Application and validation of capacitive proximity sensing systems in smart environments

dem Fachbereich Informatik
der Technischen Universität Darmstadt
vorzulegende

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

M.Sc. Andreas Braun

geboren in Aschaffenburg, Deutschland

Referenten der Arbeit:	Prof. Dr. techn. Dieter W. Fellner Technische Universität Darmstadt Prof. XXX affiliation of Prof. XXX
Tag der Einreichung:	28/04/2014
Tag der mündlichen Prüfung:	

Darmstädter Dissertation
D 17

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 28/04/2014 Andreas Braun

Abstract

Summarize the thesis in 1/2–1 page.

Zusammenfassung

Describe in German in 6–10 pages your thesis. This is compulsory for EN written thesis. Zusammenfassung auf Deutsch.

Contents

1. Introduction **1**
1.1. Motivation . . . 2
1.2. Research Challenges . . . 3
1.3. Contributions . . . 5
1.4. Structure of this work . . . 5

2. Related Work **7**
2.1. Electric field sensing . . . 7
2.1.1. Physical properties . . . 7
2.1.2. Proximity sensing versus touch sensing . . . 9
2.1.3. Measuring modes . . . 9
2.1.4. Materials and geometry . . . 10
2.1.5. Data processing . . . 13
2.2. Capacitive proximity sensing applications . . . 18
2.3. Sensor systems in smart environments . . . 25
2.3.1. RGB cameras . . . 26
2.3.2. Infrared cameras . . . 28
2.3.3. Ultrasound sensors . . . 29
2.3.4. Microphone arrays . . . 31
2.3.5. Radiofrequency sensing . . . 32
2.4. Applications in smart environments . . . 36
2.4.1. Indoor localization . . . 36
2.4.2. Gestural interaction . . . 38
2.4.3. Physiological sensing . . . 39
2.4.4. Activity recognition . . . 40
2.4.5. Smart appliances . . . 41
2.4.6. Mobile devices . . . 42
2.4.7. Autonomous systems . . . 44

3. Benchmarking model for sensors in smart environments **45**
3.1. Sensor features . . . 46
3.1.1. Sensor performance characteristics . . . 46
3.1.2. Pervasive metrics . . . 46
3.1.3. Environmental characteristics . . . 47
3.1.4. Discussion and feature matrix . . . 47
3.2. Benchmarking model . . . 49
3.2.1. Feature score and weighting . . . 49
3.2.2. Modeling . . . 50
3.2.3. Feature score normalization . . . 50

3.2.4. Scoring . . . 51
3.3. Evaluating technology popularity . . . 52
3.3.1. Scoring . . . 52
3.4. Discussion . . . 53
3.4.1. Central tendency bias . . . 54
3.5. Applications for capacitive proximity sensors . . . 55
3.5.1. Benchmark weights . . . 55
3.5.2. Benchmarking scores for capcitive proximity sensors . . . 58
3.5.3. Popularity of capacitive proximity sensors in literature . . . 58

4. Use cases for capacitive proximity sensors **59**
4.1. Use cases and associated challenges . . . 59
4.2. Processing methods . . . 62
4.2.1. Sparsely distributed sensor arrays . . . 63
4.2.2. Model-driven fitting methods . . . 66
4.2.3. Heterogeneous sensor systems . . . 70
4.2.4. Image-based processing . . . 71
4.2.5. Physiological signals in frequency- and time-domain . . . 75
4.3. Application prototypes . . . 78
4.3.1. CapFloor . . . 78
4.3.2. Smart Bed . . . 81
4.3.3. The Capacitive Chair . . . 83
4.3.4. Active Armrest . . . 89
4.3.5. Magic Box . . . 92
4.3.6. CapTap . . . 95
4.4. Other capacitive prototypes . . . 103
4.5. Capacitive prototypes from related work . . . 107
4.5.1. Indoor localization . . . 107
4.5.2. Smart Appliances . . . 108
4.5.3. Physiological sensing . . . 108
4.5.4. Gestural interaction . . . 108

5. Evaluating capacitive proximity sensors in smart environments **111**
5.1. Comparison to other sensing technologies . . . 111
5.2. Limitations . . . 112
5.2.1. Environmental Influence . . . 113
5.2.2. Physical Range . . . 113
5.2.3. Object Detection . . . 114
5.3. Benefits . . . 115
5.3.1. Versatility . . . 115
5.3.2. Unobtrusiveness . . . 115
5.3.3. Processing Complexity . . . 116
5.4. Guidelines . . . 117

6. Conclusions and Future Work **119**

A. Evaluation results **123**

A.1. Capacitive chair posture recognition - manual classifier . . . 123
A.1.1. Evaluation setup . . . 123
A.1.2. Raw results . . . 123
A.1.3. Postures . . . 124
A.2. Capacitive chair posture recognition - SVM classifier . . . 124
A.3. Capacitive Chair - working situation recognition . . . 124
A.4. CapTap evaluation . . . 124
A.4.1. Questionnaire . . . 124
A.4.2. Raw results . . . 125
A.5. Active Armrest evaluation results . . . 128
A.6. CapFloor @ EvAAL 2012 . . . 128
A.7. Smart Bed sleep phase recognition results . . . 128

B. Publications **129**

C. Supervising Activities **133**
C.1. Diploma and Master Thesis . . . 133
C.2. Bachelor Thesis . . . 133

D. Curriculum Vitae **135**

Bibliography **137**

List of Figures

1.1. Sketch of electrode placement of all capacitive sensing prototypes created in the scope of this work 1

2.1. Black box setup of a capacitive proximity sensor 7
2.2. Capacitive sensing procedure 8
2.3. Different projected capacitive sensing methods based on distance 9
2.4. Three measurement modes for capacitive proximity sensing [Smi96] 10
2.5. Spatial resolution of different materials at various distances [GPBB*13] 11
2.6. Examples of multilayer layouts for touch screens - grid (a), interlocking diamonds (b) and trademarked complex patterns (c) [BO10a] 11
2.7. Electrode placement below upholstery [GPMB11] 12
2.8. Abstracted sensor data processing pipeline 13
2.9. Example of baseline reset using a threshold rule 14
2.10. Generic pipeline of probability based methods of capacitive proximity sensing 16
2.11. *Left:* Leon Theremin playing his epnoymous electronic musical instrument [Gli00]. *Right:* The Theremini by Moog Music Inc., released in 2014 [Inc14] 18
2.12. *Left:* Concept view of passenger seat set to deploy or not deploy airbag. *Center:* Sensor readings for empty seat and adult person. *Right:* Sensor readings for front-facing child seat (FFCS) and rear-facing child seat (RFCS). [?] 19
2.13. *Left:* LaZmouse innards *Center:* Joshua R. Smith using LaZmouse [?] *Right:* Novint Falcon 3D input device [Nov14] 19
2.14. *Left:* Person interacting with the gesture wall *Right:* Air drawing results, depth estimation results and associated movements on bottom. [SWD*98] 20
2.15. *Left:* Prototype of CapToolKit [WKBS07] *Right:* Thracker prototype and visualized grasping gestures. [WHKS06] 21
2.16. *Left:* Prototype of HandSense and supported grasping types [WB09]. *Right:* Setup of time domain reflectometry sensing and example of stretchable material [WB11]. 22
2.17. *Top:* Capacitive profiles of different gestures on a door knob [SPH12]. *Bottom Left:* User identification using capacitive fingerprinting [HSP12]. *Bottom Right:* Botanicus Interacticus prototype and interaction concept [PSLS12] 23
2.18. *Top:* Capacitive profiles of different gestures on a door knob [SPH12]. *Bottom Left:* User identification using capacitive fingerprinting [HSP12]. *Bottom Right:* Botanicus Interacticus prototype and interaction concept [PSLS12] 24
2.19. A bayer pattern on a sensor in isometric perspective [Com06] 26
2.20. *Left*: Tracking of body masks using cameras [TKA06]. *Right*: Eigenfaces created from input picture set [PC00] 27
2.21. *Left*: Eulerian Video Magnification to attenuate the human pulse with original (a) and amplified (b) video sequence [WRS*12]. *Right*: FoodBoard schematics (top), underside view (second row), original, reconstructed and segmented image (third row) and final system (bottom) [PJS*13] 27

2.22. *Top Left*: ORL Active Badge [Wei91]. *Top Right*: Kinect infrared projection [Zha12]. *Bottom Left*: Kinect Fusion reconstrucion [IKH11]. *Bottom Right*: Kinect kitchen interaction [Pan12] . . 28
2.23. *Top Left*: Cricket Indoor Localization hardware [PCB00]. *Top Right*: Mid-air particle manipulation [OHR13]. *Bottom Left*: Robotic wheelchair MAid with ultrasound range sensors [PSF01]. *Bottom Right*: Activity and context recognition using ultrasound sensors [WTT13] 30
2.24. *Top Left*: Visualization of speaker count in different areas [XLL*13]. *Top Right*: Directional microphone and conference room for speech source localization [ZFBZ08]. *Bottom*: TapSense detecting different tap events [HSH11]. 31
2.25. *Top left* ZigBee node. *Bottom Left* Sensors and targets in larger room [SKOM06]. *Top right* Photo of radio tomography setup. *Bottom right* Result attenuated signal [WP10]. 33
2.26. *Top left* WiTrack antennas and signal generator. *Bottom Left* WiTrack 2D and 3D localization method [AKKM13]. *Top right* WiSee supported gestures. *Bottom right* WiSee Doppler profiles of gestures [PGGP13]. 34
2.27. *Left:* Coverage of a room by camera fields of view - AmbiTrack [BDA*13]. *Middle:* Fiduciary marker photographed by camera phones [MWBS09]. *Right:* Magnetic coils in background and receiver circuit in foreground [PL13] 36
2.28. *Left:* FTIR multitouch table [Han05]. *Middle:* DataGlove hand gesture system [ZLB*87]. *Right:* XWand interaction device [WS03] 38
2.29. *Left:* MACH conversation coach system [HCM*13]. *Right:* Study setup to collect emotion data [KMS*12] 40
2.30. *Left:* Eye movement tracker by Bulling [BWGT11]. *Right:* Hon4D - surface normals of different activities [OL13] 41
2.31. *Left:* Different MediaCup prototype [GBK99]. *Right:* Stickear system to augment appliances with audio sensing [YNR13] 42
2.32. *Left:* TiltRacer controlled by accelerometers of mobile phones [VCBE07]. *Right:* PECo environment control application on a PDA [SN04] 43
2.33. *Left:* Scenario of the PEIS system [BGS*06]. *Right:* Robots moving around in a social environment [GKIH09] 44

3.1. Benchmarking process 49

4.1. Piecewise linear hand distance estimation [BH11] 63
4.2. Principle components of a learning by example recognition framework [BDK13] 64
4.3. Wire electrode grid below floor cover attached to sensors on the border 65
4.4. Two potential person locations resulting in same sensor readings (green indicates active electrodes 65
4.5. Shapes of a standing and lying person on top of the CapFloor grid 66
4.6. Cylindrical human body model and various poses on mattress 67
4.7. Pressure distribution of a uniform cylinder 67
4.8. Data processing components [BH12] 68
4.9. Calculating centers of pressures and deviation [BH12] 68
4.10. Screenshot of the Capacitive Chair application showing the fitted 3D model on the left, posture detection on the upper right and the recognized posture on the lower right 69
4.11. Screenshot of the Capacitive Chair application showing the fitted 3D model on the left, posture detection on the upper right and the recognized posture on the lower right 70
4.12. Data processing pipeline of Active Armrest 70
4.13. Suspended peg knock detection system for CapTap [BF14] 71
4.14. Pixel array mapped from sensor values 72

4.15. Effect of different upscaling methods on shape, (A) nearest neighbor, (B) bicubic, (C) bilinear, (D) Lanczos4 - shown as thresholded binary images (pixel intensity > 30) 73
4.16. Overhead camera picture of the scene overlaid with live arm and palm reconstruction for one arm (left) and two arms (right) 74
4.17. Example of human sleep phases throughout the night 75
4.18. CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides 79
4.19. Floor mats with integrated CapFloor system used at the EvAAL 2011 competition [BHW12] . . 80
4.20. Smart Bed sketch - flexible plate electrode are attached on spring board 81
4.21. Sleep movement data over three hours in one night [DBM14] 82
4.22. Smart office chair sketch - eight electrodes three in backrest, three on seat and two in armrests . 83
4.23. Capacitive Chair electrode positions 84
4.24. Detail view of conductive thread electrode 85
4.25. Example chart of work activity data collected 87
4.26. Disturber position of a participant (left) and sitting upright (right) 88
4.27. Slouching position (left) and sitting hunched (right) 88
4.28. Active armrest sketch - six electrodes for finger gesture detection in front, two for arm detection in back 89
4.29. Active Armrest prototype, left - outside view, right - detail view of electronics 90
4.30. Active Armrest demo software, left - finger tracker, right - OSM based navigation application . . 91
4.31. MagicBox sketch - six electrodes uniformly distributed below surface 92
4.32. MagicBox conceptual rendering (left) and detail view of electronics (right) [BH11] 92
4.33. MagicBox demonstration application - 3D object viewer (left) and image viewer (right) [BH11] 93
4.34. Gesture overview module (left) and gesture recorder (right) [BDK13] 94
4.35. CapTap sketch - 24 electrodes placed under table surface and a single detector for touch events . 95
4.36. Detail views of the prototype system: left - electrodes and sensors, right - audio interface and contact microphone 96
4.37. Abstract view below the surface of the prototype including capacitive sensing electrodes and touch detection microphone 96
4.38. Views of final prototype, complete view (left), top view with markers for touch evaluation (right) 97
4.39. 64 sample FFTs and photo for a knock event (A), a finger tap (B), a finger swipe (C) and a hand swipe (D) 97
4.40. Finger tap (blue), knuckle knock (green), finger swipe (purple), hand swipe (orange) and stomping (red) spots relative to tabletop. 99
4.41. Tracks generated by Kalman filtered palm position, a sine wave (A), a rectangle (B), a circle (C), and a diagonal swipe (D) 100
4.42. Interaction speed evaluation. Different types of boxes for near layer (N), knock (K), far layer (F), and disturber (X). 101
4.43. CapDisp sketch - TV on a stand equipped with capacitive sensors hidden below a plastic cover . 103
4.44. *Top left*Person lying on the couch. *Top right*Resulting sensor value visualization. *Bottom left*Rendering of recongized posture. *Bottom right*Position of electrodes within the couch. 104
4.45. *Top left* Rendering of Honeyfish device and hands. *Top right* Image of developed GUI and pointer. *Bottom left* Swiss cheese algorithm predicting objects in interaction space. *Bottom right* Evaluation of Honeyfish at a student fair. 105
4.46. CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides 106
4.47. GestIC® Sabrewing prototyping board [Mic13] 109

5.1. Reduced angular resolution on smaller, distant objects 113
5.2. Same response to differently sized objects (left), different response to varying materials (right) . 114
5.3. Electrodes and sensors hidden below mattress of Smart Bed 116

A.1. *Top left* upright posture. *Top right* hunched posture. *Bottom left* slouched posture. *Bottom right* disturber posture 124

List of Tables

2.1. Baseline calibrations terms and methods . . . 14
2.2. Overview of high-level processing methods for capacitive proximity sensors . . . 15
2.3. Feature recognition methods . . . 17
2.4. Sensors for smart environments [CD07] . . . 25
2.5. Sensing technologies and measured properties . . . 26
2.6. Application domains and relevant works . . . 37

3.1. Feature matrix denoting capabilities required for a certain rating . . . 48
3.2. The importance weighting of different applications, based on the features. . . . 52
3.3. Feature rating of the different sensor technologies . . . 52
3.4. Regular and normalized benchmark score matrix of different applications and technologies . . . 53
3.5. Search result frequency given specific applications, sensor technologies and synonyms for ACM Digital Library (DL) and Google Scholar (GS) . . . 53
3.6. Central tendency bias correction for different exponents a . . . 55
3.7. Importance weights of applications for capacitive proximity sensors . . . 55
3.8. Feature weights for capacitive proximity sensors . . . 57
3.9. Benchmarking scores for capacitive proximity sensors in different applications . . . 58

4.1. Application domains and derived implemented use cases for capacitive proximity sensing . . . 59
4.2. Challenges associated to the different use cases for capacitive proximity sensors . . . 62
4.3. Overview of developed capacitive proximity sensing prototypes . . . 78
4.4. Results of touch detection for single and double taps (SFT, DFT), knocks (SKK, DKK), finger swipe (FS), hand swipe (HS) and stomp (STO). Noted are the overall samples, errors, no event errors, wrong classification errors and the percentage of correct classification. . . . 99
4.5. Results for interaction time in the different runs of the interaction speed test . . . 101
4.6. Measuring layout and data processing of different prototypes from related works . . . 107

5.1. Comparison between capacitive proximity sensors and other senor technologies . . . 111
5.2. Overview of capacitive proximity sensing limitations . . . 112
5.3. Overview of capacitive proximity sensing benefits . . . 115

A.1. Percentage of correctly classified postures using manually set classifier . . . 123
A.2. CapTap evaluation raw results . . . 126
A.3. CapTap questionnaire results . . . 127

1. Introduction

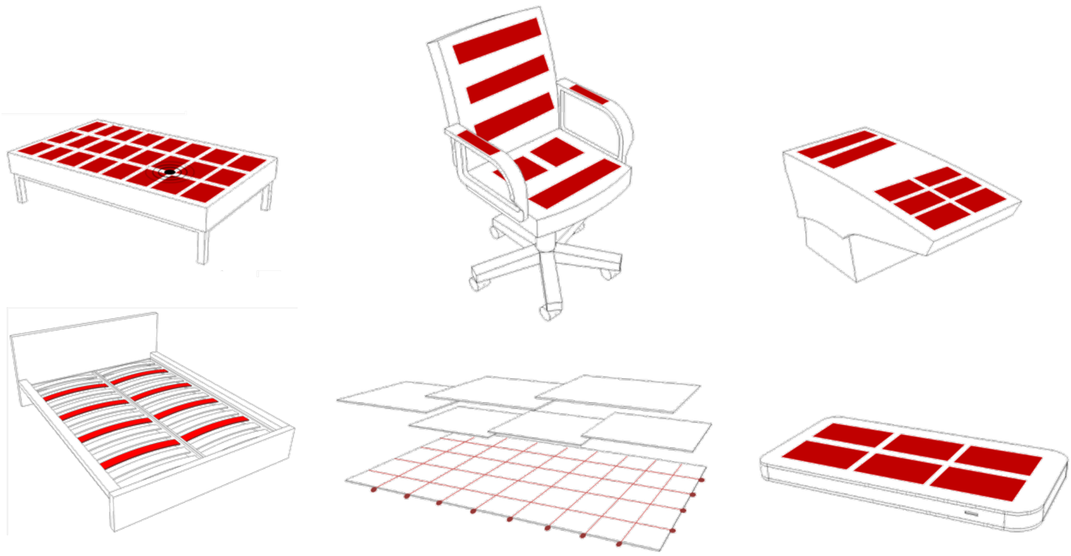


Figure 1.1.: Sketch of electrode placement of all capacitive sensing prototypes created in the scope of this work

Smart environments are comprised of numerous sensing and computing devices that are supporting a number of users in this environment on performing their tasks. Driven by advances in computing power, miniaturization of sensors and processing methods, novel devices including a plethora of functionality have been introduced into our everyday lives. In science the field has been thriving in the last decades, combining knowledge from disciplines including computer science, engineering, but also product design, in order to create systems that are integrated into the environment, have a high usability and provide information and services to the actors in smart environments. Perhaps the most cited example of this trend is the rise of the smartphone, from professional business tool to a consumer device being sold hundreds of millions times a year. Using integrated sensors and communication facilities it is possible to provide services aware of location, schedule, contacts, or preferences, that realize navigation, event planning, augmented reality, or entertainment. A different example are increasingly networked homes that are aware of energy usage, lighting levels, temperature and the status of critical devices and can be controlled by the user from a single place or autonomously using a set of specified rules.

A common aspect of all smart environments and smart devices is sensing. This includes environmental parameters but also system state and most importantly the activities of the different users. There are numerous categories of sensing devices that can realize different aspects of this sensing, ranging from cameras or accelerometers to GPS and acoustic sensors. Capacitive sensors are a category of sensors that use electric fields to sense the presence and certain properties of the human body. The most common variety is sensing the presence of fingers on touch screens, already present in billions of devices. However, there is another variety, the capacitive proximity sensor that is able to detect the presence of human body parts over a distance, providing interesting applications in smart environments, by unobtrusively integrating the sensors into different materials, environments and appliances. When designing an application of smart environments choosing the right sensors is one of the most relevant decisions that has to be taken early in the process. Even though there are numerous prototypes available, so far this process has been mostly supported by looking at previous activities.

In this work I present a benchmarking model that can support this decision process in the domain of smart environments, in order to determine relevant application areas for capacitive proximity sensors. There are numerous challenges associated to each area, regarding the details of the application. I will present a collection of existing and novel methods that support processing data generated by capacitive proximity sensors. Several prototypes created using those methods have been implemented and evaluated for performance and usability. Figure 1.1 shows sketches of the different prototypes and the placement of the electrodes attached to the capacitive proximity sensors. Based on this evaluation and the knowledge generated in the design process I am able to discuss the benefits and limitations of the technology, classify it with regards to competing technology, in order to present a set of guidelines that can aid parties interested in designing smart environment applications using capacitive proximity sensors.

1.1. Motivation

In the last decade the way we interact with computing machines has changed in a profound fashion. Today more than one billion people operate a smartphone, enabling ubiquitous access to communication tools, processing power and information. The vision of ubiquitous computing as proposed by Mark Weiser in the early 90s is inching closer to reality [Wei91]. The required technologies of

> "cheap, low-power computers that include equally convenient displays, a network that ties them all together, and software systems implementing ubiquitous applications"

are now existing in the form of smartphones and tablets that are connected to the internet, using high-speed connections such as LTE and web-based services such as Google Now, that combine numerous data sources to provide personalized services.

While the vision and underlying ideas remain similar other names have been used in research, including Pervasive Computing and Ambient Intelligence. The concept has been expanded to not only consider devices that can be directly manipulated, but include determining the situation and reacting based on it. This context-aware computing proposes

> "systems that examine and react to an individual's changing context. Such systems can promote and mediate people's interactions with devices, computers, and other people" [SAW94]

Different forms of context can be distinguished, ranging from location and the actual system state, to different activities or even the current mood of the user. In order to acquire this context, the input-and-output based systems originally proposed by Weiser, are augmented by an ensemble of devices that are very small (dust), coordinate in massive numbers (clay) or are flexible, unobtrusive extensions to everyday objects (fabric) [Pos11]. This devices can be invisibly integrated into our everyday environment and provide sensing capabilities that can be used by sufficiently smart systems. Examples of these devices are microelectromechanical systems (MEMS) or bendable technology, such as OLED screens. The number of computation and sensing devices that we carry with us is growing continuously, yet we want the technology to further disappear, allowing us to focus on the application instead of the underlying technology.

The famous science fiction author Arthur C. Clarke proposed three laws of prediction, the third of which is

> "Any sufficiently advanced technology is indistinguishable from magic." [Cla62]

Capacitive proximity sensing allows us to measure the influence of the human body (or conductive objects in general) on an electric field. While this technology is not magic per se, a peculiarity of electricity is that humans, as opposed to some animals, have no specific sensing organs for this property. Thus we remain unaware of their presence, unless the field strength is very high. Consequently, when interacting with capacitive sensors there is no awareness of what they are sensing unless it is specifically exposed to another sense of the user, such as haptic feedback on touch screens or visual feedback on touch less systems. Touch screens are the most ubiquitous application of capacitive technology, being applied in all modern smartphones and tablets, thus being used by a large number of the world's population every day. However, they are typically tuned to only register touches, while other varieties are also able to detect objects over a distance. This enables numerous other applications for this technology, ranging from industrial fluid level and material detection, to presence detection in cars. A particularly interesting domain for this sensing technology are smart environments that provide services based on unobtrusively acquired information about persons currently acting in this environment. A popular capacitive proximity sensors have been primarily used for human-computer interaction (HCI) applications, including a mouse tracking the distance of the heel of the hand or a monitor that is able to track gestures performed in front of it. Another area are smart appliances, such as an object detecting car seat or localization systems. There are numerous sensing technologies that provide similar detection capabilities. Looking at the recognition of simple activities, such as standing, walking and lying, cameras and accelerometers can lead to the same result. Thus, it is often difficult to decide what the specific sensor technology to use in a specific system. Commonly one refers to previous work and best practice, building on previously generated knowledge. However, so far there is no formal model that would allow to quickly evaluate different sensor technologies in different applications. Taking into account a specific set of features required for a specific application domain this could be an important decision support tool in the early stages of system development. As it was stated by Cook and Das [CD07]:

> "Finally, a useful goal for the smart environment research community is to define evaluation mechanisms. While performance measures can be defined for each technology within the architecture hierarchy [...], performance measures for entire smart environments still need to be established. This can form the basis of comparative assessments and identify areas that need further investigation."

Such a tool can also be used to identify specific applications for a single sensor technology, such as verifying current and developing new use cases for capacitive proximity sensors or providing a classification of the technology with regards to competing sensor systems. However, identifying a suitable sensor is just the first step in designing a smart environment prototype or product. After this decision a designer has to determine specific challenges of the domain, select suitable methods for applying the technology and processing the data and create a working system. In this design step it is helpful to have a set of methods, examples and guidelines, leading to a more rapid prototyping for researchers and shorter time-to-market for product developers. These guidelines should be the result of literature review and validating prototypes for performance and usability.

1.2. Research Challenges

In the past there have been numerous influential works that gave an overview of technologies and applications in smart environments. Cook et al. identified common technologies, frameworks and applications in this domain and give an overview of ongoing research [CD07]. Poslad specified a more detailed taxonomy of device classes, provides concepts for interaction between humans and environments and gives an overview of intelligent systems [Pos11]. A different category of previous work details the different sensing technologies that are supporting various different applications and give an overview of limitations and benefits. However, so far there has been no work that provides a benchmark that maps different sensor characteristics to applications in smart environments. An intermediate step between evaluating entire environments and low-level technologies is an application-specific benchmarking of systems. Benchmarking as a method allows us to quantify the performance of a specific process or item and allows a comparison to competing processes or items. It is common to benchmark different technologies according to their features. My proposal is to extend technology-driven benchmarks by adding an application-specific feature weighting. This approach allows to map the same set of features to different applications that have similar requirements that are catered to by divergent technologies. It will be verified by benchmarking typical applications with regards to several example applications in the domain of smart environments.

The selection of application scenarios for capacitive proximity sensors is mostly based on previous works, most notably by research groups from MIT [PLCH02], Disney research [SPH12] or the Munich University of Technology [WHKS06]. They are extending on the methods or modify existing use cases to another domain. Using the developed benchmarking method it is possible to verify the existing application areas or even determine new ones. It also allows to specify if other sensing technologies might be more suitable for a specific application. This allows to identify four relevant application domains that can be realized with capacitive proximity sensors.

Following the design process a next step is specifying the particular challenges for the given application, selecting suitable processing methods, in order to create and evaluate a system. Looking at the related works several areas are identified that can be improved using novel or adapted methods. E.g. previous systems often rely on uniform sensor arrays [Smi96] or require a large number of sensors [Rek02]. Improvements are proposed for object tracking using a low sensor count, unobtrusive localization in large areas, model-driven approaches for object fitting and heterogeneous sensor systems , fusing different geometries or sensor categories. The presented methods are realized in different prototypes that are evaluated for usability and performance. Using the knowledge generated from determining applications, finding specific challenges and creating different prototypes I am able to classify capacitive proximity sensors with regards to other sensing technologies in the domain of smart environments. It is possible to identify specific benefits and limitations and compare features between technologies. This culminates in creating a set of guidelines for parties that are interested in developing smart environment systems based on capacitive proximity sensors.

In addition to the work on capacitive proximity sensors I have also worked on various other aspects regarding interaction in smart environments and indoor localization. Pointing at devices in order to control them is an intuitive way of interaction, often unconsciously performed when switching TV stations with an infrared remote, even though it is usually not required. However, only a limited number of devices have the required facilities for this kind of interaction since it does require attaching transceivers and often results in the necessity to use multiple remote controls. I propose a system giving a user the ability to intuitively control arbitrary devices in smart environments by identifying the appliance an interaction device is pointed at and providing means to manipulate these. The system is based on identifying the position and orientation of said interaction device, registering these values to a virtual representation of the physical environment, which is used to identify the selected appliance.

Indoor localization is a base technology within smart environments, enabling a multitude of applications including augmented reality, navigation in large indoor areas, or sports [TCD*00,IHQ04,LBO11]. Some particular challenges in assisted living applications include low budgets, easy installation, privacy preservation and interoperability with other systems in the environment [CK12]. Camera-based systems are very popular in this domain, yet particularly in private settings struggle with user acceptance. A potential solution are smart camera systems that process and abstract the images before they are sent to the network, however they require efficient and robust tracking algorithms for implementation on embedded systems.

1.3. Contributions

In the following I will list briefly and concisely what are the specific contributions provided by this work on a methodological and practical level. They are distinguished into three different groups, the benchmarking model, the validation of capacitive proximity sensors and additional smart environment research topics.

I will introduce a generic and formal benchmarking model for sensor systems in smart environments. This includes the identification of relevant sensor features that should be included in the scoring process, leading to a feature matrix that links features to ratings. Based on an importance-based weighting process a benchmark score calculation is presented, including methods to normalize the score based on average feature scores. This benchmarking model is used to identify different use cases for capacitive proximity sensors in the domain of smart environments.

In order to validate the sensor technology different challenges are determined for each use case, allowing to identify processing methods that can be beneficial. Processing methods in five different groups are presented. Two methods for sparse sensor configurations that enable tracking hand gestures in three dimensions and localization and fall detection in large areas. Several model-driven fitting methods applying machine learning on single- and multi-body models. Methods for processing in heterogeneous sensor systems comprised of non-uniform array configurations, parallel processing of single data streams or fusion of different sensor systems. Considerations regarding image-based processing for loading mode sensors in uniform array configurations. Several methods to process physiological signals using different sensing modes in frequency- and time-domain.

Six different prototypes are detailed and evaluated. The MagicBox prototype, enabling expressive single-hand gestural interaction with sparse sensor distribution and machine learning gesture classification. The CapFloor system, using a novel layout for floor-based capacitive indoor localization systems, enabling unobtrusive application, easy maintenance and additional services such as fall detection. The SmartBed prototype using a model-based approach for fitting one or two persons, concurrently detecting sleep phases and breathing rate for occupants. The Capacitive Chair smart furniture that allows to detect presence, identify users, track different postures, measure breathing rate and enables novel applications for smart offices. The Active Armrest using a heterogeneous sensor layout to enable different forms of interaction in automotive environments. The CapTap prototype combining capacitive sensors and microphones in a table-based interaction device, enabling multi-hand gesture recognition in three dimensions using a multi-level interaction pattern. Three other prototypes, CapDisp, HoneyFish and GestDisp, that were created in collaborative efforts are described briefly.

The last group is comprised of other topics in smart environments focus on presenting a marker-free interaction paradigm for pointing-interaction with arbitrary devices in smart environments, showing multi-modal use cases and different bounding volume modification techniques in the virtual realm. The second system is AmbiTrack - a camera-based indoor localization system for smart environments, initially designed to calibrate and evaluate capacitive systems.

1.4. Structure of this work

After having identified the research challenges and introduced the topic the related works are specified in Section 2 - *Related Work* - in four categories. The first section gives a background on electric field sensing, including relevant historical work and the physical properties. Additional different sensing categories are outlined, before different electrode considerations and data processing methods are introduced. The second category of related works discusses different applications of capacitive proximity sensors that were created in the last decades, ranging from MIT research in the early 90s, to novel touch classifiers based on different sensing methods. The third category introduces different competing technologies that will be used in the later benchmarking. Finally we give an overview of existing work collecting and grouping applications in smart environments. This will allow us to identify candidate scenarios for capacitive proximity sensors.

Section 3 - *Benchmarking model for sensors in smart environments* - is concerned with the first main contribution of this work, the introduction of an application-specific benchmarking model for sensors in smart environments. In the first part of this section the sensor features relevant for application in smart environments are discussed. Suitable features are discussed in three different categories, discussing the rationale for inclusion or omission in the model. The next part describes the benchmarking model. The application-based feature weighting is introduced, leading to the derivation of the model itself, including the required calculation of an overall rating and a feature score normalization. After that we are using the model to score different examples and validate those using search results from scientific publication databases. Afterwards, the model is discussed and used to identify suitable applications for capacitive proximity sensors in smart environments.

Section 4 - *Use cases for capacitive proximity sensors* - describes the identified use cases for capacitive proximity sensors in smart environments. First, the use cases and associated challenges for design and processing are identified. Afterwards different processing methods for capacitive proximity sensors are presented that tackle the specific challenges. This includes methods for sparsely distributed sensor arrays, model-based data fitting, heterogeneous sensor systems, image-based processing and physiological signal processing. Six different prototypes that implement one or more of the processing methods are presented and evaluated - MagicBox, CapFloor, Capacitive Chair, Active Armrest, SmartBed and CapTap. Each of the prototypes has been evaluated for performance and usability. Additionally, three other prototypes are discussed briefly.

The knowledge gathered in designing, building and testing the prototypes and using the benchmarking model leads to Section 5 - *Evaluating capacitive proximity sensors in smart environments*, wherein the results are discussed and evaluated. This section has four parts. At first capacitive proximity sensors are compared to the other sensor classes introduced in the related works. Afterwards limitations and benefits of the technology are collected and linked to the sensor features and applications. The section concludes with a set of guidelines that may help interested parties in evaluating their application for usage with capacitive proximity sensors and give practical help when applying this technology.

The document concludes in Section 6 - *Conclusions and Future Work* - that briefly recapitulates the work and introduces potential future research stemming from this work.

There are x different appendices. Appendix A lists publications and talks. Appendix B lists Master and Bachelor Thesis that were supervised or co-supervised. Appendix C contains a short CV.

2. Related Work

In this section I will describe the most relevant works that inspired this work or are linked to a specific topic. The aim of this section is to provide a basis for both, the benchmarking model that is developed in section 3, and the capacitive proximity sensing prototypes described in section 4. The related works are distinguished into four distinct parts. At first I will give a general introduction to electric field sensing, including a discussion on different properties, physical background, the influence of materials and geometry and different data processing methods. Afterwards I will present relevant applications using capacitive proximity sensing, ranging from historical works to very recent systems. In the next section various sensing technologies are introduced that are used in smart environment systems. Finally I will identify and group different applications in smart environments, providing a basis for the benchmarking model.

2.1. Electric field sensing

Different electric charges apply either a repelling or attracting force to each other. For any point in space these forces have a distinct direction and magnitude. The resulting collection of force vectors is called the electric field. Conductive objects that are present in this area modify the properties of the field. Electric field sensing enables measuring field properties at a certain point in space. Using continuous monitoring it is possible to gather information about conductive objects passing through the field by associating measured disturbances to properties of the object. It is possible to gather a multitude of different information about a project. In this section I will give an overview of the physical background, different measurement modes and how to process data acquired by digital sensors.

2.1.1. Physical properties

A complete overview about the electrostatic principles of capacitive proximity sensing can be found in the book by Baxter [Bax96], chapters 2 and 6. We will give a very brief introduction to this topic in the following section. The basic setup of a typically used sensor is shown in Figure 2.1. The proximity capacitance C_x can be determined

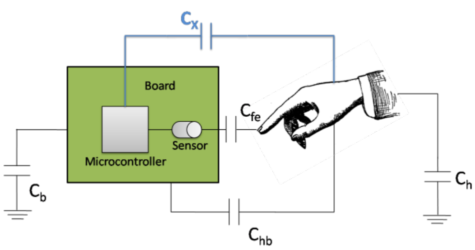


Figure 2.1.: Black box setup of a capacitive proximity sensor

using a combination of serial and parallel circuits of capacitors, resulting in the following equation:

$$C_x = \left(\left(C_{hb} + \frac{C_h C_b}{C_h + C_b} \right)^{-1} \frac{1}{C_{fe}} \right)^{-1} \tag{2.1}$$

Additionally there are parasitic capacitance components, i.e. disturbing capacitance values within the system. Sources are:

- Sensing electrode capacitance
- Capacitance between sensing electrode and ground plane
- Intercapacitance between neighboring traces on the board

The present parasitic capacitances C_{par} amount to values approximately between $10pF$ and $300pF$ and are therefore considerably larger than the value of the proximity capacitance Cx, being between $0.1pF$ and $10pF$. The total capacitance sensed is the sum of parasitic and proximity components.

$$C_S = C_X + C_{par} \tag{2.2}$$

It is obvious that this parasitic capacitance is considerably higher than the capacitance induced by an approaching object. However, this parasitic capacitance is typically static and can therefore be calibrated in a way not affecting the measurement. Now we will shortly discuss how we can estimate the capacitance of common ob-

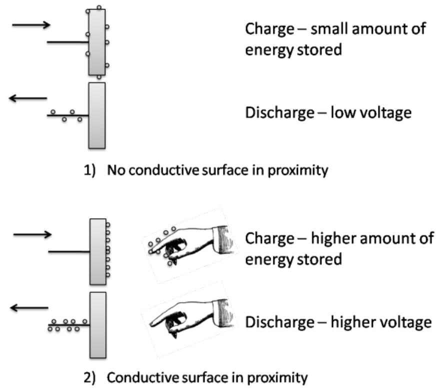


Figure 2.2.: Capacitive sensing procedure

jects that approach the sensor. Any object exhibits capacitance in respect to infinity. Surveying simple geometric shapes this capacitance is analytically determinable, e.g.:

$$C = 8\varepsilon_0 r_{Disk} \tag{2.3}$$

$$C = 4\pi\varepsilon_0 r_{Sphere} \tag{2.4}$$

ε_0 is the vacuum permittivity and r the respective radius. This free space capacitance is increasing as soon as another object is approaching, caused by the capacitance of this second object, resulting in mutual capacitance. Looking at generic formulas, determining capacitance between parallel plates this behavior can be described analytically.

$$C = \frac{Q}{V} \qquad\qquad C = \varepsilon_0 \varepsilon_r \frac{A}{d} \tag{2.5}$$

The capacitance is directly proportional to the plate area A and inversely proportional to the distance d between the plates, with ε_r being the relative static permittivity of the dielectric between the plates. Sensor electronics are grounded with the body acting as ground itself. The sensor plate is continuously charged using a constant voltage V. A higher capacitance allows the system to hold a larger charge. If the system is connected to the ground, the sensor capacitor is discharged through a resistor. The resulting voltage is depending on the available charge, shown in the equation above. Furthermore the required time to discharge the capacitor is increased. This process is symbolized in Figure 2.2.

2.1.2. Proximity sensing versus touch sensing

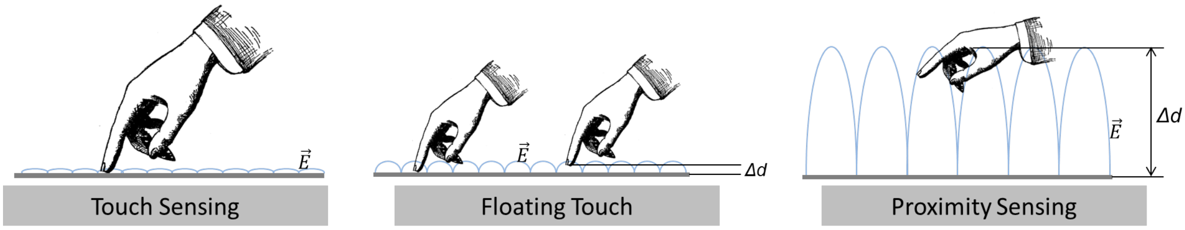


Figure 2.3.: Different projected capacitive sensing methods based on distance

The most ubiquitous usage of capacitive sensing technology can be found in touch screens. As the trend went from pen-controlled mobile systems to finger controlled devices with the first iPhone in 2007, projected capacitance touch is the most prevalent technology for touch screens. It uses various layers of transparent electrodes or nanowires to detect the mutual capacitance as objects enter the detection area [BO10b]. The commercially available devices have gained additional abilities over the last few years, leading to the development of "floating touch" systems that are able to track fingers in gloves or fingers that are hovering above the surface [Cyp12, Nok12]. Applications are the usage of mobile devices in cold outdoor temperatures or additional navigation fea-tures based on the hovering fingers. In consequence we can distinguish the three different projected capacitive sensing methods as shown in Figure 2.3:

- Touch sensing - densely distributed sensors are tuned to project a weak electric field in order to detect one or more objects touching the interactive surface. The sensors have to be close to the surface.
- Floating touch - densely distributed high-sensitivity sensors are able to detect both touches and very near objects ($< 2cm$) to enable usage using protective gear or additional navigation feature. The sensors have to be close to the surface.
- Proximity sensing - sparsely distributed sensors create a stronger electric field that propagates into space in order to detect larger objects, such as hands, that are in proximity of the interactive surface. Achievable distances are up to 30 centimeters and the sensors may be applied below thick non-conductive material.

2.1.3. Measuring modes

A classic work in the field of capacitive proximity sensing that will be referenced occasionally in this work is "Electric Field Imaging" by Joshua Smith [**?**]. One contribution was the introduction of different measurement modes in capacitive sensing, as shown in Figure 2.4. Transmit mode is using a transmitting electrode that is

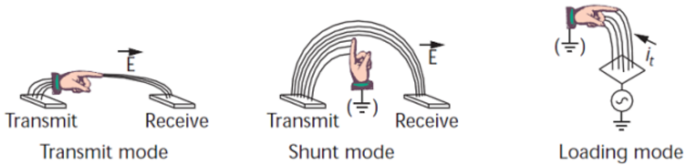


Figure 2.4.: Three measurement modes for capacitive proximity sensing [Smi96]

coupled to a conductive object; in case of interaction applications typically the human body. The properties of an electric field generated with respect to a receiving electrode will therefore be dependent on the distance of this body, thus extending the achievable range. Shunt mode similarly uses both a receiving and transmitting electrode generating a static field. However, there is no body coupled and any conductive object will ground the field, thus reducing the energy stored, which is measured. This setup is able to work with various transmitters on a single receiver, enabling a higher amount of virtual sensors using limited hardware. The third measurement mode is called loading mode. An oscillating field is induced on a single electrode measuring the capacitance relative to the environment. Any approaching grounded object results in an increased capacitance that is measured periodically.

2.1.4. Materials and geometry

Two major factors that have to be considered when designing an application based on capacitive sensors are the materials and geometry of the electrodes performing the measurements. The material of the electrode should be picked according to the desired application, i.e. if the interaction device has a flexible surface, conductive thread could be used, if it is solid and opaque, the application of solid metal electrodes is viable. Additionally there are other options for transparent materials. While we traditionally associate solid metals to antennas and electrodes this view can no longer be upheld. Transparent conductive layers have been in use for decades now, e.g. in car windows or solar technology. They typically rely on metal oxide layers, polymer layers or in recent years carbon nanotubes [MPLK05]. The most common technology for usage in displays is projected capacitive touch that uses a multi-layer design of insulated ITO electrodes that are able to detect the movement of several objects close to the surface [BO10b]. However, they are typically tuned to allow operation within a small distance of $1cm$ or less. However, they are typically tuned to allow operation within a small distance of $1cm$ or less. One recent work was evaluating different types of electrode materials in terms of their spatial resolution at different distances between object and electrode [GPBB*13], focusing on larger distance proximity measurements. They benchmarked both ITO and PEDOT:PSS. The first is a thin layer of indium-titanium-oxide, a highly conductive metal layer that possesses good optical properties. PEDOT:PSS is a conductive polymer that has a lower conductivity and slightly less appealing optical properties. In conclusion they evaluated that while copper has still the best properties, at least ITO can be considered a suitable alternative in applications that require optical clarity, as shown in the achievable spatial resolution given in Figure 2.5. The most common technology for usage in displays is projected capacitive touch that uses a multi-layer design of insulated ITO electrodes that are able to detect the movement of several objects close to the surface [BO10b]. However, they are typically tuned to allow operation within a small distance of 1cm or less. Another area that is strongly influenced by the intended application is the geometry, whereas the electrode is considered the part of the electronics directly attached to the measurement circuit. This may range from simple straight wires or plate electrodes to complex optimized multidimensional structures specifically designed for a single task. Even though it is aimed at touch or near-proximity sensing we will give a short overview of multi-layer designs for touch screens that have been reviewed by Barrett and Omote [BO10a].

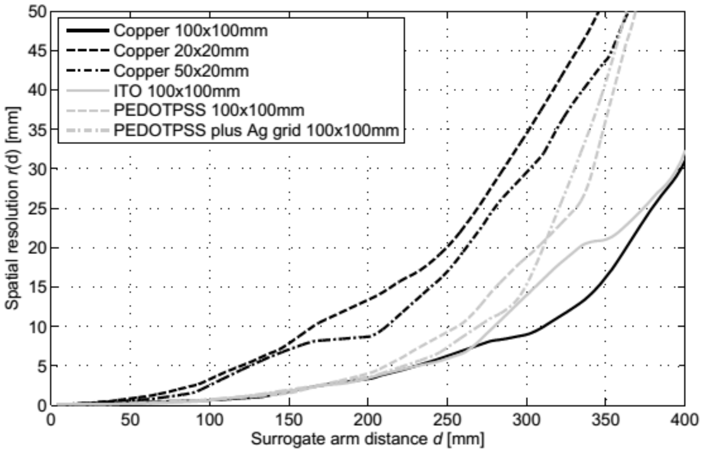


Figure 2.5.: Spatial resolution of different materials at various distances [GPBB*13]

They are designed to measure mutual capacitance, i.e. the resulting capacitive properties between a sending and a receiving electrode that are intersecting. If a sensible excitation and measuring process is used, multiple nearby objects may be reliably detected. A simple example is two layers of perpendicular straight line electrodes - used

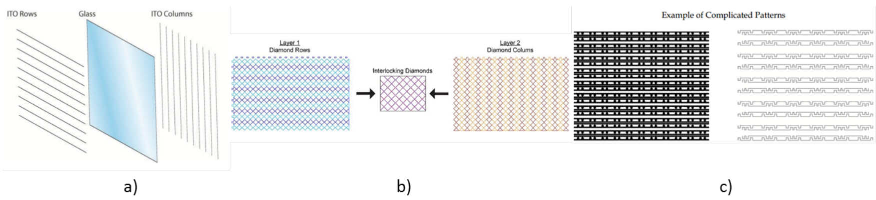


Figure 2.6.: Examples of multilayer layouts for touch screens - grid (a), interlocking diamonds (b) and trademarked complex patterns (c) [BO10a]

by the first iPhone (Figure 2.6 - a). Another example uses an interlocking diamond shape [DL01] to create a good spatial coverage (Figure 2.6 - b). Finally, there are numerous other complex patterns that are often trademarked by the companies that have developed the respective controller. One example is given in (Figure 2.6 - c).

