithm used was originally implemented in C by Andrew Slater and John Coleman [11] at Oxford University Phonetics Laboratory. The DTW algorithm non-linearly wraps one time sequence to match another given start and end point correspondence.

Sensor data was collected using a Nokia N95's embedded 3-axis STMicroelectronics LIS302DL accelerometer. The Symbian 3rd Edition SDK's Sensor API was used to gather raw sensor data using an interrupt-driven sampling method. The data was filtered using both a digital low-pass filter (LPF) and a high-pass filter (HPF). In figure 1 the raw sensor data gathered from the mobile phone's accelerometer is shown for all the three axes.

A total of 8 gestures with 10 samples per gesture were collected. As the DTW algorithm is essentially a type of template-matching technique, only one training sample per gesture was required for the DTW algorithm to perform the gesture recognition correctly. The 8 gestures used in this study can be observed in figure 2. The gestures used were obtained from a study done by Bailador et al. [12]. The DTW algorithm was able to correctly classify a total of 77 of the 80 samples, for an overall accuracy of 96.25%. The algorithm was optimized [9] for the mobile phone and the recognition time was reduced from around 1000 ms to under 200 ms.

The gesture recognition algorithm was ported to the mobile device by making use of Nokia's Open C platform [13]. Open C is a set of POSIX libraries to enable standard C programming on Symbian Series 60 devices.

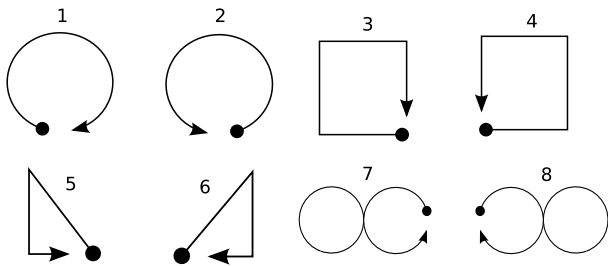


Figure 2. Gestures used in this study



Figure 3. User application running on the mobile device

A user application was implemented to test the real-world functionality of the gesture recognition algorithm. The user application was developed in the Python programming language and executed on the mobile phone using Nokia's Python for Series 60 (S60) version 1.4.1 utilities [14]. Using Python allows one to rapidly prototype a graphical user interface (GUI) and other functionality by making use of the built-in APIs to provide, for example, sound and graphics capabilities. An example of the user application running on the mobile device is shown in figure 3.

The gesture recognition algorithm (written in C) was linked into the Python program as a dynamically linked library (DLL). Wrapper code was created for the C algorithm in order to link it into the Python program. The user application was converted into a standalone Python program on the Symbian device through the Ensemble developer utilities for Symbian S60 [15]. It can also run as a script in the Python for S60 shell.

To have the system learn a new gesture, the user can select the *New Gesture* command from the pop-up menu. When the user starts moving the phone, the application records the gesture until the phone stops moving. The recorded gesture is then stored as a reference gesture on the phone. To recognize a gesture, the user selects the *Recognize* command from the pop-up menu. The application records the test gesture as soon as the user starts moving the phone. When the device stops moving, the application executes the gesture recognition algorithm and displays the recognized gesture as a graphic on the screen.

Nokia's Python for Series 60 does not provide built-in support for vibro-tactile feedback, but third-party utilities have

been developed to overcome this. For Series 60 3rd Edition devices (like the Nokia N95) a third-party module called *misty* provides vibration support, and for Series 60 2nd Edition an earlier package called *mi_so* was developed. These capabilities will probably be added to the Nokia Python library in future.

Haptic feedback was added to the user application by utilizing the vibro-tactile capabilities of the mobile phone when a gesture is recognized. Visual feedback is provided by displaying a graphic of the gesture on-screen. Auditory feedback was added by having the recognized gesture spoken out loud using the text-to-speech functionality of the Nokia Python Audio API.

Personal media viewers, such as the Myvu Crystal [16], allow for a full-screen mobile viewing experience. When combined with a mobile phone such as the Nokia N95 with an embedded accelerometer, our gesture recognition algorithm and a mobile game, the pervasive mobile gaming system as described in the previous sections becomes possible. The Myvu glasses can be connected to the Nokia N95 via the Nokia AV connector, a 3.5 mm stereo headphone plug.

It is envisioned that personal media viewers such as the Myvu will enable mobile gesture-based gaming opportunities until true see-through head mounted displays become less expensive. With the Myvu video glasses it is possible to look above or below the screen, which allows one to walk around. This makes it possible to use the video glasses for urban gaming, or other applications where the user is required to physically walk around while still wearing the video glasses.

Another possible application would be body mnemonics, an interface design concept for portable devices that uses the body space of the user as an interface [17]. Different body positions may be used as markers to remember computational functionality or information such as phone book entries. For example, the user might move the mobile phone to the shoulder or head to access a specific sub-menu or program on the phone. Continuous audio or tactile feedback relating to the user's motion or gesture trajectories may be provided. It is believed that this kind of tightly coupled control loop will support a user's learning processes and convey a greater sense of being in control of the system [18]. User interfaces or functions can now be logically or emotionally mapped to the user's body, completely changing the perception of interacting with a mobile device.

CONCLUSION

Gestures can change the way we interact with computers and mobile devices. This is evident in new user interfaces such as the multi-touch interface introduced by the Apple iPhone. The multi-touch interface adds motion gaming capabilities to the iPhone, albeit in a different sense than using accelerometer-based gesture recognition. This paper describes a cost-effective mobile system that can be implemented with readily available hardware and realizable software on a mobile phone. An optimized gesture recognition algorithm that require minimal resources was described and

implemented on a mobile phone.

