Andreas Braun, Tim Dutz, Felix Kamieth:
Capacitive Sensor-Based Hand Gesture Recognition in Ambient Intelligence Scenarios. In: PETRA '13 - Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments, ACM, May 2013. ISBN 978-1-4503-1973-7.

Capacitive Sensor-Based Hand Gesture Recognition in Ambient Intelligence Scenarios

Andreas Braun
Fraunhofer IGD
Fraunhoferstr.5
64283 Darmstadt
+49-6151-155-208
andreas.braun@igd.fraunhofer.de

Tim Dutz
Fraunhofer IGD
Fraunhoferstr.5
64283 Darmstadt
+49-6151-155-233
tim.dutz@igd.fraunhofer.de

Felix Kamieth
Fraunhofer IGD
Fraunhoferstr.5
64283 Darmstadt
+49-6151-155-576
felix.kamieth@igd.fraunhofer.de

ABSTRACT

Input devices based on arrays of capacitive proximity sensors allow the tracking of a user's hands in three dimensions. They can be hidden behind materials such as wood, wool or plastics without limiting their functionality, making them ideal for application in Ambient Intelligence (AmI) scenarios. Most gesture recognition frameworks are targeted towards classical input devices and interpret two-dimensional data. In this work, we present a concept for adapting classical gesture recognition methods for capacitive input devices by realizing an extension of the feature set to three dimensional input data. This allows more robust gesture recognition for free-space interaction and training specific to capacitive input devices. We have implemented this concept in a prototypical setup and tested the device in various Ambient Intelligence scenarios, ranging from manipulating home appliances to controlling multimedia applications.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: *User Interfaces* - Input devices and strategies

General Terms

Experimentation

Keywords

Capacitive Proximity Sensors, Gesture Recognition, User Interfaces, Input Devices, Smart Environments, Ambient Intelligence

1. Introduction

Capacitive proximity sensors allow not just the detection, but a distance estimation of conductive, grounded objects, such as the human body. In the past, this property has already been used to create devices that can estimate the position of one or more hands [23,24], or the posture of a person on different pieces of furniture [8,9]. A distinct advantage of capacitive proximity sensors is their ability to detect objects without being disturbed by non-conductive materials. They can be installed invisibly behind solid objects, which allows for an unobtrusive application. In questions of functionality and reliability, capacitive sensor-based devices used for hand tracking a comparable to gesture recognition system that are based on other technologies such as cameras [15] and accelerometers [22]. Many of the corresponding gesture recognition software frameworks are based on algorithms that use learning-by-example [17], for instance used in conjunction with pointing devices [6], accelerometer-based input devices [22] and camera-based [15] gesture recognition systems. However, to our knowledge, there is no generic framework available to support gesture recognition with capacitive input devices. In this research paper, we will thus present such a framework and evaluate it in a typical home control scenario, using a prototypical capacitive sensor based device for hand-tracking [4,13].

PETRA '13, May 29 - 31 2013, Island of Rhodes, Greece
Copyright 2013 ACM 978-1-4503-1973-7/13/05…$15.00.
http://dx.doi.org/10.1145/2504335.2504340

2. Free Space Gesture Recognition

There are various technologies available to recognize gestures in free air. Common methods include cameras [19], depth sensors [16] or capacitive systems [23]. This work is focusing on the latter. Compared to the other systems capacitive sensors can be employed unobtrusively, work through various materials and do not have a high computational cost [24]. However, there are also various drawbacks, including a lower resolution, limited detection distance sensitivity towards dynamic electric fields in the immediate, environment and shielding issues in various materials [3].

Concerning free air gestures there are some challenges that do not occur in other more confined gesture-recognition systems, such as touch-screens. The latter have a clearly constrained interaction area and can clearly distinguish between an intentional gesture and spontaneous hand movement. The methods to solve that include using initiation gestures [14], multimodal interfaces [12] or restrictions of the gesture set. Given the limited detection distance and low resolution of capacitive sensors the last method is a viable candidate and has been evaluated in this document.