Capacitive proximity sensing applications are typically less concerned about intricate designs, but instead use varying electrode sizes and placement over a larger area. As previously mentioned the purpose of capacitive proximity sensing is the detection of objects and their properties. There are numerous factors that can influence the geometrical layout, but they can be abstracted into the following categories:

- Number of objects
- Object size
- Desired spatial resolution

Going back to our example of touch screens, we have small objects, a higher number of those (usually up to 10) and require a high spatial resolution to select small items on the screen. The result is a fine multilayer grid, using mutual capacitance to simplify multi-object recognition, fine electrode spacing to achieve a high spatial resolution and thin or transparent electrodes to guarantee good optical properties. A similar rationale can be applied to other applications. If we take the smart couch by Große-Puppendahl et al. the aim is to detect the presence and posture of one or more persons on a couch [GPMB11]. This necessitates detecting large body parts such as head, torso or limbs. There is no fine-grained spatial resolution required, allowing a reduction the number of sensors and it was assumed that a maximum of two persons are on the couch. Furthermore the electrodes are placed below the upholstery, thus requiring a reasonable detection distance. The resulting electrode placement can be seen in

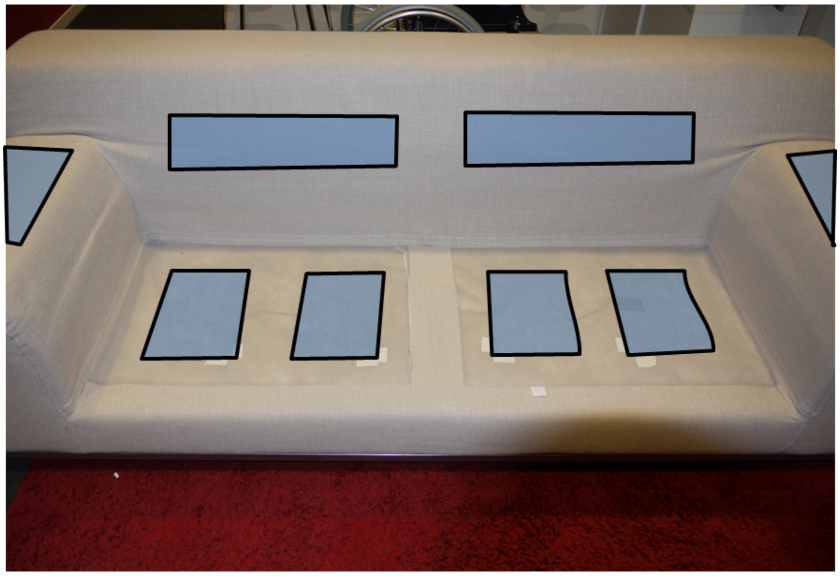


Figure 2.7.: Electrode placement below upholstery [GPMB11]

Figure 2.7. The layout was designed under the additional constriction of using a single sensor kit, supporting up to eight electrodes. Regarding placement it is most important to distinguish two persons and different sitting positions, thus four electrodes are placed below the sitting area. In the back there are two electrodes spread over the entire width to determine the presence of the upper body close to the backrest. The electrodes in the armrests determine a head and are primarily suitable for detecting lying positions. In consequence this setup is suitable for detecting multiple sitting persons, infer information about their sitting position and recognize lying persons. Regarding those postures it showed good results in the prototype's evaluation [GPMB11].

A third and final example for the rationale of electrode placement is the TileTrack system by Valtonen et al, a capacitive person tracking system using floor tiles [VMV09]. It is a transmit mode system that has the transmitting electrodes placed below the floor tiles and the receiving electrodes are placed in the walls of the area. The main goal of the system is the tracking of persons on the surface. Thus the floor area should be mostly covered by electrodes to establish a good transmission link to the bodies. The receiving electrodes should be able to pick up all signals generated by the body. Valtonen et al. picked wire or plate electrodes that went from floor level to a height of 190cm that covers most typical body sizes. While the system has some shortcomings with regard to applicability in larger rooms, the design rationale is appropriate for narrow rooms or when only movement close to walls has to be detected and had a reasonable precision in their evaluation. Looking at the above examples it becomes apparent that the proper selection of materials and geometry is highly depending of the desired application. In consequence it is difficult to give generic guidelines independent of the application. After reviewing the different application domains in the next section we will revisit this topic in section 5.4.

2.1.5. Data processing

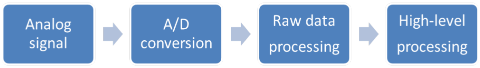


Figure 2.8.: Abstracted sensor data processing pipeline

In order to acquire usable data from any digital sensor an analog signal has to be acquired and processed. A simplified typical processing pipeline for this is shown in Figure 2.8. This basic structure is also applicable to the processing of capacitive proximity sensor data. The analog signal is the capacitance of an electric circuit that can be digitized using different methods, e.g. by using the quantized discharge time of the circuit. In the following section some typical steps of raw data processing and high-level processing for capacitive proximity sensors are presented and discussed.

2.1.5.1. Raw data processing

Raw data processing of capacitive proximity sensor data is primarily intended to compensate for sensor noise and environmental influences. Noise is an inherent property of any measurement system and describes random unwanted data that is added to a signal. Environmental parameters can have strong influence on the signal of a capacitive sensor system. These effecting factors include temperature, humidity, composition of the air, or grounded objects in close proximity. There are numerous additional preprocessing steps that can be taken, such as different multiplexing methods that may be required in some hardware settings, or signal quantization that reduces the outgoing data to a distinct set of values in order to simplify post processing of different applications. These will not be further discussed in the scope of this work.

Noise Reduction In order to deal with noise, some sort of filtering is typically applied. Filtering describes a set of methods that attenuate the parts of a signal that are relevant in a given application. In capacitive proximity sensing we are dealing mostly with high-frequency noise that is added to the signal. Therefore, low-pass filtering can be used to deal with this influence. The most typical examples are average filters that take various samples and calculate an average value, and median filters that are sorting a set of samples and select the median element. Each of those filters has a plethora of potential adaptations that are not too specific to discuss in this limited space. Some adaptations are discussed in the specific prototype sections.

Table 2.1.: Baseline calibrations terms and methods

Name	Description	Application
Initial calibration	First set-up of baseline at system start, e.g. by taking the average over various samples	Required for any application
Static baseline	Baseline that does not change at run-time	For static environments
Dynamic baseline	Baseline that changes over time	For non-static environments
Drift	Change of system response to environmental factors at run-time	-
Drift compensation	Methods to account for occurring drift, by changing the baseline value	Non-static applications
Recalibration	Change of the baseline value at a specific point in time given a set of rules	Non-static applications

Baseline Calibration A very important aspect of capacitive raw data processing is signal calibration. The generated electric field is subject to changes over time, if either intrinsic parameters change or the environment is modified. Some specific examples include the electronic components heating up, the environmental temperature changing, or objects being moved in and out of detection range. Therefore it is essential to have a well-calibrated and adaptive baseline; that is the sensor signal generated in the environment without the presence of any object that we want to detect. Again, there are numerous methods to adapt and configure the baseline. We have collected a few common terms and methods and give some pointers regarding their application. The results are shown in Table 2.1. If a dynamic baseline is used, a set of rules will have to be defined that determines at which points

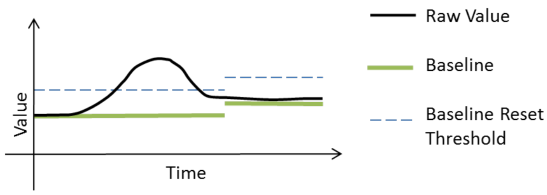


Figure 2.9.: Example of baseline reset using a threshold rule

in time the baseline has to be recalibrated, what specific methods should be used and the set of parameters that control the methods. One simple example is to define a threshold level that triggers a baseline calibration, as shown in Figure 2.9. The raw signal is above the threshold, indicating the presence of a detectable object. Afterwards, it falls back down below the threshold, yet stays for a certain time above the baseline. This triggers a reset of the baseline after a certain amount of time.

2.1.5.2. High-level processing

High-level processing assumes that we already have calibrated (and possibly normalized) sensor values that are used in further steps. The goal of any capacitive sensing application is the acquisition of information about a detectable object, e.g. its current position, the material used or the shape. In order to get this information we need to use knowledge about the object and intrinsic properties of the sensor system. In this section we will discuss methods to combine data from various sensors using the system properties, how to track the position of an object using different methods and how to recognize specific features. An overview of the methods in abridged form is given in Table 2.2.

Table 2.2.: Overview of high-level processing methods for capacitive proximity sensors

Name	Description
Sensor data fusion	Combining sensor data into a shared representational format
Uniform fusion	Sensor data fusion that combines all data into a single common format
Heterogeneous fusion	Sensor data fusion that combines groups of data to serve multiple purposes
Object tracking	Continuous identification of an object within the systems range
Single object tracking	Methods to realize object tracking for a single detectable object
Multiple object tracking	Methods to realize object tracking for multiple objects
Feature recognition	Identifying certain parameters of an object within the system range

Sensor data fusion Sensor data fusion in its most general terms describes "the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format" [Mit07]. Using the combined information from various capacitive proximity sensors we are able to generate high-level information that exceeds the capabilities of a single sensor. We can distinguish uniform fusion that uses the information from all involved sensors in one common way or heterogeneous fusion that combines groups of involved sensors that serve multiple purposes, yet are attached to a single system. A simple example for the latter would be a single large electrode sensor that detects the presence of a hand from a farther distance and then a combination of various small electrodes that track single fingers. Sensor data fusion often requires taking into account some additional information we possess about the system. A classic example is the precision or bias of the sensor. Various methods, e.g. the class of Kalman filters, use weighted information from several sensor sources [WB95]. If we know how that a certain sensor is only half as precise as another one working in collaborating, the weighting factors can be adapted accordingly.

One of the most important additional information we use when fusing data of capacitive proximity sensors, is the geometric layout of the system. This describes position and size of all electrodes that are integrated. Using this information is crucial when trying to localize an object. A simple example would be applying a weighted average algorithm on a set of sensors. In order to determine object location relative to the plane a weighted average algorithm is used. The linear object location $\overline{x}$ is calculated using the sums over sensor positions x_i and sensor values v_i as weight:

$$\bar{x} = \frac{\sum_{i=1}^{n} v_i x_i}{\sum_{i=1}^{n} v_i} \tag{2.6}$$

Using similar methods we are able to determine the location of multiple objects or additional dimensions of the position. However, it is possible to use other information in the fusion process as well. The electrode material may result in a different response and thus should be treated differently in a fused data representation and can be weighted. Another example is the shape of the electrode that may result in different responses. How to apply sensor data fusion is strongly depending on the application and the desired common representation that is most suitable for subsequent calculations.

Object tracking In the previous section about sensor data fusion we have shortly discussed a method to determine the linear position of a single object using a linear array of capacitive proximity sensor. This is a basic example of a group of methods associated to object tracking. In computer vision applications they can be defined as “the problem of estimating a trajectory of an object in an image plane as it moves around a scene” [YJS06]. The analogy to capacitive applications is viable if we consider a 3D scene and a distinct interaction space instead of a scene. Capacitive proximity sensors allow the detection of conductive objects within their range. However, as this presence is determined indirectly using the influence on an electric field it is not possible to get a direct association between the actual distance between sensor and object and the resulting sensor value. The created electric field is only analytically descriptive for very specific, theoretic classes of objects [Bax96]. Nonetheless, we are able to get a relative distance measurement. If we combine this proximity value using geometric information about the electrode location we can infer the relative position of an object in the sensing area. The weighted average method presented in the previous section is one option for relative positioning. Another method is trilateration, similar to many radio-based localization applications, that uses the known location of three or more points and the known distance to the position to be determined. In case of capacitive proximity sensing this position is determined relative to the electrodes as there is no absolute distance measurement. A more complex example for direct calculation was presented by Smith, who formulated the issue of detecting multiple objects as a forward problem and used numerical methods to estimate the position and orientation of two hands [**?**]. A

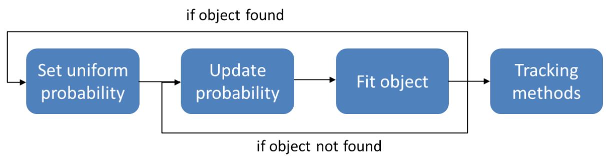


Figure 2.10.: Generic pipeline of probability based methods of capacitive proximity sensing

second class of methods to track objects is not relying on direct geometrical calculations but instead formulates a numerical solution to a probability distribution. The initial assumption is that the probability of an object to be at a certain point in the detection area is uniform. The methods then follow a few basic steps, as shown in Figure 2.10. At first the probability is updated based on the current sensor readings and a priori knowledge that we have about the system. Afterwards we try to fit the objects into the resulting probabilities. This step may or may not work, meaning that it may result in no object found. In the latter case the process will have to start at the beginning. If an object is found the probability update may use the current object location in the update algorithm, thus starting with a non-uniform probability distribution. One example for probability-based object recognition using capacitive proximity sensors was presented by Grosse-Puppendahl et al. [GPBKK13]. Using a model suggested by Smith the basic idea is using the assumption that an object may be present anywhere, remove regions where no objects can be present and then fit an object into the remaining space. This method additionally uses particle filters to track object locations over time. This also allows tracking multiple objects. Throughout the years various methods have been suggested for supporting multi-object tracking using capacitive sensors. Touch screens often use inversion of the sender signal to reliably detect the positions of multiple points; however, this method can't be used in proximity applications [WF07]. Some of the previously presented methods support the tracking of two or more objects. There are still various limitations, particularly if not only the object location but also various other features such as rotation should be tracked. This is still an area of ongoing research, leading to the next area of high-level processing - feature recognition.

Table 2.3.: Feature recognition methods

Name	Description
Data-driven methods	Directly associate input data to output features using various methods, e.g. machine learning and training data
Model-driven methods	Input data is manipulating a pre-defined model of the system that is latter mapped to the output
Neural networks	Computational models using a network of neuron-like objects that are often used in machine learning
Pattern recognition	Methods that look for certain patterns in a set of input data
Semantic mapping	Methods to realize object tracking for a single detectable object

Feature recognition Feature recognition is primarily used as a term in image processing, traditionally in computer-aided design applications to recognize specific geometric properties of an object but also picture analysis, e.g. in facial recognition [HPR00, BHK97]. In the domain of capacitive proximity sensing, feature recognition can be defined as the acquisition of non-location information from any detectable object. An important feature in industrial applications is the material of an object [Bax96]. With regards to recognizing additional features a system was presented by Wimmer et al. - Thracker [WHKS06], a prototype augmenting a regular monitor with capacitive proximity sensors. In addition to recognizing hand position the system is able to detect grasp gestures, which can be used to select items on the screen and perform pick and drop operations. Capacitive sensors can also be used to distinguish between persons and a children's seat on the passenger side of a car [GZBB09]. The methods to recognize the features can be divers, ranging from typical machine learning algorithms, to model-based approaches. An incomplete list is given in Table 2.3. In order to keep this work contained we refrain from a deeper discussion at this point.

2.2. Capacitive proximity sensing applications

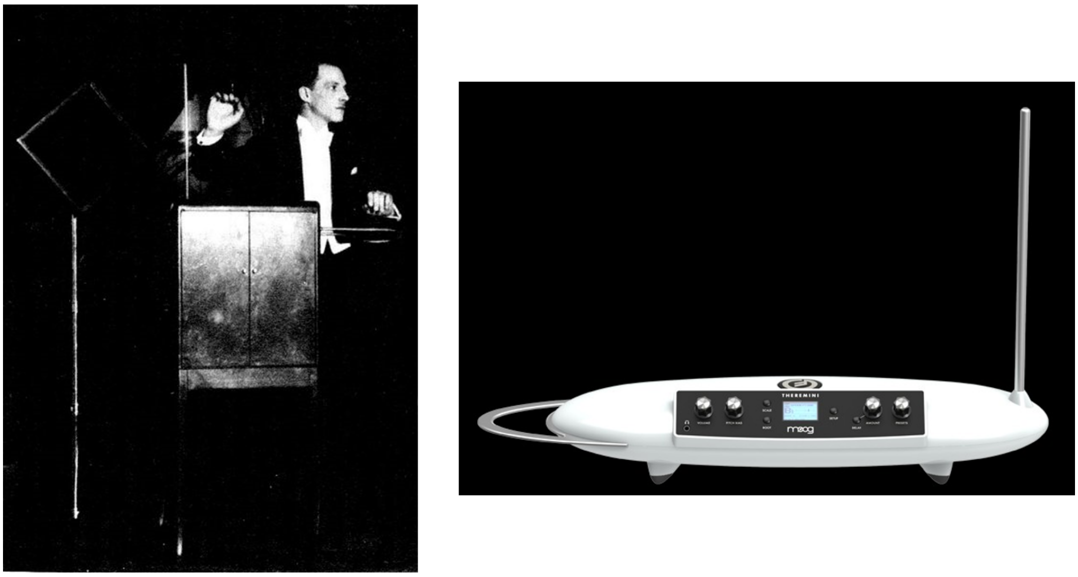


Figure 2.11.: *Left:* Leon Theremin playing his epnoymous electronic musical instrument [Gli00]. *Right:* The Theremini by Moog Music Inc., released in 2014 [Inc14]

In the last decades of the 19th and the beginning of the 20th century a considerable number of inventors and scientists performed research on the application of electric systems, sparking innovations such as electric lighting, electric motors, telegraphy, and radio communication. Lev Sergeyevich Termen or Léon Theremin in the American naming was a Russian inventor most famous for designing the eponymous theremin. This early electronic musical instrument could be played without touch. One hand is controlling the pitch and the other the volume by changing the distance to an antenna. Initially designed as a motion detector, this device is transferring the influence of the human body on an oscillating electric field to an audible sound [Gli00]. Léon Theremin can be seen playing the instrument in Figure 2.11 on the left. This instrument is still in production to this day, with the electronic music instrument company Moog releasing a new variety that simplifies the sound production by fitting the distance to a sound on a specific musical scale [Inc14]. The instrument is shown in Figure 2.11 on the right.

Electric field imaging was a research focus at the MIT in the 1990s. A research group in the Media Lab division including Joseph A. Paradiso, Thomas G. Zimmerman, Joshua R. Smith designed various sensing devices and evaluated various applications in in the domains of human computer interaction, smart appliances and reactive systems. They drew inspiration from biological precedents - various species of fish, such as Gymnotoidei can sense their surroundings using electric fields [**?**]. The changing currents created by objects with a different dielectric constant from water can be registered and thus used to avoid obstacles, even if no light source is available. Accordingly, the group named some of their prototypes after this biological precedent, including the LazyFish and School of Fish [**?**]. The research group created various different applications in the domains of human computer interaction, smart appliances and reactive systems.

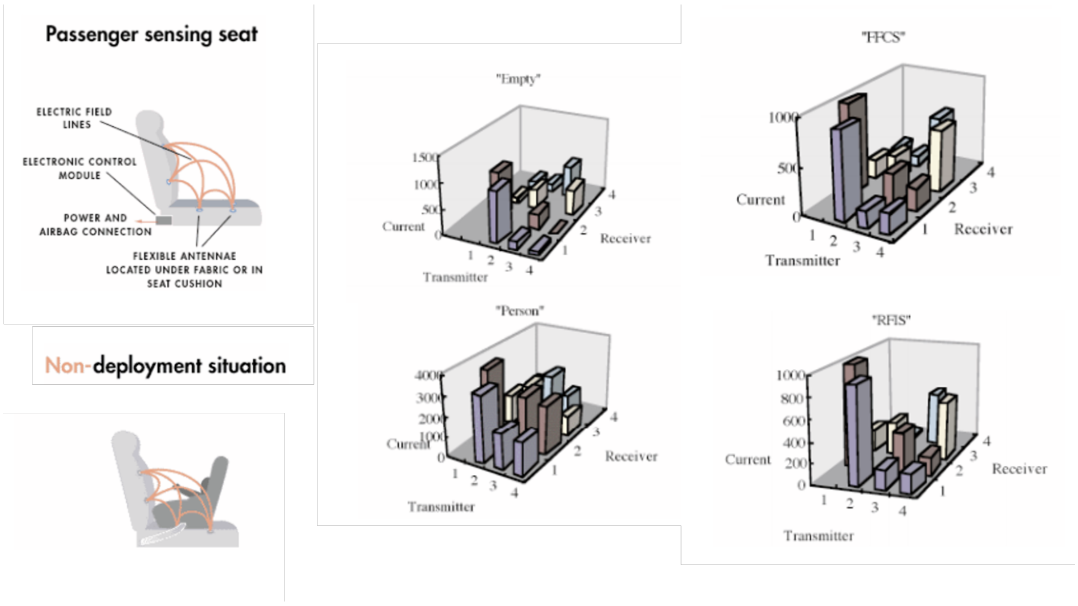


Figure 2.12.: *Left:* Concept view of passenger seat set to deploy or not deploy airbag. *Center:* Sensor readings for empty seat and adult person. *Right:* Sensor readings for front-facing child seat (FFCS) and rear-facing child seat (RFCS). [**?**]

In collaboration with NEC a smart passenger seat was created that incorporated capacitive sensors operating in shunt mode to detect if an infant seat is currently present on the passenger seat of car [**?**]. The underlying challenge is that an airbag deployment should be prevented in such cases to prevent potential injuries to the infant. The seat is able to distinguish four different states, “No passenger”, “adult passenger”, “front-facing infant seat” and “rear-facing infant seat”. It uses four sending and four receiving electrodes and classifies the situation according to the current readings - the concept and readings of the sensors in the different situations are shown in Figure 2.12.

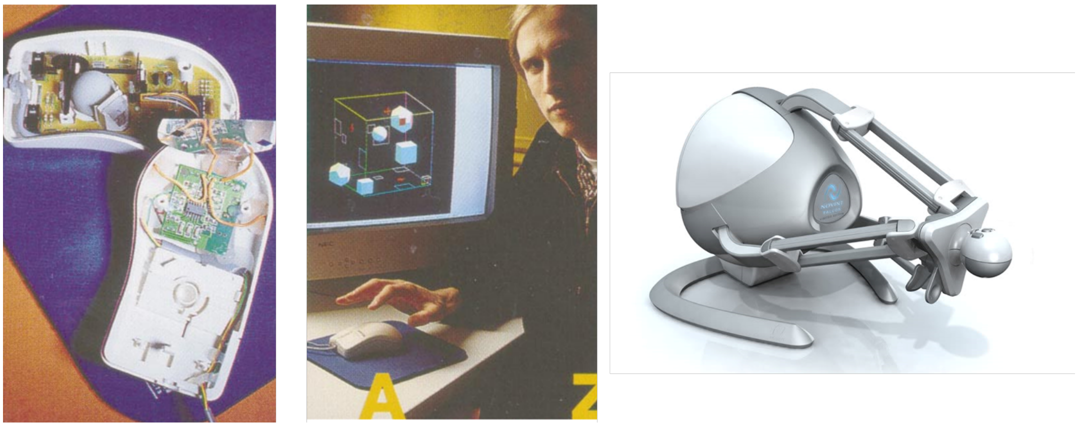


Figure 2.13.: *Left:* LaZmouse innards *Center:* Joshua R. Smith using LaZmouse [**?**] *Right:* Novint Falcon 3D input device [Nov14]

Another prototype is the LaZmouse that extends a regular mouse with shunt mode capacitive sensors, having one transmitting and two receiving electrodes, to measure the proximity between the heel of the hand from the mouse surface, thus allowing the fingers to remain in the common position and the mouse to be moved around [**?**]. Effectively this creates an input device with three degrees-of-freedom, enabling to perform interactions with a mouse that would usually require a more specialized 3D input device, such as the Novint Falcon that tracks the movement of the moved interaction sphere in three dimensions [Nov14]. Figure 2.13 shows on the left, the electronics inside the mouse, the inventor using the device and a graphical representation, and on the right the Novint Falcon.

In another work Paradiso et al. presented numerous applications for capacitive proximity sensors, including smart furniture devices [ZSP*95]. They propose a smart table, comprised of a single transmitter electrode and two receivers that is able to track the position of a hand in two dimensions, a chosen dimension on the table and height. It may be used as gesture input device or to augment video desk applications. They also installed the system in a room, whereas the floor is a single electrode and there are four receivers located on the walls. This allows to infer the location of a person, based on relative signal strength. An additional system in this work is the smart chair, using a single transmitter in the seat and four receivers in the armrests and headrest. It allows to navigate through various audio channels based on head and arm movements [SM95].

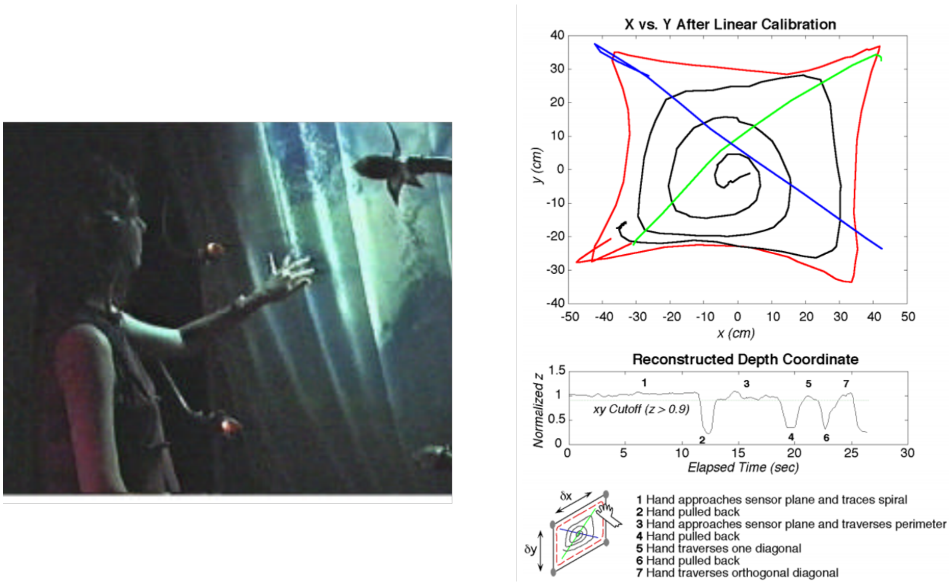


Figure 2.14.: *Left:* Person interacting with the gesture wall *Right:* Air drawing results, depth estimation results and associated movements on bottom. [SWD*98]

A final prototype of this group I would like to present is the Gesture Wall, a large interactive multimedia wall, designed for public appearances [SWD*98]. A plate on the floor in front of the screen is acting as transmitter and four receiving electrodes that protruded from the edges of the projection area, as shown in Figure 2.14. It supports interactive experiences, such as drawing in the air, controlling different audio streams and an interactive video clip.

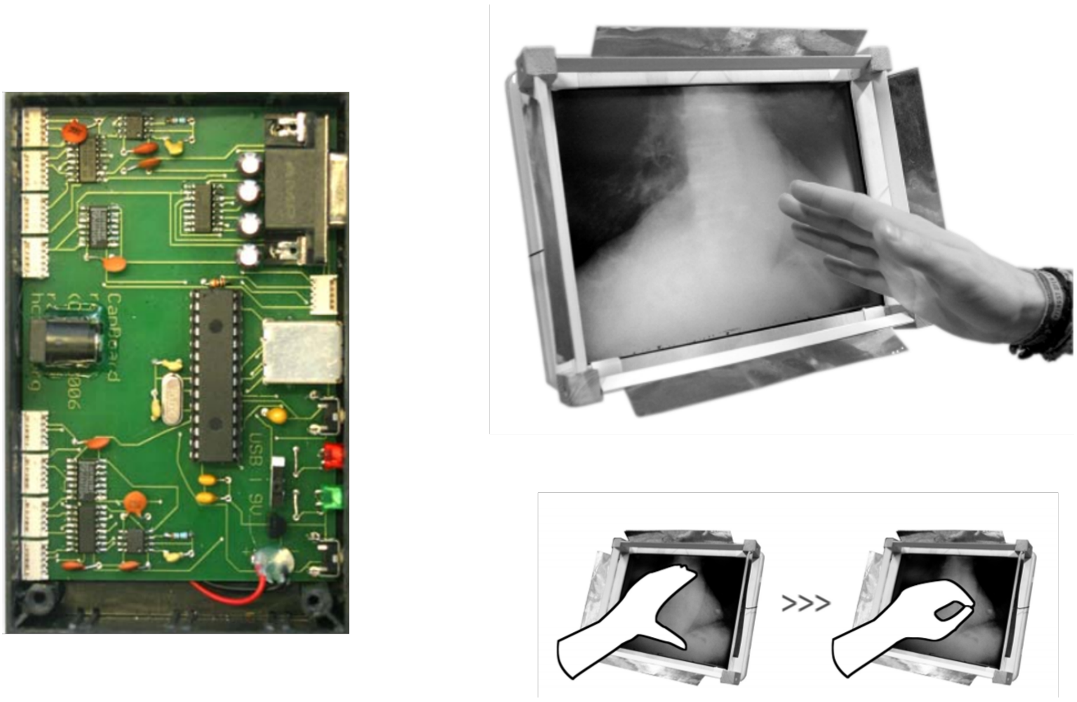


Figure 2.15.: *Left:* Prototype of CapToolKit [WKBS07] *Right:* Thracker prototype and visualized grasping gestures. [WHKS06]

Another group that was active in capacitive proximity sensing was located at the University of Munich. Raphael Wimmer and colleagues revisited capacitive sensors in the scope of human-computer interaction. They created CapToolKit, a capacitive sensing rapid prototyping toolkit that allows interfacing eight capacitive proximity sensors and enables a quick design and testing of new applications [WKBS07]. They also created Thracker - a display augmented with four capacitive proximity sensors that allows to detect the position of the hand in front of the screen and supports performed pick-and-drop gestures [WHKS06]. Both devices are shown in Figure2.15.

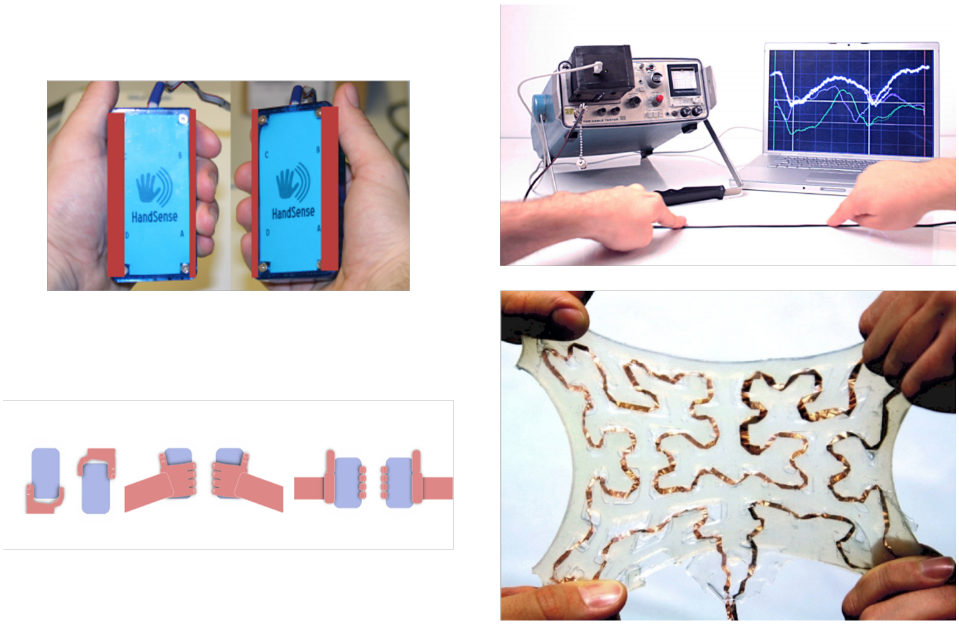


Figure 2.16.: *Left:* Prototype of HandSense and supported grasping types [WB09]. *Right:* Setup of time domain reflectometry sensing and example of stretchable material [WB11].

In other works they also presented capacitive sensing to allow discriminating the ways an interaction device is held, including distinguishing left and right hand or proximity to a body part [WB09]. This enables graphical interfaces to be adapted, based on user-handedness, grasping style and spatial cues that can be acquired from this device in collaboration with other sensors. Supported grasping styles and a prototype are shown in Figure 2.16, on the left.

A final work of this group was concerned with exploring the potential of time domain reflectometry for human-computer interaction [WB11]. This technique is sending a short electrical pulse into an electric conductor and measures the time until the signal returns. Originally intended for finding defects in long cables, such as transatlantic phone lines, high-sampling rates enable to also detect the presence of grounding objects close to much shorter conductors. Wimmer and Baudisch use an image analysis on the screen of an older reflectometer to enable applications, such as location tracking, touch detection and stretchable materials. The setup and an example of stretchable materials are shown in Figure 2.16, on the right.

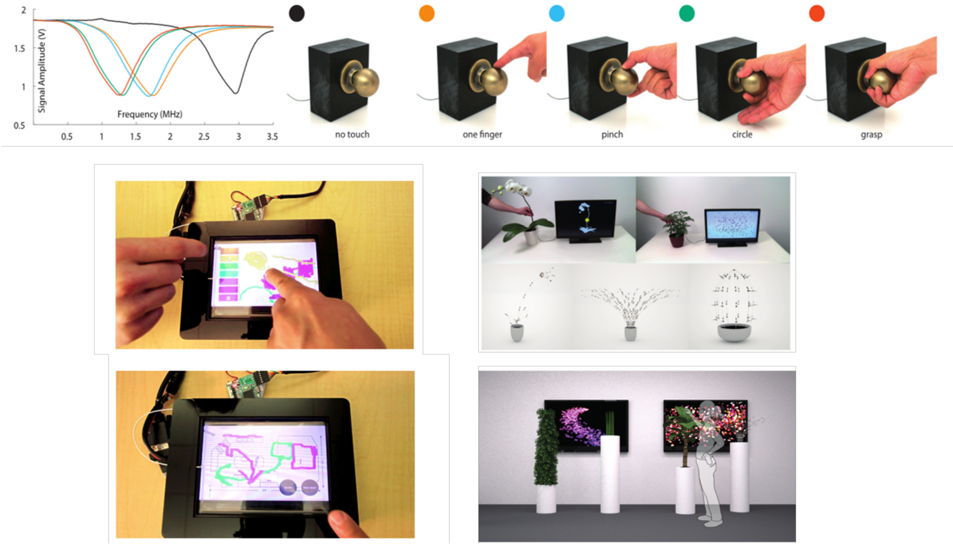


Figure 2.17.: *Top:* Capacitive profiles of different gestures on a door knob [SPH12]. *Bottom Left:* User identification using capacitive fingerprinting [HSP12]. *Bottom Right:* Botanicus Interacticus prototype and interaction concept [PSLS12]

Another research group that has been recently active with capacitive sensing systems is centered around Ivan Poupyrev. They presented Touché, proposing a novel sensing technology called swept-frequency capacitive sensing [SPH12]. Instead of relying on a single excitation frequency like typical capacitive sensors this measurement system is sweeping a larger range of frequencies, thus allowing to get more data points from a single electrode. Many systems react differently to changing frequencies, including the human body and its complex structure of bone, muscle, fat and other tissues. The result is a capacitive profile comprised of values at different frequencies, as shown in Figure 2.17 on the top. Using these profiles it is possible to distinguish numerous events based on the application, e.g. the different ways in which a mobile device can be held. Other applications proposed in this work were sensing of body postures on a table, on-body gestures using wrist-worn sensors, interaction with a body of water and grasping a door knob using different gestures. Building on this work Harrison et al. proposed a system using this technology to identify the users of a touch screen with a capacitive fingerprint [HSP12]. This allows to distinguish the touches of different users, thus enabling a variety of applications on collaboratively used touch screen devices, such as gaming and multiuser drawing tools, as shown in Figure 2.17 on the bottom left. An interesting suggestion are personalized undo stacks that allow a more seamless collaboration. A final system is the Botanicus Interacticus, that uses the same technology to realize interactive installations using plants [PSLS12]. It is possible to distinguish the location a certain plant is touched and associate it to different input events. In this first system it was used in an interactive art installation, that is shown in Figure 2.17 on the bottom right.

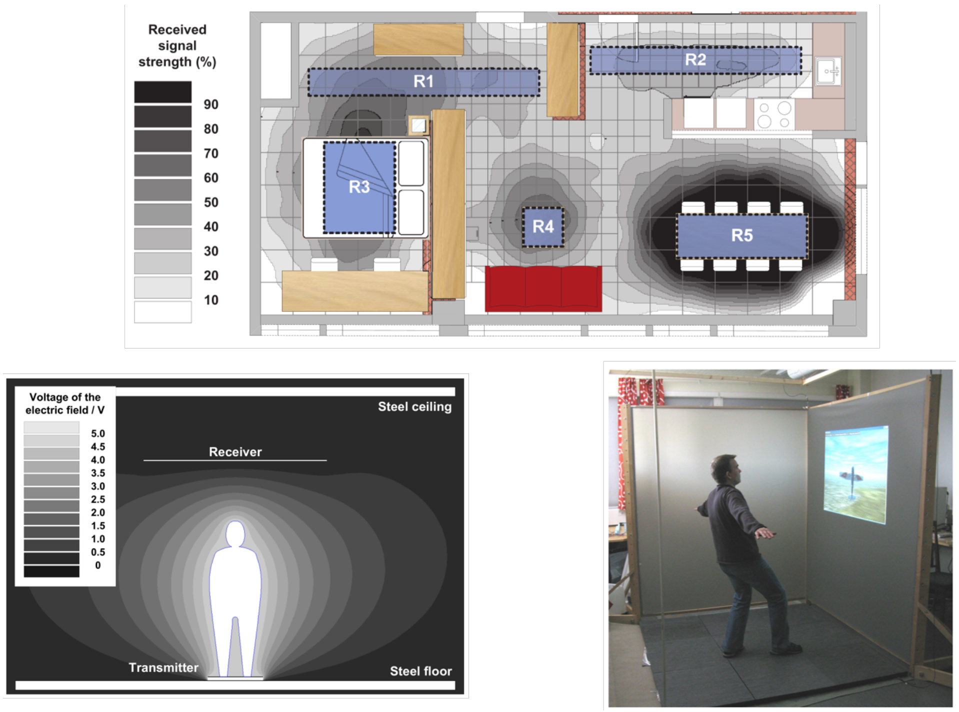


Figure 2.18.: *Top:* Capacitive profiles of different gestures on a door knob [SPH12]. *Bottom Left:* User identification using capacitive fingerprinting [HSP12]. *Bottom Right:* Botanicus Interacticus prototype and interaction concept [PSLS12]

A final group working with capacitive proximity sensing that I would like to present is located at the Tampere University of Technology. Valtonen et al. have been working on a capacitive proximity sensing system tracking the position of users in a room [VMV09, VKV12]. A transmitting electrode is placed under the floor and several receiver electrodes are hidden in the walls or other objects of the room. The system therefor is using the presented transmit mode, coupling the users present in the environment to an electrode in the floor and measuring the effect on the receiver electrodes. A downside of this system is the required proximity to a wall electrode, making application in larger rooms impossible. Accordingly, an addition was presented that hides the receiving electrodes in different items of furniture to provide a more reliable localization in living environments [VKV12]. The placement of the receiver electrodes (R) and the received signal strength in the environment are shown in Figure 2.18 on the top. An extension of this principle is using a similar electrode configuration to determine height and posture of a person between a transmitter electrode on the bottom and a receiver electrode on the ceiling [VKMV11]. A simulation of the electric field voltages created in this setup is shown in Figure 2.18 on the bottom left. The taller a person, the higher the resulting signal. Accordingly, a sitting or lying person will have a much smaller effect and can be classified. A similar setup was also used to identify different gestures performed in an interaction area, as shown in Figure 2.18 on the bottom right [VRV10]. Using a single transmitter and four wire receivers in the corners it is possible to track different body postures and control some example applications.

2.3. Sensor systems in smart environments

In the most general definition a sensor is a device that transforms a physical property into an observable signal. This definition includes traditional systems such as mercury-based thermometers or hair-based hygrometers. Yet nowadays we are usually considering digital sensors that transfer the measured property to a binary signal that can be further processed by computing devices. A common variety is the smart sensor that provides additional functionality beyond generating a correct sensing signal [Fra13]. The main goal is to simplify installation and maintenance of distributed sensing systems by having processing close to the measurement device. Early considerations in this domain were put to the standard family IEEE 1451 - IEEE Standard for a Smart Transducer Interface for Sensors and Actuators between 1997 and 2007 [iee07]. An additional concept is the Virtual Sensor that includes digital signal processing and conditioning and therefor abstracts the processing steps from devices interfacing the sensor. The number of available sensors is very high, but it is possible to restrict them based on application domain. Lewis and Cook et al. [Lew04, CD07] have proposed a collection for smart environments focused on wireless sensor networks. The overview is shown in table 2.4. This sensor categorization is based on

Table 2.4.: Sensors for smart environments [CD07]

Properties	Measurand
Physical properties	Pressure, temperature, humidity, flow
Motion properties	Position, velocity, angular velocity, acceleration
Contact properties	Strain, force, torque, slip, vibration
Presence	Tactile/contact, proximity, distance/range, motion
Biochemical	Biochemical agents
Identification	Personal features, RFID or personal ID

the property to be measured and is agnostic to the specific measurement technology. Physical properties, such as pressure, temperature, humidity and flow, can also be noted as environmental properties. They are measurements that determine the state of the smart environment, e.g. temperature in different rooms, or the current water usage. Motion properties denote the movement parameters of actors in this environment and can refer to both humans and machines. Angular velocity is important in self-localization of robots in an environment. Contact properties groups the different types of interaction between surfaces in the smart environment and actors. Presence as a group is similar to motion paramteres, but does not require a series of measurements for tracking an actor. Biochemical sensors enable measuring the presence of specifc chemical compounds in the environment and are most suited for measuring pollution or air quality. Finally, identification of actors allows to provide personalized services and can be realized with different methods ranging from tags to biometric systems.

While this listing provides a decent overview of sensing properties in smart environments it is abstracted from sensor technologies. Various types of sensors, including capacitive proximity sensors, allow us to detect multiple of these properties and thus providing a higher flexibility. Therefore it is possible to provide an inverse listing of sensor technologies that allow measuring different properties. A short overview of sensor technologies with this capabilities and that are commonly used in smart environments is given in table 2.5. In the following sections I want to give an overview on how these sensor systems are used in this domain, in order to provide a basis for the benchmarking model that will be introduced in section 3.

Table 2.5.: Sensing technologies and measured properties

Technology	Properties
RGB cameras	Motion, Presence, Identification
Infrared cameras	Motion, Presence, Contact
Ultrasound sensing	Motion, Presence, Contact, Identification
Microphone arrays	Motion, Presence, Contact, Identification
Radiofrequency sensing	Motion, Presence, Identification

2.3.1. RGB cameras

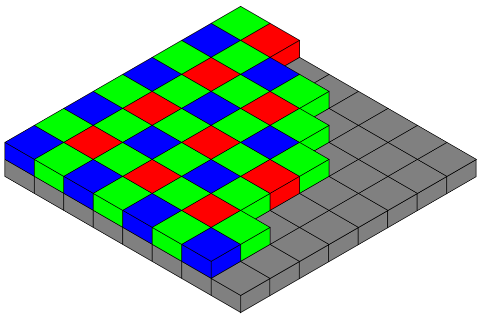


Figure 2.19.: A bayer pattern on a sensor in isometric perspective [Com06]

A RGB camera is an image processing device that processes light in the visible spectrum, similar to the human eye. Modeled after the retina it has three distinct color channels - red, green and blue. There are different methods available to distinguish these channels from visible light, such as Bayer filters (Figure 2.19) in front of a single sensor or using multiple sensors behind a prism. The usage of cameras in smart environments is very common. I will present five different examples and afterwards will specify how they are linked to the properties that were defined previously. Tabar et al. have been using a combined system of cameras, RF transmitters and wearable sensors in a home care scenario [TKA06]. The cameras are used to improve the accuracy of the accelerometer-based fall detection by eliminating false positives. Once a fall event occurs an algorithm tracks the posture of detected humans in the scene (Figure 2.20 - left). They used an edge detector to distinguish the human body from other objects and applied a heuristic to differentiate lying and standing. Additionally a face detector was used to improve the recognition of human objects. Combining this with information from the fall detecting sensor and a RF based localization system they were able to achieve a good reliability in eliminating false positive alerts.

Pentland and Choudhury provided an overview of vision-based face recognition systems in the domain of smart environments [PC00]. The systems are able to identify users and recognize facial expressions. The proposed applications in smart environments include personalized shopping experiences based on customer recognition, behavior monitoring in child care facilities and emotion-aware systems that react to the user's current awareness. The described techniques include PCA-supported, eigenvector-based classification, face-based localization and systems based on local feature analysis (Figure 2.20 - right). Newer systems are able to operate well

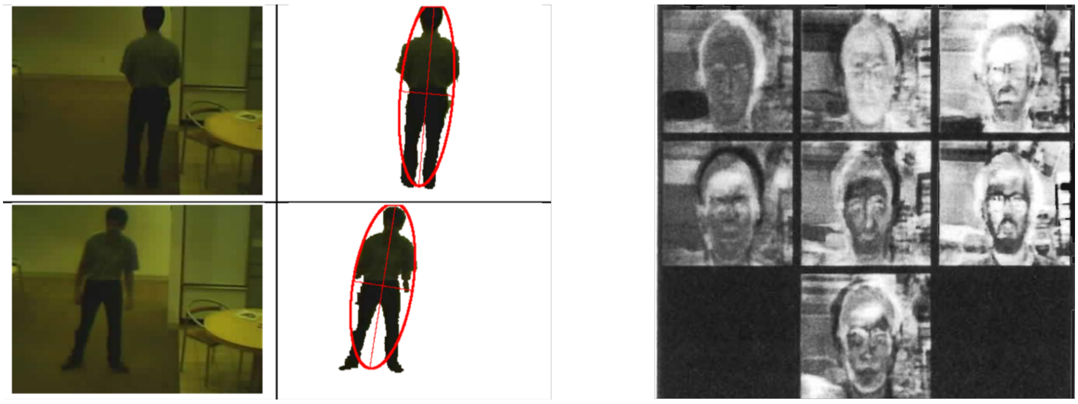


Figure 2.20.: *Left*: Tracking of body masks using cameras [TKA06]. *Right*: Eigenfaces created from input picture set [PC00]

in unconstrained environments, that include varying expression and illumination, ageing of persons, occlusion and disguise [WYG*09].