Accelerometer-based techniques have an advantage above camera-based techniques, i.e. that computationally intensive calculations are not required for accurate movement information, as measurements are directly provided by the sensors. Sensor-based techniques also have the advantage in that they can be used in much less constrained conditions and are not reliant on lighting conditions or camera calibration [21].

To provide a more immersive experience, wireless video glasses may be developed that does away with cumbersome cabling. For the video glasses to be connected to a mobile phone, the wireless technologies used will most probably have to be Bluetooth or Wi-Fi, as these technologies are already available in mobile phones. This is an avenue for further exploration, since as of this writing no true wireless video glasses have been developed.

Possible pitfalls for gesture recognition in mobile phones include user acceptability: Will a user feel comfortable waving his or her arms around in a public space? Haptic feedback is also important for user acceptance. The Nintendo Wii, for example, incorporates this by providing both auditory and vibro-tactile feedback when performing a gesture. A user must know the set of gestures that a system recognizes and gestures requiring high precision over a long period of time can cause fatigue. Therefore the gestures must be designed to be simple, natural and consistent. If the gestures prove to be tiring or strenuous, any possibility of altering the user's perceptions will be limited.

When only one accelerometer is used, the accuracy in detecting the various gestures is reduced. With the Nintendo Wii, for example, the basic motions it detects can easily be cheated with partial movement [1], which reduces the immersive perception of a video game. Utilizing multiple accelerometers increases accuracy at additional cost. Adding additional accelerometer-based sensing devices to a mobile gaming system should not be technically complex, as Bluetooth may be used for communicating with the mobile phone.

Location-based games, also known as urban gaming, can utilize a mobile phone's GPS receiver to provide a realistic, augmented reality-type gaming experience. This may be combined with the methods described in this paper to improve even further on the alteration and modification of the user's perceptions. Hand gestures can also be used in 3D virtual environments to provide a more natural and immersive user experience [2], truly altering users' perceptions in viewing and experiencing their environment.

REFERENCES

1. Crampton, N., Fox K., Johnston H. and Whitehead A. Dance Dance Evolution: Accelerometer Sensor Networks as Input to Video Games. In *Proc. IEEE HAVE 2007*, 107-112.
2. Chen Q., Petriu E.M. and Georganas, N.D. 3D Hand Tracking and Motion Analysis with a Combination Approach of Statistical and Syntactic Analysis. In *Proc. IEEE HAVE 2007*, 56-61.
3. Mueller F., Agamanolis S. and Picard, R. Exertion interfaces: sports over a distance for social bonding and fun. In *CHI '03: Proc. SIGCHI Conf. on human factors in computing systems 2003*, 561-568.
4. Khronos Group. OpenGL ES Overview. <http://www.khronos.org/opengles/>.
5. Feldman, A., Tapia, E.M., Sadi, S., Maes, P. and Schmandt, C. ReachMedia: On-the-move interaction with everyday objects. In *Proc. IEEE ISWC 2005*, 52-59.
6. Niezen G. The optimization of gesture recognition techniques for resource-constrained devices. M.Eng. thesis, University of Pretoria, South Africa, 2008.
7. Choi, E., Bang, W., Cho, S., Yang, J., Kim, D., and Kim, S. Beatbox music phone: gesture-based interactive mobile phone using a tri-axis accelerometer. In *Proc. IEEE ICIT 2005*, 97-102.
8. Pylvänäinen, T. Accelerometer Based Gesture Recognition Using Continuous HMMs. In *LNCS: Pattern Recognition and Image Analysis*, Springer-Verlag (2005).
9. Niezen G. and Hancke G.P. Evaluating and optimising gesture recognition techniques for mobile devices. *Int'l J. Human-Computer Studies*. Elsevier (submitted June 2008).
10. Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Trans. Acoustics, Speech, and Signal Processing*, 26 (1), 43-49.
11. Coleman, J. *Introducing speech and language processing*. Cambridge University Press, Cambridge, UK, 2005.
12. Bailador, G., Roggen, D., Tröster, G. and Triviño, G. Real time gesture recognition using Continuous Time Recurrent Neural Networks, In *Proc. Int. Conf. Body Area Networks 2007*.
13. Nokia Research Center. Open C: Standard-based Libraries for Symbian-based Smartphones. <http://opensource.nokia.com/projects/openc/>.
14. Nokia Research Centre. Python for S60. <http://opensource.nokia.com/projects/pythonfors60/>.
15. Ylänen, J. The Ensymble developer utilities for Symbian OS. <http://www.nbl.fi/~nbl928/ensymble.html>.

16. Myvu Corporation. Myvu Crystal.
<http://www.myvu.com/Crystal.html>.
17. Ängeslevä J., Oakley I., Hughes S. and O'Modhrain, S. Body mnemonics - portable device interaction design concept. In *UIST - Adjunct Proc. ACM Symposium on User Interface Software and Technology 2003*.
18. Strachan S., Murray-Smith R., Oakley I. and Ängeslevä J. Dynamic Primitives for Gestural Interaction. In *LNCS: MobileHCI*, Springer-Verlag (2004).
19. GestureTek Mobile.
<http://www.gesturetekmobile.com>.
20. Geer, D. Will gesture recognition technology point the way? *IEEE Computer*, 37(10), 2004, 20-23.
21. Chambers, G.S., Venkatesh, S., West, G.A.W. and Bui, H.H. Hierarchical recognition of intentional human gestures for sports video annotation. In *Proc. Int. Conf. on Pattern Recognition*, 1082-1085.