The problem of deducing a hand position from capacitive sensor data is a complex task. The irregular shape of the hands, differences in electrode geometry and environmental conditions result in a highly non-linear problem that has to be solved with numerical approximation [2]. This can be achieved using different approaches, ranging from using tomography methods [18] to simplified location-based averaging [4]. Tracking a *moving* hand - and especially the detection of a gesture - presents a much larger challenge. Noise and sensor data drift cause the distance measurement of each sensor to be relative to a variable maximum detection distance [4]. The main consequence of this is that the resolution and scale of such a hand detector vary in the different dimensions and cannot be directly transferred to absolute measurements. This limitation has to be considered by choosing a gesture recognition method that is robust against dynamic scaling of measurements and non-static paths created by noisy signals. Furthermore, the method should allow for a generic gesture definition that enables reusability in different contexts. Gesture specification by example [17] is a well-performing approach for generalizing the recognition process that has been implemented for varying gesture recognition systems. Wilson et al. [22] have evaluated different example-based methods to determine the best solution for accelerometer-based input devices. Nam et al. [15] have performed a similar study for hand-gesture recognition, using cameras.

For us, a main source of inspiration has also been the various frameworks for mouse and touch gesture recognition which are employed in many devices and desktop applications [6,10]. The aim of our research is to employ gesture recognition based on learning-by-example in the domain of capacitive proximity sensors. To this end, we have developed a framework that uses generic methods to implement gesture learning and additionally addresses the specific challenges posed when working with capacitive proximity sensors.

3. Gesture Recognition Framework

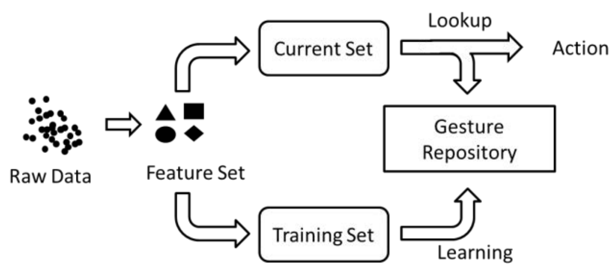

Figure 1 - Principle components of a gesture recognition framework

The general functionality of a gesture recognition framework that is using learning by example is shown in Figure 1. A set of features is extracted from incoming raw data. These are distributed to training sets that are used to associate certain features to given gestures. After this learning process is completed current feature sets that are acquired on-the-fly are tested against the training features in the repository. If certain criteria are met, these lookups lead to successfully recognized gestures and subsequently, an action may be performed.

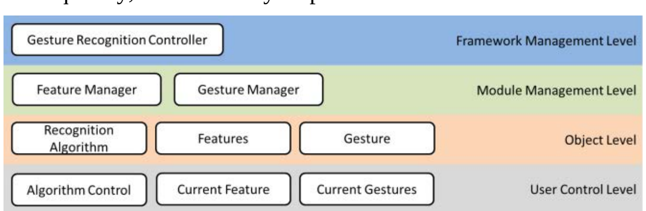

Figure 2 - Gesture recognition framework layer model

Our proposed framework is implementing gesture recognition by example, employing a multi-level design as shown in Figure 2. The lowest level - User Control - is a collection of aspects that are controllable by the user. The Algorithm Control allows setting the parameters of the gesture recognition algorithm, Current Feature allows selecting between different methods of feature acquisition from the raw data and Current Gestures is the set of gestures that can be recognized. The Object Level is encompassing all objects in the frameworks, the group of gesture recognition algorithms the user can choose from and their current settings. Furthermore, there are the available features including settings and the set of available gestures. Above is the Module Management Level that controls feature acquisition and gesture management on a contextual level, meaning that, based on the current situation, the framework can select different features and gestures to process. Finally the Framework Management Level is controlling high-level functionality of all modules and provides interfaces to external applications to control the framework and access registered gestures.

The framework is implemented in the Microsoft .NET environment (version 4.0) using C#. It requires a gesture recognition device that provides location data in three dimensions. In our cases this is an array of capacitive proximity sensors that will be detailed in the prototype section.