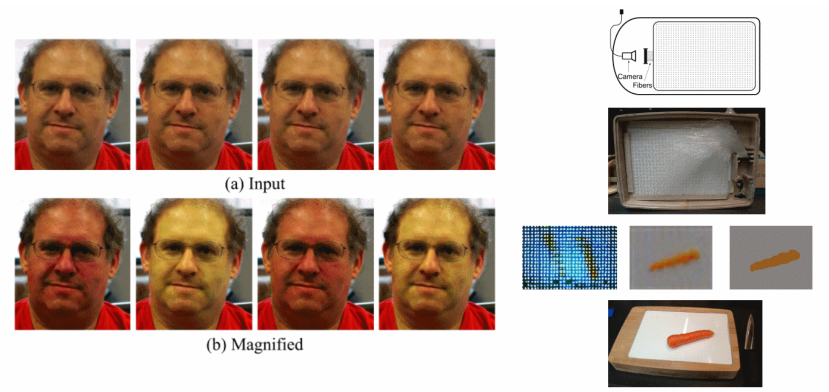


Figure 2.21.: *Left*: Eulerian Video Magnification to attenuate the human pulse with original (a) and amplified (b) video sequence [WRS*12]. *Right*: FoodBoard schematics (top), underside view (second row), original, reconstructed and segmented image (third row) and final system (bottom) [PJS*13]

An example for a novel image processing method that is useful in smart environments was presented by Vu et al. in 2012 [WRS*12]. They are using temporal variances of pixel values to exaggerate spatial movements and color changes that would typically be invisible to the naked eye. The method is called Eulerian Video Magnification and uses a combination of spatial decomposition and temporal filtering applied to adjacent frames. It can be tuned to different time-frequency bands to attenuate different classes of signals. Some of the proposed applications include the tracking of breathing rates of infants by attenuating chest movement, or the tracking of subtle movements, such as vibration in appliances. The example shown in Figure 2.21 on the left is using a magnification of colors, in order to identify the heart rate of a person. The latter can be used for personal health applications, e.g. by integrating the system into the bath room mirror to provide an unobtrusive daily measurement and give the user feedback over a longer period of time.

A final example in this section is the FoodBoard, a smart chopping board that uses image processing to recognize the food items that are put on it [PJS*13]. It is shown in Figure 2.21 on the right. To enable a thin footprint, ambient light is transferred to a camera using glass fibers. The picture is reconstructed and segmented, allowing to identify different items of food that are placed on it. The classification is based on a combination of Fast-Hessian and color histogram feature extractors. Pham et al. were able to distinguish 12 different ingredients with an accuracy between 59% and 93%. The system can be used to support dietary monitoring, give recipe guidance or support visually impaired users in identifying and tracking food.

2.3.2. Infrared cameras

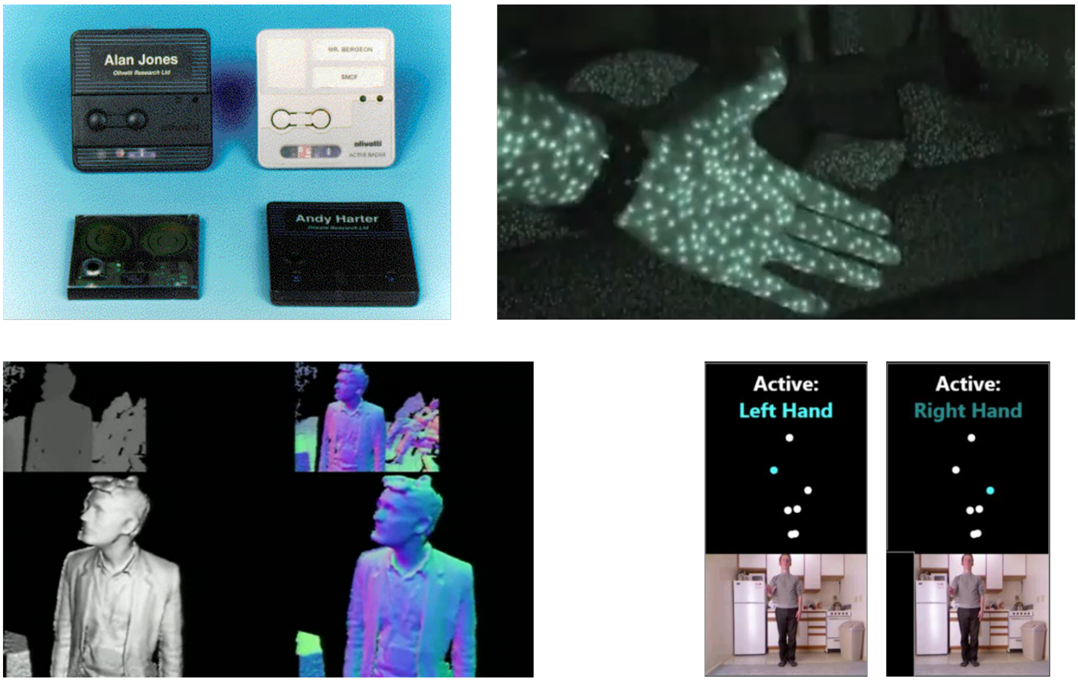


Figure 2.22.: *Top Left*: ORL Active Badge [Wei91]. *Top Right*: Kinect infrared projection [Zha12]. *Bottom Left*: Kinect Fusion reconstrucion [IKH11]. *Bottom Right*: Kinect kitchen interaction [Pan12]

Infrared imaging is using the same sensors that are suitable for visible light imaging. The difference is that they are tuned to collect electromagnetic waves of a lower wavelength that are just outside of the visible spectrum. This allows for distinct applications, such as thermal imaging, as it is possible to detect heat radiation. One of the earliest prototypes in Ubiquitous Computing designed by PARC was using the ORL Active Badge, an infrared emitter developed by Olivietti Research Laboratories that was used to identify persons operating in the environment [Wei91] (Figure 2.22 Top Left). In smart environments the most common application is using infrared cameras in combination with infrared light sources. This allows to illuminate spaces without visible artifacts to the user, thus providing imaging capabilities in dark rooms, or very specific conditions that may be required by a certain application. Another very interesting option is to use a specific projection of patterns into the scene. Analyzing the returning infrared light it is possible to infer the depth of specific pixels of the camera (Figure 2.22 Top Right). This variety is called a depth camera. Particularly in the last few years the research in this domain has expanded strongly, sparked by the availability of an affordable depth camera/RGB camera combination - the Kinect by Microsoft [Zha12]. On the following pages we will present various examples of how this device can be used in smart environments to enable different applications in interaction and activity tracking.

Sung et al. have presented a system that is tracking activities of daily life based on the movements of a skeleton that is provided by the Kinect API [SPSS11]. This skeleton model is based on a pose reconstruction algorithm developed by Shotton et al. [SFC*13] and is used in many different Kinect-based applications. The algorithm is using a method called hierarchical maximum entropy Markov model (MEMM). Each activity is considered to be composed of sub-activities. Based on this assumption a two-level graph is generated using dynamic programming. The system was tested with twelve activities performed in an office, a kitchen, a living room, a bathroom and a bedroom. If the person was part of the training set the precision was 84.3% and 64.2% for unknown persons. Some example activities of the acquired dataset are shown in Figure .

A novel method to use the Kinect for fast 3D reconstruction of scenes was presented by Izadi et al. in 2011 [IKH11] (Figure 2.22 Bottom Left). The basic premise is to use a fast registration method to combine point clouds that are generated by the system and continuously extend and optimize the current model of the scene. They are using a GPU-based ICP implementation to track the position of the camera in six degrees-of-freedom. This allows to reliably integrate the different point clouds into a single voxel grid that can be used to represent and render the scene. They proposed a number of applications ranging from phyiscal simulation of particles in the scene, to system control based on segmenting and tracking the user's hands and their interaction with arbitrary surfaces in the environment. Figure shows some results of the touch recognition in a reconstructed scene.

Galen has presented a method to use a Kinect in a kitchen to provide touch free interaction [Pan12]. Two different interaction schemes are proposed. The first allows to control the applications with messy hands, the second enables to use other limbs if the hands are currently occupied (Figure 2.22 Bottom Right). Three different test applications have been implemented. A recipe navigator that allows to open and navigate through different recipes. A timer that enables setting different alarm times, similar to a kitchen clock and a music player that can be controlled to choose different stations according to preference. A real life test with five subjects did also test installation complexity, which was deemed favorable. There were some concerns in clearly distinguishing commands from random movements performed in the kitchen.

The final system based on infrared cameras is an immersive telepresence system developed by Beck et al. [BKKF13]. Telepresence enables persons to be present as a representation in a remote place. A typical example is video conferences, but advanced system may include robotics or virtual elements. The presented system is using a single Kinect for each participant. A 3D representation is created based on the depth information of the infrared camera and on-the-fly texturing using the included RGB camera. The virtual user representations are put into a shared virutal environment and can interact with each other. Some variations were tested where local and remote users were decoupled, side-by-side or face-to-face. Different tracing and pointing gestures are supported. The supported applications included the exploration of a virtual city.

2.3.3. Ultrasound sensors

Ultrasound sensors allow detecting sound wave signals that have a frequency beyond 20kHz and are thus not audible to humans. Their propagation and reflection properties are similar to audible sound waves, thus the generated measurements can be similar. While there are natural sources of ultrasound waves the applications

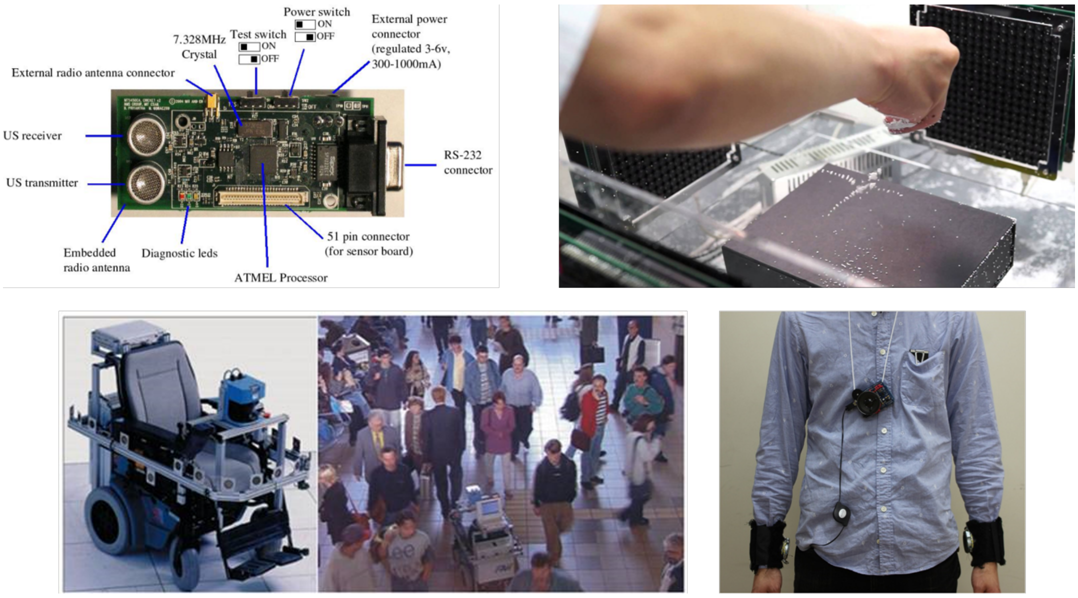


Figure 2.23.: *Top Left*: Cricket Indoor Localization hardware [PCB00]. *Top Right*: Mid-air particle manipulation [OHR13]. *Bottom Left*: Robotic wheelchair MAid with ultrasound range sensors [PSF01]. *Bottom Right*: Activity and context recognition using ultrasound sensors [WTT13]

in smart environments do rely on active systems, that combine sound generators and sensors that measure the resulting signal. By timing the time distance between sending the signal and receiving a response it is possible to measure distances between the sender and different object. If various receivers are used it is possible to localize the sound source, making ultrasound sensing a popular candidate in indoor localization systems. In Figure we can see a sketch of the basic functionality of ultrasound sensing systems on the left, and an example of localization using three receivers and a single source. We will present four different examples on how ultrasound sensors are used in smart environment applications.

The Cricket developed by Pryantha et al. is an example for an indoor localization system based on a badge the tracked object has to wear [PCB00]. The badge is periodically sending signals to a set of beacons that determine the distance and calculate a location. Initially it was supposed to solely rely on radiofrequency signals, but was modified to use a combination of RF and ultrasound. The rationale of this decision is the considerably slower speed of sound that simplifies measuring the time-of-flight and thus allows for more precise distance measurements. Consequently this also leads to a better precision in the localization algorithm. Additionally, the system uses some novel methods to deal with interference and multipath issues, that is dealing with reflected signals. As potential applications they propose space-dependent services that are provided as the user is identified in a certain region and guidance scenarios on a floor plan. The hardware is shown in Figure 2.23 (Top Left).

The robotic wheelchair MAid (Mobility Aid for Elderly and Disabled People) was designed to support and transport people with limited motion skills [PSF01]. It is based on a commercial wheelchair that has been equipped with an intelligent control and navigation system. The system includes an ultrasound-based range finder that allows to detect obstacles in the path and circumnavigate around (Figure 2.23 Bottom Left).

Watanabe et al. investigate the role of ultrasound sensors in recognizing activities and gestures of a user [WTT13]. The system is comprised of a microphone attached to a necklace and two speakers that are attached to each wrist. Based on the acquired volume and evaluation of the Doppler effect it is possible to determine both distance of the wrists from the neck and the speed of movement (Figure 2.23 Bottom Right). Watanabe et al. want to determine if this system allows for similar performance compared to other body-worn systems based on accelerometer and gyroscope data. Additionally it was evaluated if external microphones can perform as well as the neck worn microphone. The system was able to recognize 87% of activities and gestures in a set of 10 test persons if no influencing sound was present. The ultrasound did also improve results if environmental noise was present.

A recent project at the University of Tokyo evaluates the potential of ultrasound in manipulating small particles in free-air [OHR13]. Using standing waves to create sound pressure notes it is possible to apply a force to small particles that is sufficient to counteract gravity. Using a set of ultrasonic phased arrays it is possible to create these sound pressure nodes at arbitrary positions in three dimensions (Figure 2.23 Top Right). Ochiai et al. use this to move small objects around and investigate required object properties and their floating properties. So far the moved items have to be very light and smaller than 2 mm in diameter, resulting in the usage of polystyrene. The technology is also able to hold and move small amounts of fluid. Suggested applications include object manipulation in microgravity environments and projected haptic feedback systems.

2.3.4. Microphone arrays

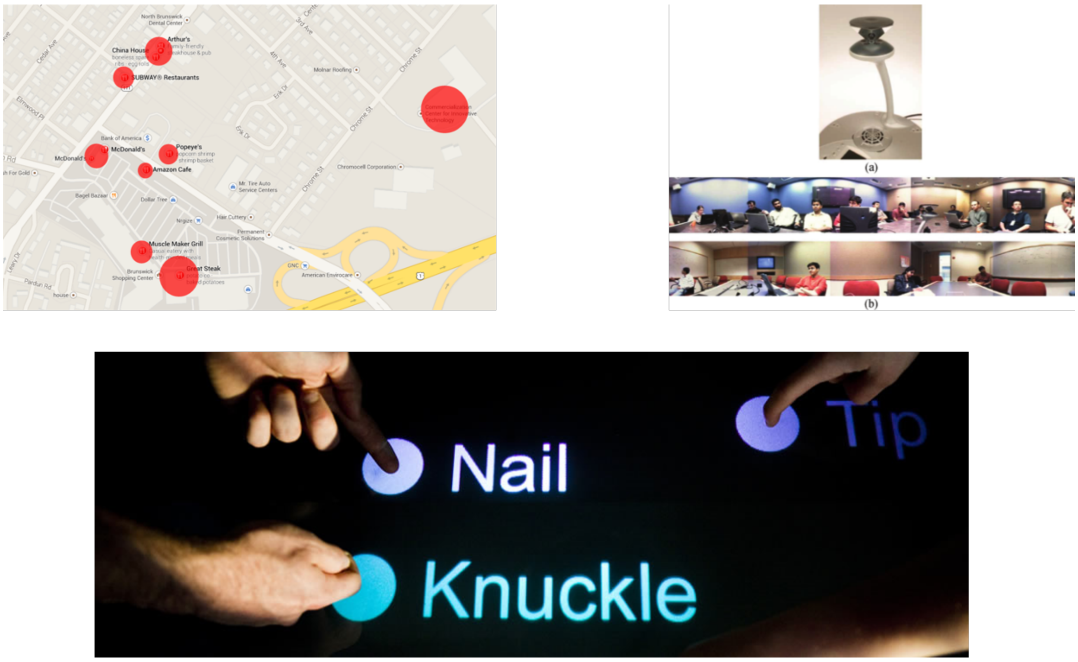


Figure 2.24.: *Top Left*: Visualization of speaker count in different areas [XLL*13]. *Top Right*: Directional microphone and conference room for speech source localization [ZFBZ08]. *Bottom*: TapSense detecting different tap events [HSH11].

Microphones are signal receivers that are tuned to detect sound frequency ranges that are audible by humans. Typically they consist of a piezo element that transfers vibration to an electric current that is amplified. Most human activities produce some kind of sound. As opposed to the presented ultrasound system, microphones typically operate passive. That is there is usually no dedicated signal source, but instead naturally occurring sounds are picked up. By combining microphones or arrays thereof with data processing systems that are aimed at analyzing specific sound patterns we are able to get feedback on human activities. Looking at smart environments there are numerous use cases that can benefit from microphones as sensors. In this section we will present four different systems that cover a large variety of different applications, ranging from breathing rate detection to estimating the number of speakers in large groups.

Detecting the breathing pattern of a person has several applications in smart environments. Apart from medical applications that require detecting abnormal breathing in risk groups it is also possible to track training progress using such a system or provide a measure for the current attention level in affective computing. Corbishley et al. investigated using very small microphones in mobile devices to enable detecting the breathing rate [CRV08]. The algorithm is designed to be applicable on single ICs, allowing for miniaturization and energy efficiency. Even with the presence of noise the combined score for true negatives and true positives was as high as 91.3%. Using small and energy efficient systems also enables unobtrusive applications in non-mobile environments, e.g. placing such a system close to the bed to detect the breathing rate while the user is sleeping.

Collaborative applications are an important aspect of smart environments, e.g. to link together meeting places at different places, similar to the presented telepresence applications. If a multitude of speakers is present it gets increasingly difficult to provide a system that enables proper speech transmission for all participants. Using an array of microphones it is possible to focus the attention on the person currently speaking and filter out environmental noise. A project at Microsoft Research investigated using a maximum likelihood of two known techniques, beamforming and speech source localization, to enable a reliable speaker selection [ZFBZ08] (Figure 2.24 - Top Right). Additionally the framework enables a good adaptation even if directional microphones are used that are placed close to each other. The method provides a real life accuracy of more than 90%.

A fairly new work was performed by Xu et al. [XLL*13], called Crowd++. Their idea is to use smartphone microphones to identify the number of speakers in crowded environments. Such system could be used to estimate the number of persons in a given place and potentially react quickly if a crowd panic may occur. The method is based on an unsupervised machine learning classification of short audio parts that are picked up by the individual microphones in the handsets. It was tested by 120 participants in 10 different environments and allowed detecting the number of persons with an average error of 1.5 persons (Figure 2.24 - Top Left). No dedicated hardware is required to achieve this precision, enabling an application using off-the-shelf smartphones.

Microphones can also be used to analyze the mechanical surface waves that occur when objects interact with each other. Harrison et al. have designed TapSense, a microphone based sensor system that improves touch interaction by classifying the sounds created by different objects hitting the surface [HSH11]. In particular different parts of the hand can be distinguished, including nail, knuckle or tip. Potential applications are improving touch interaction on touch screens by enabling different forms of interaction, but also can be adapted to mobile devices, that typically have less interaction space and increase the expressiveness of different touch gestures. The achievable accuracy was between 95% using four different input types up to 99% when using just two input sets, such as finger and pen (Figure 2.24 - Bottom).

2.3.5. Radiofrequency sensing

Radiofrequency sensing is a traditional field for sensors. Radar is a system that uses radio waves to acquire direction, speed, distance or altitude of objects and was developed at the beginning of World War II. This variety is using active sensing and emits radio waves that are reflected by objects. Most applications in smart environments similarly rely on active systems. A popular signal source are WiFi signals, intended to wirelessly transmit

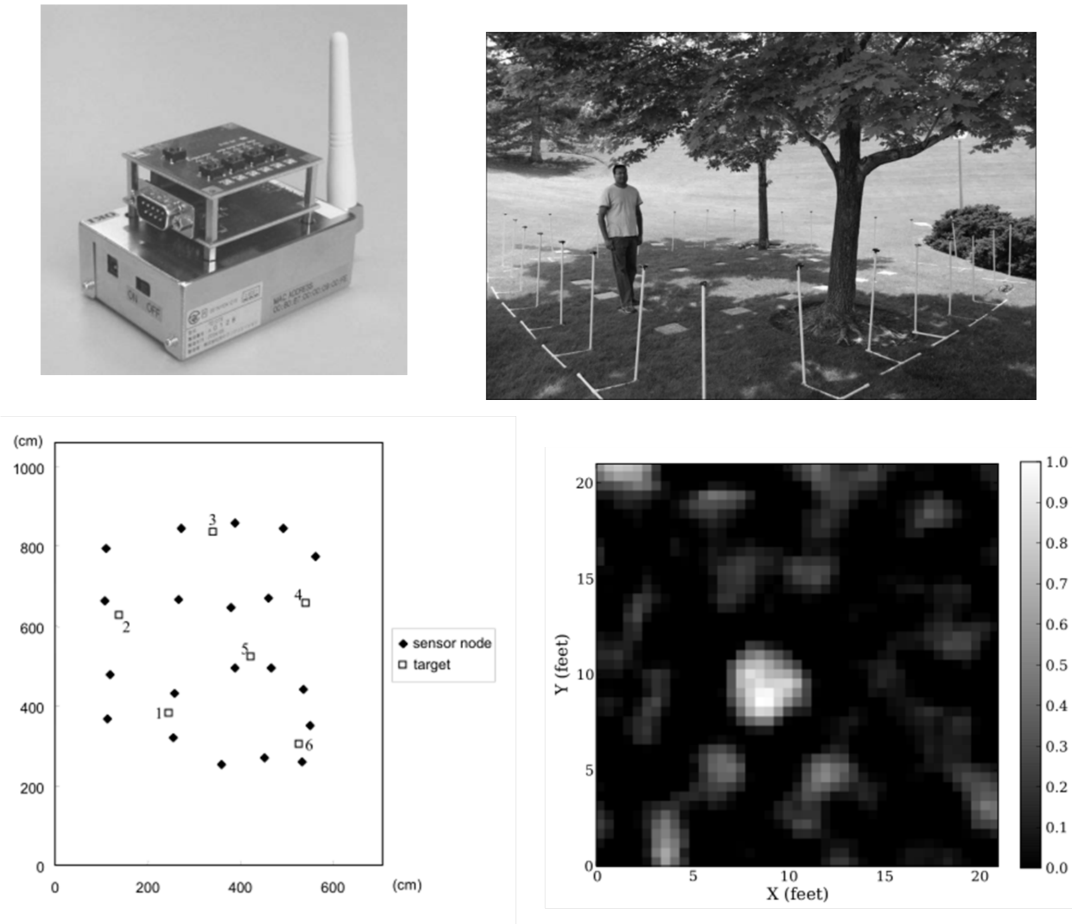


Figure 2.25.: *Top left* ZigBee node. *Bottom Left* Sensors and targets in larger room [SKOM06]. *Top right* Photo of radio tomography setup. *Bottom right* Result attenuated signal [WP10].

information between several systems. The systems are wide-spread, with all smartphones possesing two or more wireless technologies (typically 4G/3G/GSM for long range communication, WLAN for medium range communication and Bluetooth/NFC for short range communication) that can be used in coordination with stations placed in the environment. The signal processing of the wireless LAN sets generates some additional data that can be used, most notably the signal strength (RSSI). We will present different systems that show some different ways how radiofrequency sensing can be used in smart environments.

Radiofrequency based systems are very popular for indoor localization applications. We previously glimpsed at the difficulty of time-of-flight systems in the electromagnetic specturm. Thus most systems rely on a different approach, using the received signal strength (RSSI). If the initial signal strength is known and we have a good estimate how the signal propagates we can estimate the loss of signal strength at a certain distance. An accepted work that helped shaping this domain is the system created by Sugano et al. in 2006 that uses a ZigBee-based network with a limited number of nodes receiving RSSI information [SKOM06]. Based on this it is possible to locate one or more users with an error between 1.5-2m. This often is sufficient to distinguish the room a person is currently in, enabling a room-based system adaptation, which is suitable for many applications. A photo of a ZigBee node and results in a larger conference room are shown in Figure 2.25 on the left.

A different approach for RSSI based systems was presented by Wilson and Patwari in 2010 [WP10]. They are using tomography methods to determine the location of users. By placing a large number of sensors on the outer limits of the environment and creating a unique link between each, it is possible to infer the position based on the signal attenuation when a person moves in the environment. The human body absorbs some of the signal, resulting in a reduced received RSSI in the affected nodes. The error for standing persons was between 0.64cm for a single person and 1.10cm for two persons. An image of the test area and the resulting reconstructed attenuation map are shown in Figure 2.25 on the right.

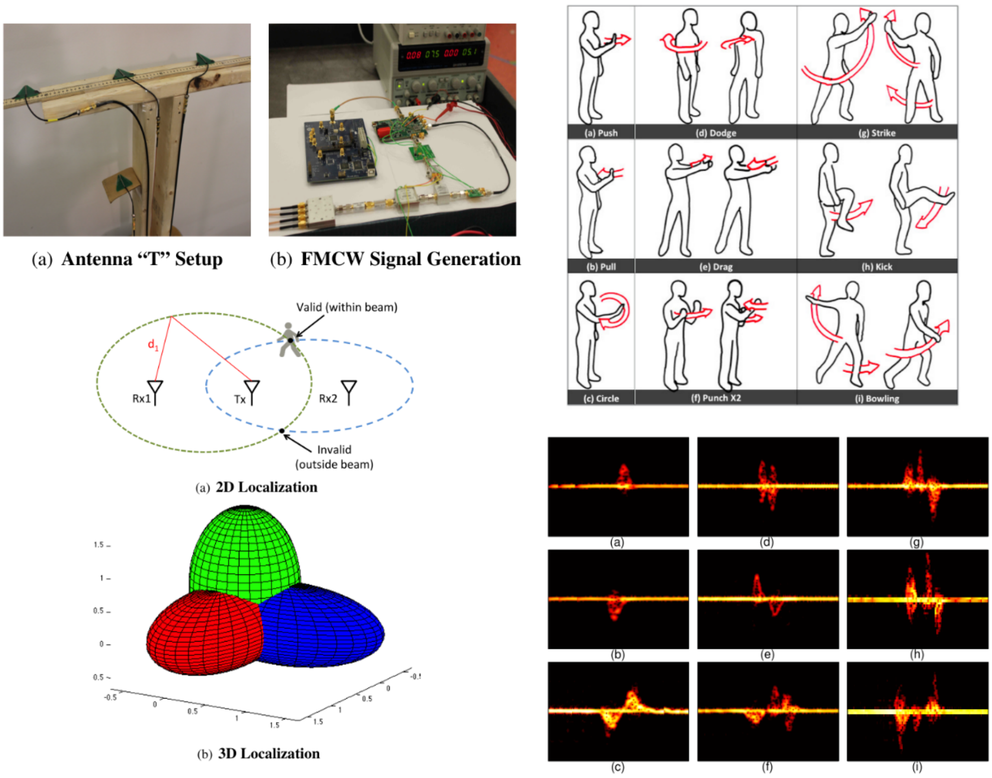


Figure 2.26.: *Top left* WiTrack antennas and signal generator. *Bottom Left* WiTrack 2D and 3D localization method [AKKM13]. *Top right* WiSee supported gestures. *Bottom right* WiSee Doppler profiles of gestures [PGGP13].

A final system that provides a localization is WiTrack, presented by Adib et al in 2013 [AKKM13]. It uses the signal reflected by the human body to provide a location estimate based on time-of-flight. As mentioned previously this is challenging due to the propagation speed within the electromagnetic field. To overcome this issue they are using a method called frequency modulated carrier wave that transfers time differences to the frequency domain. Looking at the spectrum of received signal these shifts can be analyzed. The resulting location has an accuracy of 10-13 cm in x and y and 21 cm in z dimension. Additional use cases are provided in determining coarse arm or foot gestures and enabling an accurate detection of falls of a person. Some limitations of this approach include a restriction to just one person and that the tracked object has to move. In Figure 2.26 on the left we can see pictures of the antenna setup, the signal generator and how the location is determined in two and three dimensions.

The final radiofrequency system I want to present is WiSee, developed at the University of Washington by Pu et al. [PGGP13]. They are using Wi-Fi to enable gesture recognition in large areas without requiring a line-of-sight. They are analyzing the Doppler shift resulting from human activity in order to classify different gestures. They are using MIMO (multiple input multiple output) to distinguish between non-interacting persons in the environment and the person performing gestures. To equip a whole appartment only a single receiver and two transmitters are necessary. The achievable accuracy varies according to persons present and number/type of devices involved but peaks at an accuracy of 94% for detecting nine different whole body gestures. The supported gestures and their Doppler profiles are shown in Figure 2.26 on the right.

2.4. Applications in smart environments

The field of smart environments is not strictly and conclusively limited and distinguished from other fields in technology, using influences from disciplines including electric engineering, behavioral psychology, computer science or mechanical engineering. Accordingly, it is difficult to formally list or distinguish all applications that are relevant or have been tackled in previous work. Thus, I will refer to previous collections of surveys, books and state-of-the-art in the associated disciplines smart environments, ambient intelligence and ubiquitous computing to get an informed selection of relevant applications that have been, or could potentially be supported by capacitive proximity sensors. The chosen collections of applications are taken from different collections of work that have been released in the past few years. Cook et al. that presented a survey on recent developments in smart environments research in 2007 [CD07]. Augusto et al. edited a book on ambient intelligence in 2009, including chapters on various domains and applications. Another source is the book *Ubiquitous Computing* by Poslad, released in 2011 [Pos11]. Additionally, I will take into account most recent focus areas, special issues and sessions of conferences and journals that are active in this regard, such as CHI, UbiComp, Ambient Intelligent International, the Journal of Ambient Intelligence and Smart Environments, or the International Journal of Human-Computer Studies.

The selected areas-of-interest are collected in Table 2.6. I will briefly introduce relevant work in those domains and outline existing links to capacitive sensing applications.

2.4.1. Indoor localization

The domain of indoor localization has been briefly touched, when discussing the different sensing technologies used in smart environments in section 2.3. The reliable tracking and localization of multiple users is a major challenge in smart environments. It is one of most important contextual information that can be used to adapt the behavior of the environment using a set of deployed actuators. Often basic motion sensors are used, e.g. if just a single person should be detected or if there is a single area of interest that should be covered. However, determining a more exact location or following multiple users requires more sophisticated systems. If a varying number of non-relevant actors are moving in the environment, e.g. pets the challenge increases further. Together with Tim Dutz, I participated in the EvAAL competition 2013 with our AmbiTrack system, reaching a second place, [BD13]. The system was based on a set of cheap off-the-shelf cameras connected to different processing nodes that employ an efficient tracking algorithm based on background subtraction, ray intersections and clustering methods [BDA*13]. It also supports analyzing the coverage of areas where it is installed, as shown in Figure 2.27.

Figure 2.27.: *Left:* Coverage of a room by camera fields of view - AmbiTrack [BDA*13]. *Middle:* Fiduciary marker photographed by camera phones [MWBS09]. *Right:* Magnetic coils in background and receiver circuit in foreground [PL13]

Table 2.6.: Application domains and relevant works

Application Domain	Relevant Works
Indoor Localization	Braun - AmbiTrack camera system [BD13] Chintalapudi - EZ Indoor localization [CPP10] Mulloni - signpost markers [MWBS09] Pirkl - resonant magnetic coupling [PL13]
Gestural interaction	Zimmerman - hand gesture [ZLB*87] Wilson - XWand pointing interaction [WS03] Majewski - visual feedback [MBMK13] Liu - uWave [LZWV09] Pu - WiSee [PGGP13]
Physiological sensing	Cowie - emotion recognition [CDCT*01] Khosrowabadi - EEG emotion [KQWA10] Wöllmer - multimodal emotion recognition [WME*10] Hoque - MACH [HCM*13] Koelstra - DEAP database [KMS*12]
Activity recognition	Bao - user-annotated acceleration data [BI04] Bulling - electrooculography [BWGT11] Lasecki - real-time crowd labeling [LSKB13] Oreifej - Hon4D [OL13]
Smart appliances	Gellersen - MediaCups [GBK99] Yeo - StickEar [YNR13] Tsai - Smart Medication Dispenser [TCY*11] Dementyev - Bistable Display Tags [DGT*13]
Mobile devices	Ballages - mobile devices in ubiquitous computing [BBRS06] Dearman - Bluetone [DT09] Vajk - TiltRacer [VCBE07] Nazari Shirejini - PECo [SN04] Olsson - mobile augmented reality [OKLVO12]
Autonomous systems	Coradeschi - symbiotic robotic systems [CS06] Broxvall - PEIS ecology [BGS*06] Arndt - robotic frameworks [AWD*13] Huijnen - companion robotics and smart homes [HBvdH*11]

There are various indoor localization and tracking approaches that have been proposed in smart environments, including specific competitions that benchmark the different solutions against one another [CK12]. A common discrimination of localization systems is based on the type of object an actor has to wear in order to be successfully recognized. Active marker-based solutions require the person to carry a token that actively sends out signals that are received by stations placed within the environment. Passive marker-based solutions require a token that is uniquely identifiable by external sensors, but is not sending any signal. Finally, there are marker-free solutions that do not require the actor to carry anything and instead rely on external sensors finding unique features of the actor and following those within the environment.

A common method is using radiofrequency sensing to track a mobile token, e.g. based on 802.11 WiFi networks. A recent work in this domain has been presened by Chintalapudi et al. [CPP10]. Their EZ system relies on an existing WiFi infrastructure, the user carrying a mobile device with WiFi that also has periodic access to a GPS signal. It does not require any prior mapping or knowledge about the specific location or transmit power of the access points, but instead relies on a genetic algorithms to determine potential locations from a limited set of known locations and distances to access points calculated using the received signal strength (RSSI).

Another system was presented by Mulloni et al. [MWBS09]. They are using a set of fiduciary markers that are placed on signposts and can be photographed with smartphones. Based on an integrated mapping application the software will provide navigation from the current signpost to the desired destination. Thus, this is an example for an indoor localization system that relies on external markers, requires the user to wear a token in form of a smartphone and active user participation when the markers are photographed. The setup is shown in Figure 2.27 in the middle.

A final example is a system created by Pirkl and Lukowicz based on resonant magnetic coupling [PL13]. Based on the magnetic field created by static coils an arbitrary number of mobile receivers can localize themselves by measuring the field strength and estimating the distance from the different static coils. The system can be calibrated to ignore other magnetic objects in vicinity and enables a planar localization of about $1m^2$ accuracy. The static coils can be seen in Figure 2.27 in the background with the receiver circuit in the foreground.

2.4.2. Gestural interaction

Gesture recognition enables the detection of meaningful expressions of motion by a human body, including the hands, arms, face, head and body [MA07] . If these expressions are translated into machine commands the result is gestural interaction. The most expressive and explicit form of gestures are performed by the hands, further distinguished into free-air gestures and touch gestures that typically involve one or more fingers interacting with a surface, the latter being called multi-touch.

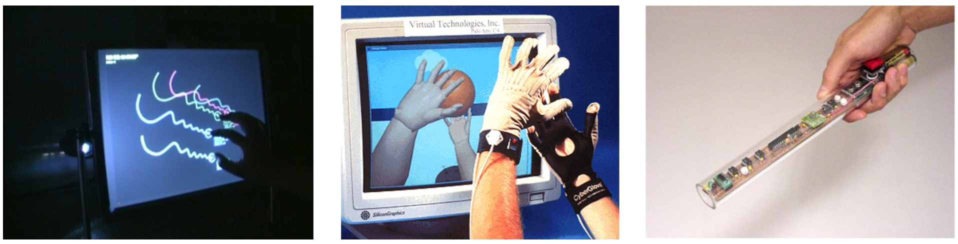


Figure 2.28.: *Left:* FTIR multitouch table [Han05]. *Middle:* DataGlove hand gesture system [ZLB*87]. *Right:* XWand interaction device [WS03]

Jeff Han showed a low-cost system based on frustrated total internal reflection (FTIR) of infrared light. This system allows tracking ten or more objects in real-time on large surface areas [Han05], as shown in Figure 2.28

on the left. Acoustic systems are another popular technology in this domain. Surface acoustic wave (SAW) uses the signal decrease of ultrasonic waves as they pass through an object touching the surface to infer its location [AAB98].

Throughout the years there have been various attempts to enable the tracking of gestures in free air. A common attempt in the late 1980s were data gloves, sensor-augmented gloves put on the hand that translated finger movements to gestural input, e.g. the system presented by Zimmerman [ZLB*87]. It uses optical sensors to measure the flex of the fingers and ultrasonic sensors to detect the absolute position and orientation of the hand. It is shown in Figure 2.28 in the middle. Applications included the evaluation of hand impairments and object manipulations in three-dimensional scenes.

A different approach is performing gestural interaction supported by interaction devices that can sense orientation and position in the room. A popular example is the Nintendo Wii Remote that is used in gaming applications. A predecessor was the XWand by Wilson et al. [WS03]. This interaction device is using accelerometers to sense orientation and has infrared diodes that are tracked by an external camera system to determine the absolute position in the room. It is shown in Figure 2.28 on the right. Using knowledge about the location of different appliances within this room, it is possible to control them by pointing in this direction. They later extended the system with a pointing device based on a laser that indicated the position in the environment currently pointed at [WP03]. We have used a similar method, based on skeleton tracking provided by a Kinect and different varieties of pointing gestures [MBMK13].

An extension of accelerometer-based gesture systems was presented by Lui et al. [LZWV09]. Their uWave system is capable of building a personalized gesture set using dynamic time warping and HMM classification, enabling a high recognition rate of almost 99% using just one training sample. Additional applications of this approach include using personal gestures for authentication, reaching an error rate of approximately 3%. However, this method can be bypassed if the movements are observed and mimicked and thus is not suitable for critical applications.

Another interesting system for whole body gestures is WiSee presented by Pu et al. [PGGP13]. Using two sources of wireless signals, they are using the Doppler shift caused by the human body moving in the area and reflecting the signal to determine gestures. The system was able to detect nine different full-body gestures with a precision of 94%. The system uses the MIMO capabilities of modern WiFi systems to distinguish users and providing the option to support multiple users within an environment.

2.4.3. Physiological sensing

In the 1990s researchers began to distinguish different channels when interacting with machines. The explicit channel, whereas the user gives distinct commands to the user, and the implicit channel that comprises information about the user himself [CDCT*01]. One interesting parameter to consider for this implicit interaction is emotion. Based on physiological cues it is possible to determine the current state of the user and translate it to an input machines. Thus, an area has emerged that uses acquired physiological signals of the users to provide additional input. In this section I will present four different works of recent years that proposed important technologies and methods.

Figure 2.29.: *Left:* MACH conversation coach system [HCM*13]. *Right:* Study setup to collect emotion data [KMS*12]

Khosrowabadi et al. presented a machine learning approach to discriminate four different emotions from EEG readings [KQWA10]. The subjects were asked to fill out self-assessment questionnaires regarding their emotional state with the states being induced by a set of pictures. The signals were classified using a self-organizing map and nearest-neighbor classification. The system was able to correctly classify 84.5% of four different emotional states.

A multimodal emotion recognition system based on speech and facial expression was introduced by Wöllmer et al. [WME*10]. They apply a method based on Long Short-Term Memory Networks that allows to model long-range temporal dependencies between features. They combine 46 facial markers and a variety of waveform speech features that are combined using a correlation-based feature subset selection that leads to a selection between 66 and 224 features. They are able to improve the recognition performance compared to HMM-based classifiers by 4% to 6%.

An example application for emotion-aware systems has been presented by Hoque et al. [HCM*13]. *MACH: My Automated Conversation coacH* is a training system that tries to improve the social skills of its users. It collects facial expressions, speech data and analysis prosody from an attached camera to create a personalized feedback of the users behavior when talking to a virtual agent. This agent is shown in Figure 2.29 on the left. A study with 90 participants showed a significant performance improvement as opposed to the control group.

In this domain it is also very important to build databases of labeled emotions and physiological signals that allows other researchers to try new methods. Koelstra et al. [KMS*12] created a database by collecting EEG information, playing different video clips and collecting the perceived emotions in a self-assessment. This database includes the physiological signals of 32 users and frontal face videos of 22 users that are reacting to 40 different videos. The setup of this study can be seen in Figure 2.29 on the right.

2.4.4. Activity recognition

Activity recognition, also called situation awareness, describes the interpretation of sensor data into higher domain-level information that relates to the given situation [YDM12]. To give an example, a single temperature reading of 15 °C can be combined with the information that the sensor is in the sleeping room, the time is 21 : 00 and we know that the inhabitant of the environment is typically going to sleep at 22 : 00. In this case we can infer the activity *user will go to bed soon* and trigger the action *heat up sleeping room*. There is a plethora of different concepts and methods on how to recognize the situations and activities for numerous domains. A good overview of the topic can be found in the survey of Ye et al. [YDM12]. I will present four different works that have been presented in recent years that provide a different selection of technologies and methods that are commonly used.

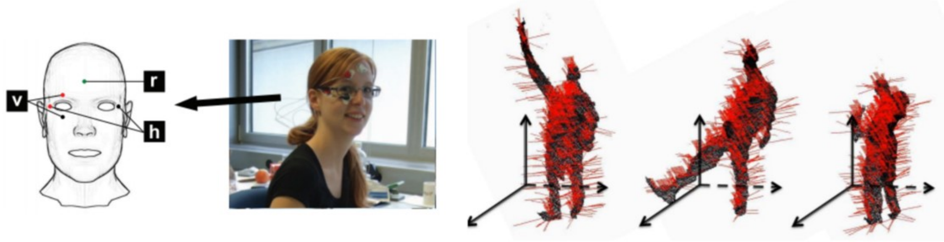


Figure 2.30.: *Left:* Eye movement tracker by Bulling [BWGT11]. *Right:* Hon4D - surface normals of different activities [OL13]

When trying to determine human activities wearable sensors are an important approach that has been used extensively. It is possible to use either physiological measurements or movement data. A classic work in this domain presented by Bao et al. is associating activities based on movement data generated by accelerometers [BI04]. This sensor category is able to detect the acceleration in multiple directions, e.g. commonly used to detect the orientation of mobile devices. Using five different sensors attached to arms, legs and hip, they were able to associate 20 different activities performed by 20 different subjects with an accuracy of approximately 80%.

In some application domains that have a specific set of tasks other approaches might be suitable. An interesting system was presented by Bulling et al. that tries to determine activities based on the movement of the eyes [BWGT11]. They are attaching a set of electrodes to the user's head to track the activity of muscles around the eye, without using any external sensors such as gaze trackers. The setup and electrode placing is shown in Figure 2.30 on the left. In a study with eight users they tried to distinguish five different typical office work activities using SVM classification - copying text between documents in a two monitor setup, reading a document on the table, writing on a page on the table, watching a video and browsing the internet. The achieved average precision was 76% and recall 71%.

A common problem in situation classification is labeling of the acquired data to a given situation. Typically this is performed manually by the researcher. Lasecki et al. introduced Legion:AR, a system that allows to crowd-source this process to a large group of workers that individually label activities from a video feed [LSKB13]. They could show that while in complex situations a group of five persons labeled 90% of activities and objects correctly, while a single person only reaches 48% on average. The process is aided by integrating the input from different persons that are currently labeling.

A final example in the domain of activity recognition is Hon4D, a system that infers situation from depth camera sequences [OL13]. They suggest the histogram of normal orientation in depth, time and spatial coordinates as a feature for activities from depth data. Examples of surface normals for different activities and body surfaces are shown in Figure 2.30 on the right. The advantage of this operator is that it takes into account the movement of the overall surface of recognized persons, as opposed to silhouettes or the reconstructed skeleton joints. This approach reaches classification precision between 89% and 97% when used on common activity depth data sets.

2.4.5. Smart appliances

Smart appliances are devices that are attentive to their environment [SV01]. This is usually achieved by integrating different sensors and actuators to provide additional functions and services to a user. Some examples include intelligent furniture that can detect their occupation, internet-connected household items, or single-purpose devices, e.g. providing reminder services. An overview of different examples can be found by Park et al. [?].

Figure 2.31.: *Left:* Different MediaCup prototype [GBK99]. *Right:* Stickear system to augment appliances with audio sensing [YNR13]

One example for common household items augmented with additional features is the MediaCup by Gellersen et al. [GBK99]. This coffee cup is augmented with temperature and acceleration sensors, a processing unit and communication using an infrared system. Some of the prototypes are shown in Figure 2.31 on the left. It is able to sense if there is fresh coffee in the cup, if it currently used for drinking, stationary or being played with. The applications were focused on remote colleague awareness, whereas the activities of the MediaCup were transmitted to a remote location.

StickEar by Yeo et al. is a wireless device that adds different capabilities to objects it is attached to [YNR13]. They integrate a rotary switch, buttons, microphone, speaker and processing and communication components into a small portable package, as shown in Figure 2.31 on the right. Some supported interactions are control of devices using sound, autonomous response to sound events, or remote triggering of sound. The system can be controlled using an app for mobile devices.

There is a number of smart appliances in the medical domain. They try to provide different services related to health and well-being. A common example are smart medication dispensers, such as the one presented by Tsai et al. [TCY*11]. Based on a medication schedule it will dispense the right medication and dose at the specified times. They include a few different algorithms for heuristic scheduling based on a collaborative approach between scheduler and the controller of the dispenser.

When deploying smart appliances it is necessary to provide energy to the systems. If there is no socket nearby batteries or other methods have to be used. If the system is capable to independently harvest the energy from environment sources. One example are the bistable display tags by Dementyev et al. [DGT*13]. They rely on e-paper displays that keep their screen content unless a refresh is triggered. Using an energy harvesting IC it is possible to transfer enough information and the new screen content via NFC. In an example application screenshots from a smartphone could be transferred to the display.

2.4.6. Mobile devices

In his famous article, Mark Weiser coined three different forms of devices - tags, tabs and boards [Wei91]. They are primary distinguished by their size and how they can be used for interaction with the system. The prevalet smart phones and tablets nowadays resemble closely the envisioned smart tab - handheld devices that provide sensing, interaction and communication facilities supported by sufficient processing power. Consequently, they are used very often in smart environment applications.

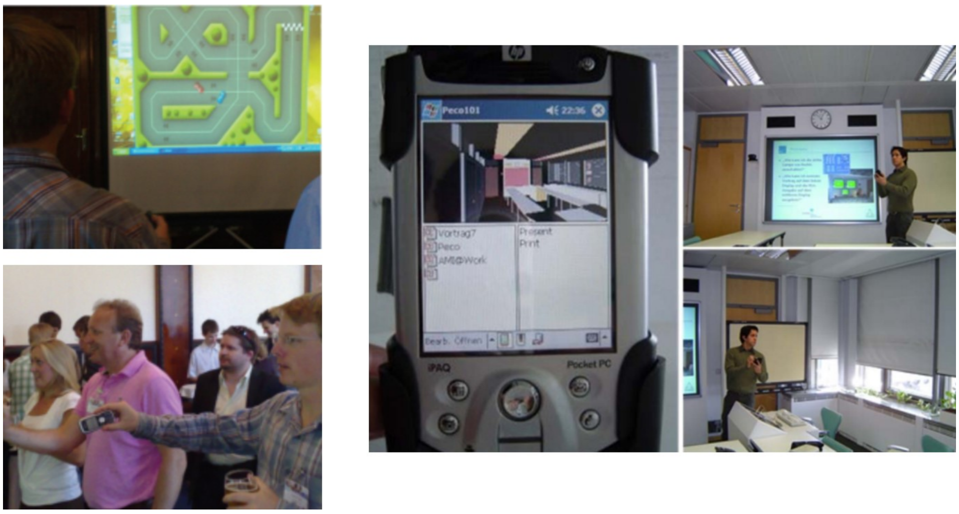


Figure 2.32.: *Left:* TiltRacer controlled by accelerometers of mobile phones [VCBE07]. *Right:* PECo environment control application on a PDA [SN04]

Ballages et al. collected and investigated numerous ways how smart phones can be used to in ubiquitous computing [BBRS06]. They provided an overview of different positioning, orientation and selection methods, including using the camera on the mobile phone and different processing methods to control a cursor. An important contribution was a classification of potential mobile phone interactions and a summary of of position, orientation and selection techniques.

Bluetone is a system created by Dearman and Truong that allows to control public displays using mobile phones via Bluetooth communication [DT09]. In order to avoid installing any additional application they are using the Bluetooth headset profile and transfer sounds instead of packaged information. Similarly Vajk et al. proposed using the accelerometers in a phone to control applications on a large public display [VCBE07]. One example application was TiltRacer that could be controlled by different users standing in front of a display, as shown in Figure 2.32 on the left.

Nazari Shirejini presented PECo, an environment controller based on a PDA device [SN04]. Based on a 3D model of the current environment it was possible to control different appliances by selecting them on the mobile device. Additionally, a concept was presented that allows to transfer documents from the PDA to different suitable devices, by using simple drag & drop operations. The practical use case was implemented in a lecture room and connected to the controlling system, e.g. allowing to display documents on a projector by dropping them on the 3D model of the projecting screen. The system is shown in Figure 2.32 on the right.

A final example is in the popular domain of augmented reality systems for mobile devices. There are numerous popular applications that overlay additional information on a live camera image of mobile phones. Olsson et al. tried to evaluate user experience and acceptance of different mobile augmented reality applications [OKLVO12]. They found that information applications were better received than entertainment applications. In addition, information flood, loss of autonomy and virtual replacement of actual items were seen as most negative aspects of this technology.

2.4.7. Autonomous systems

Autonomous systems are an important future application in smart environments. While factory complexes started using robots decades ago, the trend towards home robotics is fairly recent, as the processing and sensing capabilities of the systems increased, while the price could be reduced significantly. There are numerous potential use cases, ranging from vacuuming or gardening robots, to full-service robots that can be used in care systems for the elderly. In this section I will present four different examples, how autonomous systems can be used in smart environments.

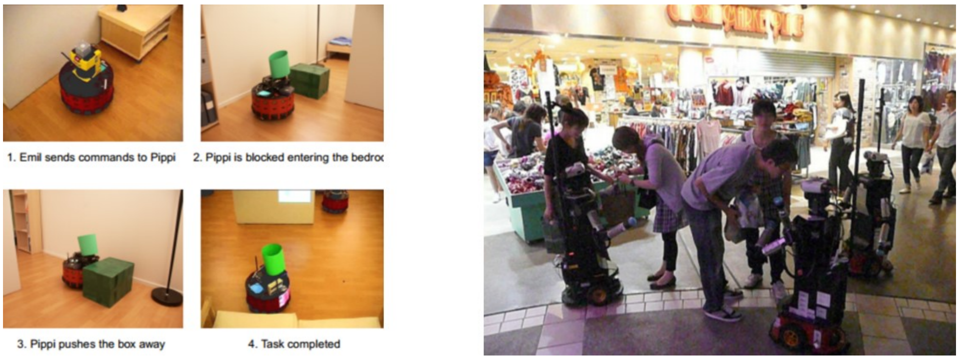


Figure 2.33.: *Left:* Scenario of the PEIS system [BGS*06]. *Right:* Robots moving around in a social environment [GKIH09]

Coradschi and Saffiotti propagated the symbiotic properties of robots operating in a smart environment [CS06]. Human, robot and smart environment are modeled as three distinct actors that share resources and information between each other. The distributed systems in the environment, e.g. sensors, tags and actuators, can deliver additional information to the control unit of the autonomous system that can be used to optimize any strategies. Similarly, users can benefit from the coordination between robot and environment to achieve common goals.

An implementation of a similar concept was published by Broxvall et al. [BGS*06]. PEIS, a network of heterogeneous smart devices that operate in a smart environment. In the example scenario shown in Figure 2.33 on the left. A coordinating system (Emil) sends commands to a robot (Pippi) that moves around the house but has his path blocked by a parcel. As this parcel is a smart object and knows its contents this information is forwarded to the robot, who then safely moves the parcel away. Thus, a collaboration between smart environments and robots can support similar scenarios that reduces the requirements for both environments and robots.

Glas et al. investigated how robots can track people and localize themselves in social environments, that are frequented by a larger number of persons [GKIH09]. They combine laser range-scanners placed in the environment that provide wide coverage and are able to track the trajectories of moving persons and robots. This is combined using Kalman filtering with the odometry data generated by the robots. As a combined solution this enables localization of both people and robots in a shared global coordinate system. One example application system is shown in Figure 2.33 on the right, whereas robots provide directional information to shoppers in a mall.

A final example that evaluates user experiences gathered in different research projects was presented by Huijnen et al. [HBvdH*11]. Within the projects CompanionAble and Mobiserv different service robots were deployed in smart home environments for elderly persons. The user trials involved testing interaction and usability, user acceptance and privacy. The robots user - Hector and Kompai - are designed to provide assistance for persons with Mild Cognitive Impairments or early stages of dementia. One finding was that the appearance of the robot was not particularly important for this user group, whereas the functionality is those users, even though

3. Benchmarking model for sensors in smart environments

When designing a new application or system for a specific purpose, the parties involved have to make a number of decisions regarding the different components, processes and methods that are to be used. Benchmarking is a method mostly used in business practice to compare the performance of processes, products and market entities against one another. A single or a set of different indicators are used to act as metric or calculate an overarching metric of performance that can be compared to other entities [CC89]. This tool is widely used for supporting decisions in different domains. Looking at smart environments, a common challenge is to select a specific sensor technology for any given application. While the majority of systems are following a structured approach in the design process, e.g. by ranking available systems or performing an iterative trial & error routine, so far there has been no generic model that would allow to evaluate the expected performance of a system based on a specific sensor technology. Using a set of common features and an adaptive weighting model it is possible to cover a high number of different applications in a specific domain and thus support the decision process at an early stage of the application design.

In this chapter I will introduce a formal benchmarking model that allows estimating the performance of applications in smart environments based on a specific sensor technology. To build a base for this modeling it is necessary to look into previous works related to technology benchmarking and the selection of specific metrics. Additionally, it is necessary to find a common set of features relevant for smart environments that can be applied to a number of different sensor technologies. The method is evaluated by benchmarking several example technologies and applications based on querying large scientific databases. Finally, the benchmarking method is used to determine different applications suitable for capacitive proximity sensors.

Benchmarking is a tool that is widely used in computing technology [LC85]. Hardware benchmarks compare the performance of different single systems, often seen for GPUs or CPUs to evaluate both theoretical and real-life performance. Some metrics that are used for theoretical comparison in CPUs are FLOPS (floating point operations per second), e.g. measured by Linpack [DLP03], or MIPS (million instructions per second), e.g. measured by Dhrystone [Wei84]. Regarding GPUs the benchmarks include Texel rate (how many triangles can be processed per second) and Pixel rate (how many pixels are processed per second. Real-life benchmarks for CPUs typically included timing specific tasks on applications that are demanding for certain aspects of the CPU, such as video processing, image processing or audio encoding. For GPUs many PC games provide benchmarking tools that allow evaluating the real-life performance of different graphics cards at different settings, e.g. resolution or detail level. The typical metric here are FPS (frames per second) that denote how often the screen content can be rendered in a second. System benchmarks are a step up from single component benchmarks and combine the performance measurements of various components in different scenarios to evaluate the estimated behavior in numerous real-life situations. There are several standardized test suites that provide this functionality, such as SPEC [Hen00]. A common single index that is available for all newer Windows machines (Vista and beyond) is the Windows System Assessment tool that calculates the WEI (Windows Experience Index), a combined score of CPU performance, 2D and 3D graphics performance, memory performance and disk performance [Mic]. The lowest score of all single metrics is chosen to determine a lower bound for expected real-life performance. Ranganathan et al. introduced benchmarking methods and a set for pervasive computing systems [RAMB*05]. They distinguish system metrics, configurability and programmability metrics and human usability metrics. Finally, if different systems of the same category are compared, technology reviewers often use a single index that is calculated based on various aspects of the system. Smith introduced different potential combined metrics that can be used for this purpose [Smi88]. Three different approaches are mentioned, geometric mean, arithmetic mean and harmonic mean. Additionally varieties with a specific weighting are mentioned. Another example for benchmarking whole systems is the EvAAL competition that aims at evaluating different technologies that are applicable in Ambient Assisted Living [CK12]. There are various tracks, including indoor localization and activity recognition. Apart from technical metrics, such as precision, a focus of this competition is on a more holistic approach and thus includes metrics like installation time, user acceptance and interoperability of the solution. Finally there has been considerable work in the domain of identifying suitable metrics for a given benchmark. Crolotte argued that the only valid benchmark for decision support systems is the arithmetic mean of different single benchmark streams, as it is valid for normalized and time-relevant benchmarks [Cro09]. Jain and Raj dedicate several chapters of their book to introduce methods and considerations for metric selection in benchmarking computer systems [Jai91].

3.1. Sensor features

One of the most challenging aspects of benchmarking is selecting the appropriate metrics to be included in the scoring process. In order to identify relevant sensor features for technologies to be applied in smart environments, inspiration is taken from sensor technology overviews [Wil04] and the pervasive model presented by Ranganathan et al. [RAMB*05]. Accordingly, it is possible to identify three different groups of sensor features: sensor performance characteristics, pervasive metrics and environmental influence. These different groups are detailed in the following sections. An overview of the different potential members of this groupsis given, their relevancy for the benchmarking model is discussed and a feature matrix is created that builds the basis for the feature scoring model.

3.1.1. Sensor performance characteristics

This group of sensor features is related to specific technical properties of the given sensing device, as they would be typically put into the datasheet. A first important characteristic is the sensitivity or resolution of a sensor, which is the smallest change of a measured quantity that is still detectable. For example an accelerometer might be able to only detect changes that are above 0.1g. Another important characteristic is the update rate of a sensor. This denotes the number of samples the sensor is able to measure in a certain timeframe. Typically, the number of samples in a second is noted as frequency, thus a sensor may have an update rate of 20 Hz, generating 20 samples in a second. Another factor that is particularly important for embedded systems or wearables is the power consumption of the sensor that may limit the time it can operate on battery, independent of a power source. A last example is the detection range, denoting the maximum distance between the quantity to be measured and the sensing device.

3.1.2. Pervasive metrics

Pervasive metrics can be identified as features that specify how well a given sensor system will perform in collaboration with smart environments, when networked with other devices and when placed into existing surroundings. An example for the latter is the obtrusiveness of a sensor device. If it is clearly visible when applied, if there are disturbing signals generated, or if certain privacy concerns are associated to the sensor device, the acceptance by the user and thus the applicability is reduced. If the sensor is operating in a larger network of other devices, the bandwidth required to submit signal to an analyzing node should be kept low. Equally, if the processing capabilities are limited, less complex data processing is preferable. The overall system cost is increasing if single sensors are particularly expensive, thus limiting the potential applications. The system and attached sensors should be robust, both in terms of physical design and quality of service. Finally, the sensors are more readily applicable if the systems are interoperable to each other.

3.1.3. Environmental characteristics

The third group is the environmental characteristics of a sensor system. Any sensor is affected by a certain disturbance caused by factors in the environment that are similar to the measured quantity, also called noise. For example an optical sensor is influenced by ambient light sources. In this context it is relevant how frequent those influences are in a certain environment and how robust the sensor is against noise. In many cases the presence of noise can be detected and counteracted with a calibration towards the changed environmental factors. The complexity of this calibration is another interesting factor in this regard. Finally, all sensors have some unique limitations, e.g. specific materials that absorb certain wavelengths of the electromagnetic field are difficult to detect for sensors that work in this specific frequency range.

3.1.4. Discussion and feature matrix

To keep the benchmarking constrained the three most relevant features of each category are chosen. This allows a more manageable overall model, however, requires an informed selection of the presented features. Of the sensor performance characteristics group resolution, update rate and detection range are selected. Resolution is a critical feature in any application, determining precise any detection is and if particular objects may be detected at all. Update rate is equally important if fast objects are to be detected and if a reactive systems is desired that respond in real-time. The importance of detection range correlates with the size of the environment and may lead to a reduction of required sensors. Of the mentioned features power consumption is omitted. The actual power consumption of a whole system is a more interesting metric but very difficult to predict from the energy usage of a single sensor [LWG05]. Of the pervasive metrics group unobtrusiveness, processing complexity and robustness are chosen. Unobtrusiveness of the sensor device is a desired feature in many different scenarios, where it should not impede the environment. While microprocessors are becoming ever faster processing complexity is still crucial if the number of sensors is increasing. A dedicated chip will require a more complex architecture and lead to more cost, higher energy usage and more potential points of failure, leading to the final chosen feature of robustness, both against physical abuse, but also in terms of system design, where it is supposed to be resilient towards failure of single components.

Additionally the required bandwidth was omitted, as this metric is not important for many sensors, as they have low bandwidth requirements in general, but also the available bandwidth in wired and wireless systems is increasing continuously. In the last group of environmental characteristics frequency of the disturbing factor, calibration complexity and unique limitations are chosen. If the disturbing factor occurs only rarely it is not critical and therefore should be part of the benchmark. Calibration complexity combines both the processing complexity and time that is required to recalibrate the system. This is highly important in real-time systems that have to monitor the environment continuously. Finally, unique limitations are a rather broad metric that is difficult to quantify. However, in many scenarios it is obvious that a specific limitation might arise, e.g. if the smart environment is in an area with a lot of human noise, microphones could be regularly disturbed. Including this metric allows modeling those applications into the benchmark with a strong weight penalizing unsuited sensors. From the selected metrics a feature matrix is created that allows to associate specific capabilities to a

Table 3.1.: Feature matrix denoting capabilities required for a certain rating

Feature	–	-	o	+	++
Resolution	very coarse	coarse	normal	fine	very fine
Update Rate	less than once per second	slower real-time	real-time	faster real-time	more than 100 times per second
Detection Range	touch	less than one meter	less than 5 meters	less than 20 meters	more than 20 meters
Unobtrusiveness	open large system	open small system	hidden system, large exposure	hidden system, small exposure	invisible
Processing Complexity	single sensor CPU	10+ sensors CPU	single sensor embedded chip	10+ sensors single embedded chip	no further processing
Robustness	single point of failure	error detection	quality of service	self-recovery	fully redundant
Disturbance Frequency	very frequent	frequent	average	unlikely	highly unlikely
Calibration Complexity	very hard	hard	normal	easy	very easy
Unique Limitations	very critical	critical	average	not critical	none

specific rating that will be used later in the scoring process of the benchmark model. Each feature is mapped to five different ratings on an ordinal rating scale comprised of the items “least favorable” (–), “not favorable” (-), “average” (o), “favorable” (+) and “very favorable” (++). This leads to the feature matrix shown in Table 3.1, which will be discussed briefly.

- *Resolution* is ranging from “very coarse” to “very fine”. This unspecific rating is used, as the range may vary strongly between different sensor types. A mapping to actual should depend on the application and object that has to be detected. If the object is large a sensor that would be ranked “coarse” for smaller objects can be ranked as “fine”.
- *Update Rate* is rated around real-time performance that is often rated at around 20 samples per second. Slower sensors might miss various events, while faster sensors allow detecting highly dynamic events.
- *Detection Range* is rated around the 5m distance mark, that is typically enough to cover the entirety of a single apartment room. For larger rooms sensors with a higher detection distance are favorable, many sensors only react to touch.
- *Unobtrusiveness* is ranging from exposed systems placed in the environment (one example would be the Microsoft Kinect) to invisible systems that integrate seamlessly into the environment.
- *Processing Complexity* has a range from dedicated CPUs that are required to process the data of a single sensor to smart sensors that require no further processing, which allows to apply numerous sensors without adding additional processing capabilities to the environment.
- *Robustness* is following criteria for quality of service. The least favorable system fails, if only a single node is present and failing. The preferred system is fully redundant.
- *Disturbance Frequency* is ranging from frequently occurring disturbing signals, to highly unlikely disturbing signals, resulting in a better rating.

- *Calibration Complexity* is a combined metric including the calibration time, the required processing capabilities and if external aid is required in the calibration process, leading to a rating from "very hard" to "very easy".
- *Unique Limitations* should be ranked according to their criticality, as previously explained they may be suitable to penalize certain sensors or emphasize the prevalence of a disturbing factor in a noisy environment.

Now that the feature matrix is complete, the next step is presenting the formalized benchmarking model and how to use the presented features and their rating to calculate a benchmark score that allows comparing different sensor categories with regard to different applications.

3.2. Benchmarking model

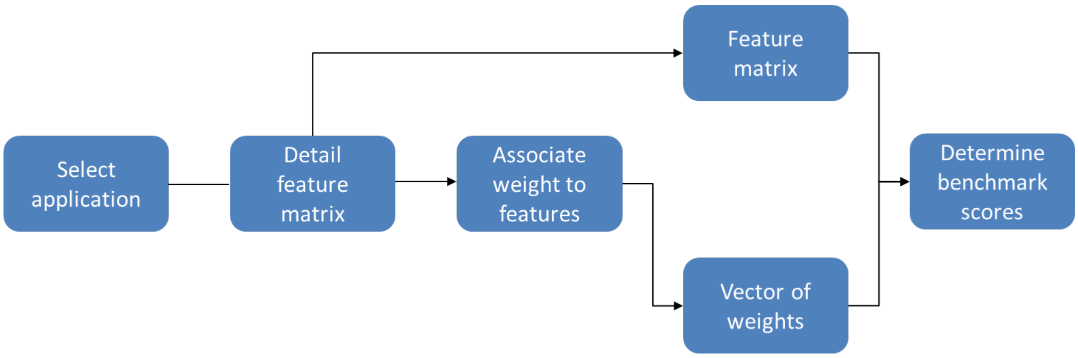


Figure 3.1.: Benchmarking process

In this section I will describe a formal model that will allow us to determine a benchmark score for a given application and a given sensor technology. As previously explained the different applications are distinguished by applying a different set of weights to the known features. As a start the process of this feature weighting is discussed and some examples about proper application are given. Afterwards, the formal model is introduced that deduces a single score benchmark for any sensor technology and any application. The overall process is shown in Figure 3.1 and will be detailed in the following sections, including some example calculations.

3.2.1. Feature score and weighting

The presented feature matrix has some ratings that need detailing in order to be quantifiable in the specific application. The ordinal measurements of the feature matrix should be assigned a quantifiable measure. Taking "Unobtrusiveness" the open system can be detailed as "visible by users" and "large system" as size larger than 100 x 100 x 100 mm. Similar level of details can be applied to the other features leading to the application-specific detailed feature matrix that is used in the scoring process. The different ratings are assigned different numeric values, namely 0.00 (–), 0.25 (-), 0.5 (o), 0.75 (+) and 1.00 (++). The weight of the features for the specific application is also rated on a 5-point ordinal scale, denoted as "not important" (numeric value 0.0), "less important" (0.25), "moderately important" (0.5), "important" (0.75) and "very important" (1.00). Thus for each application there is a distinct detailed feature matrix and a vector of associated weights that can be applied to a set of sensor technologies in order to calculate the benchmark score. The next step is to formally introduce the model that allows to determine a benchmarking score from feature ratings and application weights.

3.2.2. Modeling

The model is supposed to formalize a benchmark for any application and any sensor technology in any domain. As a start the following definitions are given:

- Set of n domains $D = \{d_1, ..., d_i, ..., d_n\}$
- Set of m applications $A = \{a_1, ..., a_j, ..., a_m\}$
- Set of o features $F = \{f_1, ..., f_k, ..., f_o\}$
- Set of p sensor technologies $S = \{s_1, ..., s_l, ..., s_p\}$

In any domain d_i there is have a set of potential applications $A_{d_i} \subseteq A$ and a set of relevant features $F_{d_i} \subseteq F$. For each feature $f_{k,d_i} \subseteq A$ there is the associated feature score $r_{F_{d_i}}$ as explained in the previous section. Each sensor technology s_l has a specific feature score $r_{s_l,F_{d_i}}$ with normalization $\left\|r_{s_l,F_{d_i}}\right\| \varepsilon[0,1]$.

$$\overrightarrow{r_{s_l,F_{d_i}}} = \begin{pmatrix} r_{s_l,f_{1,i}} \\ \vdots \\ r_{s_l,f_{o,i}} \end{pmatrix} \tag{3.1}$$

The weights w_{f_o} associated to a specific application a_j in a domain d_i have the same cardinality $|w|$ as the vector of feature scores $\left|\overrightarrow{r_{s_l,F_{d_i}}}\right|$. The values are determined, so $\|w_{f_o}\| \varepsilon[0,1]$.

$$\overrightarrow{w_{a_j}} = \begin{pmatrix} w_{f_1,a_j} \\ \vdots \\ w_{f_o,a_j} \end{pmatrix} \tag{3.2}$$

The feature scores and associated weights allows to determine a benchmark score b_{s_l} for a specific sensor technology s_l for any application a_j by using the scalar product of feature score and respective weight and apply normalization with regard to the weight.

$$b_{s_l} = \frac{\overrightarrow{r_{s_l,F_{d_i}}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^{o} w_{f_k,a_j}} \tag{3.3}$$

Thus it is possible to compare different sensor technologies by calculating and comparing the different benchmark scores for a given set of sensor technologies $S_p \subseteq S$ and receive a set B_{S_p} with $t = |s_p|$.

$$B_{S_p} = \{b_{s_l,1}, \ldots, b_{s_l,t}\} \tag{3.4}$$

In order to determine the optimal (chosen) sensor technology b_c for an application a_j and given the prerequisites regarding non-negativity of weights and feature scores we can evaluate the set for the maximum element.

$$b_c = max(B_{S_p}) \tag{3.5}$$

3.2.3. Feature score normalization

With regards to actual benchmarking the problem of bias towards a specific technology may occur. If the average features ratings are different between two technologies the calculated benchmark score will increase. In many instances this might be beneficial, yet if comparing numerous technologies to a set of different applications a trend might be more important than absolute scores. Thus, an optional step is provided that calculates the normalized feature vector $r_{s_l,F_{d_i},norm}$ with regard to the average associated value of 0.5, using the following equation.

$$\overrightarrow{r_{s_l,F_{d_i},norm}} = \begin{pmatrix} r_{s_l,f_{1,i}} \\ \vdots \\ r_{s_l,f_{o,i}} \end{pmatrix} \cdot \frac{o \cdot 0.5}{\sum_{p=1}^{o} r_{s_l,f_{p,d_i}}} \tag{3.6}$$

The feature-normalized benchmark score is accordingly determined with the following equation.

$$b_{s_l,norm} = \frac{\overrightarrow{r_{s_l,F_{d_i},norm}} \cdot \overrightarrow{w_{a_j}}}{\sum_{k=1}^{o} w_{f_k,a_j}} \tag{3.7}$$

3.2.4. Scoring

Now with the formal model and the available set of feature matrix and weights it is possible to calculate the benchmarking score for a set of sensor technologies. As an example the application indoor localization in a public shopping area is chosen, in order to monitor customer behavior. The requirements include a tracking accuracy of 50 cm, with a large area to cover and potentially fast moving persons. Thus the importance ratings for performance characteristics are moderately important for resolution, important for update rate and very important for detection range. The system can also be used for security purposes, thus unobtrusiveness is less important. There can be dedicated servers, so processing complexity is not important, but the system should be difficult to disturb, thus robustness is important. Disturbance frequency is moderately important, as a large number of per-sons is monitored, leading to statistically significant results, even if single measurements are disturbed. The environment is fairly static, thus calibration complexity is less important. It is possible that a crowded shop produces a lot of acoustic noise, therefore no unique limitations towards acoustic disturbances should be present and this is moderately important. The resulting vector of weights is:

$$\overrightarrow{w_a} = \begin{pmatrix} 0.50 & 0.75 & 1.00 & 0.25 & 0.00 & 0.75 & 0.50 & 0.25 & 0.50 \end{pmatrix}^{-1} \tag{3.8}$$

Based on previous experiences and best practice a camera-based system is evaluated. The system has high resolution cameras, with an update rate of 30 samples per second and a high detection range of more than 20 meters. The cameras are external, not hidden from view but attached on the ceiling. The processing complexity is very high, requiring a dedicated system per camera. Since they are out of reach they are robust towards human intervention and independent from each other. In the given setting visual disturbance is unlikely, calibration is difficult but not required regularly and the system is not disturbed by acoustic noise. This results in the following rating vector:

$$\overrightarrow{r_{s,f}} = \begin{pmatrix} 1.00 & 0.75 & 1.00 & 0.25 & 0.00 & 0.50 & 0.75 & 0.25 & 1.00 \end{pmatrix}^{-1} \tag{3.9}$$

Using those two vectors the final scoring for this sensor system can be calculated, using the equations of the previous section, leading to $b_s \approx 0.78$ and a feature-normalized score of $b_{s,norm} \approx 0.63$. Determining the feature rating vector for other technologies is possible in a similar fashion. The optimal technology would have the highest score b_s or $b_{s,norm}$.

3.3. Evaluating technology popularity

In order to evaluate the method I propose a verification based on previous successful works in the domain of smart environments. Three different application areas are selected and for each benchmark three different sensor technologies. In order to estimate how popular a certain technology is for a given application the ACM Digital Library is used to query scientific publications with respective author keywords. This method is limited, as the chosen keywords may not catch all relevant publications. Therefor the focus is slightly increased by using multiple associated search terms for each application and technology. Additionally, respective searches are also performed using the Google Scholar database that has a much broader scope. The advantages of the latter are the huge collection of scientific resources and no strong selection bias. However, there are various associated issues that may affect the method. The search results vary on the search term, additionally there will be results that mention the search term but do not necessarily rely on the technology for their respective system. Therefore, the results should be considered as an indicator for popularity in the research community. Similar to the ACM DL search different synonyms are considered and an average between the search results is calculated.

The chosen applications are hand gesture interaction, a marker-based identification system and obstacle avoidance for an autonomous system. The technologies are camera systems, radio-based systems, depth or stereo cameras and ultrasound devices.

3.3.1. Scoring

Table 3.2.: The importance weighting of different applications, based on the features.

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Hand Gesture	++	++	-	-	-	o	+	o	o
Identification	–	-	++	++	o	++	+	-	+
Obstacle Avoidance	-	+	-	o	+	+	++	++	+

At first the weights of the different applications with regards to the features are determined. The results are shown in Table 3.2. For the tables in this section the short notation of the features is used, in order of appearance in Section 3.2.1. The rating of the different technologies and the resulting score is shown in Table 3.3. Here

Table 3.3.: Feature rating of the different sensor technologies

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Camera	++	o	+	-	o	o	o	-	o
Radio	-	+	++	+	o	o	o	o	-
Depth camera	+	o	o	-	-	o	-	o	o
Ultrasound	-	+	o	o	+	o	+	o	o

it is possible to follow different strategies regarding the rating. In terms of unbiased comparison looking at the equations it would be necessary that all technologies have the same average feature rating. The second strategy is to apply an absolute ranking to all technologies, independent of the given application. This might lead to certain technologies being unsuited for a given task, or technologies that have the best benchmark score regardless of application. In this specific case the average rating is 0.53 for cameras, 0.58 for radio, 0.44 for depth cameras and

0.56 for ultrasound devices. Table 3.4 displays the different calculated benchmark scores for the combinations between applications and technologies. As numerous technologies and applications are compared, the feature-normalized benchmark score is also included.

Table 3.4.: Regular and normalized benchmark score matrix of different applications and technologies

		Camera	Radio	Depth Camera	Ultrasound
Hand Gesture	b_{s_l}	0.58	0.53	0.49	0.55
	$b_{s_l,norm}$	0.54	0.45	0.55	0.50
Identification	b_{s_l}	0.49	0.64	0.40	0.57
	$b_{s_l,norm}$	0.46	0.55	0.45	0.51
Obstacle Avoidance	b_{s_l}	0.47	0.56	0.42	0.59
	$b_{s_l,norm}$	0.44	0.48	0.47	0.53

The effect of the normalization is easily visible. Particularly radio has a high feature rating and is negatively affected by the normalization. The only example with a negative average feature rating is the depth camera. After applying the normalization it becomes competitive in some applications.

Finally, Table 3.5 shows the search results regarding the different technologies and applications. Particularly the ACM DL keyword search can generate empty results if the search terms are too specific. Thus, the search terms used are “gesture“, “identification and “obstacle” in this regard and add synonyms for the different technologies. For each sensor category the following synonyms are allowed. “Camera” and “video” for the first technology, “radio”, “rf” and “wifi” for the second, “depth camera”, “stereo camera” and “Kinect” for the third and “ultrasound” as well as “ultrasonic” for the last one. All search results were averaged according to the number of synonyms used. For the Google Scholar search more specific terms were used: “hand gesture”, “user identification” and “obstacle avoidance” with the same synonyms to prevent an excessive number of search results and prevented inclusion of patents and citations. All searches were performed on January 30th, 2014.

Table 3.5.: Search result frequency given specific applications, sensor technologies and synonyms for ACM Digital Library (DL) and Google Scholar (GS)

	Camera		Radio		Depth Camera		Ultrasound	
	DL	GS	DL	GS	DL	GS	DL	GS
Hand Gesture	66	14100	27	7350	32	6850	3	1660
Identification	81	5590	162	4920	10	3957	5	599
Obstacle Avoidance	8	24000	1	13017	17	12278	8	14500

3.4. Discussion

In this evaluation both benchmark score types are included to outline their differences. “Camera”, “radio” and “ultrasound” have a feature rating above average, whereas “depth cameras” had a lower than average rating. The feature-normalized benchmark score is thus adapted accordingly. Regarding the application of “hand-gesture recognition” this leads to “depth cameras” being considered the optimal technology as opposed to “cameras” that had a higher score before normalization. For the other applications there is no change in optimal technology. The preferred strategy for applying feature-normalized or non-feature-normalized benchmark scoring should depend on the specific benchmarking process. If numerous technologies and applications are benchmarked in a single process, the feature-normalization might be helpful to get a tendency regarding the optimal system. However, if the application is very specific it might be preferred to get a clear ranking and penalize unsuited technologies, regardless of their average feature weight. Accordingly, it is possible to refrain from normalization. Looking at the search results several conclusions can be drawn. The prevalence is unequally distributed between the different technologies. Both in keywords and general occurrence cameras are the most commonly occurring sensor device, with radio and depth camera ranked behind. Ultrasound on the other hand is less frequently occurring. This may be explained by the higher versatility of the other options. Regarding the “hand gesture” application, cameras have both the highest benchmark scores and most results in the database searches. The benchmark score for “user identification” and “radio” are matched for the ACM DL. However, there are more GS results for “camera”. As already mentioned cameras are more com-monly used, yet, the difference in keyword search results is significant. “Obstacle avoidance” is least common in the ACM DL, however quite popular in GS. Accordingly, “ultrasound” sensors are significantly more common in both searches, as opposed to the previous applications. Nonetheless, “stereo cameras” are the most common sensor device for this application. They are commonly used in automotive scenarios, where the detection range of ultrasound is insufficient, as the objects are moving fast [BB98]. Therefore, the application scenario might have to be redefined for fast-moving object detection in open areas as opposed to obstacle avoidance for robots in home scenarios. Additionally, the method of using database searches for verifying the benchmarking method has to be discussed, as opposed to expert opinion. Surveys of a specific application or certain technologies are common in scientific literature. However, while they might be comprehensive and cite several hundred different applications, the ACM DL database covers more than 2.2 million entries and GS searches can lead to more than 9.7 million results. Therefore, the index searches are preferable in terms of broadness. The search for keywords in ACM DL results in few hits compared to the database size. As they are chosen by the authors there is a large variety in word choice, spelling or number of keywords. While extending the number of different searches might lead to more results overall, it may also lead to additional overshoot, including work that do not cover the desired topics. The GS searches are very prone to overshooting, and should be preferably used to discover trends in data, as opposed to narrowly clustered results.

3.4.1. Central tendency bias

I want to briefly discuss the tendency of the benchmark scores to crowd around 0.5. While the benchmark may result in any number between 0 and 1 the two normalization processes the average is close to 0.5. Thus, even slower differences close to this average may have a higher significance. This effect is called central tendency bias and is a common occurrence on Likert-scale questionnaires and rating systems [CHE00]. Experts scoring technologies, just like survey respondents have a tendency to avoid extreme responses to a question.

While experience of the person executing the benchmarking process might avoid this problem, it is also possible to use a corrective term in the calculation of the final benchmarking score. The primary purpose of this corrective term is to make the comparison between different scores easier to the reader. The following equations can be used to fix either regular or normalized benchmarking scores, resulting in the modified benchmarking score m_b, respectively $m_{b,norm}$.

$$m_b = (b_{s_l} + 0.5)^a \qquad m_{b,norm} = (b_{s_l,norm} + 0.5)^a \tag{3.10}$$

The exponent a should be a value higher 1 and chosen according to the level of adjustment that is desired. As an example, Table shows adaptations of b_{s_l} and $b_{s_l,norm}$ for cameras and ultrasound, taken from Table 3.4. The different values for a are 1 , 5 and 10.

Table 3.6.: Central tendency bias correction for different exponents a

		Camera			Ultrasound		
		a=1	a=5	a=10	a=1	a=5	a=10
Hand Gesture	m_b	1.08	1.47	2.16	1.04	1.28	1.63
	$m_{b,norm}$	1.04	1.22	1.48	0.99	1.00	1.00
Identification	m_b	0.99	0.95	0.90	0.96	1.40	1.97
	$m_{b,norm}$	0.96	0.82	0.66	0.97	1.05	1.10
Obstacle Avoidance	m_b	0.97	0.86	0.74	0.94	1.54	2.37
	$m_{b,norm}$	0.94	0.73	0.54	0.50	1.16	1.34

3.5. Applications for capacitive proximity sensors

In the previous sections I have presented a generic method to evaluate different sensor technologies, with respect to various applications and provided evidence based on analyzing the frequency of technology popularity in scientific literature. In the following section I will use this method to verify potential application domains for capacitive proximity sensors. In section 2.4 - *Applications in smart environments*, seven different application domains within smart environments have been introduced. Using the specified benchmarking method I will specify the different feature weights and calculate the resulting scores. Afterwards, I will perform another frequency analysis of capacitive sensing in literature to determine if the existing scores are suitable.

3.5.1. Benchmark weights

Table 3.7.: Importance weights of applications for capacitive proximity sensors

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Indoor localization	o	+	++	+	-	+	o	+	-
Gestural hand interaction	++	++	-	+	o	o	+	+	-
Physiological sensing	++	o	–	+	-	+	-	o	o
Wearable activity recognition	o	-	-	+	+	+	+	o	o
Person-sensing smart appliances	-	+	+	++	o	+	+	+	+
Mobile augmented reality	o	++	+	-	o	+	-	o	+
Obstacle detection	-	+	-	o	+	+	++	++	+

Indoor localization will not typically require a high resolution, as it is in most cases sufficient to identify the approximate position of the different actors in the environment. However, a high update rate is required as it is necessary to follow the trajectories of the different persons. Achieving a high detection distance is often the most important aspect, e.g. if it is intended to cover large areas. The system should be unobtrusive and not disturb the actors, while processing complexity is less relevant, since it can be integrated and hidden in the existing environment. The system should be robust. The frequency of disturbances is less of an issue, as indoor surroundings are more controlled, as opposed to outdoor environments. However, if a disturbance occurs it should be calibrated quickly to resume normal operation. The unique disturbances are usually considered in system design and should not affect the overall situation too much.

Gestural hand interaction requires high resolution and high update rate, as it is intended to capture fine movements of the fingers. The detection distance is not critical, since it is assumed that the capturing device can be put in close proximity of the hands. Most systems are designed for explicit interaction and it is not necessary to be unobtrusive. Likewise, enough processing capabilities are typically included. The system should be fairly robust, as reliable input is expected and it should not be disturbed, resulting in unwanted actions. The calibration in case of disturbance should be ensured, or at least a feedback has to be given to the user. There are no critical unique disturbances in this specific application.

Physiological sensing often has to work on weak signals produced by the body and thus needs a high resolution. The required update rate likewise should be high as various of those signals are fast, such as heart rate or brain functions. The detection distance is weighted least favorably, as they mostly require a direct touch. However, even then it should be fairly unobtrusive, leading to wearable systems that have restricted processing capabilities, but nonetheless should be very robust. The signals are typically noisy with high frequency of disturbance, yet recalibration is not very costly and due to the touch properties no particular unique disturbers should be present.

Wearable activity recognition is mostly performed by wearable fitness trackers that have to be very conscious about energy usage. Thus the resolution is average and the update rate low. The system has to be close to follow the movements directly. However, it is preferable if there is an integration into other devices, or a small form factor, so they are less obtrusive. Processing capabilities are limited, yet the design should be very robust to avoid damage during typical activities. Disturbance mostly occurs due to unforeseen movements that can be misinterpreted, though are not very difficult to calibrate. Likewise, there are no particular unique disturbances apart from movement that we have to consider.

Person-sensing smart appliances are designed to just detect the presence of a human being, without providing any additional information apart from its own location. Therefore, the requirements for resolution are low, while this information should be available with a high update rate and preferably over a fairly large detection distance. The systems should be unobtrusive and robust, while the processing complexity of the signal is not critical, given the simple measurements that can be handled even by very low-powered devices. The systems should distinguish between persons and other moving objects, thus being reliable towards unique disturbances that may occur often. A recalibration can be fully automated.

Mobile augmented reality is typically executed on a smart phone. The localization is based on GPS using the limited resolution. However, the other sensors should clearly and precisely identify orientation, leading to mixed requirements on resolution. The update rate and detection distance has to be high, particularly regarding the orientation estimate. The systems can't be unobtrusive, as an additional device is needed to overlay the visual information. The processing complexity is average and can be performed even by slower smart phones. The system should be robust and provide its service at all times. The system is difficult to disturb, can be easily calibrated and there are no obvious unique disturbers that have to be considered.

Obstacle detection is a primary concern of autonomous systems operating in a smart environment. The resolution is less critical than the update rate, as there are various strategies to deal with encountered obstacles, but they have to be detected swiftly and robustly. Thus a higher detection distance is preferred. It is not necessary to be unobtrusive and it can be assumed that enough processing capabilities are present. For this application we have to account for disturbing factors, as it is mandatory that the system detects obstacles of all materials, shapes and most sizes. For most sensor categories there are materials that will prevent a proper detection. Thus unique disturbers have to be factored in, according to their prevalence in the given scenario.

Capacitive proximity sensor feature rating

Table 3.8.: Feature weights for capacitive proximity sensors

	res	upd	det	unob	proc	robu	disfr	calco	uniqd
Capacitive proximity sensors	**-**	**+**	**-**	**++**	***o***	***o***	***o***	**+**	**-**
Camera	++	o	+	-	o	o	o	-	o
Radio	-	+	++	+	o	o	o	o	-
Depth camera	+	o	o	-	-	o	-	o	o
Ultrasound	-	+	o	o	+	o	+	o	o

As the next step in the benchmarking process the feature weights of capacitive proximity sensors have to be determined. Following the same process used for the different technologies discussed previously the result can be seen in Table 3.8.

Looking at the group of sensor performance characteristics the resolution has to be considered coarse. While it is possible to create highly sensitive and precise sensors, this is typically restrained to very close distances. The majority of capacitive proximity sensor applications are operating over a certain distance. However, the sensor layouts that allow detection in this area do not have the resolution to precisely distinguish precise object locations without extensive post-processing. There is no set limitation to the update rate of capacitive sensors, as also high-frequency electric circuits can be evaluated, e.g. we created an exemplary application that has an update rate of about $1kHz$ [GPBB*13]. This will affect the achievable resolution and distance. Most setups operate in the range between real-time $20Hz$ up to $100Hz$ leading to a favorable update rate. The detection distance of the sensors is limited. While there are some systems that operate in distances of more than $1m$ [Mac04], they are restricted in their sensing abilities. Typically the systems are operating in a range somewhere between $10cm$ and $50cm$.

Discussing briefly the pervasive metrics, the major advantage of capacitive proximity sensors is their ability to be installed completely invisible. They also operate in a frequency spectrum that is not considered biologically active and at low voltages. The processing complexity varies, based on applications. They will require pre-processing and calibration, that is more complex compared to other sensors. However, as there are not many sensor value the post-processing is simpler, e.g. compared to cameras. Some data processing methods can use fairly complex statistical methods if multiple objects are to be tracked or there is additional gesture recognition [GPBKK13]. Overall, the processing complexity is rated average. Similar to the other sensor systems presented robustness and quality of service depend on the particular application. It is possible to design systems that fail easily, or that are fully redundant. In consequence, they receive an average rating.

Finally, the environmental characteristics are discussed. The frequency of unique disturbances is average. The sensors can be disturbed by electrical signals that are present in the environment. The majority of potential disturbing signals can be compensated in an initial calibration phase and by using appropriate measures to isolate the measuring system from the existing electric circuits. The remaining disturbances are either caused by devices that are brought into the environment after, or by irregular disturbances in the electric supply. E.g. in our laboratory we had a persistent issue caused by a faulty power supply in a neighboring building that was affecting the stability of the supply frequency. The complexity of calibrating the sensors is low. There is no need to use any external measure and it is sufficient to consider a time-series of previous measurements for calibration.

The example above about a single power supply disturbing all capacitive sensors is one example why unique disturbers have to be taken into account. While these instances are limited the rating is reduced to "not favorable".

3.5.2. Benchmarking scores for capcitive proximity sensors

Using the methods of the previous pages it is now possible to calculate the different benchmark score for capacitive proximity sensors and the various applications. In this case there is no comparison between different technologies. To provide a complete overview all four varieties are calculated and discussed. The results are shown in Table 3.9.

Table 3.9.: Benchmarking scores for capacitive proximity sensors in different applications

	b_{s_l}	$b_{s_l,norm}$	m_b (a=10)	$m_{b,norm}$ (a=10)
Indoor localization	0.56	0.53	1.79	1.34
Gestural hand interaction	0.51	0.49	1.10	0.90
Physiological sensing	0.57	0.54	1.97	1.48
Wearable activity recognition	0.55	0.52	1.63	1.22
Person-sensing smart appliances	0.57	0.54	1.97	1.48
Mobile augmented reality	0.50	0.47	1.00	0.74
Obstacle detection	0.55	0.53	1.63	1.34

As seven different applications without comparison of different sensing technologies, the normalized scores have the purpose of putting the calculated scores around the 0.5 average feature rating. Additionally, the correction term for central-tendency bias is included for both regular and normalized scores.

The lowest scores are associated to mobile augmented reality and gestural hand interaction

are tested it can be assumed that the non-normalized benchmark scores will have a tendency towards the highest-rated application. In fact there is a spread between physiological sensing having an average feature rating of 0.53 and person-sensing smart appliances with an average feature rating of 0.69. The non-normalized score for both is 0.57 and diverges to

3.5.3. Popularity of capacitive proximity sensors in literature

4. Use cases for capacitive proximity sensors

After having defined the potential application domains for capacitive proximity sensors, in the following section I would like to evaluate the actual use cases by presenting associated challenges, presenting data processing methods on how to tackle these, and present a number of prototypes that have been created implementing these methods. This allows to gather the information required to discuss the application of capacitive proximity sensors in smart environments and validate their use in Chapter 5. In addition to the application specific challenges there are also numerous implementation challenges that are detailed in the descriptions of the associated prototypes.

4.1. Use cases and associated challenges

Table 4.1.: Application domains and derived implemented use cases for capacitive proximity sensing

Application Domain	Applying capacitive proximity sensors	Implemented use cases
Indoor Localization	Sensing system hidden in environment	Capacitive sensing below floor cover
Smart Appliances	System detecting presence and other parameters of human bodies in range	Posture recognizing office chair, occupation sensing bed, arm detecting armrest
Physiological Sensing	Determine physiological parameters associated to movement	Breathing rate detection via chest movement, long-term movement analysis
Gesture Interaction	Hand interaction in near range	Finger gestures, single hand 3D gestures, combined multi-hand and touch tracker

Looking at the previously defined application domains for capacitive proximity sensing we can have a closer look and think about actual use cases that belong in the different domains. While it is also possible to associate the different systems presented in the related works I will focus on the implemented prototype systems that are presented in the subsequent sections. Table 4.1 shows the different application domains, how capacitive proximity sensors can be applied and the use cases that were derived from it. In this section I will discuss the creation of this table and create a list of challenges that become apparent when designing the specific systems. Based on these challenges it is possible to identify a number of steps in the processing of capacitive proximity sensor data that can be improved, in order to enable the presented applications. These specific contributions to processing methods will be detailed in the following section.

Indoor localization has been presented as one of the application examples for smart environments in the related works section. The main advantage of capacitive systems is their unobtrusive application in the environment as presented in TileTrack [VMV09] and SensFloor [LS09]. Capacitive indoor localization systems can be hidden below any non-conductive material and enable tracking of users on a distance. Particularly interesting is the ability to place the sensing equipment below the floor cover as an additional layer, e.g. when installing a new carpet or wood parquet. SensFloor as commercially available solution is designed to be placed as an additional layer and integrates sensors and communication chips in this layer. While this enables a precise and reduced noise sensing close to the walking persons, it is costly and can lead to maintenance issues once the sensors integrated into the layer fail. Instead, it is also a viable option to separate sensor hardware and electrodes, e.g. by placing the sensors on the borders of the indoor area and use a specific electrode layout below the floor. The electrodes can be made of any conductive material and can be protected using non-conductive isolation to prevent corrosion and physical damage. The system components that are most prone for failure are the connections between electrode and sensor and the sensor hardware and its communication channels. Those can e.g. be placed within the border covers.

The main challenge of this solution is to balance the number of sensors and the required resolution. Using a limited set of sensors placed on the border it should nonetheless be possible to determine the positions of one or more persons on the area above and potentially additional information, such as the status of a person, if it is standing, sitting or lying. In addition to the cost factor a larger number of sensors also causes several other issues, such as cross-talk between the different electrodes that has to be avoided using a variety of multiplexing methods. The achievable resolution of the single electrodes is depending on several factors, including the measurement time, the applied voltage, distance between electrodes and floor surface, or the geometric layout of the electrodes. Thus, it is important to find an electrode layout and processing methods that achieve this balance.