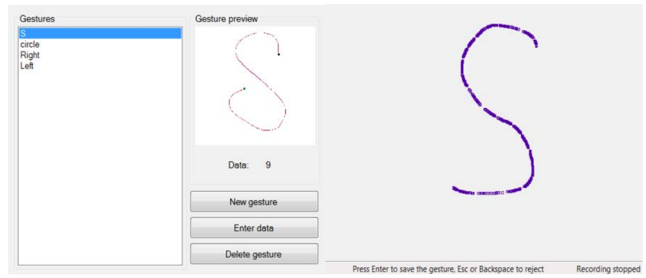

Figure 3 - Screenshots of gesture manager and gesture recorder

The key aspect of gestures by example – providing examples - is realized in a debug application. It provides a simple way to record exemplary movements and associate them to gesture sets. The main screens realizing this functionality are shown in Figure 3. On the left side we can see the management screen that allows adding and deleting of gestures, as well as a preview window that is an average of the sample data associated to this gesture. The process of entering data is shown on the right side where several samples can be recorded and associated to the selected gesture and the user can decide, whether the current movement should be stored or discarded.

4. Home Automation Connection

The 20-year old vision of Ubiquitous Computing, famously described by Mark Weiser [21], is based on a seamless integration of technical devices and interfaces into a user's environment. The vision later evolved into the paradigm of Ambient Intelligence [1] that in recent years has been realized in pilot sites around Europe, e.g., the Great Northern Haven assisted living facility in Ireland [5]. As Ambient Intelligence systems usually incorporate various heterogeneous devices, they tend to become rather complex and thus, they require sophisticated human-computer-interfaces that allow access to all services. In this context users should be enabled to communicate with AmI systems in a natural manner, utilizing multiple modalities such as voice commands and gestures [20]. It is desirable that devices are installed discreetly and hidden from view, particularly for AmI systems that are used to enhance the private homes of their users.

AmI systems require platforms, specifically developed for the purpose of connecting and controlling heterogeneous technical devices that come from different areas of application, such as home automation devices, entertainment devices and whiteware. One example is the universAAL platform [11], which is currently being developed in the EU-funded research project of the same name[1]. Its semantic approach, based on shared ontologies, supports the quick and reasonable integration of additional devices into an existing system, thus allowing it to easily adapt to its users' changing needs. This includes both input devices used

[1] http://www.universaal.org

for explicit interaction - such as a light-switch or a microphone - and those that participate in the implicit interaction between the user and the AmI system, that is, devices gathering information about the context of the user, such as, movement sensors or cameras. Capacitive proximity sensors are suited for designing both types of devices [4,8]. They can be readily installed underneath a variety of surfaces such as wood, plastic and ceramic. Capacitive proximity sensors can be seen as generic proximity detectors and are a basis for a multitude of different interfaces, as presented on a previous PETRA conference [3].

5. Prototype System

We tested the concept presented in this work with a prototypical capacitive sensor based input device informally dubbed the CapTab [4]. It is comprised of six sensor units by Cypress Semiconductor (AD7124) that are combined into an array and attached to six electrodes fixed to the inner frame of the upper surface.

The sensors communicate wirelessly with a base station on a PC using a custom 2.4GHz protocol. Two different filter methods for raw sensor data and two filter methods for the calculated hand position are supported. A moving average filter is using dynamic windows to flatten the sensor values and a median filter allows resiliency towards outliers. The hand position is filtered either by a fixed-window average filter or a derivate filter that requires a minimal amount of movement before subsequent locations are determined. The setting of which filter to choose depends on the application.

Concerning our previously presented gesture recognition framework two points are of interest, finding suitable features and recognition algorithms and most importantly determine a set of gestures that can be distinguished by our system and reliably used to control the different appliances. Regarding features it has turned out that using a three dimensional filtered position vector is suitable. The number of features determining a gesture is flexible, however it has turned out that using a set of eight features is sufficient. The recognition algorithm is using the spatial distance between the objects to distinguish gestures. For that all locations are normalized.

Some gestures that are regularly used on other systems, such as circles and multi-finger/multi-hand gestures are difficult to realize using this prototype, because of its limited resolution and a high signal-to-noise ratio that is affecting the localization algorithm. Furthermore supporting a high number of gestures can be counter-intuitive as it increases the required learning time and cognitive load. The gesture recognizer requires a specific threshold before a gesture is matched. This can be used to discard atypical movements and significantly reduce the number of false-positives.