Smart appliances as presented in the related works can be a very diverse group of devices that are in the current environment. There is a huge variety of sensor categories and processing that can be applied to any given task. Looking at capacitive proximity sensors, the major advantage is the invisible application that allows to create smart appliances that are indistinguishable from systems without sensor devices. Using different conductive materials for the electrodes this integration can range from solid antennas hidden within the appliance to conductive threads that can be woven into fabric. The main application for capacitive proximity sensors in smart appliances is the sensing of different parameters of persons interacting with the system. For example the sensors can be used to recognize the posture of a user and use it to adapt certain parameters of the appliance or the environment. This type of interaction has also been called implicit interaction, as the user does not directly attempt to manipulate the environment, but instead the activities are interpreted as input according to the given situation [SV01]. In many instances it is sufficient to get information about the presence of the user. A simplified version of the posture recognition can be used to detect presence or occupation, based on the data acquired by one or more sensors. Finally, it is often also interesting to detect if certain body parts are currently at a given location, e.g. the arm resting on the armrest of a chair, indicating a specific situation that the system can react on.

The challenges in this domain are manifold. Existing posture recognition systems might rely on a different sensor category, supporting hundreds of measurement spread over a larger area. Again, capacitive proximity sensors are distributed sparsely and need methods that enable gaining a similar amount of higher-level information. Here it is necessary to create models of the human body that are suited for processing of capacitive data. According to the parameters that are supposed to be detected the models can be more or less complex and thus improving the required processing time. A sensing bed that wants to detect how a person is lying on it would require a simpler model, as opposed to a sensing chair that would require to detect a larger variety of different postures. Often the capacitive systems are combined with other systems, or use a custom non-uniform distribution of electrodes in the device that require methods of data fusion and processing of heterogeneous signals to acquire higher level information, e.g. an armrest that combines the detection of an arm and an interactive area that allows gesture interaction.

Physiological sensing allows us to measure signals generated by the different process of the human body. One common application is for athletes that track the effects of training on their body parameters and might measure heart rate or respiration. There are numerous medical applications ranging from long-term blood pressure sensing, to blood glucose level sensing or tracking the quality of sleep throughout the night. Additionally, there are physiological signals derived from long-term monitoring, such as movement-based sleep-phase detection. Measuring electric properties is the most common variety to detect physiological parameters, ranging from EEG, measuring the brain activity, or ECG measuring the heart rate, or sensors for skin conductance that can infer the stress level. For all these applications electrodes are placed very close to the measured property, often even requiring contact. Capacitive proximity sensors on the other hand are by definition used over a distance. However, the systems can be designed to enable a high resolution that can track very small movements of the body. Rob MacLachlan has created a spread spectrum system that is able to measure the chest movement associated to breathing over a distance of more than 30cm [Mac04]. Other examples include the detection of swallow movements [CAL10].

There are various challenges when trying to gather physiological signals from capacitive proximity sensors. A major problem is to distinguish the measured property from other signals generated by movement of the body. Here it is possible to use the effect that the measured properties often are prevalent in a specific frequency range. Thus, if the signals are analyzed in the frequency domain it is possible to extract the physiological properties from the overall signal, e.g. when analyzing the chest movement associated to respiratory rate and focusing on the frequency areas most important. Regarding long-term physiological signals, capacitive proximity sensors can be used to aggregate data on movements. In this regard, it is interesting how the sensor data in time-domain can be associated to particular movements that can be used in long-term analysis of the user's physiological patterns, e.g. to detect sleep phases. Based on the particular setup of the system a different set of features has to be selected and evaluated.

Gesture interaction is a very diverse application area that reaches from the acquisition and interpretation of whole body gestures to small movements of the fingers registered on surfaces. It is maybe the most thoroughly researched domain of capacitive proximity sensors, starting with Leon Theremin's musical instrument. The MIT research group experimenting with capacitive interaction in the 90s created some concepts for touchless interaction, e.g. the Field Mouse that allowed to control the third dimension in certain applications [Smi96], or an art installation that could be controlled using a set of gestures [SWD*98]. A new category of interaction devices such as Wii remote, Kinect or Leap motion led to the proclamation of more natural interaction between human and machine [Val08]. While capacitive touch sensors have become ubiquitous in mobile devices, the proximity variety is less frequently used. Wimmer integrated several sensors into a table to enable a regional interaction on the surface [WHKS06].

While the area has been well-researched there is still a number of challenging aspects. In many instances the capacitive interaction devices will have a different resolution according to the direction. In the last years there has been a rise in methods that allow a generic recognition of gestures in two dimensions, e.g. from mouse cursor movement and finger movement on touch screens. It is interesting to evaluate if this is also possible for 3D positions acquired from capacitive proximity sensors. The acquisition of the hand position is also challenging, as the sensors can't distinguish between hands and other conductive parts of the body. Thus it is interesting to investigate different methods of fitting arms and hands, particularly on larger area interaction devices. Additionally, it is challenging to enable gestures via multiple hands and arms. In many gesture interaction applications fatigue is a challenge, if the hands have to be moved too much or the arms have to be held in free air for a longer time. Designing specific graphical user interfaces that are suited for this. If the capacitive proximity sensors are placed under thicker layers of non-conductive material it is difficult to detect touch events from capacitance data alone. It becomes interesting to combine capacitive proximity sensors with other sensor categories that can detect touch or even different touch events, thus allowing a richer interaction.

In consequence there is a large number of specific challenges that can be tackled in the different domains. They can be associated to the identified use cases using Table 4.2. In the following section I will present the

Table 4.2.: Challenges associated to the different use cases for capacitive proximity sensors

Use cases	Challenges
Capacitive floor sensors	Sparse sensor distribution in large areas, geometric electrode layout
Posture chair	Multi-body models, electrode material
Occupation sensing bed	Single-body models, movement tracking
Armrest supporting gestures	Heterogeneous capacitive arrays
Breathing rate detection	Frequency spetrum analysis
Sleep phase detection	Long-term movement features
Finger micro gestures	Small 3D movements
Multi-arm tracking	Arm and hand fitting, interaction design
Combined touch sensing	Combining position tracking and touch events

methods that have been developed in this regard and how they contribute to the different challenges in processing the data generated from capacitive proximity sensors.

4.2. Processing methods

After analyzing the challenges associated to the different use cases for capacitive proximity sensor I will use the following section to specify a number of novel or improved data processing methods that can be used in this context. They are grouped into five specific areas:

Sparsely distributed sensor arrays refer to configurations that have a limited number of sensors spread over larger areas. In this regard it is important to find methods that allow acquiring sufficient information about the object to be detected. Typically information about the electrode geometry and interpolation methods are used to meet these requirements.

The second area are model-driven fitting methods. Using a simplified model of the object to be detected, it is possible to fit these to the received sensor data. The area can be distinguished according to the complexity of the models that are either comprised of a single body or multiple parts that are connected to each other.

Heterogeneous sensor systems can be described as a combination of multiple sensors that are not uniform. In terms of capacitive proximity sensors this can refer to either arrays of different capacitive sensors that use a geometric layout of varying sizes and shapes or the combination of capacitive sensors with other categories of sensing systems in a meaningful fashion.

Image-based processing describes the method of creating an image from capacitive sensor data and applying different algorithms associated to visual computing. An uniform array of capacitive proximity sensors resembles an array of light sensors on a different frequency interval of electromagnetic radiation. Thus, with a few limitations it can be treated similar to a camera system with operations applied on a pixel level.

A last group is the processing of physiological signals in time domain and frequency domain. Many physiological activities rely on the movement of muscles, e.g. the beat of the heart or the chest movement associated to breathing. These movements have an effect on the electric field generated by capacitive proximity sensors and can be analyzed using a variety of different methods.

4.2.1. Sparsely distributed sensor arrays

Sparsely distributed sensor arrays refer to layouts that limit the number of available sensors either by environmental parameters or by design. This limits the information that can be gathered about the detectable object, or reduces the number of different objects that can be distinguished. To compensate this limitation a number of interpolation methods can be used that take into account our knowledge about the position and shape of the electrodes used in the current setup. One example for this sparse distribution is the previously presented Thracker system that uses only four electrodes to acquire a hand position and detect gestures at certain positions [WKBS07]. In this section I will present two different contributions - a new method to recognize single-hand gestures in free-air using just six different sensors and an indoor localization system based on a coarse grid of wire electrodes that can be hidden below different floor surfaces.

4.2.1.1. 3D location tracking and gesture interaction

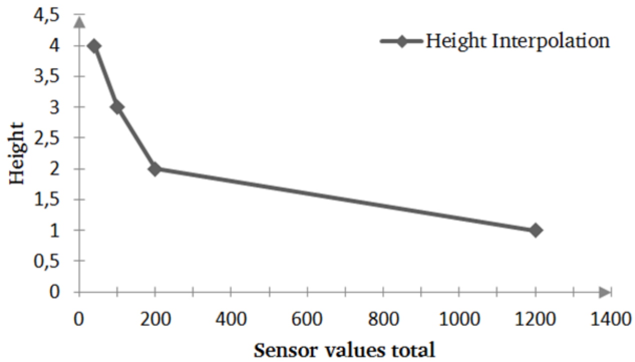


Figure 4.1.: Piecewise linear hand distance estimation [BH11]

Gesture recognition can comprise a large number of different body movements, including sign language that uses movements and position of hands and fingers, the posture of the whole body, or as in our case the movement of the hand in three dimensions. This requires two distinct processing steps. At first it is necessary to precisely localize the position of the hand in the interaction space. Afterwards, a time series of theses positions has to be analyzed and attributed to different gestures. The localization method was first presented in a publication from 2011 [BH11]. The static gesture recognition method used there was later extended by adapting algorithms used to detect mouse gestures for movements in three dimensions [BDK13]. The first data processing step is the planar localization of the hand, following a weighted average algorithm, whereas n is the number of sensors, (x_i, y_i) the location of the electrode centers and v_i the value of the given sensor.

$$\bar{x} = \frac{\sum_{i=1}^{n} v_i \cdot x_i}{\sum_{i=1}^{n} v_i} \qquad \bar{y} = \frac{\sum_{i=1}^{n} v_i \cdot y_i}{\sum_{i=1}^{n} v_i} \tag{4.1}$$

In order to calculate the distance of the hand from the plane we are using a piecewise linear interpolation, that resembles the response curve of a single sensor [BH11]. In this case four different thresholds t_i are used to calculate the proximity, based on the sum of sensor values. t_1 indicates the closest distinguishable proximity, e.g. touch, with all higher value sums associated to this. t_4 represents the maximum distance in which the sensors can

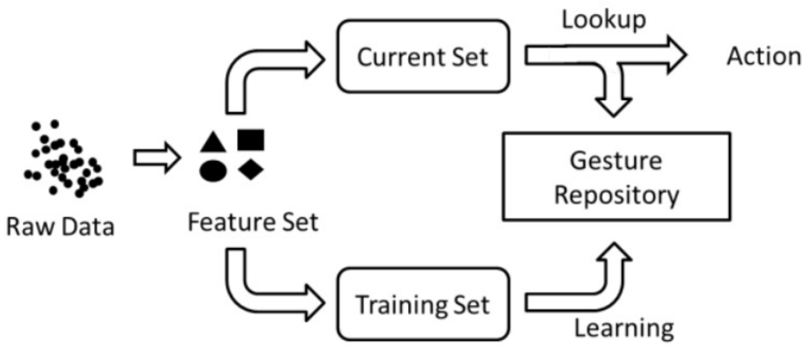


Figure 4.2.: Principle components of a learning by example recognition framework [BDK13]

detect a hand. One example with fixed points at values 40, 100, 200 and 1200 is shown in Figure 4.1. Initially a static gesture recognition method was implemented that used a series of five subsequent locations and simple heuristics to determine a small number of gestures. Thus, a generic gesture recognition module was created, based on learning by example [BDK13]. The general functionality of a gesture recognition framework that is using learning by example is shown in Figure 4.2. A feature set is extracted from incoming raw data. Collections of these are distinguished into training sets that are used to associate certain features to given gestures. After a learning process the current feature sets, acquired on-the-fly, are tested against the training sets in a repository. These look-ups can lead to successful gesture recognition and association to certain actions. This association method is also called classification. There are numerous approaches, e.g. neural networks (NN) or support vector machines (SVM).

4.2.1.2. Large-area location tracking

There are several systems that use capacitive proximity sensing to track the location of one or more persons in an environment. A common challenge in large areas is achieving a suitable coverage with electrodes at all positions. Lauterbach et al. overcome this problem in their SensFloor system by integrating sensors and electrodes in an underlay that can be placed below the upper layer of the floor [LS09]. TileTrack, the system developed by Valtonen et al. requires large emitter electrodes under the floor and receivers placed in the walls [VMV09]. To overcome this limitation they later integrated receivers into different pieces of furniture.

While SensFloor allows to cover large areas by having the sensing electrodes near all surface areas, a limitation is maintenance. If a sensor breaks below the floor covering it is difficult to replace. TileTrack in its initial installation requires proximity to walls, or later to specific pieces of furniture that are in the environment. This is difficult to guarantee in many instances and requires an initial calibration of the environment, according to placement of the furniture.

I proposed a system based on a rectangular grid of long wire electrodes that are placed below the top floor cover, with sensors attached at the edge of the area, e.g. in skirting boards [BHW12]. As the system is based on loading mode there is no need for dedicated receivers, but instead relies solely on the electrodes below the floor. The system is akin to a larger variety of projected capacitive touch screens that are partially also using grid layouts [BO10a]. Using long, straight wire electrodes have different effects on the measurement. One effect of this is the limited detection distance that is not comparable to large plate electrodes. Particularly if thick floor

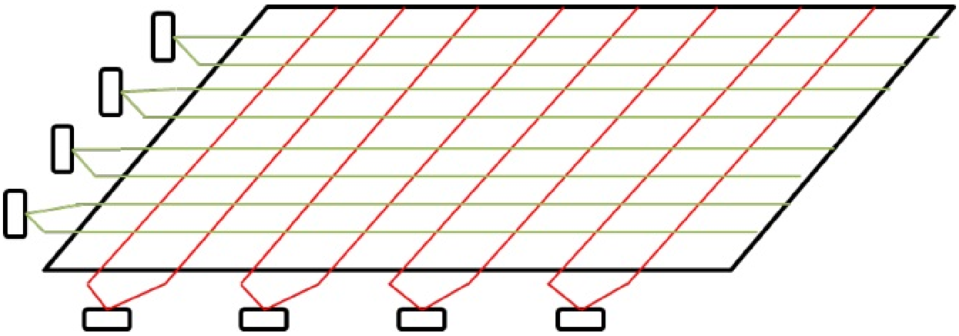


Figure 4.3.: Wire electrode grid below floor cover attached to sensors on the border

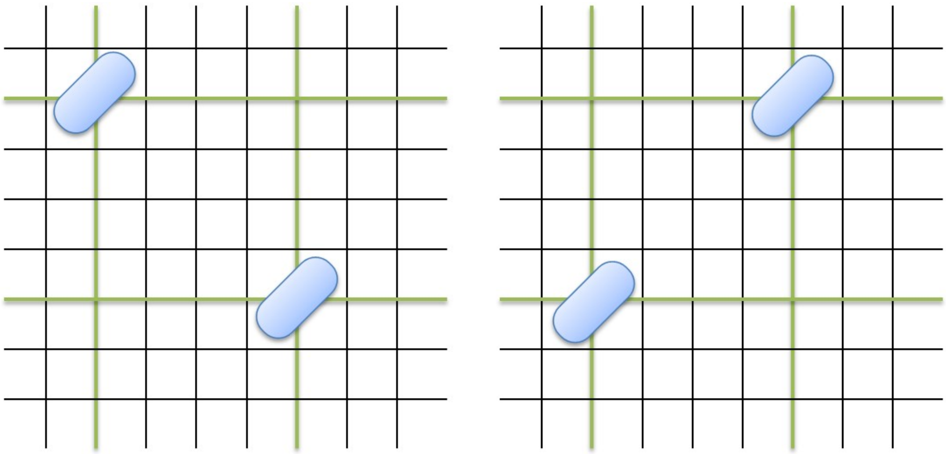


Figure 4.4.: Two potential person locations resulting in same sensor readings (green indicates active electrodes

covers are used the grid has to be fairly dense. Another effect is the sensitivity towards noise and influence from outside electric fields. Therefore the system requires preprocessing to reduce the noise and achieve a more robust high-level data processing. In order to localize persons the system uses an adapted weighted average algorithm, similar to the variety presented in the previous section. Each electrode is considered to only have a single coordinate in either x or y direction, allowing to easily calculate the center-of-gravity. However, it can occur that only two electrodes are active at a certain point in time, while two persons are present and too far away from any other electrode to be detected. In these cases there is a certain ambiguity as each (x,y) value combination can result in two potential intersection points, as shown in Figure 4.4. To overcome this problem it is possible to either use a mix of sending and receiving electrodes operating in shunt mode and specific measurement cycles. Another option is to analyze the time-series of previous locations to discard unlikely positions. Similar to SensFloor the concept also supports detecting additional information about the persons present, most notably fall detection. This is based on a time-series analysis of aggregated values of the sensors that are currently detecting an object. This method is using the assumption that the overall sensor response is roughly equivalent to the shape of the object that is closest to the surface, resulting in a higher capacitance of the overall system, similar to the plate capacitor model. This effect is shown in Figure 4.5. The sum s of all n sensor values r is the closest equivalent

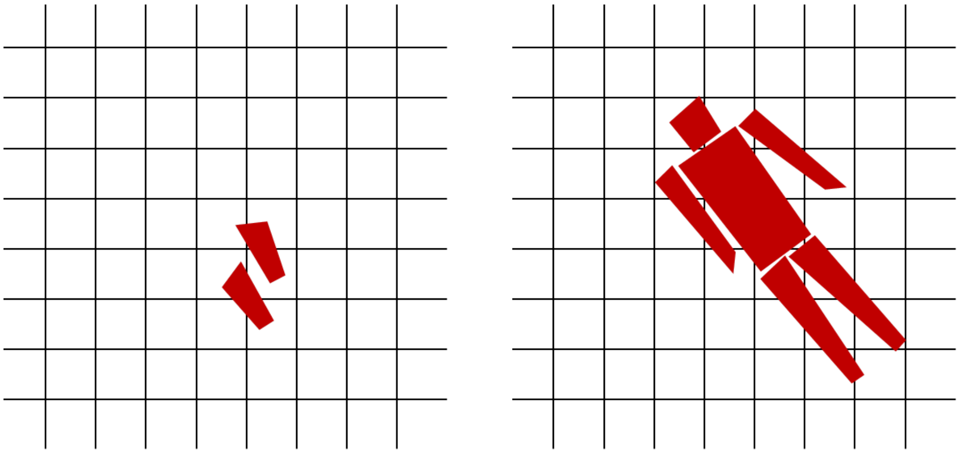


Figure 4.5.: Shapes of a standing and lying person on top of the CapFloor grid

to the system capacitance and therefore a viable measure. If the overall value is beyond a certain threshold v_l we can consider a lying person p_l.

$$s = \sum_{i=0}^{n} r_i \qquad\qquad p_l = \begin{cases} 1, & s \geq v_l \\ 0, & s < v_l \end{cases} \tag{4.2}$$

In order to increase the robustness this threshold has to be exceeded for a certain amount of time t_m. In consequence a fall f is detected if the following equation is 1.

$$f = \prod_{j=0}^{t_m} p_{l,t_j} \tag{4.3}$$

4.2.2. Model-driven fitting methods

When acquiring sensor data from physical objects it is often difficult or even impossible to analytically describe the resulting value, as there are numerous environmental factors influencing the signal and the properties of the object might not be clearly determined. Considering the human body, there is a mostly unconstrained number of sizes, shapes and biological properties that influence the response to an electric field. Thus, in order to fit sensor outputs to the potential object configurations, simplified models can be used that resemble the actual physical effects and can be described analytically. Regarding capacitance of the human body relative to a single sensor, a common abstraction is a sphere having a diameter close to the height of an average human [Sea97]. Models based on a single geometric objects are considered single-body, while connected geometric objects that comprise a single model can be called multi-body. Smith used a model of multiple spheres to approximate arm position and rotation above an array of capacitive proximity sensors [SWD*98]. Another possibility is adapting the models to a derived physical effect. Harada et al. are using the projected pressure distribution of a virtual skeleton and body model on a flat surface to create a pressure distribution that can be compared to the actual pressure effect generated by an actual human body resting on a set of sensors [HSM00]. In this section I will describe two novel methods to fit abstracted models of the human body to sensor readings acquired from smart furniture systems. The first method uses a cylindrical human body model to match the posture of one or two bodies on a bed, the second method uses a multi-body skeleton that is fitted to sensor readings determining posture on a chair.

4.2.2.1. Single-body models

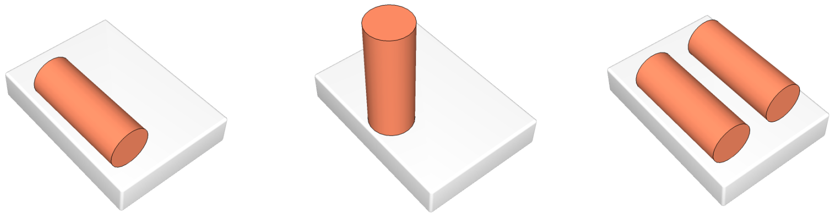


Figure 4.6.: Cylindrical human body model and various poses on mattress

To identify occupation and positioning we use a very simple model for estimating the effect of a human body on the sensor values. As mentioned earlier the sensors act on both presence and pressure applied. We model the human body as an approximately cylinder shaped object that is on the mattress either lying or sitting, either one or two objects, a few potential poses shown in Figure 4.6. We assume that the sensors are analyzing the pressure distribution on the mattress, a sitting person will cause a high pressure on a small region, a lying person a moderate pressure on a larger region. Determining the position and orientation of this cylinder from a limited amount of sensor readings can be postulated as an inverse problem. If we assume a constant density of the cylinder the idealized pressure distribution is uniform as shown in Figure 4.7.

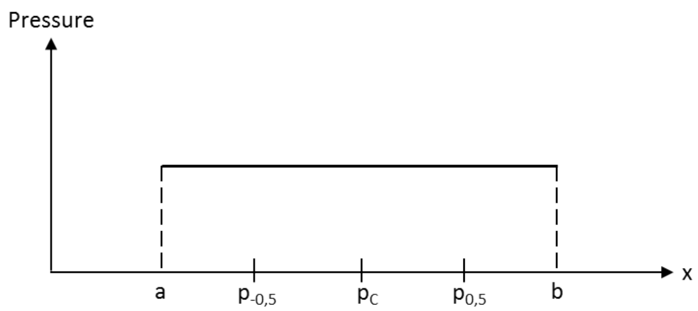


Figure 4.7.: Pressure distribution of a uniform cylinder

We further simplify the system by using two distinct values of the pressure distribution. p_c is the center of pressure, $p_{-0.5}$ and $p_{0.5}$ the points enclosing half of the pressure distribution. Using the calculations of a regular uniform, continuous distribution we get the following equations:

$$p_c = \frac{a+b}{2} \qquad \sigma = \sqrt{\frac{(b-a)^2}{12}} \tag{4.4}$$

$$p_{-0.5} = p_c - \sigma \qquad p_{0.5} = p_C + \sigma \tag{4.5}$$

The raw data from the sensor is considered as random, uniform sampling, a discretization of the continuous distribution. We calculate the center of pressure and the standard deviation of using the geometric meta-information, the position of the sensor $\overrightarrow{x}$.

$$p_c = \frac{\sum_{i=1}^{n} v_i \overrightarrow{x}}{\sum_{i=1}^{n} v_i \overrightarrow{x}} \tag{4.6}$$

$$\sigma = \sqrt{\frac{1}{n}(\sum_{i=1}^{n}\vec{x}^2 - \frac{1}{n}(\sum_{i=1}^{n}\vec{x})^2} \tag{4.7}$$

Using this model we have determined a set of potential poses that cover the most common situations. We distinguish between potential poses for one and two occupants. One person may sit at a certain location or lie on the bed in various angles. It is assumed that the head is always at the upper part of the bed. Two persons may either both sit, both lie down, or one is sitting and one lying. The limitations of this model concerning the actual system are the non-uniform pressure propagation throughout the mattress, as well as the non-linear sensor response on different pressure levels. Therefor we do not expect the deviations to strictly adhere to the theoretical model but instead use configurable thresholds that allow for increased robustness in exchange for precision.

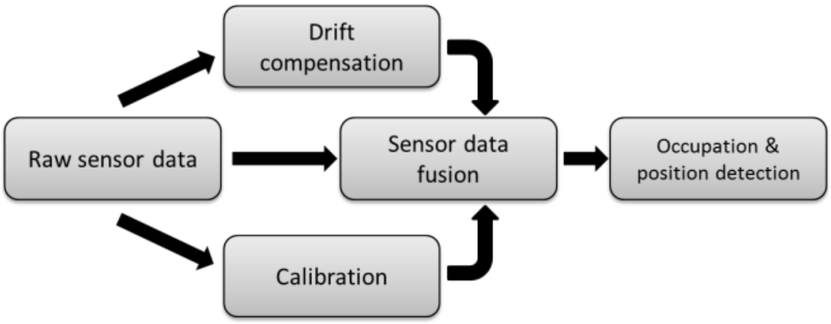


Figure 4.8.: Data processing components [BH12]

The different components of the Smart Bed data processing are shown in Figure 4.8. Raw sensor data is distributed to three different modules, the calibration which is determining the initial parameters for the sensor data fusion, the drift compensation that alters those parameters according to long term trends and finally the sensor data fusion module that processes the data and does feed it to the occupation & position detection. Calibration and drift compensation follow the previously presented model [BH12].

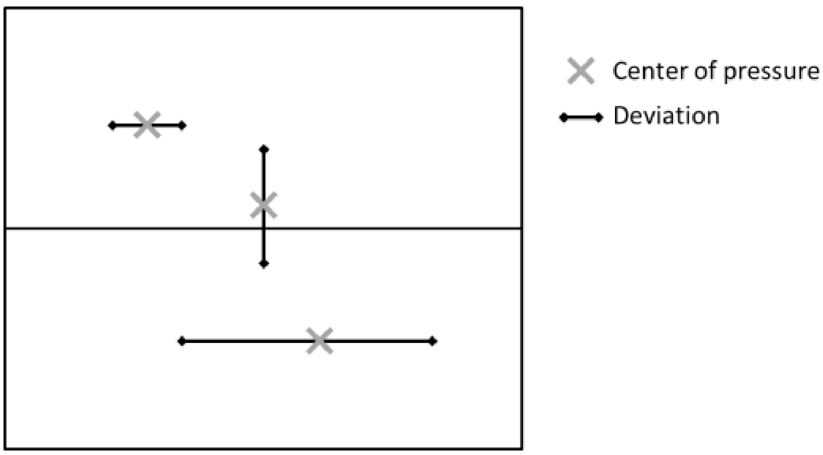


Figure 4.9.: Calculating centers of pressures and deviation [BH12]

Occupation and position detection is performed by dividing the two person bed into left and right and individually calculating for each side the total sensor values, assumed center of pressure using weighted average and the standard deviation (Figure 4.9). The same calculation is done between the two sides to distinguish where is activity or if one person is lying diagonally. Using these six intermediate values we can now map various poses. If all activity is on one side and the horizontal deviation is low, we can assume that one person is sitting. We can additionally use the intermediate values to calculate more information, e.g. the exact location a person is sitting at. The data processing for the sleep phase recognition is based on detecting the sensor data variations in order to analyze movement. Discriminating between sleep phases using movement is a common approach that has been used in the past [SL86]. Using a sparse set of sensors it is possible to detect movement by comparing subsequent sensor readings and associate it to different sleep phases using different activity profiles. The system is based on the same prototype as the posture recognition system [DBM14].

4.2.2.2. Multi-body models

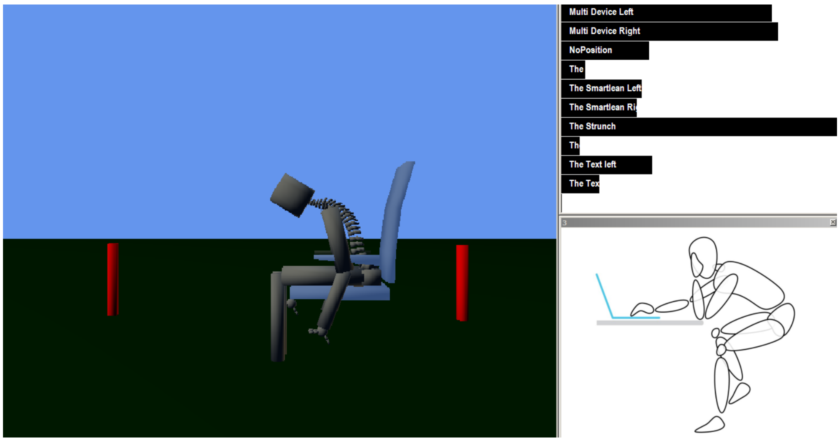


Figure 4.10.: Screenshot of the Capacitive Chair application showing the fitted 3D model on the left, posture detection on the upper right and the recognized posture on the lower right

In Figure 4.10 we can see a screenshot of the Capacitive Chair debug application. On the left side we see a 3D model that is fitted to a chair model according to the current sensor values, in the middle the results of the machine learning module and the recognized posture and on the right side the currently running breathing rate detection as both Fourier analysis and signal deviation analysis. All processing methods work on filtered and normalized sensor data. The difference in shape, material and size of the electrodes necessitates slight adaptations to noise filtering and data processing. As an example only the conductive thread backrest electrode is used in the breathing rate detection. The 3D model is using a simplified human joint model comprised of 13 connected components. Based on the current sensor readings, single parts or groups of components are fitted to the virtual chair. The process is a mix of posture mapping as found in the smart bed and modification of the dynamic links between the single components [BF14].

Figure 4.11.: Screenshot of the Capacitive Chair application showing the fitted 3D model on the left, posture detection on the upper right and the recognized posture on the lower right

We use a simple RBF neural network and training data collected by two different persons to match the input from eight sensors to nine potential output postures that are associated to different working situations. An early observation is that certain postures are difficult to distinguish given the limited number of sensors and the similarity of the postures on the rigid chair. Either a higher number of sensors or a more versatile chair could be used that allows gathering additional information required to distinguish the different poses more reliably.

The breathing rate detection is operating on a single electrode that is integrated into a mesh on the backrest using conductive thread. The setup is shown in Figure 4.11. Consequently the surface of the electrode is large and able to pick up the chest movement. Two different methods of data processing are used and fused to get the final breathing rate. Using a fast Fourier transformation the signal is transformed into the frequency space. We are looking for significant signal portions in frequency areas that can be associated to breathing, between $0.2Hz$ and $10Hz$. The second method is to look for zero-crossings of the sensor signal through an adaptive baseline. If a person is breathing in the sensor value will decrease resulting in the signal dropping below the long-term average, and rise above when the person is breathing out. Accordingly the breathing rate can be calculated by counting the zero-crossings.

4.2.3. Heterogeneous sensor systems

4.2.3.1. Heterogeneous capacitive arrays

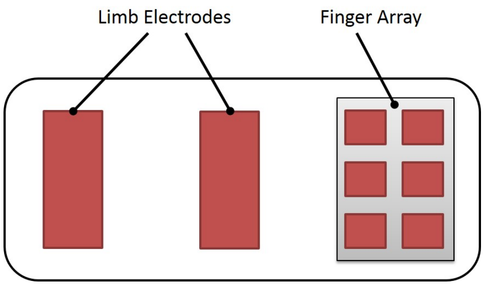


Figure 4.12.: Heterogeneous sensor array for limb detection and finger tracking

Figure 1 Active armrest sketch - six electrodes for finger gesture detection in front, two for arm detection in back We briefly mentioned the challenge of using the armrest as an interactive zone. It is necessary to clearly distinguish between intended control gestures by the driver or if he is just resting the arm. Our approach is to use the status of both arm and hand to identify the intention. Capacitive proximity sensors create a weak electric field that is disturbed by conductive objects, e.g. limbs and fingers. They can be used to detect posture and realize free-air gesture interfaces. We are combining both methods to enable both intention recognition and small finger gestures. 3.1 Interaction concept

Figure 2 Postures of limbs on armrest - resting position (left), arm raised position (middle), hand raised position (right). In Figure 2 we can see the three different positions arm and hand can have on the armrest. On the left arm and hand are in resting position with both close to the surface. The middle image shows the arm raised position and fingers touching the front of the armrest. The right picture shows the arm resting on the back and the hand in proximity of the front area. The latter two positions are suitable for finger-based gestural interaction as they can be moved freely. The system should therefore be able to distinguish between the three different positions. The interaction for both active positions is a bit different. Regarding the arm raised position the person will typically want to interact using familiar touch gestures. In the hand raised position it is necessary to track gestures that are performed in the air. In both cases we assume that a single finger is used. A set of four different gestures has been defined for both interaction methods. The number is sufficient to control the user interface that we have developed and support both navigation and selection. The type of gestures has been defined after looking at previous research into touch and hand gestures [3, 14]. For the touch interaction we support left and right swipes performed with either one or multiple fingers. Regarding the free-air interaction we are using left and right swipes, as well as circles either clockwise or counter-clockwise. The gestures are mapped to typical navigation and selection options required to trigger the different actions of the GUI that we will describe in the prototype section. 3.2 Data processing

Figure 3 Arm model and detection of posture based on distances to two sensors and finger array for resting position (1), hand raised position (2) and arm raised position (3) The data processing of the Active Armrest requires three distinct steps. At first we have to determine the three potential limb postures specified in the previous section. Afterwards, we need to calculate the position of the fingers in or above the interaction area, and finally perform a time-series analysis of subsequent positions to infer different gestures. A stylized view of the posture detection is shown in Figure 4. The two distinct arm sensors are able to determine single distance values. In addition the aggregated data of the finger detection array in the front is used to detect a third distance value. To map the different postures we are using a set of thresholds that determine if the arm or hand is touching the armrest surface, or is hovering above it. As we are acquiring sensor data proportional to distance it is also possible to calculate orientation angles of the arm and use it as input. However, we are not using that option in this work. The calculation of the finger position in three dimensions is adapted from the method presented by Braun et al [4] that uses a combination of weighted average for planar location and stepwise linear interpolation to determine the height. An addition is the distinction of one and multiple finger touch events, distinguished by an additional threshold. To classify the gestures we are using points from a distinct start to a distinct stop. In case of the free-air interaction this is determined by the finger moving (exceeding a certain gradient between subsequent points) and stopping to move. In case of the touch interaction it is determined also by a finger starting and stopping to move, however only when in touch distance. The points between start and stop position are normalized to a specific time-scale and we are using five significant positions in this time frame that are fed into a SVM classifier. There are distinct classifiers for the two different methods that are triggered according to the selected interaction pattern. The SVM is trained using the sequential minimal optimization method by Platt [11].

As we already mentioned, the Active Armrest electrodes are put into two groups. The data processing for both groups is distinctly different. In order to detect the presence of the arm using the two-electrode group a simple threshold on the accumulated values is used. The six sensor array in the front (touch area) is using the presented weighted average method to calculate finger positions. Additionally a threshold is used to distinguish one and two fingers. Overall there is a data processing pipeline as shown in Figure 4.29. The finger tracking and gesture recognition will be inactive until it is ensured that no arm is present.

4.2.3.2. Heterogeneous sensor fusion

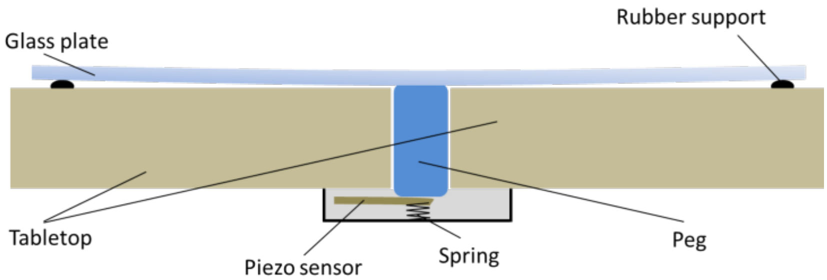


Figure 4.13.: Suspended peg knock detection system for CapTap [BF14]

The hand location of the CapTap is similar to the methods presented for the MagicBox. We add the additional component of knock detection to provide selection events when touching the surface. Figure 4.35 shows a sketch of the knock detection system. The table has a glass plate that is suspended on some rubber supports. In the center of the table we attach a small peg (enlarged in sketch) that creates a connection between the glass plate and a piezo sensor. If the glass plate starts vibrating from a touch we can measure this using the piezo sensor [BF14]. If a notable vibration is measured we are collecting the next 50 samples, resulting in a window of 250 milliseconds. To distinguish single and double knocks we calculate the weighted average within this window to get a measure for the distribution of sensor values within. If the average is closer to the beginning of the window the resulting event should be a single knock, and a double if the average is closer to the end of the window. Hand localization and knock detection are working independently and are combined later in the software. It is reasonable to combine this, e.g. to ignore knock events that are occurring without a hand present. They may be indicative of a person doing a strong step close to the table.

4.2.4. Image-based processing

Their ability to detect changes in the electric field over a distance has led to capacitive proximity being regarded as similar to cameras. Smith et al. consequently called their approach electric field imaging, as particularly shunt mode measurements and their constrained electric fields allow applying certain image processing methods, e.g. tomography [**?**]. They were critical of using similar methods for shunt mode, noting the following statement.

> Loading mode measurements can be likened to images formed without a lens, since only one "end" of each field line is constrained by the measurement. [SWD*98]

Nonetheless, loading mode has certain advantages, particularly if all electrodes are in a single plane and we would like to have a higher sensitivity at a distance from the plane it is advantageous if there is no receiving potential nearby. One example for this planar electrode setup is large area gesture interaction devices, e.g. a table that is able to track the position of arms and hands in three dimensions. There is a plethora of image-based object detection and tracking algorithms that can be also used for capacitive proximity sensor data processing. There is a short process that I propose to realize this arm and hand tracking that includes some general steps that can be used to identify a variety of objects. The process is distinguished into four distinct steps:

- Creating a grayscale image from the acquired sensor data
- Apply a feature-preserving image upscaling method
- Find the contours of the present objects according to pixel values
- Analyze the image moments of the contour areas and fit human arms

Figure 4.14.: Pixel array mapped from sensor values

The most challenging aspect of the first step is the low resolution of a reconstructed image. In order to achieve a mid-range distance resolution that allows detecting objects within 30 or 40 cm it is necessary to use electrodes that are sufficiently large. Thus, an example device uses an array of 6x4 sensor electrodes, resulting in an image of only 24 pixels. Typically the sensor values are an integer value in a range between 0 and 15000. Accordingly we can create a single-channel image with a channel depth of two bytes. In our case we use a linear mapping of sensor values to pixel intensities. An exemplary result image of this mapping is shown in 4.14 (with enlarged pixels). In this format it is difficult to gather information about the exact position of the arms and thus we need to apply further processing before finding the contours and fitting arm objects.

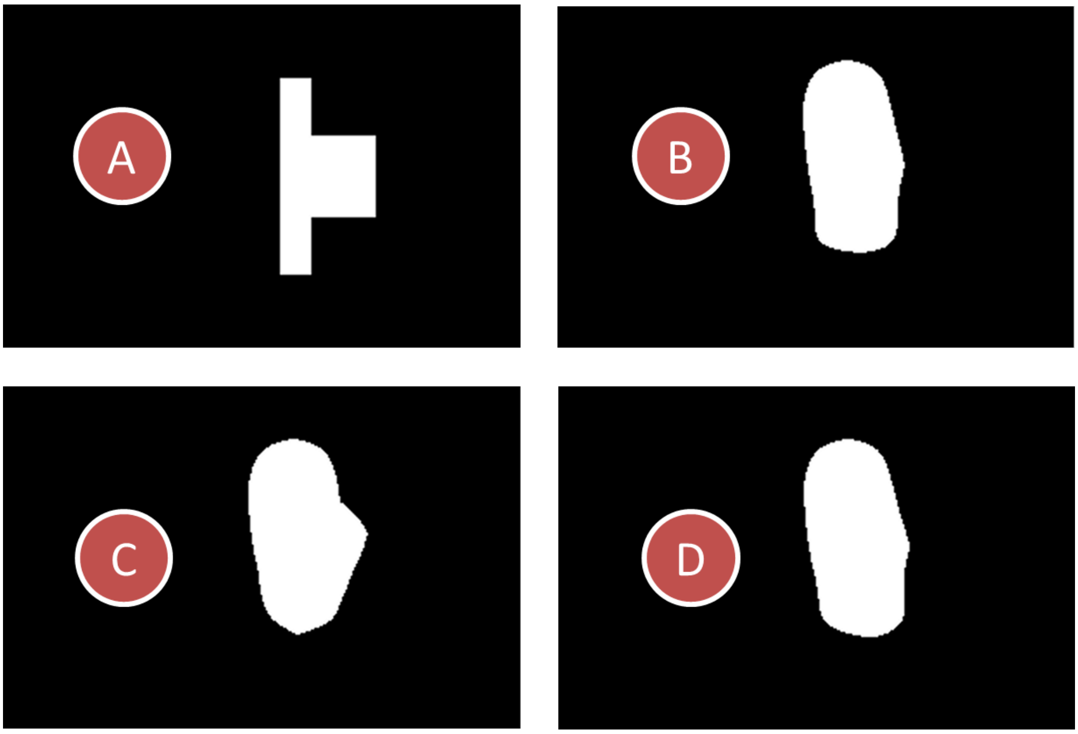


Figure 4.15.: Effect of different upscaling methods on shape, (A) nearest neighbor, (B) bicubic, (C) bilinear, (D) Lanczos4 - shown as thresholded binary images (pixel intensity > 30)

4.2.4.1. Acquire and optimize contours

In order to get the relevant contours of objects in the interaction area we have to apply some further processing. The first step is to enlarge the image using a feature-preserving scaling method. As all sensors are prone to environmental noise we apply some thresholding based on the pixel intensities before looking for contours. The result is an enlarged binary image of black and white pixels. We have tested four different image scaling methods, nearest neighbor, bilinear interpolation, bicubic interpolation and Lanczos interpolation. Exemplary results are shown in 4.15. The Lanczos interpolation showed the best results but is most processing intensive. However, since we are dealing with small images it is reasonable for CapTap. The contours are calculated based on those binary images, defined as the borders between black and white regions. For further processing we are looking into the distribution and the intensities of the pixels within the specified region.

4.2.4.2. Palm and arm fitting

The last step of identifying and tracking the arms is to fit the position and orientation of the palms and arm into the acquired object contours. For this task we are analyzing the image moments within the contours. These are certain particular weighted averages of pixel intensities, or a function thereof [Hu62]. They can be calculated using the following equation, whereas j and i define the order and $I(x,y)$ is the pixel intensity at a given position. We can use this to calculate the center point $(\bar{x},\bar{y})$, leading to the central moments mu_ji that are required to

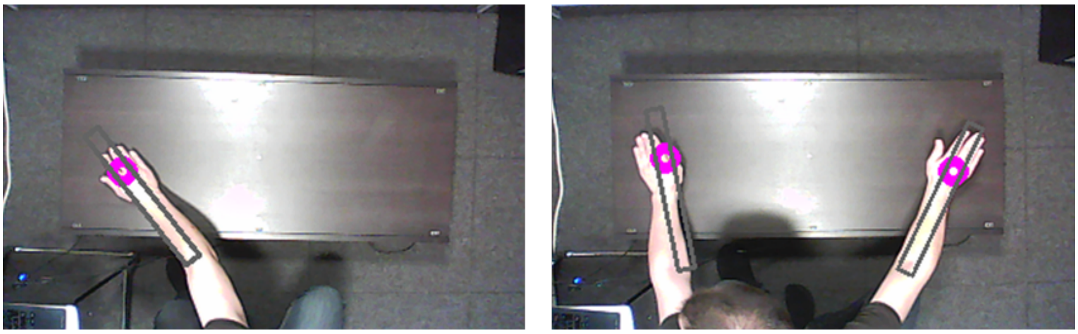


Figure 4.16.: Overhead camera picture of the scene overlaid with live arm and palm reconstruction for one arm (left) and two arms (right)

determine the orientation of the contour as angle γ.

$$m_j i = \sum_{(x,y)} I(x,y) x^j y^i \tag{4.8}$$

$$\bar{x} = \frac{m_1 0}{m_0 0}, \bar{y} = \frac{m_0 1}{m_0 0} \tag{4.9}$$

$$mu_{ji} = \sum_{(x,y)} I(x,y)(x-\bar{x})^j (y-\bar{y})^i \tag{4.10}$$

$$\gamma = 0.5 \cdot arctan \frac{2 \cdot mu_{11}}{mu_{20} - mu_{02}} \tag{4.11}$$

We use the center point and orientation to calculate the estimated position of the palm of the hands. These points are the basis for the subsequent gesture recognition. Additionally, we are using separate Kalman filters for smoothing the different palm positions and arm orientation. The resulting arm reconstruction and the actual arm position in a photo are shown in 4.16. We installed a simple webcam above the table and registered the table position to the camera image.

The arm reconstruction so far is mostly used to determine the arm position. Another potential use of the arm orientation is to improve the merging of two hands. While the system can't distinguish from a single sensor if one hand is close or two hands are further away, we can use the presence of two arms to identify the overall number of objects in the detection range.

4.2.4.3. Intensity-based elevation estimate

A distinct challenge of the capacitive hand tracking is the considerable directional difference in available resolution. While we can use the presented image analysis to track the planar position of the arms over the whole table area of 80cm width and 50cm depth, estimating the elevation of the arm above the table is restricted by the proximity range of the single sensor. Typically the achievable range maxes out at around 35cm, depending on environmental conditions. In a plate capacitor system the distance d is proportional according to size of the plates A and resulting capacitance C. Due to the linear mapping of sensor capacitance measurements to pixel intensities I we can use the image moment within a contour S as estimate of the actual capacitance, and calculate the elevation e according to the following equations:

$$d \propto \frac{C}{A} \qquad\qquad S \propto \frac{m_{00}}{\int S} \tag{4.12}$$

The same thresholds discussed in the contour retrieval phase apply to this step, thus leading to discarding objects at a larger distance that are difficult to detect. Starting from this threshold we normalize the resulting elevation according to a maximum threshold for m00 that denotes a very close object (such as touch). The actual touch recognition is performed using acoustic methods. As previously explained the sensors are prone to environmental influences, thus this just allows to get an estimate of the actual elevation and no absolute distance value. Therefore, the interaction should not be designed to require a highly precise discrimination of different elevation values, but instead use more of a 2.5D paradigm. Our take on this will be presented in the application section.

4.2.5. Physiological signals in frequency- and time-domain

4.2.5.1. Respiratory rate

4.2.5.2. Sleep phase recognition

The most reliable way to track sleep phases is by using an electroencephalography (EEG); that is measuring the electrical activity of the brain by placing electrodes on the scalp. Various different types of neural oscillations can be distinguished - the most important for sleep phase detection are alpha waves, theta waves, delta waves and sleep spindles. The American Academy of Sleep Medicine (AASM) distinguishes three different phases of non-rapid eye movement sleep (NREM) and REM phase [11].

- Stage 1 - occurs mostly in the beginning of sleep. It has slow eye movement, alpha waves disappear and the theta wave appears.
- Stage 2 - dreaming is very rare and no eye movement occurs. The sleeper is quite easily awakened. EEG recordings have a tendency for characteristic "sleep spindles"
- Stage 3 - was previously divided into stages 3 and 4. It is slow-wave sleep (SWS) or deep sleep. Stage 3 used to be the transition between stages 2 and 4 where delta waves began to occur, while delta waves are dominant in stage 4.
- REM sleep - is a phase of sleep characterized by random and rapid movement of the eyes. It is considered the lightest phase of sleep and occurs all through the night but gets longer close to morning.

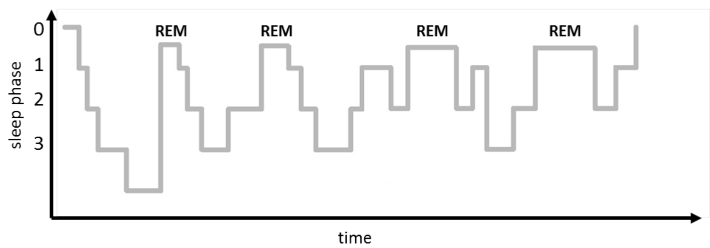


Figure 4.17.: Example of human sleep phases throughout the night

A typical distribution of sleep phases throughout the night is shown in Figure 4.17. It can be easily seen that the sleep is distributed into different cycles, whereas the sleeping person is moving through the different sleep phases until having a REM phase and then going back to deep sleep. If the only available data is body movements it is becoming more difficult to reliably determine the sleep phase. Studies have shown that the magnitude of movement is typically associated to the follow-ing phases in decreasing order: wake, stage 1, REM, stage 2, stage 3 [12]. Another method is distinguishing between awake phase, active sleep and quite sleep and takes into account the order of those phases. This information allows to correlate the actual sleep phases with good certainty [13]. We have chosen this method for our system.

Capacitive proximity sensors enable us to detect the presence of suitable object and their relative proximity to the electrode. Consequently a moving object will cause a change of sensor values. If we aggregate these data deviations from an array of sensors we get a reliable measure of objects moving above the electrodes. In the case of MoviBed we can assume that there is a limited number of persons moving on top of the sensors and thus it is possible to associate the sensor values to movement. In the following we will present a suitable method to achieve a reliable detection of the movements of a sleeping person. We are following a similar approach as Salmi and Leinonen [13]. At any given time t a set of the latest values of all n sensors can be stored as a tuple in the following form:

$$\overrightarrow{s_t} = \begin{pmatrix} s_{1_t} \\ s_{2_t} \\ \vdots \\ s_{n_t} \end{pmatrix} \tag{4.13}$$

As capacitive proximity sensors are particularly susceptible to external influences, such as temperature, humidity and other electric fields it is necessary to apply filtering on the sensor values. A suitable candidate is a median filter - a low-pass filter method that selects the median object of a sorted set of values, thus discarding outliers and strongly deviating values. This is particularly suited if transmission errors may occur. If a person is moving on the bed the value of all sensors in detection distance of the moved body parts will change accordingly, the most relevant example in our case being a person moving in its sleep. We can generate a measure of movement intensity by comparing the values at time t with those at time t-1 resulting in:

$$\overrightarrow{d_t} = |\overrightarrow{s_t} - \overrightarrow{s_{t-1}}| = \begin{pmatrix} |s_{1_t} - s_{1_{t-1}}| \\ |s_{2_t} - s_{2_{t-1}}| \\ \vdots \\ |s_{n_t} - s_{n_{t-1}}| \end{pmatrix} \tag{4.14}$$

In subsequent calculations we will use $\overrightarrow{d_t}$ as combined measurement. For distinguishing between wake, active sleep and quiet sleep we are solely interest in the most intense movement. Thus we are testing for the largest value over a set of m samples, generating the value b_t.

$$b_t = max(\overrightarrow{d_t 1, d_t 2, \ldots, d_t m} \tag{4.15}$$

The value b_t is affected by changes in the speed of movement. Therefor as a final step we generate a centered average value of order $2q-1$:

$$\overline{b_t} = \frac{1}{2q-1} \sum_{i=-1}^{q} b_{t-i} \tag{4.16}$$

The resulting value $\overline{b_t}$ allows us to quantify the intensity of movements over a given period. In order to extract an actual body movement from this value we have to quantify a threshold $s(t)$ that is determined by the average of q previous values of $\overline{b_t}$ multiplied with a factor f that has to be evaluated individually for each configuration of bed and sensors. This threshold $s(t)$ allows us to identify a movement m at any time t. This behavior is denoted in the following equations:

$$s(t) = \left(\frac{1}{q} \sum_{i=1}^{q+1} \overline{b_{t-1}} \right) \cdot f \tag{4.17}$$

$$m_t = \begin{cases} 1, if\ \overline{b_t} > s(t) > \overline{b_{t-1}} \\ 0, else \end{cases} \tag{4.18}$$

As previously mentioned it is difficult to determine sleep phases solely by monitoring the movement. Instead following the example of Salmi and Leinonen and distinguish three phases - wake, active sleep and quiet sleep [13]. These are determined by dividing the sleep time into a three-minute epochs e_{i_a} and qualify these as active or quiet by counting the number of movements occurring in those intervals and comparing it to the average amount of movements in all epochs $\overline{e_a}$ determined by the following equations:

$$e_{i_a} = \sum_{e_{i_{Start}}}^{e_{i_{End}}} m_i \qquad\qquad \overline{e_a} = \frac{1}{n} \sum_{i=0}^{n} e_{i_a} \tag{4.19}$$

In consequence we determine the status of any epoch with this final equation:

$$e_{i_a} = \begin{cases} active, if\ e_{i_a} > \overline{e_a} \\ quiet, if\ e_{i_a} \leq \overline{e_a} \end{cases} \tag{4.20}$$

These active and quiet periods can be semi-autonomously interpreted by humans in order to determine the actual sleep phases. For example initial activity for 20 to 40 minutes followed by a quiet period can be attributed to a person falling asleep. Following quiet phases are a good indicator for deep sleep phases.

4.3. Application prototypes

Table 4.3.: Overview of developed capacitive proximity sensing prototypes

Name	Description	Application Areas	Measuring Layout	Data Processing
CapFloor	Capacitive system for indoor localization and fall detection based on electrode grid below the floor.	Indoor Localization	Loading mode, variable number of sensors based on area size	Binary activity association and using geometry for positioning. Monitoring of overall value for falls.
Smart Bed	Capacitive sensors placed below mattress able to determine sleeping postures and breathing rate.	Smart Appliances, Physiological Sensing	Loading mode, four sensors on each side of bed	Posture fitting using a static model. Fourier analysis for breathing rate recognition.
The Capacitive Chair	Office chair equipped with capacitive sensors to distinguish different typical work postures and stress levels.	Smart Appliances, Physiological Sensing	Loading mode and shunt mode, eight sensors, heterogeneous sensing capabilities	Model fitting using a dynamic model. Fourier analysis for breathing rate detection. Posture recognition using machine learning.
Active Armrest	Heterogeneous system for finger gesture recognition and arm rest identification for automotive applications.	Smart Appliances, Gestural Interaction	Loading mode, heterogeneous layout	Finger positioning using direct calculation. Binary arm presence detection.
MagicBox	Mobile 3D gesture interaction device using an array of electrodes.	Gestural Interaction	Loading mode, six wireless sensor nodes	Geometric detection of hand relative to plane. Adapted mouse methods for gesture recognition.
CapTap	Table capable of detecting 3D gestures and knocks to realize tactile interaction in a living room.	Smart Appliances, Gestural Interaction	Loading mode, 24 capacitive sensors and a single touch detecting microphone	Image-based hand and arm detection. Independent touch detection. Tracking of multiple objects.

In the last few years I have created a number of different prototypes using capacitive proximity sensors in various usage scenarios within smart environments. They tackle specific application domains and implement one or more of the data processing methods that have been specified in the previous section. A short overview can be found in Table 4.3. During the next few pages I will describe in detail how the prototypes have been created, how they implement the different data processing methods and outline the results of any technical evaluation and usability study that has been performed.

4.3.1. CapFloor

CapFloor is a capacitive system for indoor localization and fall detection that is based on a grid array of sensing electrodes placed below a floor covering [BHW12]. A sketch of the system is shown in Figure 4.18. The grid is comprised of insulated wires that are placed orthogonal to each other. Sensors are placed on two sides of the room. Each sensor is performing loading mode measurements. The system is intended to act as both indoor localization system and fall detector. CapFloor can be placed below any non-conductive material, like wood, tiles and PVC, if the distance between the wires and the floor surface is not too high. It can discriminate between a foot being above an electrode or a whole body. Combining this information from various sensors we are able to get a reliable detection of lying, sitting and standing persons. Using only two sides of the room for sensors it is possible to cut the wires without considerably affecting the signal; allowing easy installation in non-rectangular

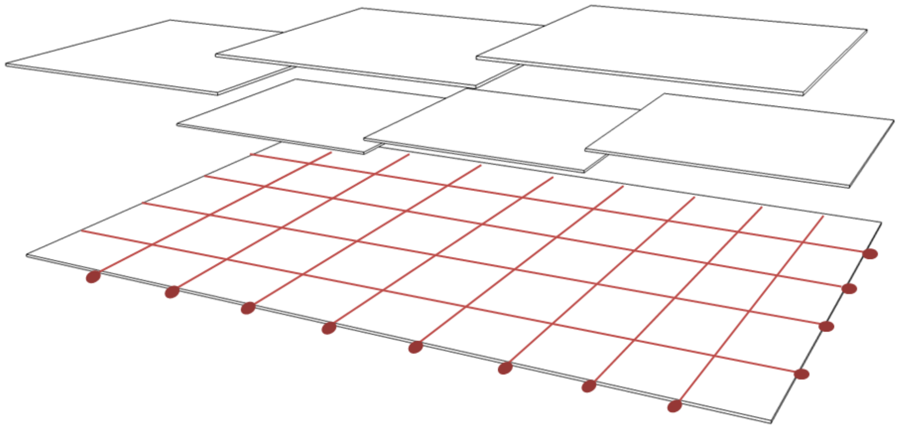


Figure 4.18.: CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides

rooms. Accordingly CapFloor is able to be used in various application scenarios. Indoor Localization in the home domain can be useful in energy saving and fall prevention by appropriately activating and deactivating the environment lighting. It can also be used in security-restricted areas to detect unauthorized movement. The fall detection should be used in a system that has various levels of escalation. E.g. it is not easy to distinguish between a person doing exercises on a floor and a person that has fallen down. Accordingly the system should query if the person is well and not autonomously call for outside help.

4.3.1.1. Evaluation

The CapFloor system was evaluated in the scope of the Indoor Localization Track of EvAAL 2011, where it participated out of competition [CK12]. In Figure 4.19 we can see a picture of the demonstration setup installed in the living lap using the system integrated into different mats that are placed in the environment. The system was tuned to detect a single person and was able to perform this reasonably in the areas covered. The resolution of the system is strongly depending on the given density of electrode wires. While there is a certain measure of proximity, it is not possible to detect objects that are more than a few centimeters away from the wires. Later iterations of the system are using higher voltages and shunt mode measurements to improve the tracking reliability and enhance the fall detection.

Figure 4.19.: Floor mats with integrated CapFloor system used at the EvAAL 2011 competition [BHW12]

4.3.2. Smart Bed

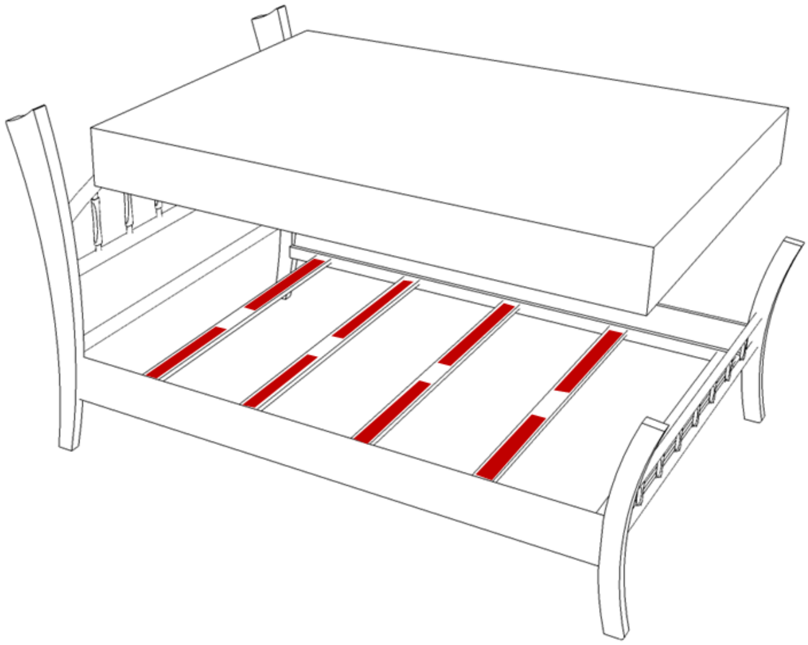


Figure 4.20.: Smart Bed sketch - flexible plate electrode are attached on spring board

The Smart Bed is a regular bed frame that has been equipped with capacitive proximity sensors in order to determine occupation, posture and sleep phases [BH12] [DBM14]. A sketch can be seen in Figure 4.20. The electrodes are comprised of copper foil that is attached to the flexible wooden panels of the slatted frame. This allows the electrodes to be sensitive to both proximity and applied pressure, resulting in a superposed combined sensor value that is considerably higher as opposed to proximity measure on its own. The electrodes are equally distributed, with four being on both sides of the two person bed. The system is able to determine different sitting and lying postures of one or two persons, including less regular lying positions such as diagonal or orthogonal to the long side of the bed. Using an analysis of the movement gathered by variation in the sensor signal the sleep phases can be analyzed, similar to accelerometer-based systems that are popular for smartphones [KJJ11].

The Smart Bed can be used for various purposes. A main application is connecting the occupation detection to a home automation system and timer in order to activate ambient lighting if the person is get-ting up in the night, presumably to find the way to the restroom. Accordingly, in a single person household the lights in unoccupied rooms could be turned off in order to conserve energy. In the domain of personal health the Smart Bed is able to give the user a feedback on sleep quality based on the sleep phase measurement performed in the night. Another potential application is to use the acquired pressure distribution as indicator for back-friendly lying positions that may be harmful over a longer period of time [HB10]. The occupation and posture detection relies on a simplified body model to approximate the pressure distribution and sensor values to a certain posture [BH12].

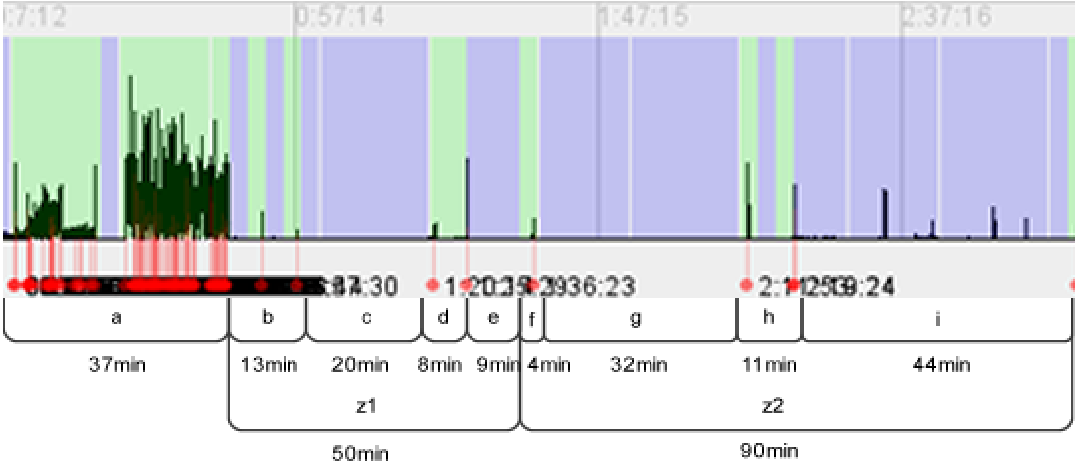


Figure 4.21.: Sleep movement data over three hours in one night [DBM14]

4.3.2.1. Evaluation

The Smart Bed posture recognition is able to successfully distinguish eight typical sitting and lying states. Using adaptation of the intermediate values it is possible to fit the state to an actual position on the bed, e.g. a *person sitting on the right side of the bed* state can be modified to any location on that specific side of the bed. Regarding the detection of sleep phases there has been an evaluation and benchmarking of three nights [DBM14]. The Smart Bed was able to achieve a comparable performance to smartphone applications that detect sleep phases based on accelerometers. Figure 4.21 gives an example of movement recordings using the capacitive proximity sensors over one night. The activities are grouped into distinct chunks that are later associated to the sleep phases. Currently breathing rate detection is added to the Smart Bed that can be used to improve the sleep phase detection and also can potentially detect anomalies that may be indicative of a certain health risk.

4.3.3. The Capacitive Chair

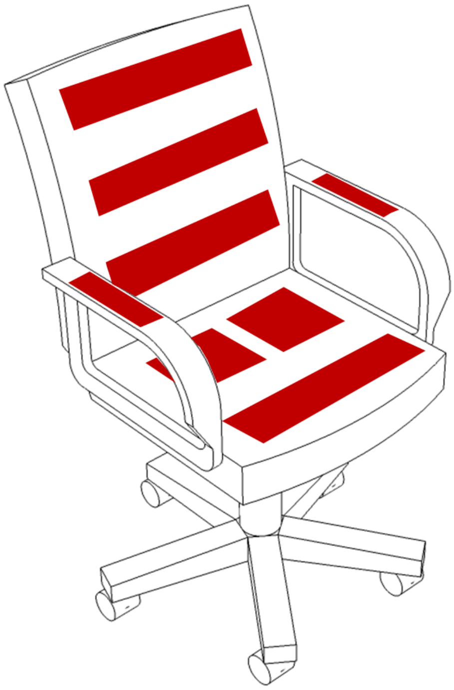


Figure 4.22.: Smart office chair sketch - eight electrodes three in backrest, three on seat and two in armrests

The Capacitive Chair is a regular office chair equipped with eight capacitive proximity sensors that can detect different sitting postures and work-related stress levels by examining movement and breathing rate [BF14]. Seven solid copper electrodes that are placed below the covering are augmented by a single conductive thread electrode that is placed in a mesh on the backrest. In the past smart chairs have used pressure sensors to infer posture and occupation [TSPM01]. Combining presence and proximity sensing it is possible to directly infer postures where parts of the body do not touch the surface, e.g. if the body is arched towards the front, or if an arm is raised from the armrests. Additionally higher area electrodes in the backrest allow detecting the breathing rate by measuring the movement of the chest.

The Capacitive Chair aims at providing different services to a typical office worker and office managers. Using the occupation detection it is possible to advise for some type of physical activity, if the time spent in front of the screen was too long. The system can also advise the user to change to a more back-friendly posture or regularly switch the stance to achieve a more general workout. Using the breathing rate detection we are able to get some sort of measure of the current stress level associated to the given working situation. By adapting the environment it is possible to improve the working atmosphere and reduce stress. The Capacitive Chair uses a multifacetted data processing approach. A machine learning algorithm is associating the sensing data to one of nine different typical sitting positions, inspired by a recent study of sitting positions for modern device usage [Inc13]. An adaptive body model that is fitted to the current sensor values allows for fine grained adaptation of those postures. Finally a combination of Fourier and data variation analysis is calculating the current breathing rate [BF14].

4.3.3.1. Capacitive layout

The Capacitive Chair is based on a single OpenCapSense board that supports eight different electrodes. In order to get the posture measurements we need to distribute the electrodes equally on the different areas of the seat. The measurement of the breathing rate requires a larger electrode near the chest area. Consequently the electrodes are placed as follows:

1. Electrode on the upper part of the backrest (covered by faux leather)
2. Electrode in the central part of the backrest (using conductive thread)
3. Electrode in the lower part of the backrest (covered by faux leather)
4. Electrode below the right armrest
5. Electrode below the left armrest
6. Electrode for the left hip area below the left part of the seat
7. Electrode for the right hip area below the right part of the seat
8. Electrode for detecting both legs below the front part of the seat

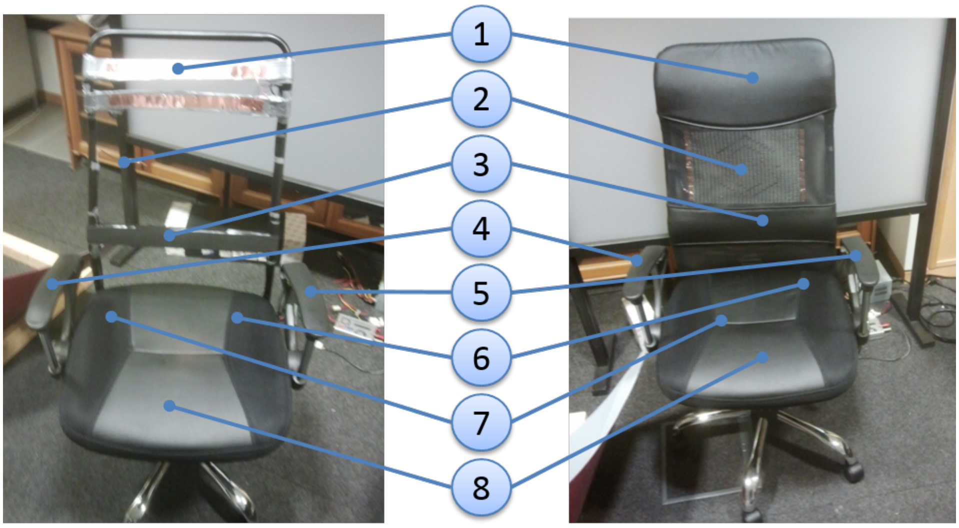


Figure 4.23.: Capacitive Chair electrode positions

The electrode is connected to channel 0 (CH0) of the OpenCapSense evaluation board. The following figure shows the layout of the electrode (2) including sensing electronics (5). The shield electrode is additionally the support structure for the whole setup here. The shield electrode is comprised of copper sheet bedded in duct tape. On the duct tape there are strips of copper sheet applied using conductive glue (2). The copper sheet is the sensing electrode connected to the sensor (5) using the blue wire (4). The shield electrode is connected using the red wire. The frame of the backrest is indicated using the number (6). This electrode in the lower part of the backrest is connected to CH2 of the OpenCapSense evaluation board. The layout is analog to the one on the upper part of the backrest, comprised of shield electrode (2) covered by duct tape and a copper sheet electrode. The electrode on the right armrest is connected to CH3, the one on the left side to CH4 of the OpenCapSense evaluation board. Both electrodes are comprised of a copper sheet fixed to the armrest using duct tape. The electrode below the right hip area is connected to CH5, the one below the left hip area to CH6 and the leg electrode to CH7. The figure above shows the electrodes. All of them are made of unprocessed, two-layer copper PCBs. They are isolated to the environment using duct tape. The electrode for the leg area is comprised of two distinct PCBs (2,3) that are connected using copper wire (5). The hip electrodes (1) are similarly comprised of copper PCBs. The wires are guided through the wooden seat using small drill holes (4,6). The red wire leads to the sensing electrode while the grey wire leads to the shielding.

Figure 4.24.: Detail view of conductive thread electrode

The electrode in the central part of the backrest is connected to CH1 of the OpenCapsense evaluation board. The electrode (1) is comprised of conductive thread that was woven into the covering of the backrest. The ends of the conductive thread are connected to a conductive copper foil. This foil is formed in such a way (4 left) that a terminal (4 right) can be applied and connected to the sensor electronics (3). This type of electrode does not support any shielding. In order to remove the covering the terminal should be disconnected from the electrode first.

4.3.3.2. Processing

The first step of data processing is filtering. We have implemented different types of filters, including static average and floating average filters. In this case we are using a median filter that is taking the median value of eight previous samples. After filtering we are building the baseline for each sensor channel. The baseline is the minimal sensor value that is created by sampling the environment without any object present. There is a plausibility check in this step to discard values that deviate too far from the norm. The maximum values of a channel are collected on run-time. Again we are using a plausibility check. The final step of pre-processing is a normalization based on acquired minimum and maximum values. For further processing we additionally need information about short-term value variance that we gather by calculating the difference quotient using a sample of ten average filtered measurements. Afterwards we are performing a fast fourier transformation (FFT) to get information about the frequency spectrum of the sensor values, in order to perform breathing rate detection.

Breathing rate detection

We are using the FFT values of sensors attached to the central backrest to get the current breathing rate. A binning operation is performed to look for significant signals in a reasonable frequency interval (0.1Hz-3Hz). In order to increase the reliability of the breathing rate detection we use a second method. Based on the normalized values a mean value curve is calculated. The intersection points of this mean value curve and the current sensor values are additionally stored. We are using a dynamically weighted combination of both values to increase the reliability of the breathing rate detection.

Posture recognition, kinematics of the human body

The processed values of all sensors are compared to previously trained sitting positions of a user. The position with the lowest deviation is considered the current posture. Currently the system supports nine different postures; however it can be dynamically extended or reduced. Based on the normalized sensor values and geometric positions of the sensors the data is interpreted as position of the different joints of a user. 4.2.4 Output The GUI allows displaying of raw and processed data. In the following section we are presenting the different forms of interaction.

Figure 13 GUI with four opened windows Figure 11 gives on overview of the GUI. Selecting the desired output in the ToolBox (1) opens the associated window. In this case we can see the data display of sensor channel 1 (2), the recognized breathing rate (4), the FFT of sensor channel 1 (3) and the recognized postures and their deviations (5).

Figure 14 GUI with two windows Figure 12 shows two additional windows. On the left side (1) we can see a picture depicting the currently recognized posture; on the right side (2) we can see the human model with recognized joint positions. The 3D joint recognition is still in strong development and will be remodeled in the future. Additional screens that have not been shown in this overview are a serial monitor that displays the raw data acquired from the USB connection, the collection of measurements using software queries, the display of all sensor values in table format and a repositioning of the different windows.

Distinguish work activity levels

Figure 15 Work Activity aggregation over a single work day (mock-up) Figure 13 shows a mock-up of a typical work day activity over a single work day. We assume the work day of a typical office worker and support three different aggregated activities: Active work as indicated by a certain level of movement while on the chair Passive work as being present on the chair while not moving a lot Not present at desk, whereas no one is currently sitting on the chair.

4.3.3.3. Evaluation

The Capacitive Chair was partially supported by the EIT ICT Labs project Cognitive Endurance during 2013. In this scope it was evaluated in two distinct studies. The first aimed at testing the aggregated recognition of working activities with several persons over various days. The second study was testing the posture recognition with various users that were additionally queried about their general impression of the system. In this section we are presenting results of both studies.

Working situation recognition

The sensing chair supports distinguishing two different working situations that are determined using the method described in the previous section. The system also supports sending to the Cognitive Endurance server.

Figure 4.25 shows an example of this generated activity log. We have performed a test over 3 days between December 4th 2013 and December 6th 2013 on a typical work day in the office. The resulting activity logs were used to generate a chart as shown in the previous section. An example chart is shown in Figure 15.

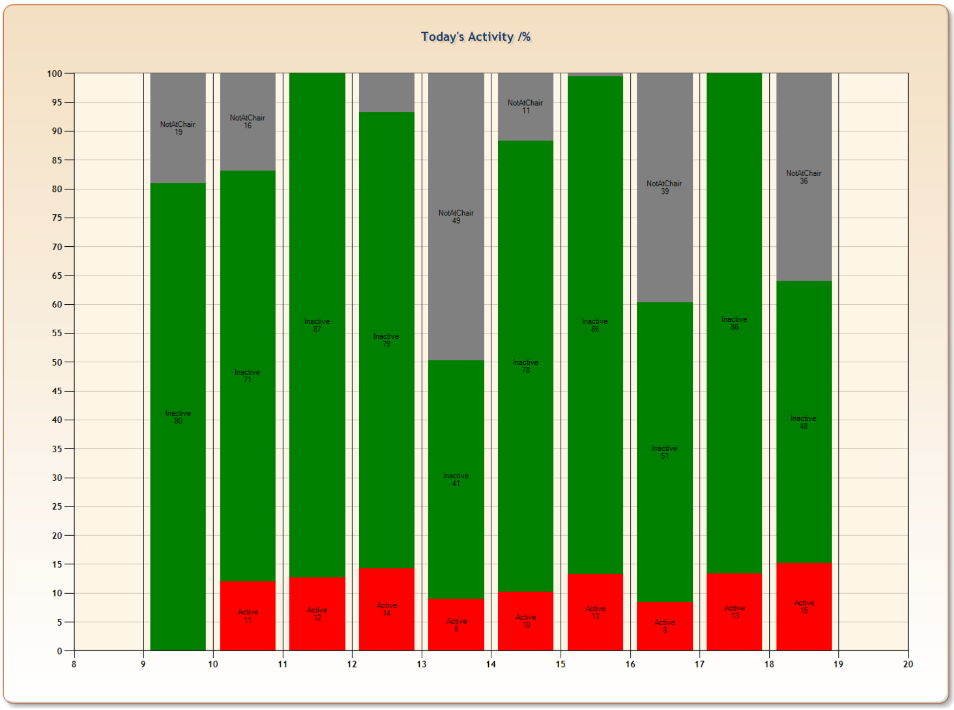


Figure 4.25.: Example chart of work activity data collected

We can clearly see some phases of not at chair - usually for lunch break or some meetings and the work is distributed between active work, such as writing and typing and longer phases of inactivity (such as reading).

Posture recognition - test 1

In a second evaluation we were testing the posture recognition of the chair in a short study with 10 participants. Our system was tuned to distinguish three poses and a non-pose:

- Sitting upright
- Sitting hunched
- “Slouching on chair”

- Close to chair - disturber

The persons were given a short introduction, the different postures were displayed, and finally the persons were asked to perform the postures in order. When testing "close-to-chair" the subjects were asked to rattle at the chair, stand close, move it around and thus disturb the potential sensor readings. Each class was tested for 10 seconds, collecting 200 samples. Some impressions can be found in the following pictures:

Figure 4.26.: Disturber position of a participant (left) and sitting upright (right)

Figure 4.27.: Slouching position (left) and sitting hunched (right)

Overall the results were very convincing. Of the 40 different measurements series only two were not achieving 100% accuracy. The Upright and Disturbance positions were classified correctly for all candidates. A single candidate had an 86% rating on the hunched posture. A different candidate had a 55% rating on the slouching position. The average of correctly classified postures is 98,5%.

4.3.4. Active Armrest

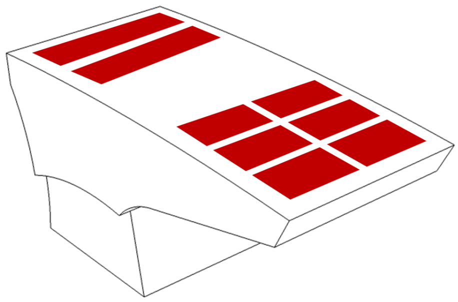


Figure 4.28.: Active armrest sketch - six electrodes for finger gesture detection in front, two for arm detection in back

Touch screens are by now one of the most ubiquitously used interaction method in HMI with billions of finger-controlled smartphones being in use [9]. This trend is also apparent in vehicles, with touch screens and touch pads becoming more common. The Tesla Model S provides a large area touch screen that completely replaces conventional button-based interfaces. BMW augmented their iDrive control system with a touchpad that is able to recognize finger gestures. However, touchscreens have been identified as potentially distracting for the driver [12] and systems like iDrive rely on mechanical control systems that protrude from the other features of the car interior. Capacitive touch sensing is the most common technology for creating touchscreens [2]. A variety is capacitive proximity sensing that enables to detect the presence of conductive objects, such as the human hand over a distance [14]. It can be used to create gesture input devices that recognize free-air hand or finger movements [4]. As it is based on measuring the electric field it can be applied below any non-conductive material, e.g. plastic, wood, or leather that are commonly used in car interiors. Thus, we can create interactive surfaces and interactive zones in free-air that sense the movement of the hand, yet are completely integrated into a car environment. A suitable area for creating an interactive zone is the armrest, as it is the intended resting position in the first place. However, this creates an additional challenge. As the majority of interactions between arm and armrest are not intended to control aspects of the car system, we need concepts to infer the intention of the driver to interact with the car. We propose two different methods that utilize the capability to detect the presence and distance of the arm from the surface. The first option enables interaction only when the arm is raised and fingers are touching the interactive area in front of the armrest. The second option assumes that the arm is resting on the surface and the fingers are performing small gestures above the interactive area in the front. To test the validity of the invisible interactive areas and the two interaction concepts, we have created the Active Armrest, a prototype comprised of an aftermarket armrest with an integrated heterogeneous array of eight capacitive proximity sensors - two using large electrodes for detecting arm proximity and orientation and six using small electrodes to create an interaction area in front of the armrest. We have adapted methods to classify both free-air and touch gestures for capacitive proximity sensors using a SVM classification method and created a demonstration application that mimics typical multimedia functions in a car. We have performed an evaluation of the classification precision and a usability test of both methods in a study with ten participants. 2. RELATED WORK The list of research into automotive interfaces is extensive. Alpern and Minardo presented a study on different forms of gesture interaction within a car, performed by the right hand while the left stays on the wheel [1]. The effects of the interaction were displayed on a HUD. They found that gestural interfaces in their setup were preferred to traditional interfaces for secondary tasks, if the visual feedback was appropriately designed.

Bach et al. compared three different interaction techniques in cars and their effects on driving performance and visual distraction. They found that interaction on touch screens is fastest, but most distracting and that gestural interaction, while slower, does not lead to an increase in interaction errors [8]. Regarding large touchscreens, e.g. used in the Tesla Model S, Rümelin and Butz explore different interfaces, including adding a knob as physical element [11]. They discovered that there was an advantage in learnability and task completion time for all direct touch interfaces, as opposed to remotely controlled systems and that driving performance was not affected negatively. Döring et al. integrated a multi-touch system into a common steering wheel to enable distraction free control of comfort functions in a car [6]. The system also allows creating different visual interfaces in the steering wheel. Their findings include a significantly reduced visual load as opposed to center console touch navigation and button controlled radio. Burnett et al. compared rotary controls, touchpads and touchscreens for their performance in different tasks [5]. They found that rotary controls performed the worst and that preference for touchscreen or touchpad is depending on the task, with the first having an advantage in more complex menus and the latter being preferred in tasks that can be completed using simple commands. Pfleging et al. combined speech input and gestures performed on a tablet integrated into the steering wheel [10]. They showed that this approach has similar distraction compared to traditional interfaces, while providing a higher degree of flexibility.

4.3.4.1. Evaluation

Figure 4.29.: Active Armrest prototype, left - outside view, right - detail view of electronics

In order to evaluate the Active Armrest we have built the prototype shown in Figure 4.12. An aftermarket armrest was equipped with an OpenCapSense toolkit. The demonstration application is based on the SenseKit debug software supplied with the toolkit. As of now there is a simple USB connection to a nearby PC. Figure 4.30 shows a screenshot of the finger tracking application on the left, with a two-finger touch registered on the upper left part of the touch area. It is interfaced with a TUIO [KBBC05] based maps application using OpenStreetMap [HW08] data. The map is moved around using simple swipe movements of the finger that are directly associated to pan-features of the demonstration application. Zooming is activated by two-finger hold gestures on the upper or lower part of the touch area. We have used public displays of this prototype to get an idea of how easily unaffiliated persons learn to use the system. While the majority agreed on the potential of the application, there have been some reservations regarding the current gesture set, particularly that a closer relationship to smartphone touch screen gestures would be welcome.

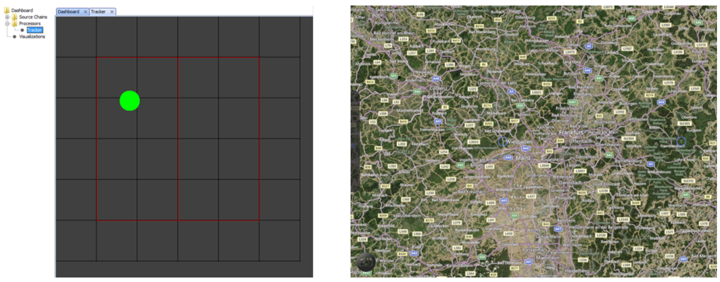


Figure 4.30.: Active Armrest demo software, left - finger tracker, right - OSM based navigation application

4.3.5. Magic Box

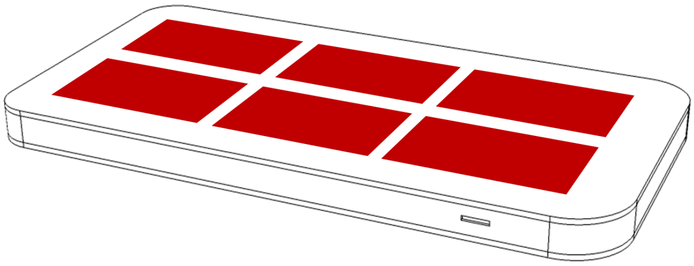


Figure 4.31.: MagicBox sketch - six electrodes uniformly distributed below surface

The so-called MagicBox was our first attempt to create an interaction device based on capacitive proximity sensing. It is using an array of six individual wireless capacitive sensors that communicate to a central station [BH11]. The electrodes are using a large surface area and are made of aluminum foil. A sketch is shown in Figure 4.31. The system is able to track the position of a single hand in three dimensions up to a distance of approximately 20*cm*, and uses different methods to infer gestures from the hand movement. It is designed to be a generic interaction device that can potentially be hidden below non-conductive surfaces. As it can be used without touching it is also applicable in sterile environments. A suite of demonstration applications has been created that showcase typical scenarios for the MagicBox. This includes multimedia applications, like image viewer and media player but also a 3D object viewer intended as demonstrator for potential medical applications, allowing a surgeon to check MRT or CT images in a sterile environment without touching any surface.

4.3.5.1. Evaluation

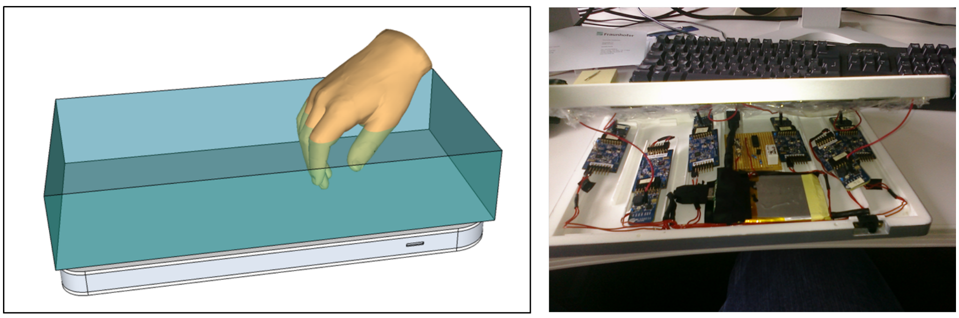


Figure 4.32.: MagicBox conceptual rendering (left) and detail view of electronics (right) [BH11]

The MagicBox prototype is based on the Cypress First Touch starter kit [Cor13] and combines six capacitive sensors communicating wirelessly to a single base station, that are put together with a USB-rechargeable power supply into a casing. A concep-tual rendering showing the interaction area and a detail view of the

Figure 4.33.: MagicBox demonstration application - 3D object viewer (left) and image viewer (right) [BH11]

prototype electronics are shown in Figure 4.32. The different iterations of the MagicBox have been evaluated in conjunction with various demon-stration applications. A usability study with 18 per-sons led to general approval of the system [BH11]. Two of the applications used in this study are shown in Figure 4.33. On the left is a 3D object viewer that has to be controlled by a combination of menu and direct manipulation of the screen content. On the right side there is an image viewer that was controlled by gesture to trigger the next/previous images or perform zooming operations. The most common positive remarks gathered in this study can be roughly put into three groups:

- The device very intuitive to use
- The idea of interacting this way is novel and interesting
- It is easy to control applications with those gestures

Likewise we identified three main groups for negative comments about the prototype:

- The device is not very precise
- The interaction speed is slow
- It can be tiring for the arm

Later iterations have been trying to improve some of the weaknesses presented above, e.g. by using a more sophisticated gesture recognition system and faster sensor refresh rates. Accordingly there were fewer complaints about interaction speed and precision [BDK13]. However, the final complaint about the device being tiring for the arm, requires a different approach, that we are investigating in the final prototype to be presented in this system.

The overall method is similar to mouse gesture recognition, albeit adapted for three dimensional locations. The developed system allows defining an arbitrary set of potential gestures and adding training data. In Figure 4.34 the defined gestures can be seen u. The module is looking for matches based on the most recent set of locations.

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.

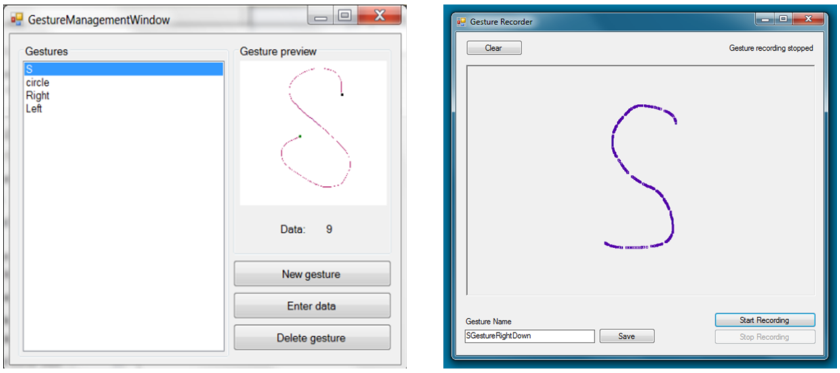


Figure 4.34.: Gesture overview module (left) and gesture recorder (right) [BDK13]

4.3.6. CapTap

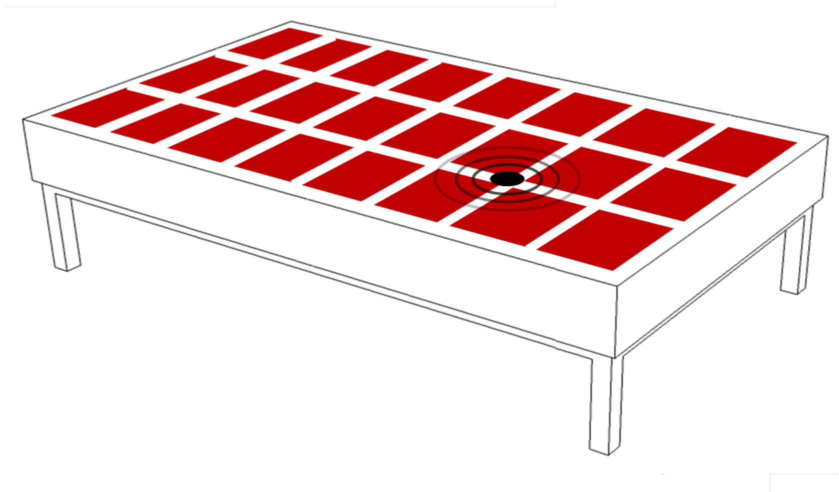


Figure 4.35.: CapTap sketch - 24 electrodes placed under table surface and a single detector for touch events

A general insight of free-air gestural interaction that became apparent even in early works is the physical demands of prolonged interaction with such systems [BBL93,LBT02] and the difficulty to adapt selection events to gestural input - the latter typically being realized by time- or position-based gestures [BBL93, KOR*02]. There is no trivial solution to this challenge and any approach has to take into account the specific application scenario covered. Some systems attempt to provide specifically adapted graphical interfaces, while others include additional input devices assisting the interaction [WB03,ZLB*87]. A major point is decreasing the required time for interaction, e.g. by adding a tactile feedback to the interaction system, preventing time-based selection gestures. CapTap is a regular living room table that includes a capacitive proximity sensor array for tracking the position of one or more arms. As it is difficult for this sensor category to detect touch, if the interaction surface is at a distance from the electrodes, a hidden acoustic system is added that allows recognizing different touch events. CapTap tracks hand and arm position using the image-based object recognition previously presented and fuses this data with different touch events generated by a single contact microphone that analyzes the audio signals in frequency space. This allows to significantly reduce interaction time, as opposed to systems relying on time-based dwell gestures. The system was created in 2013 and 2014 with collaboration of several students, most notably Sebastian Zander-Walz and Stefan Frank [BZWK*14]. It is used and further developed within the European research project POSEIDON that aims at providing technical solutions to help persons with Down's syndrome on planning their day, as input device that allows interaction regardless of motoric skill level.

4.3.6.1. Capacitive layout

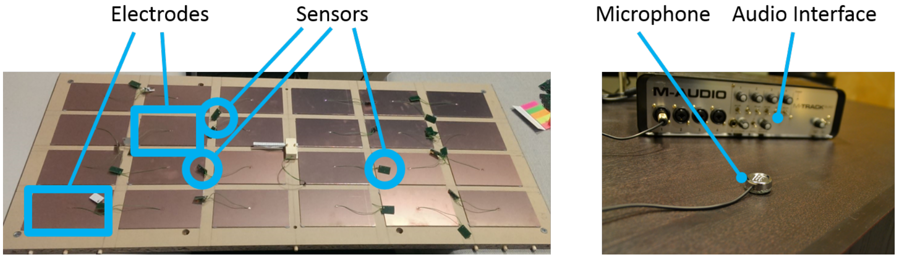


Figure 4.36.: Detail views of the prototype system: left - electrodes and sensors, right - audio interface and contact microphone

Our goal with CapTap is to enable tracking of hands and arms that move above a table. Therefore, the electrodes are placed in an uniform array that provides similar sensing properties for the whole surface. It is realized as a prototype installed in a regular living room table. It is comprised of an array of capacitive proximity sensors, a contact microphone for touch event detection and a miniature PC. All devices can be integrated into the table in a way that it is not distinguishable from the not-augmented piece of furniture. Figure 4.36 shows some detail views of the disassembled prototype. The left image shows the back of the wooden tabletop. The electrodes are arranged in a 6x4 array and each one is attached to a single sensor. The right picture shows the touch detection microphone. It is attached in the center of the surface, as to avoid non-uniform sound distribution over the surface area that would be more difficult to train. Placing the microphone below the surface has no strong influence. However, a specific training phase is required for any novel surface that is equipped with the touch detection devices.

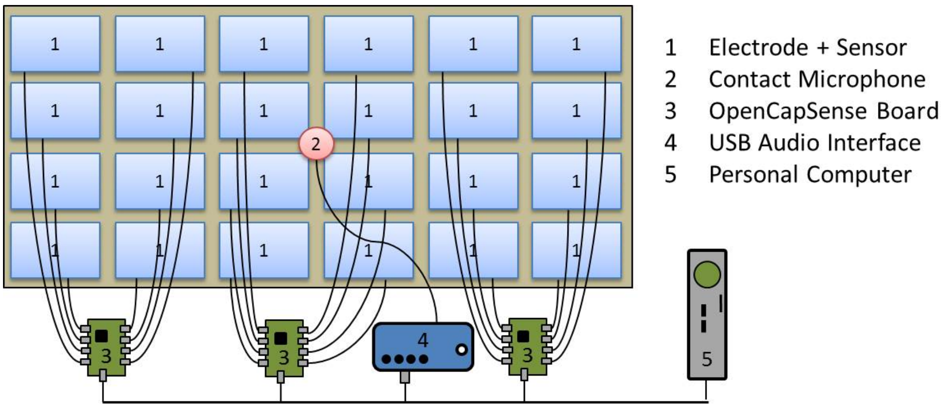


Figure 4.37.: Abstract view below the surface of the prototype including capacitive sensing electrodes and touch detection microphone

The setup is visualized in Figure 4.37. The system is comprised of 24 electrode & sensor pairs that are connected to three different measurement boards. The microphone is attached to a USB Audio Interface. Overall there are four different boards connected via USB to a PC that executes and merges the different types of data processing and links it to the software suite. The prototype is based on OpenCapSense, a more advanced prototyping system presented by Grosse-Puppendahl et al. [GPBB*13]. The boards are performing some prefiltering, whereas the image-based hand tracking is realized on an attached PC. The microphone is attached to an USB audio interface (M-Audio M-Track Quad) that transfers data acquired by up to four microphones and provides various pre-sampling functions. All four devices are attached to a Mini-PC that is performing subsequent data processing and is running the demonstration and testing applications.