However this approach is only feasibly for a low set of gestures. Consequently we finally settled on a set of eight relatively simple gestures that can be used for controlling other devices – the gestures are shown in Figure 4.

In order to evaluate the implementation, the CapTab device was connected to the universAAL-based AmI system of a Living Lab that features the layout and interior of a regular one-bedroom apartment and thus allows the evaluation of AmI technologies in real-world scenarios.

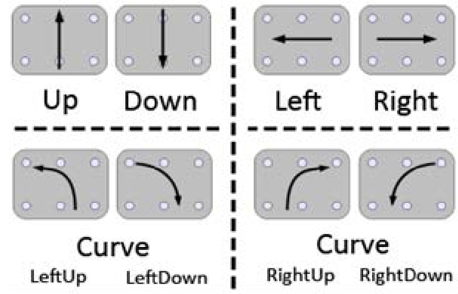

Figure 4 - Selected gesture set

A simple exemplary application for capacitive proximity sensors is their integration into the living room table. In the scope of this work we have realized three different scenarios - using an image viewer application, switching lamps and controlling a window. Using our gesture recognition framework on top of the universAAL platform we are able to associate different gestures to the various functions used in all scenarios. The image viewer scenario is a C# application that is linked to universAAL via TCP socket and a small Java application interpreting the events. The gesture recognition framework allows switching between images and enhancing details by zooming in and out of images.

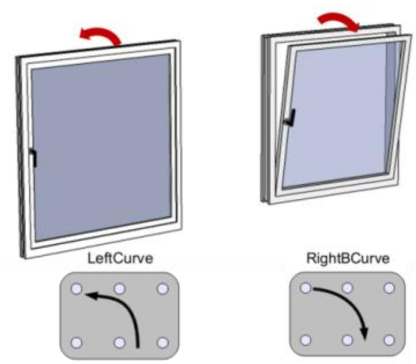

Figure 5 - Gestures associated to window opened/closed

The light switching scenario and window control scenario are realized using a direct connection between gesture recognition framework and the AmI platform. Gestures and functions for the window scenario are shown in Figure 5.

6. Evaluation

In order to evaluate our results, we placed the CapTab device onto a living room table and associated certain gestures to functions made available by the AmI platform. Swiping of a hand above the CapTab should lower and raise the electric blinds of the windows, respectively, while performing a half-curve gesture (either with the left, or with the right hand) should open and close the windows. The purpose of this evaluation was to determine if it is possible to use such a system in AmI applications and determine the user acceptance.

We found that by limiting the set of possible gestures to only a handful, the performance of our gesture detection framework connected to the CapTab device is sufficient for reliably controlling basic home control scenarios. Furthermore, for the selected gestures, the presented framework is resilient to gestures deviating from the norm, i.e. gestures performed rather sloppily. By limiting the set of possible gestures to only a few diverse, one also induces an inherent error-tolerance, as even hastily and imprecisely performed gestures can still be identified.

7. Conclusion

In this work, we have presented a generic framework for hand gesture recognition that is tailored to input devices based on arrays of capacitive proximity sensors. This framework has been used in conjunction with a software suite that processes data acquired by a prototypical capacitive sensor system. We have determined a suitable collection of gesture recognition parameters that were used to create a set of gestures that enable controlling devices in typical home automation.

However, while in theory the recognition rate is high enough even for complex gestures such as the drawing of letters or numbers, there are practical limitations of the tested system. Even though the framework is intended for three-dimensional tracking of hand positions, this has proven difficult with our current prototype due to strong differences in resolution of the dimensions. As such, our central finding is that the number of potential gestures being usable is not as high as anticipated. One reason is considerable sensor noise that slows down the tracking process and essentially limits the gesture set to a few easily distinguishable gestures.

Consequently, we plan on updating the system with an enhanced sensor platform called OpenCapSense [7] that was recently finalized and improves signal-to-noise ratio We are planning on evaluating our gesture recognition framework with other installments of capacitive proximity sensor input devices, using different electrode configurations aimed at improving resolution in the third dimension.

8. Acknowledgments

The authors would like to thank Andreas Franek, Sebastian Jeckel, and Pia Weiss for their vital contributions to this work.