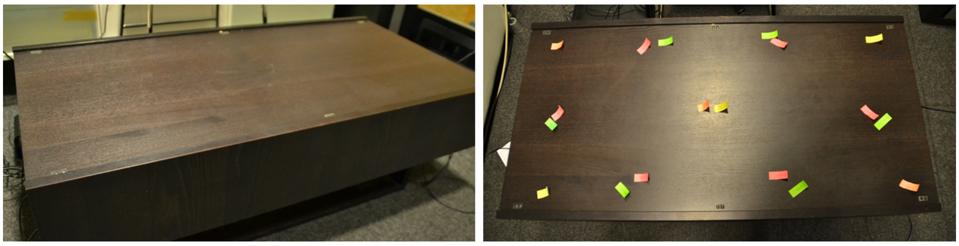


Figure 4.38.: Views of final prototype, complete view (left), top view with markers for touch evaluation (right)

The final table prototype can be seen in Figure 4.38. On the left side the table is seen without any additional markers - on the right side we see the table equipped for the evaluation using a set of markers for different touch and swipe events. The debug software was developed with C# using the .NET 4.5 framework. We are using the Emgu CV library based on OpenCV for image processing and application of the Kalman filter to the determined palm locations. The sound processing is implemented in C++ and Java using a modified version of ChucK for audio sampling and the WEKA framework to apply the machine learning on top. We are using sockets to transfer data between the different modules. The debug application allows a fine control of the various processing steps in both image and audio signal processing.

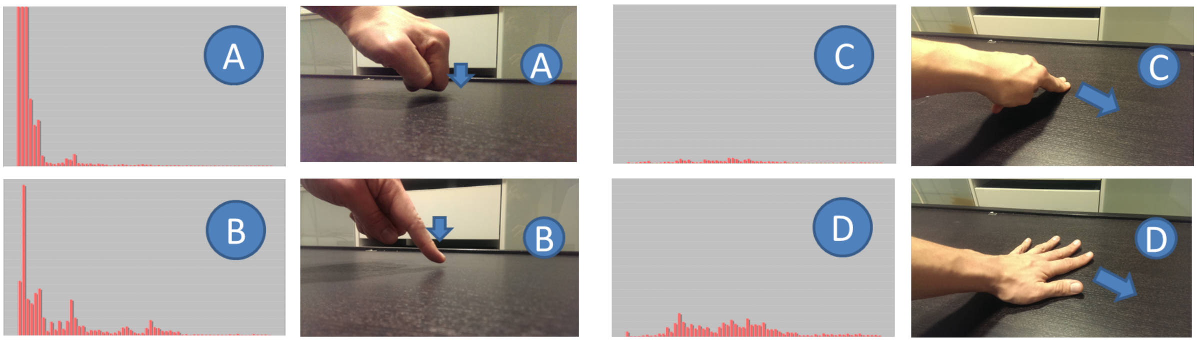


Figure 4.39.: 64 sample FFTs and photo for a knock event (A), a finger tap (B), a finger swipe (C) and a hand swipe (D)

4.3.6.2. Processing

CapTap is using the image-based algorithm for tracking palms and arms that was presented in the processing section. An addition is the touch recognition based on acoustic tracking. The method was inspired by the works of Harrison et al. [HSH11] with various modifications to allow identifying both impact and swipe events. At first the audio signal is acquired using a 96kHz sample rate and a feature extraction rate of 375Hz (using a Hanning type sliding window of 4096 (and 256 samples overlapping) samples per extraction. In order to perform a classification over this signal we have to look at a variety of different features.

Feature vector

The signal differences are most significant in the frequency domain, thus we are performing a FFT over 4096 samples, looking at the first 512 of 2048 magnitude values, thus covering the frequency range up to 12kHz. We are collecting the mean value, the standard deviation and the index of the highest value within the frequency range. This process is repeated for a downsampled FFT of 64 values, similar to the method used by Harrison et al. Another frequency domain-feature we are using is the centroid, i.e. the weighted mean of the present frequencies. Additionally, we are using two time-domain features, the RMS power (root mean square), i.e. the average magnitude within the current frequency band and the number of zero crossings of the signal.

Classification

We have to distinguish two different classifiers that are used for impact and swiping events. Even though knocks and taps are temporal gestures they are short in duration, while the swipe gestures are constant for a longer time period. Example 64-value FFTs are shown in Figure 4.39, with impact events and their low-frequency peaks on the left and swipe events and their fairly constant value on the right. For impact events we are using some metrics to identify the point at which the features shall be analyzed. A simple preprocessing identifies increasing power in lower frequencies and begins to store all feature vectors until a maximum is reached or the power is decreasing again. Not relevant secondary power increases (that are prevalent on stronger knocking events) are ignored. The feature vector corresponding to the maximum is then put into the classifier. This is a trained SMO classifier comprised of a support vector machine and a polynomial kernel that matches five different impacts - single knock, double knock, single finger tap, single double tap, and stomps that are classified but not used any further. The classification of different swipe events is realized using a sliding window over a set of previous feature vectors. The derived feature set is comprised primarily of average and standard deviations of the single items within the previous feature vectors. The FFT values are most relevant. This combined feature set is fed into a decision tree that is using several thresholds to decide if the swipe was performed by a finger or the whole hand. We are using the effect that swipes have fairly constant values in the frequency band between 2kHz and 8kHz. This classification is performed each 256 samples. In order for a swipe gesture to be identified a number of subsequent positive classifications have to occur. For example 10 classifications that indicate a constant movement of 26ms or more are identified as a swipe gesture.

4.3.6.3. Evaluation

In order to evaluate our system we have performed a combined study by 10 users who were invited to test the accuracy of the touch detection and benchmark the interaction speed. They predominantly had plenty of experience with touch devices (all questionnaire questions refer to Likert-scale 1 to 10, $\mu = 9.40, \sigma = 1.90$). Experience with gesture interaction systems like the Kinect or Leap Motion was less prevalent and had a higher variation ($\mu = 6.00, \sigma = 2.71$).

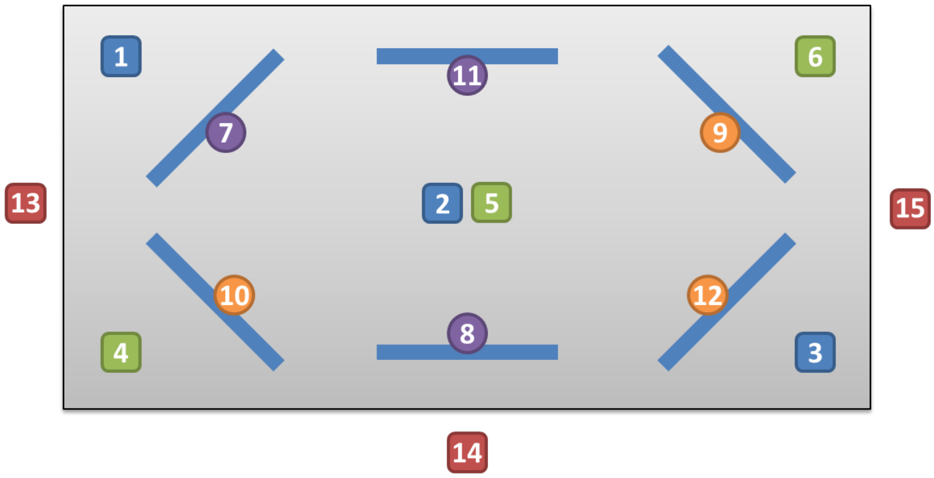


Figure 4.40.: Finger tap (blue), knuckle knock (green), finger swipe (purple), hand swipe (orange) and stomping (red) spots relative to tabletop.

Touch detection accuracy

One of the main interesting aspects for us is the accuracy of the touch detection with a classification that is only trained by a limited number of users. Six different types of touch events are tested by the different users - finger tap, double finger tap, knuckle knock, double knuckle knock, finger swipe and hand swipe. In addition we want to get an idea if outside influences can disturb the signal, thus we are letting the users stomp at three different locations in close proximity of the table. Overall we have 12 different areas on the table that have to be touched in different ways by the users. These are executed three times each, leading to 54 touch samples and 9 stomp samples. The locations shown in Figure 12 are (double) finger taps (1,2,3), (double) knuckle knocks (4,5,6), finger swipes (7,8,9), hand swipes (10,11,12) and stomps (13,14,15). The main purpose of this evaluation was to test a pre-trained algorithm that does not require any training efforts by the user. The subjects were shown all different supported touch gestures just once in a live example. The results are shown in Table 4.4. The system was very well capable of recognizing the different taps having a success rate of 96% or more. The results were not as good for knock detection, with only 81% correct classification of single knocks and 60% correct classification of double knocks. However, it should be noted that there was a high variance in results.

Table 4.4.: Results of touch detection for single and double taps (SFT, DFT), knocks (SKK, DKK), finger swipe (FS), hand swipe (HS) and stomp (STO). Noted are the overall samples, errors, no event errors, wrong classification errors and the percentage of correct classification.

	SFT	DFT	SKK	DKK	FS	HS	STO
Samples	90	90	90	90	90	90	90
Errors	1	3	17	36	18	2	22
No event	1	0	0	0	6	0	0
Wrong Class	0	3	17	36	12	2	22
Percentage correct class	98,89	96,67	81,11	60	80	97,78	75,56

Two users accounted for half the errors of double knock recognition as none of their double knocks were recognized. Thus with some additional training and adaptation it should be possible to detect all knocks. The classification of finger swipes showed similar results. The majority of errors (15 of 18) were produced by a small set of subjets (3 of 10), leading to a detection rate of only 80%. Again, training the different gestures should lead to an improvement. The results for hand swipe were very favorable with a detection rate of almost 98%. Stomps were able to disturb the system in about 25% of all events. However, in practical applications they are not highly relevant, as we can rule out any events where no hand is present above the table. Our tests showed that it is also very important to consider the uniformity of the table. The recognition rate of finger swipes at position 8 (93.33%) was considerable better than the recognition at positions 7 and 9 (73.33% for both), indicating that it is important to test each touch gesture at various positions and adjust the algorithm accordingly.

Hand tracking and interaction speed

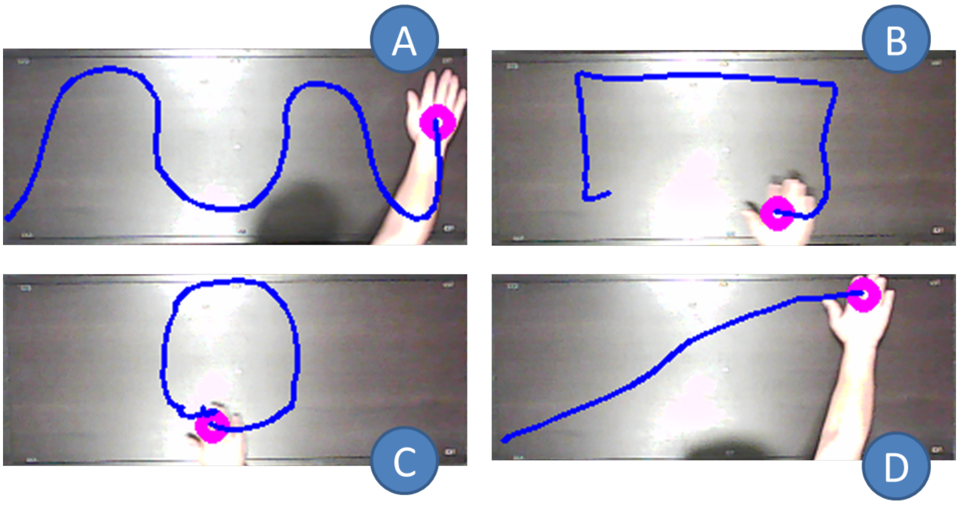


Figure 4.41.: Tracks generated by Kalman filtered palm position, a sine wave (A), a rectangle (B), a circle (C), and a diagonal swipe (D)

In order to properly identify gestures the recognized paths of the hands are the most important measure. We have added a visualization module to the registered camera image introduced previously that allows us to show the tracks followed by one or more hands. Figure 4.41 shows several of the paths that were generated this way using single hand tracking. The hand is in this case moving about 10-15 cm above the table surface. While we have not connected the system to a generic path-based gesture recognition system, we can see that the system can create smooth trajectories that can be analyzed further.

A major point of interest for us was to check if users could successfully use the layer interaction pattern introduced before and if the option for adding unobtrusive touch detection would have any influence on the interaction speed. For this we used a small game whereas the participants had to put a cursor into a box and perform either a dwell (approximately 300ms) or touch activity for selection. The cursor reflected the current interaction layer by color coding. We counted the time it took to complete a run of 15 boxes.

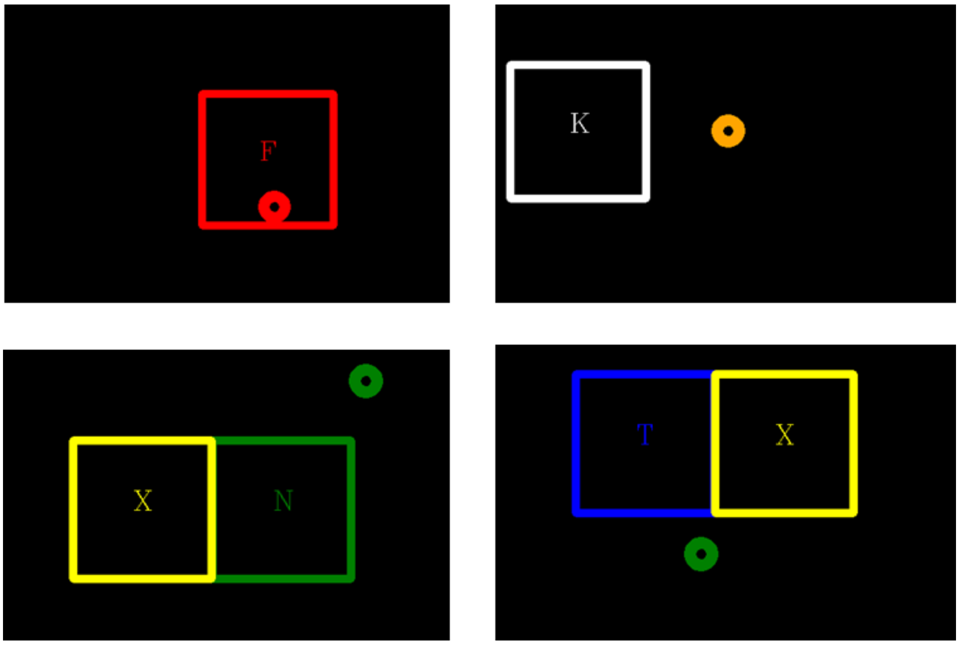


Figure 4.42.: Interaction speed evaluation. Different types of boxes for near layer (N), knock (K), far layer (F), and disturber (X).

Some example boxes are shown in Figure 4.42. There were six different types referring to dwelling in the three different layers, knock, tap and disturber. Each participant performed four runs. First a training run, then a run with all interaction layers and the two touch events, then a run with dwelling boxes only (without layers), and finally a run with tap boxes only. The dwell and tap runs also included some disturber boxes to slightly increase the challenge. The order of the runs were switched equally between the different participants. Regarding the full interaction run we wanted to know if the user understands the different layers and can achieve a good interaction speed. The main hypothesis we wanted to test was that tapping improves the interaction speed as opposed to dwelling and may reduce the overall interaction times, thus reducing potential fatigue. We expected the tap run to be shorter than the dwell run as selection events can be performed faster. All runs had the same overall distance and the last two had inverse order to account for Fitts' law.

Table 4.5.: Results for interaction time in the different runs of the interaction speed test

	Full run	**Dwell run**	**Tap run**
Average Time	40.12	37.97	33.42
Shortest Run	27.28	32.20	24.29
Longest Run	51.38	48.11	47.72
Standard Deviation	7.22	5.27	7.93

The results are shown in Table 4.5. Running a paired t-test on dwell and tap run the resulting p value is 0.0071, indicating a high statistical significance that the interaction using taps is faster than the interaction using dwelling. This fits expectations from literature [24]. While this can be countered by reducing the dwell time, the risk of wrongful selection of nearby objects increases significantly. Questionnaire Finally, we asked all participants to fill in a questionnaire with Likert questions in the style mentioned at the beginning of this section. Most users considered the device to be easy and precise enough to use ($\mu = 8.1, \sigma = 1.52$) and considered it highly intuitive ($\mu = 8.9, \sigma = 0.74$). Regarding the layer model they had no problem using it in the full test run ($\mu = 8.6, \sigma = 0.97$) but were critical to adding more layers ($\mu = 3.9, \sigma = 2.08$). They clearly preferred ($\mu = 2.6, \sigma = 2.72$) finger taps (Likert score 1) to knocks (Likert score 10). There was no clear preference for finger swipes (score 1) or hand swipes (score 10) ($\mu = 4.6, \sigma = 3.06$). The participants considered CapTap to be an interesting interaction device ($\mu = 9.1, \sigma = 1.20$) and could even imagine using it for longer periods ($\mu = 6.9, \sigma = 1.97$). Asking for particular points that they liked about the current version of CapTap the comments mentioned the ease of use and the high variety of different input commands that are supported. Points that were disliked are the usage of knocks that were considered unpleasant after a short while, even during the 10 minute study that only included few knocks. The hand tracking at this point is sometimes disturbed by the user's knees that enter the generated electric field around the table.

4.4. Other capacitive prototypes

In collaboration with other partners from industry and different students some additional prototypes have been created that are discussed briefly in this section.

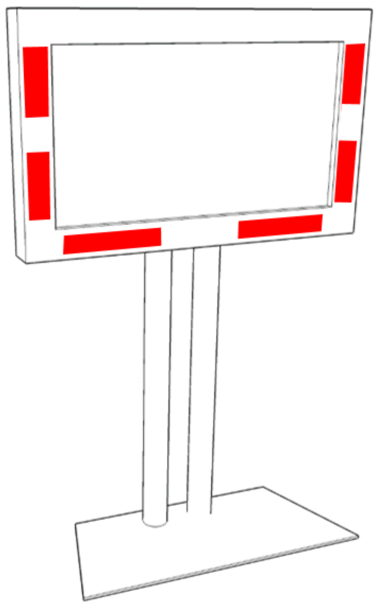
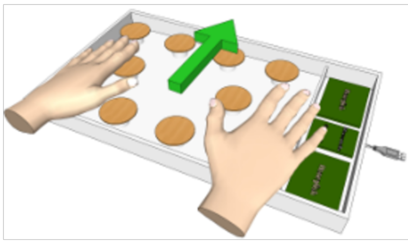


Figure 4.43.: CapDisp sketch - TV on a stand equipped with capacitive sensors hidden below a plastic cover

CapDisp is a presentation display augmented with capacitive sensors to enable touch-free control of several applications. It was created as a prototype for Hessen IT GmbH in 2010. The system is comprised of six Cypress CY3271 capacitive sensors that are powered via USB and interfaced to a Mini-PC that is attached behind a 42" display on a presentation stand, as shown in a sketch in Figure 4.43. The display is set in an additional case that hides the six electrodes that are made of copper foil. As shown in Figure , they are placed on the bottom and right part of the screen, allowing to detect four different swiping gestures, left and right on the bottom of the screen, up and down on the left side of the screen. The gestures can be performed at a distance of up to 20cm in front of the electrodes. Since the primary purpose of this device is showing presentations, some additional dwell gestures were added, that allow jumping to the first or to the last slide by holding the hand in front of a specific sensor for a certain time. Additionally, two other applications were included, a gesture-controlled image viewer and a video player.

Figure 4.44.: *Top left*Person lying on the couch. *Top right*Resulting sensor value visualization. *Bottom left*Rendering of recongized posture. *Bottom right*Position of electrodes within the couch.

The smart couch was created by then-student Tobias Grosse-Puppendahl in scope of a practical course in 2010/11, supervised by Alexander Marinc and me. The results were later published at the AmI 2011 conference [GPMB11]. Using an array of eight capacitive proximity sensors that are unobtrusively placed inside a couch it is possible to determine the posture of one or more persons that are currently occupying the system. The sensor readings are calibrated and normalized and fed into the WEKA machine learning framework for classification. Three different classifiers have been tested, decision trees, Naïve Bayes and RBF networks, whereas the latter provided the best results. The system was evaluated with 18 users and 8 resulting postures (6 with one person, 2 with two persons), such as sitting left or right, or lying in a specific direction. The resulting measurements was distinguished in a training set from 9 persons and a test set from 9 persons. The resulting precision was 97.5%, the recall 97.2%. The system is still working as a demonstrator in our living lab, implicitly controlling different networked systems based on the detected postures, e.g. activating ambient lighting as soon as the person is lying down.



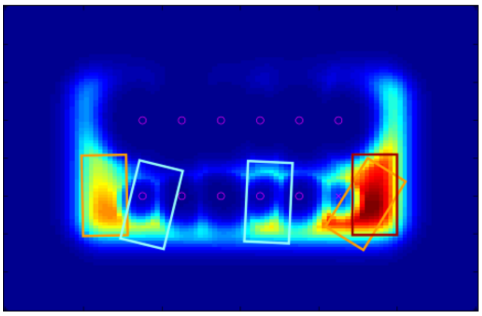


Figure 4.45.: *Top left* Rendering of Honeyfish device and hands. *Top right* Image of developed GUI and pointer. *Bottom left* Swiss cheese algorithm predicting objects in interaction space. *Bottom right* Evaluation of Honeyfish at a student fair.

Honeyfish is a gesture interaction device based on capacitive proximity sensors operated in shunt mode. It was created by Tobias Grosse-Puppendahl in scope of his Master's thesis that I supervised in 2012. It led to two different publications focusing on the provided contributions in multiplexing and object detection [GPB12] [GPBKK13]. The system is using eight different transmitters and two receivers, leading to a set of 16 virtual sensors that are set in the middle of each receiver-transmitter combination. As shown in Figure 4.45 on the top left, the receivers are in the center and the transmitters are placed on the outside. Using a frequency division multiplex it is possible to generate 50 samples from each virtual sensor. The system is using a method of object tracking that extends on a proposal by Smith, dubbed Swiss-cheese, which is based on the premise that each sensor not recognizing an object or a distant object is cutting a (ellipsoid) hole in the object presence probability space, thus leading to a probability distribution visually resembling a Swiss cheese [Smi96]. The remaining probability space can be analyzed to fit hand shaped objects, thus enabling gesture detection. A particle filter is used to track the position of the hands. The system is able to track multiple hands and was used to control applications ranging from an image viewer, the GNU TuxRacer game to a remote-controlled vehicle.

Figure 4.46.: CapFloor sketch - grid layout of electrodes is placed below a floor layer with sensors attached on the sides

GestDisp is the final prototype discussed in this work. It was created by Yannick Berghöfer in 2013 as part of his Master's thesis that was supervised by me and Tobias Grosse-Puppendahl. The basic idea is to enable gestures performed in front of a screen by applying capacitive proximity sensors on the screen surface. Consequently a different type of electrode material has to be used, that combines conductivity and transparency. Different materials have been evaluated, including ITO (indium-tin oxide) and PEDOT:PSS (a conductive polymer) in order to find a suitable electrode candidate. ITO was chosen and attached to the screen including shielding electrodes, reducing the effect of the electric components used to create this display. Nonetheless, the complex and highly disturbing environment drastically limits the distance in which gestures may be performed. The gestures are recognized using a Hidden-Markov-Model classifier that was trained by the developers. Using this it was possible to distinguish swipe gestures in four directions, selection gestures as indicated by dwelling at a certain position and combined swipe-and-hold gestures, whereas the hand is resting in the interaction area after performing a swipe, e.g. used for continuous scrolling or increasing the volume. The gesture recognition includes a specific garbage-gesture that allows to distinguish arbitrary movements in the interaction area from deliberate gestures and has to be trained separately. This approach allows a precision and recall of about 95% on a collected training set. A demonstration application was created that mimics a multimedia system in a car, allowing to control different radio stations or select music of choice.

4.5. Capacitive prototypes from related work

Table 4.6.: Measuring layout and data processing of different prototypes from related works

Name	Description	Application Areas	Measuring Layout	Data Processing
SensFloor [LS09]	System for indoor localization and fall detection as floor underlay	Indoor Localization	Loading mode, variable number of sensors based on area size	Individual coding of zones on floor - analysis of activity based on trajectories
TileTrack [VMV09]	Indoor localization using transmitters below floor and receiver electrodes in wall or furniture	Indoor Localization	Transmit mode, large transmitter electrodes below floor, different receiving electrodes	Location by calculating center-of-gravity on most active tiles
Touché [SPH12]	Swept-frequency sensing to detect different types of touches on a conductive material	Smart Appliances	Swept-frequency sensing, single electrode	SVM classification using features in different frequency ranges
Botanicus Interactus [PSLS12]	Using plant tissue as conductive material as application for swept-frequency sensing	Smart Appliances	Swept-frequency sensing, electrode coupled to plant tissue	SVM classification of touches that are transferred to input events
Active capacitive sensing [CAL10]	Conductive textile electrodes to sense different parameters of the human body, based on location	Physiological Sensing	Loading mode, single electrode attached to body part	Different filtering methods, based on electrode position, activity classification using LDA
Spread spectrum sensor [Mac04]	Single electrodes using a spread spectrum technique for improved sensitivity	Physiological Sensing	Loading mode, single electrode placed remotely	Spread spectrum technique to improve SNR, amplitude measurement for respiratory rate
School of Fish [?]	Array of shunt mode sensors that can track 3D position and orientation of two hands	Gesture Interaction	Shunt mode, flexible array of sensors	Modeling hands as collection of spheres and fit into area based on sensor values and position
Thracker [WHKS06]	Four electrodes placed around display that can sense spatial position of hand in front of display and certain gestures	Gesture Interaction	Loading mode, four electrodes placed spatially around display	Position based on distance to electrodes or gesture based on nearest object to electrode
GestIC [Mic13]	Shunt mode array enabling near distance gesture interaction above sensing area	Gesture Interaction	Shunt mode, four or five receiver electrodes	Positioning based on single proximity values and HMM-based gesture recognition

While I already have presented capacitive systems in the related work section, they are briefly revisited here, to classify them given the specified application domains. Similar to the created prototypes I will give some additional detail regarding their measuring layout and data processing. The prototypes are shortly listed in Table 4.6.

4.5.1. Indoor localization

One example system based on capacitive sensing is the previously presented TileTrack that uses a combination of transmit mode and center-of-gravity calculation between different floor tiles to calculate the position of multiple persons [VMV09]. A second, already commercialized system is SensFloor that uses an integrated solution of capacitive sensors and wireless communication hidden below a floor covering that is able to detect the position of several users and other parameters such as falls, based on analyzing activity above single sensor areas or the movement trajectories over time [LS09].

4.5.2. Smart Appliances

Sato et al. have presented Touché, a swept-frequency capacitive sensor that allows distinguishing different types of touches on any suitable surface and medium [SPH12]. Some examples include recognizing different hand postures in liquids and touching different body parts to control mobile devices. Their system is based on analyzing a broader range of frequencies that have a different effect on the resulting capacitance. Using a classification method they are able to distinguish different categories of events. An-other example of this technology is touching different parts of a plant to control an interactive art installation [PSLS12]. Capacitive sensing provides the ability to add interactive features to many different appliances and allows for unobtrusive placement.

4.5.3. Physiological sensing

Capacitive proximity sensors can be used to measure various physiologi-cal parameters that are related to movement of dif-ferent body parts, including internal organs, most notably the heart. Cheng et al. have presented a system that allows measuring motions and shape changes of body parts using capacitive sensors em-bedded in garment [CAL10]. They were able to detect swallowing and breathing rate. One example for an industrial application is non-contact electrocardiogram (ECG) sensing in cars, intended to detect drowsiness in drivers. MacLachlan presented a system that detects the respiratory rate of a person lying on a bed from a distance of up to 50cm using a single electrode and a highly sensitive sensing method based on spread spectrum methods that are commonly used in wire-less communication [Mac04]. Generally, capacitive proximity sensing is a powerful technology that is able to gather physio-logical information over a distance, while being unobtrusively integrated into various appliances. In applications that require this information, e.g. to detect the state of alertness or fitness of a user it is a viable alternative to body-worn sensors that are more intrusive by nature.

4.5.4. Gestural interaction

Throughout the years there have been various at-tempts to enable the tracking of gestures in free air. Capacitive proximity sensors have been first presented almost 100 years ago by the Russian physicist Leon Theremin, who invented the eponymous touch-free electronic instrument [60] The theremin uses two electrodes to control pitch and volume of a generated sine wave. Capacitive hand tracking has been a research interest at MIT in the 1990s [**?**] and has been investigated recently by other groups, enabling touch control even through thick-er non-conductive materials [32], [61]. We introduced Thracker in section 2.6.2, that allows to either detect moving hand ges-tures or detect grasp actions in front of a screen [WHKS06].

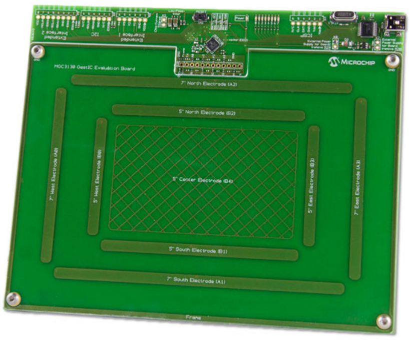


Figure 4.47.: GestIC® Sabrewing prototyping board [Mic13]

GestIC® by Microchip Technologies Inc. is a controller for electric near field 3D tracking based on capacitive proximity sensing [Mic13]. A prototyp-ing board is shown in Figure 4.47. Using a set of several electrodes and on-board processing it is capable of tracking the 3D position of a hand and provides gesture recognition at a distance between 0 and 15 cm.

5. Evaluating capacitive proximity sensors in smart environments

In the previous sections we have presented back-ground information on capacitive proximity sensors and various prototypes of this technology in different application domains within smart environments. In the following section we are building on the collected information to perform a meta-analysis of the acquired data, discussing benefits and limitations of the technology, compare it to competing technologies and give some guidelines to parties interested in developing further applications in this domain.

5.1. Comparison to other sensing technologies

Table 5.1.: Comparison between capacitive proximity sensors and other senor technologies

Name	Application Domains	Environmental Influences	Detection Range	Processing Complexity	Unobtrusiveness
Capacitive proximity sensing	indoor localization, smart appliances, physiological sensing, gestural interaction	electric fields, conductive objects	near distance (< 100cm)	Few high dynamic range data sources	invisible integration possible
Capacitive touch sensing	smart appliances, physiological sensing, gestural interaction	electric fields, conductive objects	touch	Few binary sensors	thin cover above electrodes
RGB cameras	indoor localization, smart appliances, physiological sensing, gestural interaction	occlusion, external lights	far distance (> 10m)	Complex image processing based on resolution	pinhole lenses
Infrared cameras	indoor localization, physiological sensing, gestural interaction	occlusion, external infrared light	medium distance (< 5m)	Complex image processing based on resolution	infrared source and camera
Ultrasound sensing	indoor localization, smart appliances, gestural interaction	acoustic occlusion, absorbing materials	medium distance (< 5m)	Few low dynamic range data sources	emitter and senders with exposed pinhole speaker, microphone
Microphone arrays	indoor localization, smart appliances, physiological sensing	environmental noise, absorbing materials	medium distance (< 5m)	Very high dynamic range data sources	exposed pinhole microphones
Radiofrequency sensing	indoor localization, smart appliances, gestural interaction	other RF devices	far distance (> 10m)	Few low dynamic range data sources	hidden emitters and senders possible

In order to properly place capacitive proximity sensing in the smart environment domain it is neces-sary to include a comparison to other sensing tech-nologies. We have chosen systems that have a broad applicability and have been used in various smart environment applications. A short overview can be found in Table 7. We have included a comparison of application domains, environmental influences, de-tection range, processing complexity and unobtru-siveness of the technology. Capacitive touch sensing, as opposed to capacitive proximity sensing relies on an electrode being touched instead of an object being in proximity and is ubiquitous in touch screen applications. RGB cameras are a class of image sensors operating in the same frequency domain as the human eye. They are capable of processing different colors. Infrared cameras operate in near light frequencies that are invisible to the human eye. This allows for application in dark environments and we can project infrared light into the scene without disturbing the user. Ultrasound sensing is using a low frequency range just above the audible limit of human hearing. The waves propagate similar to sound signals and we can perform reflection measurements or time-of-flight methods. Microphone arrays detect signals in the range of human hearing, and thus work with audible signals, such as human speech. Radiofrequency (RF) sensing uses signals in a range between several hundred kHz up to 5GHz, typically used for wireless communication. Commonly the signal strength or time of flight is used to gather information about the environment. Most technologies are capable of supporting multiple application domains. Some non-intuitive examples include WiSee that enables whole-body gestural interaction using WiFi signals [86] or MoGees that uses a single microphone to enable gesture interfaces on various surfaces [87]. Capacitive sensors are disturbed by conductive objects and electric fields, whereas cameras struggle with occlusion and additional light sources. Occlusion is a weak point, and a line of sight is required. Sound sensors are prone to dampening materials and environmental noise interfering with the signal. RF signals usually propagate well through most materials and only external sources may be an issue. The detection range of the technologies varies strongly. RF ranges before light, sound and electric fields. However, this again strongly depends on apxplication and layout of the sensing devices. It is not easy to find a good measure about the processing complexity associated to a different sensing technology. We are using a simplified model, taking the dynamic range of a sensor and the number of sensors typically required. Dynamic range is the difference between the smallest detectable value and the largest detectable value. Microphones have a high dynamic range measuring over a larger frequency scale, whereas touch sensors only have two different states. Finally capacitive sensors and RF sensors can be applied completely invisible. Cameras, microphones and ultrasound need a direct connection to the out-side world. However, there are very small variants available that are barely visible to the naked eye.

5.2. Limitations

Table 5.2.: Overview of capacitive proximity sensing limitations

Name	Examples
Environmental influence	Static electric fields, dynamic electric fields, temperature, humidity, conductive objects
Physical range	Small differences in capacitance, reduction due to influences, physical limitations
Object detection	Small number of data points, a priori knowledge

Despite the potential that has been described in the previous sections there are various limitations of capacitive proximity sensing that we can put into the different groups of environmental influence, physical range and object detection that will be described in more detail in the following section. A short overview is given in Table 5.2.

5.2.1. Environmental Influence

One of the main limitations of capacitive proximity sensors is their sensitivity towards environmental influences. Any factor that modifies an electric field will also affect the measurement of a capacitive sensor. The current environmental parameters, like temperature and humidity are having a considerable effect on the atmosphere in which the electric field propagates. However, those changes are usually over a longer period of time and can be compensated using a factor for drift, as described in the previous sections about noise reduction. A more challenging factor is the other electric devices in the environment that emit stronger electromagnetic fields. While persistent sources, such as permanent electric installations can usually be countered using a galvanic isolation there are other non-obvious challenges. E.g. we noticed that certain plasma TVs are able to disturb the measurement and increase noise levels consider-ably. This change is even varying according to screen content. A minor effect is the presence of high-frequency fields that are getting more prevalent in modern IT equipped environments. Instead of the 2.4GHz and 5GHz ranges that are often used in wire-less communication, capacitive proximity sensors can operate in the range of a few kHz to one MHz. An additional issue might arise when placing sensors close to each other. The created electric fields may disturb the measurement if some electrodes are charged and create fields to adjacent electrodes while they are discharged for measurement. Consequently, specific charge-discharge cycles or multiplexing methods have to be used to counter this effect. A major challenge is dealing with conductive objects that are permanently placed in the immediate sensing environment. It is difficult to distinguish the object we want to detect from a disturbing object, if their influence on the electric field is similar. Long term data analysis may help in performing a successful detection. The CapFloor prototype is affected by environmental influences the most, given the small size of the electrodes relative to the interaction area and the changing environment on top of the floor. We are using a strong noise reduction algorithm and drift compensation to create a more stable result while reducing the detection range.

5.2.2. Physical Range

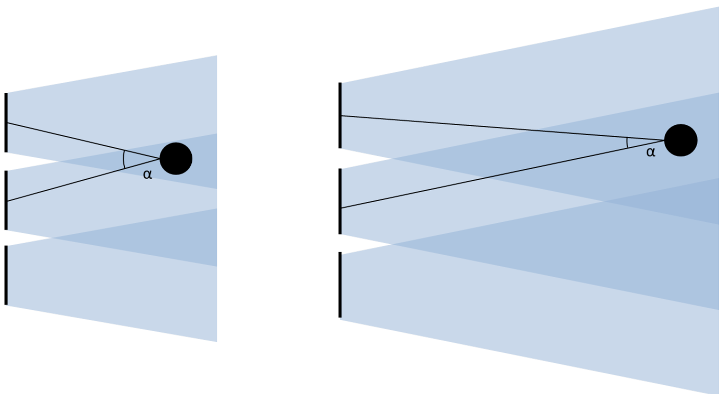


Figure 5.1.: Reduced angular resolution on smaller, distant objects

The physical range of the generated electric field is one of the main limiting factors of capacitive proximity sensing. In order to detect objects that are further away we have to increase the electric field strength sufficiently. This is easier the larger the electrode is, as its potential capacitance is higher. However, this also leads to distant objects having an ever smaller influence on the overall capacitance, and we need more precise measurement circuits and longer measurement times to improve the signal-to-noise ratio. Additionally, looking at smaller objects the angular resolution will decrease as shown in Figure 5.1. This makes it more difficult to get a precise localization as the immanent noise leads to an angular error. While this can be compensated using more sensors, the far distance would require us to use large electrodes that have to be placed further apart resulting in a huge area that would have to be equipped with sensor electrodes. In general the achievable resolution is not comparable to vision based system and has to be taken into consideration when designing the specific application. A balance between electrode size, physical range and achievable resolution has to be found. The MagicBox size does not allow an integration of very large electrodes. Instead we are optimizing the available space in order to achieve a detection that lets us detect hands in a distance between 15 and 20 centimeters.

5.2.3. Object Detection

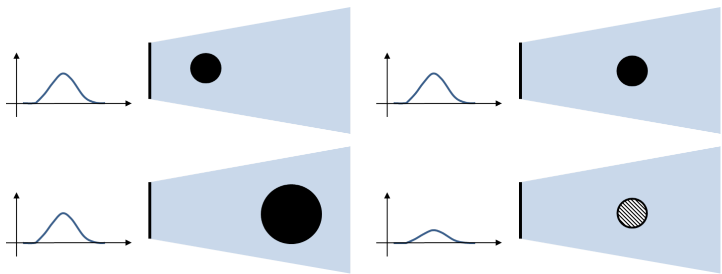


Figure 5.2.: Same response to differently sized objects (left), different response to varying materials (right)

Object detection using capacitive sensors can be partially compared to object detection using camera systems, with a single sensor being equivalent to a single photo sensor. The light intensity measure is comparable to field intensity and likewise we can't distinguish if the measurement is caused by a weak source in close proximity or a strong source at a further distance. As a practical example the capacitive sensor can't decide if one hand is close to the sensor or two hands are a bit further away. This effect makes it challenging to provide object detection and we usually have to combine the information from various sensors to get a good idea about object shape and size. Due to the presented challenges in physical range and electrode size, capacitive proximity sensing systems do not have the same level of scalability as opposed to cameras, where millions of photo sensors can be placed in very small areas. Additionally, the effect of an object on the electric field is not always closely correlated to the object dimensions, but instead based on conductivity, material and other factors. We may get the same response to different objects at different distances or get a varying on similarly sized objects made of different materials, as shown in Figure 5.2. The Active Armrest has gestures for one and two fingers that are distinguished using a simple threshold. If another object is entering the field or the person has larger fingers the system will fail to properly differentiate gestures. Accordingly some other compensation methods should be used.

Table 5.3.: Overview of capacitive proximity sensing benefits

Name	Examples
Versatility	Flexible electrode design, scalability, different sensing methods
Unobtrusiveness	Invisible application, non-disturbing frequency range
Processing Complexity	Small number of sensors, variable dynamic range

5.3. Benefits

After discussing the various limitations of capacitive proximity sensing, the following section will give an overview of the benefits. Similar to the previous section we have three groups, namely versatility, unobtrusiveness and processing complexity. Some examples within these groups are shown in Table 6.

5.3.1. Versatility

A main benefit of capacitive proximity sensing is the versatility in which they can be applied. The flexibility of electrode materials, size and geometry allows specifically creating highly individual applications. Example electrodes include transparent metal oxide layers, woven conductive thread, copper wires, PCB boards or simple aluminum foil. The sensors systems are also highly scalable. By choosing appropriate voltages and frequencies it is possible to add a high number of sensors to a single object. Using smart measurement windows and different multiplexing methods, sensors can be placed close together and electrodes may act as both sender and receiver. The different sensing methods presented - loading mode, shunt mode and transmit mode enable a variety of different sensing patterns. The human body can be used as both sender and receiver and smart electrode layouts allow using a smaller number of processing units. In conclusion, it is possible to add capacitive sensing to most everyday objects to enable different forms of interaction, create natural interfaces and smart objects. Our prototypes are using different electrode materials, flexible or solid electrodes, conductive thread, wires, shielded or non-shielded layouts.

5.3.2. Unobtrusiveness

Electric fields are not usually perceived by persons, unless they are of exceptional strength. Furthermore they propagate through many materials that we are typically using in our environment, including most plastics, wood or tiles. This allows us to invisibly apply capacitive proximity sensors without a strong effect on the measurement. Application below several centimeters of covering is possible, if the electrodes are designed properly for sensing in this distance. The frequency range in which the sensors are operating is usually not in an interval that disturbs other electronic systems. Thus it is feasible to use capacitive sensing even in environments, where non-disturbance is a main requirement. Additionally the used frequencies are not considered to be biologically active, and good results can be achieved using small currents. It is possible to equip most conductive objects directly with capacitive proximity sensors and hide them below non-conductive objects with minimal spatial requirements. Our Smart Bed and Active Armrest prototypes are using sensor sets that are completely invisible

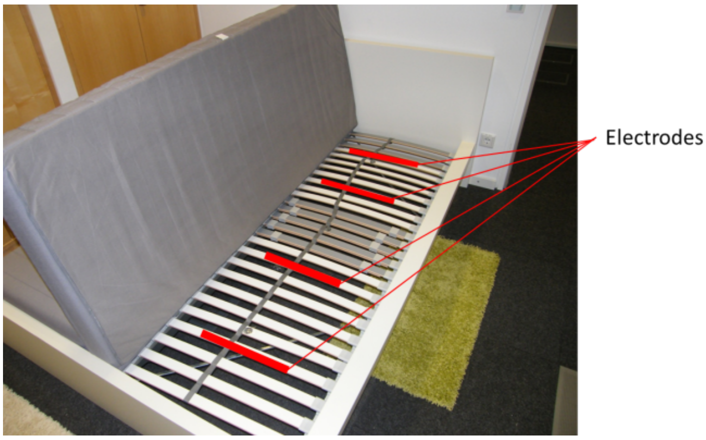


Figure 5.3.: Electrodes and sensors hidden below mattress of Smart Bed

from the outside and communicate wirelessly to a PC only using a power supply. Figure A.1 shows the electrodes and sensors hidden below the mattress of the Smart Bed.

5.3.3. Processing Complexity

An appropriate analogy of capacitive proximity sensors is a single photodiode. As opposed to a light intensity we are measuring capacitance. While the information we can gain from such a measurement is limited, the processing required to analyze the signal is also low. Performing signal analysis on an array of 16 capacitive sensors is comparable to processing the image of a 4x4 pixel camera. Therefor it is easy to create highly integrated systems with very low-power devices for performing any subsequent data analysis. While it is possible and in many cases beneficial to use complex data processing algorithms for object detection it is in most cases still possible to replace them with simpler methods for a comparable result. In many applications it is even viable to opt for a quantized capacitance measurement. In the case of a touch sensor a single binary measure is sufficient. However, it is also possible to select various different levels and reduce the dynamic range to an easily computable value that is 4 or 8 Bit long. Depending on the chosen algorithm this dynamic range reduction can occur either in pre-processing or high level processing. With the exception of the Capacitive Chair our prototypes are using simple data processing methods that can be easily applied on embedded systems. A preferred method for object localization is the weighted average algorithm. Regarding model-based data processing, even very simple cylindrical models, such as the one used for the Smart Bed, are capable to reliably predict numerous postures that are relevant in real world applications. In general, the low requirements for data preprocessing, allows dedicating more resources to high level data processing algorithms if the specific application is resource con-strained. The OpenCapSense toolkit that is the base for most of our prototypes has a fairly powerful micro-controller that is able to implement all of the processing steps - thus enabling highly integrated, low-power capacitive proximity sensing prototypes that can be used in smart environment applications.

5.4. Guidelines

After discussing the limitations and benefits of capacitive proximity sensors, the final section of this chapter will give some general guidelines on their application. The first step of this process is a decision if capacitive sensors technology is suitable for the given application. This part should be driven by three questions. What do I need to measure in my application scenario? Capacitive proximity sensors can measure the presence and properties of conductive, grounded objects. This includes the various application scenarios shown in the previous sections. However, if the application requires measuring properties of unsupported objects that are non-conductive, a different technology should be chosen.

What sensing technologies are supporting the required measurements?

It may be the case that multiple technologies sup-port the measurements required in this specific applications. Cameras often can provide similar recognition as capacitive sensors, e.g. in indoor localization applications. In this step all potential sensing technologies should be collected. Are capacitive proximity sensors beneficial for my scenario? An evaluation of the different candidates is the final step and should lead to a decision about the most suitable sensing technologies. If the distance is too high for capacitive proximity sensors or enough processing power is available and lighting conditions are static, cameras might be more suitable. This should be driven by the different benefits and limitations of the technologies. If there is a decision in favor of capacitive sensors the next step is to design the specific electrode layout. Similar to technology selection we can use a few basic questions to get an idea of what layout to use.

How many sensors are required to get the measurement?

The number of sensors required is depending on the area we want to cover, the specific object parameters that have to be determined and the desired resolution. The electrodes are inherently limited in size, as a single sensor can only charge and discharge to a specific maximum capacity. Therefore, if a large area has to be covered more electrodes and sensors are necessary. If we just want to measure the presence of a hand a single electrode may suffice. If orientation and position are interesting we need to combine measurements from various sensors.

What should be the size and geometry of the electrodes?

This is closely related to the previous question. If the application is not restricting the available space, the electrode should be approximately of the same size as the object that is to be detected. This generates the highest difference in capacitance when the distance is changing.

What is the best electrode material to use?

Copper is always a good first choice to create electrodes. If elasticity is necessary we can use copper foil and solid copper if that is of no concern. For transparent electrodes we will have to use one of the previously presented materials, such as ITO. If electrodes have to be integrated into cloth, conductive thread is a good candidate. Any conductive material will act as an electrode, thus the application and budget should be the primary driver of this decision.

Does my application require any shielding?

Shielding allows detecting only objects approaching from a certain direction. If the application re-quires this additional hardware, because it is anticipated that other objects might disturb the measurement, shielding should be used. Finally, if the hardware is designed as desired the different variations of data processing have to be chosen and configured according to the application. Using baseline calibration is beneficial in the vast majority of applications. Having a distinct starting point simplifies all further steps of high-level data processing, such as normalization and setting different thresholds. This step may only be omitted in very stable environments and if the system has sufficient a priori information to operate on raw data. Drift compensation should be handled in a similar fashion. The common methods are not computationally expensive and having a stable baseline over time allows the same algorithms to be applied in a more robust fashion. The method and configuration of noise reduction are strongly depending on the specific case. Some form of noise reduction might be required in most applications. Yet, according to the type of noise different methods can be used. If outliers are an issue a median filter is appropriate; if a smoother signal is desired an average filter can be used. Regarding high-level data processing there are manifold variations of methods. Data-driven machine learning algorithms are a good method if we have a small set of potential outcomes of our applications, e.g. the different postures that could be recognized on a chair or couch. If our application has many different potential outcomes, e.g. the thousands of potential locations in a hand tracking system, it is typically beneficial to use a model-driven approach. However, these models may be supported by data-driven algorithms, such as particle filters. One example is the Swiss-Cheese object tracker by Grosse-Puppendahl et al. [GPBKK13]. The data processing examples shown in the previous sections give an idea of the decision rationale in various application domains. We can say in conclusion that capacitive proximity sensors are a viable, or even, ideal solution for a considerable number of different applications. However, a certain level of preparation is required in the design process to create a system that benefits from the technology.

6. Conclusions and Future Work

This chapter summarizes what a great job you did and what could be done if you could do as a second PHD thesis :)

Acknowledgments

While many consider writing a PhD to be a mostly personal endeavor there are always various sources of discourse, collaboration, support and inspiration. So in no particular order there are various persons or groups of persons that deserve credit:

A. Evaluation results

This section will give an overview of the raw results gathered by the evaluations performed on the different prototypes.

A.1. Capacitive chair posture recognition - manual classifier

This section provides raw data of the first posture recognition test performed using the capacitive chair.

A.1.1. Evaluation setup

Short study with 10 participants. Three poses and a non-pose:

- Sitting upright
- Sitting hunched
- Slouching on chair
- Close to chair - disturber

The persons were given a short introduction, the different postures were displayed, and finally the persons were asked to perform the postures in order. When testing “close-to-chair” the subjects were asked to rattle at the chair, stand close, move it around and thus disturb the potential sensor readings. Each class was tested for 10 seconds, collecting 200 samples.

A.1.2. Raw results

Table A.1.: Percentage of correctly classified postures using manually set classifier

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Avg
Upright	100	100	100	100	100	100	100	100	100	100	100
Hunched	100	100	100	100	86	100	100	100	100	100	98,6
Slouch	100	100	100	100	100	100	100	100	55	100	95,5
Disturber	100	100	100	100	100	100	100	100	100	100	100

A.1.3. Postures

Figure A.1.: *Top left* upright posture. *Top right* hunched posture. *Bottom left* slouched posture. *Bottom right* disturber posture

A.2. Capacitive chair posture recognition - SVM classifier

A.3. Capacitive Chair - working situation recognition

A.4. CapTap evaluation

A.4.1. Questionnaire

- G1 Experience with touch screen systems (none = 1, daily usage = 10)

- G2 Experience with gesture recognition systems (none = 1, daily usage = 10)
- Q1 Do you agree that the required tasks were easy and precise to perform? (Not agree = 1, Strong agree = 10)
- Q2 Was the CapTap to be intuitive in its usage? (Not agree = 1, Strong agree = 10)
- Q3 I could control the different layers in the painter application? (Not agree = 1, Strong agree = 10)
- Q4 I could control the different interactions in the test run? (Not agree = 1, Strong agree = 10)
- Q5 Do you prefer finger swipes or hand swipes? (Finger swipe = 1, Hand swipe = 10)
- Q6Do you prefer finger taps or knuckle knocks? (Finger taps = 1, Knuckle knocks = 10)
- Q7 The CapTap should support more than three interaction layers? (Not agree = 1, Strong agree = 10)
- Q8 Do you agree that CapTap is an interesting form of interaction device? (Not agree = 1, Strong agree = 10)
- Q9 Would you consider using an input device like CapTap for a longer period of time? (Not at all = 1, I would like to use it = 10)
- Q10 What did you particularly like about the CapTap?
- Q11 What did you particularly dislike about the CapTap?

A.4.2. Raw results

The table denotes the results of the different touch points as explained in the descriptive section. The T's refer to the times of the three different interaction speed runs.

Table A.2.: CapTap evaluation raw results

Subject	1	1D	2	2D	3	3D	4	4D	5	5D	6	6D	7	8	9	10	11	12	13	14	15	T1	T2	T3
S1	3	3	3	3	3	3	1	3	2	0	1	2	3	3	3	3	3	3	3	3	3	45,38	41,57	33,94
S2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	43,12	41,83	36,94
S3	3	2	3	3	3	3	3	2	2	2	2	2	2	3	0	3	3	3	1	3	3	51,38	34,42	31,33
S4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	27,28	34,17	24,29
S5	3	2	2	3	3	3	3	1	3	3	3	1	2	3	3	3	3	3	2	3	2	34,66	48,11	45,35
S6	3	3	3	3	3	3	3	3	3	2	2	1	3	3	3	3	3	3	1	3	1	39,28	42,62	47,72
S7	3	3	3	3	3	3	3	3	3	2	3	3	0	1	3	1	3	3	1	3	1	40,87	33,56	25,95
S8	3	3	3	3	3	3	2	0	1	1	3	0	3	3	3	3	3	3	3	3	1	47,14	37,43	32,72
S9	3	3	3	3	3	2	2	3	1	0	2	2	0	3	0	3	3	3	1	3	2	39,71	33,78	26,63
S10	3	3	3	3	3	3	2	0	3	0	2	0	3	3	1	3	3	3	2	1	1	32,35	32,2	29,31
Ergebnis	30	28	29	30	30	29	25	21	24	16	24	17	22	28	22	28	30	30	20	28	20	40,117	37,969	33,418

Table A.3.: CapTap questionnaire results

Subject	G1	G2	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
S1	10	5	6	9	9	8	8	1	6	10	9	unsichtbare Sensorik im vertrauten Möbelstück, Gesten über Tisch	Präzision nicht gut genug
S2	10	9	6	8	6	8	7	3	3	7	6	Nice idea of embedding interaction into everyday furniture	area is too large, interaction is exhausting, demonstrator table not suitable
S3	10	4	10	10	5	10	10	1	1	10	7	tactility, intuitive and invisible	some areas hard to reach when sitting in front
S4	10	8	8	9	8	9	2	1	4	10	8		some errors in interaction
S5	10	3	10	8	8	8	2	1	6	9	8	touchpad like control on table	knocking, interaction disturbed by knee
S6	10	8	8	9	7	8	3	3	3	9	8	unobtrusive integration in furniture, intuitive interaction	delay in recognition
S7	10	10	10	10	5	10	3	2	5	10	8	Eingaben mit und ohne Kontakt, Auswahl über Tap intuitiv, Schnelle Eingewöhnung in Layer mit Farben	Beine werden erkannt, große Interaktionsarea, Verzerrungen im Zeichenprogramm
S8	10	3	7	8	7	9	2	2	1	7	4	ease of use, low learning curve, big variety of input modalities	interaction less precise in some areas
S9	10	3	8	9	4	7	7	10	3	9	3	knockin on heaven's door	tap didn't work - required more than one tap to recognize
S10	4	7	8	9	7	9	2	2	7	10	8	different interaction possibilities	detection of knees, difficulty to stay in layer
Ergebnis	9,4	6	8,1	8,9	6,6	8,6	4,6	2,6	3,9	9,1	6,9		

A.5. Active Armrest evaluation results

A.6. CapFloor @ EvAAL 2012

A.7. Smart Bed sleep phase recognition results

B. Publications

The thesis is partially based on the following publications:

2013 **OpenCapSense: A Rapid Prototyping Toolkit for Pervasive Interaction Using Capacitive Sensing** (Tobias Grosse-Puppendahl, Yannick Berghoefer, Andreas Braun, Raphael Wimmer, Arjan Kuijper), *In IEEE International Conference on Pervasive Computing and Communications (PerCom)*, volume 18, pp. 22, 2013.

2012 **V2me: Evaluating the first steps in mobile friendship coaching** (Salla Muuraiskangas, Anja K Leist, Andreas Braun, Kerstin Klauß, Peter H M P Roelofsma, Reiner Wichert, Peter Klein, Dieter Ferring), *In Journal of Ambient Intelligence and Smart Environments*, IOS Press, volume 4, pp. 517-534, 2012.

2011 **User requirements in ICT-based social media use: Acceptance of a virtual coach** (Anja Leist, Dieter Ferring, Kerstin Klauss, Peter Klein, Andreas Braun, Reiner Wichert), *In* , 2011.

2014 **V2me - Virtual Coaching for Seniors** (Andreas Braun, Silvana Cieslik, René Zmugg, Reiner Wichert, Peter Klein, Sven Havemann), *In Wohnen – Pflege – Teilhabe - Besser leben durch Technik. 7. Deutscher AAL-Kongress*, VDE Verlag, 2014.

2013 **User requirements for navigation assistance in public transit for elderly people** (Stefanie Müller, Felix Kamieth, Andreas Braun, Tim Dutz, Peter Klein), *In Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 55, 2013.

2013 **Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing** (Tobias Grosse-Puppendahl, Andreas Braun, Felix Kamieth, Arjan Kuijper), *In Proceedings of the 2013 ACM annual conference on Human factors in computing systems*, pp. 1401-1410, 2013.

2013 **Context-based bounding volume morphing in pointing gesture applications** (Andreas Braun, Arthur Fischer, Alexander Marinc, Carsten Stocklöw, Martin Majewski), *In HCII'13 Proceedings of the 15th international conference on Human-computer interaction*, Springer Berlin Heidelberg, pp. 147-156, 2013.

2013 **Capacitive sensor-based hand gesture recognition in ambient intelligence scenarios** (Andreas Braun, Tim Dutz, Felix Kamieth), *In Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 5, 2013.

2013 **Marker-Free Indoor Localization and Tracking of Multiple Users in Smart Environments Using a Camera-Based Approach** (Andreas Braun, Tim Dutz, Michael Alekseew, Philipp Schillinger, Alexander Marinc), *In Distributed, Ambient and Pervasive Interactions*, 2013.

2012 **Honeyfish - a high resolution gesture recognition system based on capacitive proximity sensing** (Tobias Grosse-Puppendahl, Andreas Braun), *In Embedded World Conference 2012*, Haar: WEKA Fachmedien, 2012 (Design & Elektronik), pp. 10pp, 2012.

2012 **Context recognition using capacitive sensor arrays in beds** (Andreas Braun, Henning Heggen), *In Technik für ein selbstbestimmtes Leben - 5. Deutscher AAL-Kongress*, VDE VERLAG GmbH, 2012.

2012 **Visual Support System for Selecting Reactive Elements in Intelligent Environments** (Martin Majewski, Andreas Braun, Alexander Marinc, Arjan Kuijper), *In International Conference on Cyberworlds*, pp. 251-255, 2012.

2012 **CapFloor – A Flexible Capacitive Indoor Localization System** (Andreas Braun, Henning Heggen, Reiner Wichert), *In Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking (Stefano Chessa, Stefan Knauth, eds.)*, Communications in Computer and Information Science, pp. 26-35, 2012.

2011 **Empowering and integrating senior citizens with virtual coaching** (Andreas Braun, Peter H. M. P. Roelofsma, Dieter Ferring, Milla Immonen), *In Ambient Intelligence (David V. Keyson, Mary Lou Maher, Norbert Streitz, Adrian Cheok, Juan Carlos Augusto, Reiner Wichert, Gwenn Englebienne, Hamid Aghajan, Ben J. A. Kröse, eds.)*, Springer Berlin Heidelberg, volume 7040, pp. 369-370, 2011.

2011 **Interactive personalization of ambient assisted living environments** (Alexander Marinc, Carsten Stocklöw, Andreas Braun, Carsten Limberger, Cristian Hofmann, Arjan Kuijper), *In Proceeding HI'11 Proceedings of the 2011 international conference on Human interface and the management of information*, volume Part I, pp. 567-576, 2011.

2011 **Passive identification and control of arbitrary devices in smart environments** (Andreas Braun, Felix Kamieth), *In HCII'11 Proceedings of the 14th international conference on Human-computer interaction: towards mobile and intelligent interaction environments (Julie A. Jacko, ed.)*, Springer-Verlag, pp. 147-154, 2011.

2011 **Adaptive implicit interaction for healthy nutrition and food intake supervision** (Felix Kamieth, Andreas Braun, Christian Schlehuber), *In HCII'11 Proceedings of the 14th international conference on Human-computer interaction: towards mobile and intelligent interaction environments*, Springer-Verlag, pp. 205-212, 2011.

2011 **Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments** (Tobias Grosse-Puppendahl, Alexander Marinc, Andreas Braun), *In Ambient Intelligence (David V Keyson, Mary Lou Maher, Norbert Streitz, Adrian Cheok, Juan Carlos Augusto, Reiner Wichert, Gwenn Englebienne, Hamid Aghajan, Ben J A Kröse, eds.)*, Springer, volume 7040, pp. 314-323, 2011.

2011 **Designing a multi-purpose capacitive proximity sensing input device** (Andreas Braun, Pascal Hamisu), *In Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments PETRA 11*, ACM Press, 2011.

2010 **Analyse des Schlafverhaltens durch kapazitive Sensorarrays zur Ermittlung der Wirbelsäulenbelastung** (Pascal Hamisu, Andreas Braun), *In 3. Deutscher AAL Kongress*, VDE VERLAG GmbH, pp. 3-6, 2010.

2009 **Using the human body field as a medium for natural interaction** (Andreas Braun, Pascal Hamisu), *In Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments - PETRA '09*, ACM Press, pp. 1-7, 2009.

2012 **Synergieeffekte aus der Kombination verschiedener AAL Lösungen** (Kerstin Klauß, Stefanie Müller, Andreas Braun, Tim Dutz, Felix Kamieth, Peter Klein), *In Mensch & Computer 2012–Workshopband: interaktiv informiert–allgegenwärtig und allumfassend!? (H. Reiterer, O. Deussen, eds.)*, Oldenbourg Verlag, pp. 61-67, 2012.

2013 **Building Up Virtual Environments Using Gestures** (Alexander Marinc, Carsten Stocklöw, Andreas Braun), *In Chapter in Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life*, Springer Berlin Heidelberg, pp. 70-78, 2013.

2013 **Providing Visual Support for Selecting Reactive Elements in Intelligent Environments** (Martin Majewski, Andreas Braun, Alexander Marinc, Arjan Kuijper), *In Chapter in Transactions on Computational Science XVIII*, Springer, pp. 248-263, 2013.

2013 **Unobtrusive Recognition of Working Situations** (Tobias Grosse-Puppendahl, Sebastian Benchea, Felix Kamieth, Andreas Braun, Christian Schuster), *In Chapter in Distributed, Ambient, and Pervasive Interactions*, Springer Berlin Heidelberg, pp. 115-121, 2013.

2013 **AmbiTrack-Marker-free Indoor Localization and Tracking of Multiple Users in Smart Environments with a Camera-based Approach** (Andreas Braun, Tim Dutz), *In Chapter in Evaluating AAL Systems Through Competitive Benchmarking*, Springer Berlin Heidelberg, pp. 83-93, 2013.

2012 **Dynamic User Representation in Video Phone Applications** (Andreas Braun, Reiner Wichert), *In Chapter in Constructing Ambient Intelligence*, Springer Berlin Heidelberg, pp. 184-188, 2012.

C. Supervising Activities

The following list summarizes the student bachelor, diploma and master thesis supervised by the author. The results of these works were partially used as an input into the thesis.

C.1. Diploma and Master Thesis

1. Große-Puppendahl, Tobias - Multi-hand Interaction Using Custom Capacitive Proximity Sensors - MSc TU Darmstadt 2012
2. Berghöfer, Yannick - Human-Machine-Interfaces in Automotive Environments using Capacitive Proximity Sensors - MSc TU Darmstadt 2013
3. Krepp, Stefan - Unobtrusive Surface Touch Recognition using Acoustic Tracking - MSc TU Darmstadt 2014

C.2. Bachelor Thesis

1. Fischer, Arthur - Unterstützung von zielbasierter Interaktion durch gestenerkennende Zeigegeräte - BSc TU Darmstadt 2012
2. Majewski, Martin - Visual-aided Selection of Reactive Elements in Intelligent Environments - BSc TU Darmstadt 2012
3. Neumann, Stephan - Automotive interfaces using an interactive armrest - BSc TU Darmstadt 2014

D. Curriculum Vitae

Personal Data

Name	Andreas Braun
Birth date & place	04.10.1982 Aschaffenburg, Germany
Nationality	German

Education

2008 – 2010	Master of Science in Computational Engineering at Technical University of Darmstadt, Germany
2004 – 2008	Bachelor of Science in Computational Engineering at Technical University of Darmstadt, Germany
2002 – 2004	Study of Physics at Julius-Maximilians Universität in Würzburg, Germany

Work Experience

2010 –	Researcher, Competence Center Interactive Multimedia Appliances, Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany, Focus: HCI applications in smart environments
2008 – 2009	Student Assistant, Competence Center Interactive Multimedia Appliances, Fraunhofer Institute for Computer Graphics Research, Darmstadt, Germany, Focus: Sensor applications and interactive systems

Bibliography

[AAB98] ARMSTRONG D., ASHER J., BENITEZ S.: Surface acoustic wave touchscreen with housing seal, 1998. 39

[AKKM13] ADIB F., KABELAC Z., KATABI D., MILLER R. C.: 3D Tracking via Body Radio Reflections. xii, 34

[AWD*13] ARNDT M., WILLE S., DE SOUZA L., REY V. F., WEHN N., BERNS K.: Performance evaluation of ambient services by combining robotic frameworks and a smart environment platform. *Robotics and Autonomous Systems* (2013). 37

[Bax96] BAXTER L. K.: Capacitive Sensors. *Sensors Peterborough NH*, 1 (1996), 1–17. 7, 16, 17

[BB98] BERTOZZI M., BROGGI A.: GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on 7*, 1 (1998), 62–81. 54

[BBL93] BAUDEL T., BEAUDOUIN-LAFON M.: Charade: remote control of objects using free-hand gestures. *Communications of the ACM 36*, 7 (July 1993), 28–35. 95

[BBRS06] BALLAGAS R., BORCHERS J., ROHS M., SHERIDAN J. G.: The smart phone: a ubiquitous input device. *Pervasive Computing, IEEE 5*, 1 (2006), 70–77. 37, 43

[BD13] BRAUN A., DUTZ T.: AmbiTrack-Marker-free Indoor Localization and Tracking of Multiple Users in Smart Environments with a Camera-based Approach. In *Evaluating AAL Systems Through Competitive Benchmarking*. Springer Berlin Heidelberg, 2013, pp. 83–93. 36, 37

[BDA*13] BRAUN A., DUTZ T., ALEKSEEW M., SCHILLINGER P., MARINC A.: Marker-Free Indoor Localization and Tracking of Multiple Users in Smart Environments Using a Camera-Based Approach. In *Distributed, Ambient and Pervasive Interactions* (2013). xii, 36

[BDK13] BRAUN A., DUTZ T., KAMIETH F.: Capacitive sensor-based hand gesture recognition in ambient intelligence scenarios. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments* (2013), ACM, p. 5. xii, xiii, 63, 64, 93, 94

[BF14] BRAUN A., FRANK S.: The Capacitive Chair. 2014. xii, 69, 71, 83

[BGS*06] BROXVALL M., GRITTI M., SAFFIOTTI A., SEO B.-S., CHO Y.-J.: PEIS ecology: Integrating robots into smart environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 212–218. xii, 37, 44

[BH11] BRAUN A., HAMISU P.: Designing a multi-purpose capacitive proximity sensing input device. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments PETRA 11* (2011), ACM Press. xii, xiii, 63, 92, 93

[BH12] BRAUN A., HEGGEN H.: Context recognition using capacitive sensor arrays in beds. In *Technik für ein selbstbestimmtes Leben - 5. Deutscher AAL-Kongress* (Berlin, 2012), VDE VERLAG GmbH. xii, 68, 81

[BHK97] BELHUMEUR P. N., HESPANHA J. A. P., KRIEGMAN D. J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 19*, 7 (1997), 711–720. 17

[BHW12] BRAUN A., HEGGEN H., WICHERT R.: CapFloor – A Flexible Capacitive Indoor Localization System. In *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking* (2012), Chessa S., Knauth S., (Eds.), Communications in Computer and Information Science, pp. 26–35. xiii, 64, 78, 80

[BI04] BAO L., INTILLE S. S.: Activity Recognition from User-Annotated Acceleration Data. *Pervasive Computing 3001* (2004), 1–17. 37, 41

[BKKF13] BECK S., KUNERT A., KULIK A., FROEHLICH B.: Immersive Group-to-Group Telepresence. *Visualization and Computer Graphics, IEEE Transactions on 19*, 4 (2013), 616–625. 29

[BO10a] BARRETT G., OMOTE R.: Projected capacitive touch screens. *Inf. Disp. Mag* (2010). xi, 10, 11, 64

[BO10b] BARRETT G., OMOTE R.: Projected-capacitive touch technology. *Information Display* (2010). 9, 10

[BWGT11] BULLING A., WARD J. A., GELLERSEN H., TROSTER G.: Eye movement analysis for activity recognition using electrooculography. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 4 (2011), 741–753. xii, 37, 41

[BZWK*14] BRAUN A., ZANDER-WALZ S., KREPP S., RUS S., WICHERT R., KUIJPER A.: CapTap - Combining Capacitive Gesture Recognition and Knock Detection. 2014. 95

[CAL10] CHENG J., AMFT O., LUKOWICZ P.: Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In *Pervasive Computing*. Springer, 2010, pp. 319–336. 61, 107, 108

[CC89] CAMP R. C., CAMP ROBERT C.: Benchmarking: the search for industry best practices that lead to superior performance. 45

[CD07] COOK D. J., DAS S. K.: How smart are our environments? An updated look at the state of the art. *Pervasive and mobile computing 3*, 2 (2007), 53–73. xv, 3, 25, 36

[CDCT*01] COWIE R., DOUGLAS-COWIE E., TSAPATSOULIS N., VOTSIS G., KOLLIAS S., FELLENZ W., TAYLOR J. G.: Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE 18*, 1 (2001), 32–80. 37, 39

[CHE00] CRAWFORD L. E., HUTTENLOCHER J., ENGEBRETSON P. H.: Category effects on estimates of stimuli: Perception or reconstruction? *Psychological Science 11*, 4 (2000), 280–284. 54

[CK12] CHESSA S., KNAUTH S.: *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking*, vol. 309 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 4, 38, 46, 79

[Cla62] CLARKE A. C.: Hazards of prophecy: The failure of imagination. *Profiles of the Future, Gollancz, London* (1962). 2

[Com06] COMMONS W.: A bayer pattern on a sensor in isometric perspective/projection, 2006. xi, 26

[Cor13] CORPORATION C. S.: CY3271 PSoC® FirstTouchTM Starter Kit with CYFiTM Low-Power RF, 2013. 92

[CPP10] CHINTALAPUDI K., PADMANABHA IYER A., PADMANABHAN V. N.: Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking* (2010), ACM, pp. 173–184. 37, 38

[Cro09] CROLOTTE A.: Issues in benchmark metric selection. In *Performance Evaluation and Benchmarking*. Springer, 2009, pp. 146–152. 46

[CRV08] CORBISHLEY P., RODRÍGUEZ-VILLEGAS E.: Breathing detection: towards a miniaturized, wearable, battery-operated monitoring system. *Biomedical Engineering, IEEE Transactions on 55*, 1 (2008), 196–204. 32

[CS06] CORADESCHI S., SAFFIOTTI A.: Symbiotic robotic systems: Humans, robots, and smart environments. *Intelligent Systems, IEEE 21*, 3 (2006), 82–84. 37, 44

[Cyp12] CYPRESS: *Cypress TrueTouch® Touchscreen Solution Drives "Floating Touch" Navigation Feature in New Xperia™ sola Smartphone from Sony Mobile Communications*. Tech. rep., 2012. 9

[DBM14] DJAKOW M., BRAUN A., MARINC A.: MoviBed - sleep analysis using capacitive sensors. In *Distributed, Ambient, and Pervasive Interactions*. 2014. xiii, 69, 81, 82

[DGT*13] DEMENTYEV A., GUMMESON J., THRASHER D., PARKS A., GANESAN D., SMITH J. R., SAMPLE A. P.: Wirelessly powered bistable display tags. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (2013), ACM, pp. 383–386. 37, 42

[DL01] DIETZ P., LEIGH D.: DiamondTouch. In *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01* (New York, New York, USA, Nov. 2001), ACM Press, p. 219. 11

[DLP03] DONGARRA J. J., LUSZCZEK P., PETITET A.: The LINPACK benchmark: past, present and future. *Concurrency and Computation: practice and experience 15*, 9 (2003), 803–820. 45

[DT09] DEARMAN D., TRUONG K. N.: BlueTone: a framework for interacting with public displays using dual-tone multi-frequency through bluetooth. In *Proceedings of the 11th international conference on Ubiquitous computing* (2009), ACM, pp. 97–100. 37, 43

[Fra13] FRANK R.: *Understanding smart sensors*. Artech House, 2013. 25

[GBK99] GELLERSEN H.-W., BEIGL M., KRULL H.: The MediaCup: Awareness technology embedded in an everyday object. In *Handheld and Ubiquitous Computing* (1999), Springer, pp. 308–310. xii, 37, 42

[GKIH09] GLAS D. F., KANDA T., ISHIGURO H., HAGITA N.: Simultaneous people tracking and localization for social robots using external laser range finders. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (2009), IEEE, pp. 846–853. xii, 44

[Gli00] GLINSKY A.: *Theremin: Ether music and espionage*. University of Illinois Press, 2000. xi, 18

[GPB12] GROSSE-PUPPENDAHL T., BRAUN A.: Honeyfish - a high resolution gesture recognition system based on capacitive proximity sensing. In *Embedded World Conference 2012* (2012), Haar: WEKA Fachmedien, 2012 (Design & Elektronik), p. 10pp. 105

[GPBB*13] GROSSE-PUPPENDAHL T., BERGHOEFER Y., BRAUN A., WIMMER R., 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. xi, 10, 11, 57, 96

[GPBKK13] GROSSE-PUPPENDAHL T., BRAUN A., KAMIETH F., KUIJPER A.: Swiss-cheese extended: an object recognition method for ubiquitous interfaces based on capacitive proximity sensing. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems* (2013), ACM, pp. 1401–1410. 16, 57, 105, 118

[GPMB11] GROSSE-PUPPENDAHL T., MARINC A., BRAUN A.: Classification of User Postures with Capacitive Proximity Sensors in AAL-Environments. In *Ambient Intelligence* (2011), Keyson D. V., Maher M. L., Streitz N., Cheok A., Augusto J. C., Wichert R., Englebienne G., Aghajan H., Kröse

B. J. A., (Eds.), vol. 7040 of *Lecture Notes in Computer Science*, Springer, pp. 314–323. xi, 12, 104

[GZBB09] GEORGE B., ZANGL H., BRETTERKLIEBER T., BRASSEUR G.: Seat occupancy detection based on capacitive sensing. *Instrumentation and Measurement, IEEE Transactions on 58*, 5 (2009), 1487–1494. 17

[Han05] HAN J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05* (New York, New York, USA, Oct. 2005), ACM Press, p. 115. xii, 38

[HB10] HAMISU P., BRAUN A.: Analyse des Schlafverhaltens durch kapazitive Sensorarrays zur Ermittlung der Wirbelsäulenbelastung. In *3. Deutscher AAL Kongress* (2010), VDE VERLAG GmbH, pp. 3–6. 81

[HBvdH*11] HUIJNEN C., BADII A., VAN DEN HEUVEL H., CALEB-SOLLY P., THIEMERT D.: "Maybe it becomes a buddy, but do not call it a robot"–seamless cooperation between companion robotics and smart homes. In *Ambient Intelligence*. Springer, 2011, pp. 324–329. 37, 44

[HCM*13] HOQUE M. E., COURGEON M., MARTIN J.-C., MUTLU B., PICARD R. W.: Mach: My automated conversation coach. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (2013), ACM, pp. 697–706. xii, 37, 40

[Hen00] HENNING J. L.: SPEC CPU2000: Measuring CPU performance in the new millennium. *Computer 33*, 7 (2000), 28–35. 45

[HPR00] HAN J., PRATT M., REGLI W. C.: Manufacturing feature recognition from solid models: a status report. *Robotics and Automation, IEEE Transactions on 16*, 6 (2000), 782–796. 17

[HSH11] HARRISON C., SCHWARZ J., HUDSON S. E.: TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), ACM, pp. 627–636. xii, 31, 32, 97

[HSM00] HARADA T., SATO T., MORI T.: Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. In *Human Motion, 2000. Proceedings. Workshop on* (2000), IEEE, pp. 99–106. 66

[HSP12] HARRISON C., SATO M., POUPYREV I.: Capacitive fingerprinting: exploring user differentiation by sensing electrical properties of the human body. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), ACM, pp. 537–544. xi, 23, 24

[Hu62] HU M.-K.: Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on 8*, 2 (1962), 179–187. 73

[HW08] HAKLAY M., WEBER P.: Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE 7*, 4 (2008), 12–18. 90

[iee07] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, 2007. 25

[IHQ04] INGRAM S. J., HARMER D., QUINLAN M.: Ultrawideband indoor positioning systems and their use in emergencies. In *Position Location and Navigation Symposium, 2004. PLANS 2004* (2004), IEEE, pp. 706–715. 4

[IKH11] IZADI S., KIM D., HILLIGES O.: KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. *Proceedings of the 24th ...* (2011), 559 – 568. xii, 28, 29

[Inc13] INC. S.: Global Posture Study, 2013. 83

[Inc14] INC. M. M.: Theremini, 2014. xi, 18

[Jai91] JAIN R.: *The art of computer systems performance analysis*, vol. 182. John Wiley & Sons Chichester, 1991. 46

[KBBC05] KALTENBRUNNER M., BOVERMANN T., BENCINA R., COSTANZA E.: TUIO: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation* (2005). 90

[KJJ11] KREJCAR O., JIRKA J., JANCKULIK D.: Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection. *Sensors* (2011). 81

[KMS*12] KOELSTRA S., MUHL C., SOLEYMANI M., LEE J.-S., YAZDANI A., EBRAHIMI T., PUN T., NIJHOLT A., PATRAS I.: Deap: A database for emotion analysis; using physiological signals. *Affective Computing, IEEE Transactions on 3*, 1 (2012), 18–31. xii, 37, 40

[KOR*02] KRUM D. M., OMOTESO O., RIBARSKY W., STARNER T., HODGES L. F.: Speech and gesture multimodal control of a whole Earth 3D visualization environment. 195–200. 95

[KQWA10] KHOSROWABADI R., QUEK H. C., WAHAB A., ANG K. K.: EEG-based emotion recognition using self-organizing map for boundary detection. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (2010), IEEE, pp. 4242–4245. 37, 40

[LBO11] LESER R., BACA A., OGRIS G.: Local positioning systems in (game) sports. *Sensors 11*, 10 (2011), 9778–9797. 4

[LBT02] LENMAN S., BRETZNER L., THURESSON B.: Using marking menus to develop command sets for computer vision based hand gesture interfaces. *Proceedings of the second Nordic conference on Human-computer interaction* (2002), 239–242. 95

[LC85] LEWIS B. C., CREWS A. E.: The evolution of benchmarking as a computer performance evaluation technique. *MIS Quarterly* (1985), 7–16. 45

[Lew04] LEWIS F. L.: Wireless sensor networks. *Smart environments: technologies, protocols, and applications* (2004), 11–46. 25

[LS09] LAUTERBACH C., STEINHAGE A.: SensFloor ® - A Large-area Sensor System Based on Printed Textiles Printed Electronics. In *Ambient Assisted Living Congress* (2009), VDE Verlag. 59, 64, 107

[LSKB13] LASECKI W. S., SONG Y. C., KAUTZ H., BIGHAM J. P.: Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work* (2013), ACM, pp. 1203–1212. 37, 41

[LWG05] LANDSIEDEL O., WEHRLE K., GÖTZ S.: Accurate prediction of power consumption in sensor networks. In *Proc. of the Second Workshop on Embedded Networked Sensors* (2005), Citeseer. 47

[LZWV09] LIU J., ZHONG L., WICKRAMASURIYA J., VASUDEVAN V.: uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing 5*, 6 (2009), 657–675. 37, 39

[MA07] MITRA S., ACHARYA T.: Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37*, 3 (2007), 311–324. 38

[Mac04] MACLACHLAN R.: Spread Spectrum Capacitive Proximity Sensor, 2004. 57, 61, 107, 108

[MBMK13] MAJEWSKI M., BRAUN A., MARINC A., KUIJPER A.: Providing Visual Support for Selecting Reactive Elements in Intelligent Environments. In *Transactions on Computational Science XVIII*.

Springer, 2013, pp. 248–263. 37, 39

[Mic] MICROSOFT: What Is the Windows Experience Index. 45

[Mic13] MICROCHIP TECHNOLOGY INC.: *GestIC ® Design Guide : Electrodes and System Design MGC3130*. Tech. rep., 2013. xiii, 107, 109

[Mit07] MITCHELL H. B.: *Multi-sensor data fusion*. Springer, 2007. 15

[MPLK05] MOON J., PARK J., LEE T., KIM Y.: Transparent conductive film based on carbon nanotubes and PEDOT composites. *Diamond and related materials 14*, 11 (2005), 1882–1887. 10

[MWBS09] MULLONI A., WAGNER D., BARAKONYI I., SCHMALSTIEG D.: Indoor positioning and navigation with camera phones. *Pervasive Computing, IEEE 8*, 2 (2009), 22–31. xii, 36, 37, 38

[Nok12] NOKIA: *Puremotion Technology White Paper*. Tech. rep., 2012. 9

[Nov14] NOVINT TECHNOLOGIES INC: Novint Falcon, 2014. xi, 19, 20

[OHR13] OCHIAI Y., HOSHI T., REKIMOTO J.: Three-dimensional Mid-air Acoustic Manipulation by Ultrasonic Phased Arrays. *arXiv preprint arXiv:1312.4006* (2013). xii, 30, 31

[OKLVO12] OLSSON T., KÄRKKÄINEN T., LAGERSTAM E., VENTÄ-OLKKONEN L.: User evaluation of mobile augmented reality scenarios. *Journal of Ambient Intelligence and Smart Environments 4*, 1 (2012), 29–47. 37, 43

[OL13] OREIFEJ O., LIU Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on* (2013), IEEE, pp. 716–723. xii, 37, 41

[Pan12] PANGER G.: Kinect in the kitchen: testing depth camera interactions in practical home environments. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (2012), ACM, pp. 1985–1990. xii, 28, 29

[PC00] PENTLAND A., CHOUDHURY T.: Face recognition for smart environments. *Computer 33*, 2 (2000), 50–55. xi, 26, 27

[PCB00] PRIYANTHA N. B., CHAKRABORTY A., BALAKRISHNAN H.: The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (2000), ACM, pp. 32–43. xii, 30

[PGGP13] PU Q., GUPTA S., GOLLAKOTA S., PATEL S.: Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (2013), ACM, pp. 27–38. xii, 34, 35, 37, 39

[PJS*13] PHAM C., JACKSON D., SCHÖNING J., BARTINDALE T., PLÖTZ T., OLIVIER P.: FoodBoard: surface contact imaging for food recognition. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (2013), ACM, pp. 749–752. xi, 27, 28

[PL13] PIRKL G., LUKOWICZ P.: Resonant magnetic coupling indoor localization system. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (2013), ACM, pp. 59–62. xii, 36, 37, 38

[PLCH02] PARADISO J. A., LEO C. K., CHECKA N., HSIAO K.: Passive acoustic knock tracking for interactive windows. In *CHI '02 extended abstracts on Human factors in computing systems - CHI '02* (New York, New York, USA, Apr. 2002), ACM Press, p. 732. 4

[Pos11] POSLAD S.: *Ubiquitous computing: smart devices, environments and interactions*. Wiley. com, 2011. 2, 3, 36

[PSF01] PRASSLER E., SCHOLZ J., FIORINI P.: A robotics wheelchair for crowded public environment. *Robotics & Automation Magazine, IEEE 8*, 1 (2001), 38–45. xii, 30

[PSLS12] POUPYREV I., SCHOESSLER P., LOH J., SATO M.: Botanicus Interacticus: interactive plants technology. In *ACM SIGGRAPH 2012 Emerging Technologies* (2012), ACM, p. 4. xi, 23, 24, 107, 108

[RAMB*05] RANGANATHAN A., AL-MUHTADI J., BIEHL J., ZIEBART B., CAMPBELL R. H., BAILEY B.: Towards a pervasive computing benchmark. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on* (2005), IEEE, pp. 194–198. 45, 46

[Rek02] REKIMOTO J.: SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* (2002), ACM, pp. 113–120. 4

[SAW94] SCHILIT B., ADAMS N., WANT R.: Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on* (1994), IEEE, pp. 85–90. 2

[Sea97] SEAVER A. E.: Human Body capacitance: A review. In *ESA Annual Meeting Proceedings, Laplacian Press: Morgan Hill, CA* (1997), pp. 16–28. 66

[SFC*13] SHOTTON J., FITZGIBBON A., COOK M., SHARP T., FINOCCHIO M., MOORE R., KIPMAN A., BLAKE A.: Real-time human pose recognition in parts from single depth images. *Communications of the ACM 56*, 1 (June 2013), 116–124. 29

[SKOM06] SUGANO M., KAWAZOE T., OHTA Y., MURATA M.: Indoor localization system using RSSI measurement of wireless sensor network based on ZigBee standard. *Target 538* (2006), 50. xii, 33

[SL86] SALMI T., LEINONEN L.: Automatic analysis of sleep records with static charge sensitive bed. *Electroencephalography and Clinical Neurophysiology 64*, 1 (1986), 84–87. 69

[SM95] SCHMANDT C., MULLINS A.: AudioStreamer: exploiting simultaneity for listening. In *Conference companion on Human factors in computing systems* (1995), ACM, pp. 218–219. 20

[Smi88] SMITH J. E.: Characterizing computer performance with a single number. *Communications of the ACM 31*, 10 (1988), 1202–1206. 46

[Smi96] SMITH J. R.: Field mice: Extracting hand geometry from electric field measurements. *IBM Systems Journal 35*, 3.4 (1996), 587–608. xi, 4, 10, 61, 105

[SN04] SHIREHJINI A., NAZARI A.: A novel interaction metaphor for personal environment control: Direct manipulation of physical environment based on 3d visualization. *Computers & Graphics 28*, 5 (2004), 667–675. xii, 37, 43

[SPH12] SATO M., POUPYREV I., HARRISON C.: Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (May 2012), pp. 483–492. xi, 4, 23, 24, 107, 108

[SPSS11] SUNG J., PONCE C., SELMAN B., SAXENA A.: Human Activity Detection from RGBD Images. In *Plan, Activity, and Intent Recognition* (2011). 29

[SV01] SCHMIDT A., VAN LAERHOVEN K.: How to build smart appliances? *Personal Communications, IEEE 8*, 4 (2001), 66–71. 41, 60

[SWD*98] SMITH J., WHITE T., DODGE C., PARADISO J., GERSHENFELD N., ALLPORT D.: Electric field sensing for graphical interfaces. *Computer Graphics and Applications, IEEE 18*, 3 (1998), 54–60. xi, 20, 61, 66, 71

[TCD*00] THOMAS B., CLOSE B., DONOGHUE J., SQUIRES J., DE BONDI P., MORRIS M., PIEKARSKI W.: ARQuake: An outdoor/indoor augmented reality first person application. In *Wearable Computers, The Fourth International Symposium on* (2000), IEEE, pp. 139–146. 4

[TCY*11] TSAI P.-H., CHEN T.-Y., YU C.-R., SHIH C.-S., LIU J. W. S.: Smart medication dispenser: design, architecture and implementation. *Systems Journal, IEEE 5*, 1 (2011), 99–110. 37, 42

[TKA06] TABAR A. M., KESHAVARZ A., AGHAJAN H.: Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks* (2006), ACM, pp. 145–154. xi, 26, 27

[TSPM01] TAN H. Z., SLIVOVSKY L. A., PENTLAND A., MEMBER S.: A sensing chair using pressure distribution sensors. *Mechatronics, IEEE/ASME Transactions on 6*, 3 (2001), 261–268. 83

[Val08] VALLI A.: The design of natural interaction. *Multimedia Tools and Applications 38*, 3 (2008), 295–305. 61

[VCBE07] VAJK T., COULTON P., BAMFORD W., EDWARDS R.: Using a mobile phone as a “wii-like” controller for playing games on a large public display. *International Journal of Computer Games Technology 2008* (2007). xii, 37, 43

[VKMV11] VALTONEN M., KAILA L., MÄENTAUSTA J., VANHALA J.: Unobtrusive human height and posture recognition with a capacitive sensor. *Journal of Ambient Intelligence and Smart Environments 3*, 4 (2011), 305–332. 24

[VKV12] VALTONEN M., KIVIMÄKI T., VANHALA J.: Capacitive 3D user tracking with a mobile demonstration platform. In *Proceeding of the 16th International Academic MindTrek Conference* (2012), ACM, pp. 61–63. 24

[VMV09] VALTONEN M., MAENTAUSTA J., VANHALA J.: TileTrack: Capacitive human tracking using floor tiles. In *IEEE International Conference on Pervasive Computing and Communications* (Mar. 2009), IEEE, pp. 1–10. 12, 24, 59, 64, 107

[VRV10] VALTONEN M., RAULA H., VANHALA J.: Human body tracking with electric field ranging. In *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments* (2010), ACM, pp. 183–186. 24

[WB95] WELCH G., BISHOP G.: An introduction to the Kalman filter, 1995. 15

[WB03] WU M., BALAKRISHNAN R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST ’03* (New York, New York, USA, Nov. 2003), ACM Press, pp. 193–202. 95

[WB09] WIMMER R., BORING S.: HandSense: discriminating different ways of grasping and holding a tangible user interface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction* (2009), ACM, pp. 359–362. xi, 22

[WB11] WIMMER R., BAUDISCH P.: Modular and deformable touch-sensitive surfaces based on time domain reflectometry. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), ACM, pp. 517–526. xi, 22

[Wei84] WEICKER R. P.: Dhrystone: a synthetic systems programming benchmark. *Communications of the ACM 27*, 10 (1984), 1013–1030. 45

[Wei91] WEISER M.: The Computer for the 21st Century. *Scientific American 265*, 3 (1991), 94–104. xii, 2, 28, 42

[WF07] WILSON T. V., FENLON W.: How the iPhone Works - Multi-touch systems, 2007. 17

[WHKS06] WIMMER R., HOLLEIS P., KRANZ M., SCHMIDT A.: Thracker - Using Capacitive Sensing for Gesture Recognition. In *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)* (2006), IEEE, pp. 64–64. xi, 4, 17, 21, 61, 107, 108

[Wil04] WILSON J. S.: *Sensor technology handbook*. Access Online via Elsevier, 2004. 46

[WKBS07] WIMMER R., KRANZ M., BORING S., SCHMIDT A.: A Capacitive Sensing Toolkit for Pervasive Activity Detection and Recognition. *Fifth Annual IEEE International Conference on Pervasive Computing and Communications PerCom07* (2007), 171–180. xi, 21, 63

[WME*10] WÖLLMER M., METALLINOU A., EYBEN F., SCHULLER B., NARAYANAN S. S.: Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In *INTERSPEECH* (2010), pp. 2362–2365. 37, 40

[WP03] WILSON A., PHAM H.: Pointing in intelligent environments with the worldcursor. In *INTERACT International Conference on HumanComputer Interaction* (2003). 39

[WP10] WILSON J., PATWARI N.: Radio tomographic imaging with wireless networks. *Mobile Computing, IEEE Transactions on 9*, 5 (2010), 621–632. xii, 33, 34

[WRS*12] WU H.-Y., RUBINSTEIN M., SHIH E., GUTTAG J., DURAND F., FREEMAN W.: Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics 31*, 4 (July 2012), 1–8. xi, 27

[WS03] WILSON A., SHAFER S.: XWand: UI for intelligent spaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2003), no. 5 in CHI '03, Microsoft Research, ACM, pp. 545–552. xii, 37, 38, 39

[WTT13] WATANABE H., TERADA T., TSUKAMOTO M.: Ultrasound-based movement sensing, gesture-, and context-recognition. In *Proceedings of the 17th annual international symposium on International symposium on wearable computers* (2013), ACM, pp. 57–64. xii, 30

[WYG*09] WRIGHT J., YANG A. Y., GANESH A., SASTRY S. S., MA Y.: Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 2 (2009), 210–227. 27

[XLL*13] XU C., LI S., LIU G., ZHANG Y., MILUZZO E., CHEN Y.-F., LI J., FIRNER B.: Crowd++: unsupervised speaker count with smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing* (2013), ACM, pp. 43–52. xii, 31, 32

[YDM12] YE J., DOBSON S., MCKEEVER S.: Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing 8*, 1 (2012), 36–66. 40

[YJS06] YILMAZ A., JAVED O., SHAH M.: Object tracking: A survey. *Acm Computing Surveys (CSUR) 38*, 4 (2006), 13. 16

[YNR13] YEO K. P., NANAYAKKARA S., RANSIRI S.: StickEar: making everyday objects respond to sound. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (2013), ACM, pp. 221–226. xii, 37, 42

[ZFBZ08] ZHANG C., FLORÊNCIO D., BA D. E., ZHANG Z.: Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *Multimedia, IEEE Transactions on 10*, 3 (2008), 538–548. xii, 31, 32

[Zha12] ZHANG Z.: Microsoft kinect sensor and its effect. *Multimedia, IEEE 19*, 2 (2012), 4–10. xii, 28, 29

[ZLB*87] ZIMMERMAN T. G., LANIER J., BLANCHARD C., BRYSON S., HARVILL Y.: A hand gesture interface device. In *ACM SIGCHI Bulletin* (May 1987), vol. 18, ACM, pp. 189–192. xii, 37, 38, 39, 95

[ZSP*95] ZIMMERMAN T. G., SMITH J. R., PARADISO J. A., ALLPORT D., GERSHENFELD N.: Applying electric field sensing to human-computer interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95* (New York, New York, USA, 1995), no. May, ACM Press, pp. 280–287. 20