REFERENCES

[1] Aarts, E. and Marzano, S. *The New Everyday View on Ambient Intelligence*. Uitgeverij 010 Publishers, 2003.

[2] Baxter, L.K. *Capacitive Sensors: Design and Applications*. IEEE Press, 1997.

[3] Braun, A. and Hamisu, P. Using the human body field as a medium for natural interaction. *Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments - PETRA '09*, ACM Press (2009), 1–7.

[4] Braun, A. and Hamisu, P. Designing a multi-purpose capacitive proximity sensing input device. *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments PETRA 11*, (2011), 1.

[5] Doyle, J., O'Mullane, B., O'Hanlon, A., and Knapp, R.B. Requirements gathering for the delivery of healthcare data in aware homes. *5th International Conference on Pervasive Computing Technologies for Healthcare*, (2011), 254–257.

[6] Dulberg, M.S., Amant, R.S., and Zettlemoyer, L.S. An Imprecise Mouse Gesture for the Fast Activation of Controls. *INTERACT*, (1999), 1–10.

[7] Grosse-Puppendahl, T., Berghoefer, Y., Braun, A., Wimmer, R., and Kuijper, A. OpenCapSense: A Rapid Prototyping Toolkit for Pervasive Interaction Using Capacitive Sensing. *IEEE International Conference on Pervasive Computing and Communications (PerCom) 18*, (2013), 22.

[8] Große-Puppendahl, T., Marinc, A., and Braun, A. Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments. *Ambient Intelligence 7040*, (2011), 314–323.

[9] Hamisu, P. and Braun, A. Analyse des Schlafverhaltens durch kapazitive Sensorarrays zur Ermittlung der Wirbelsäulenbelastung. *3. Deutscher AAL Kongress*, (2010), 3–6.

[10] Han, J.Y. Low-cost multi-touch sensing through frustrated total internal reflection. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*, ACM Press (2005), 115.

[11] Hanke, S., Mayer, C., Hoeftberger, O., et al. universAAL - An Open and Consolidated AAL Platform. *Fourth Ambient Assisted Living Congress*, Springer (2011), 127–140.

[12] Hauptmann, A.G. Speech and gestures for graphic image manipulation. *ACM SIGCHI Bulletin 20*, SI (1989), 241–245.

[13] Kessler, G.D., Hodges, L.F., and Walker, N. Evaluation of the CyberGlove as a whole-hand input device. *ACM Transactions on Computer-Human Interaction 2*, 4 (1995), 263–283.

[14] Manresa, C., Varona, J., Mas, R., and Perales, F. Hand tracking and gesture recognition for human-computer interaction. *Electronic letters on computer vision and image analysis 5*, 3 (2005), 96–104.

[15] Nam, Y. Recognition of space-time hand-gestures using hidden Markov model. *ACM symposium on Virtual reality software and*, (1996).

[16] Ren, Z., Meng, J., Yuan, J., and Zhang, Z. Robust hand gesture recognition with kinect sensor. *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, ACM Press (2011), 759.

[17] Rubine, D. Specifying gestures by example. *ACM SIGGRAPH computer graphics 25*, 4 (1991), 329–337.

[18] Smith, J.R. Electric field imaging. 1999.

[19] Utsumi, A. and Ohya, J. Multiple-hand-gesture tracking using multiple cameras. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, IEEE Comput. Soc, 473–478.

[20] Valli, A. The design of natural interaction. *Multimedia Tools and Applications 38*, 3 (2008), 295–305.

[21] Weiser, M. The Computer for the 21st Century. *Scientific American 265*, 3 (1991), 94–104.

[22] Wilson, A. and Shafer, S. XWand: UI for intelligent spaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ACM (2003), 545–552.

[23] Wimmer, R., Holleis, P., Kranz, M., and Schmidt, A. Thracker - Using Capacitive Sensing for Gesture Recognition. *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)*, IEEE (2006), 64–64.

[24] Zimmerman, T.G., Smith, J.R., Paradiso, J.A., Allport, D., and Gershenfeld, N. Applying electric field sensing to human-computer interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, May (1995), 280–28