Augmented Reality Interface for Home Automation and Control

Matthias Braun

BACHELORARBEIT

Nr. 0810237004-A

eingereicht am
Fachhochschul-Bachelorstudiengang

Mobile Computing

in Hagenberg

im Juni 2011

Diese Arbeit entstand im Rahmen des Gegenstands

Bachelorarbeit 1

im

Wintersemester 2010/11

Betreuer:

Prof. (FH) Dr. Clemens Holzmann

Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Wien, am 7. Juni 2011

Matthias Braun

Contents

Abstract

This bachelor thesis describes the deliberations and technologies behind building a vision-based Augmented Reality interface for home automation on mobile devices. The emerging fields of Augmented Reality as well as home automation are explored and their synergies resulting from combining these two technologies are investigated by discussing a prototype software that combines a home automation control with an Augmented Reality interface: the seeing remote control.

Chapter 1

Introduction and Scope

Current home automation implementations rely in terms of user interaction mostly on switches and panels built into one or multiple walls of a home [1]. Although mobile interfaces for home automation exist on smartphones, tablets and laptops, they only provide the same or a very similar interface as that of the non-portable panel (e.g., the mobile interface on the iPhone for the Gira home automation system offers the same menus on a more confined display), therefore not fully leveraging the possible benefits proper to mobile software like context-awareness.

Under the assumption that users prefer to control home appliances in their vicinity (as proposed by Hammond et al. [17, Cha. 2.3]) , a thinkable gain to home automation interfaces would surely be to offer users a means of controlling specifically those devices of the room they are currently in. Furthermore controlling a home automation system presently consists mostly of choosing the desired feature from a list and then adjusting some parameters. In order to make the numerous and often complex features of home automation simpler to control from a user perspective, an *Augmented Reality* (AR) interface can be employed putting the control element (e.g. a gauge or an on/off button) as close to the household gadget as possible and therefore making the association between the virtual control element and the real object immediate.

AR applications have gained widespread popularity through the advent of smartphones. The bulkiness of head-mounted displays and wearable computers of the past are no longer necessary for a user to experience Augmented Reality. Due to the advancement in processor speed and sensor capability of smartphones like the iPhone or Android-based phones, AR applications have been successfully implemented on these handheld devices. As Wagner [38, Cha. 1] argues, the pervasion of mobile devices in the consumer market is a strong indication that AR applications will be mostly used via a mobile phone or a tablet rather than a *head-mounted display*.

[1] Such panel interfaces are provided by enterprises like Siemens or Gira

To display virtual objects and/or information at the right spot in reality, there are at least two distinct methods to do so. Many popular AR applications rely on GPS tracking and magnetometer as means of localizing the user (e.g. *Layar*, *Wikitude*, and *3D Compass*) to then superimpose the desired content onto the real world according to the position of the user. Such an approach may be suited for applications that are designed to be used outdoors, but inside a building it is impossible to pinpoint a user's location via GPS due to the natural attenuation of the walls. Reitmayr and Drummond have shown that outdoor-oriented AR software can also be realized by using *Computer Vision* (CV) as an additional technology to estimate the position of the user [34]. See table 1.1 for an overview of the advantages and disadvantages of CV in indoor tracking. As illustrated by Rolland et al. [35] there are numerous possibilities to track the position of a device indoors in order to build an AR application, but the discussion of such tracking methods are out of the scope of this paper.

A different way to realize an indoor AR application is to use Computer Vision. As Kalkusch et al. have shown, indoor navigation is well within the possibilities of CV-based software [22]. The application is freed from the necessity of tracking its host device and instead uses pattern recognition algorithms to identify objects of interest learned beforehand and to determine the position of the host device relative to the detected object. Although recognizing three dimensional objects in real time has been proven possible by Hinterstoisser et al. [19] this paper focuses on the detection of planar objects, as dealing with three dimensional objects is still too resource-intensive for current smartphones.

Assuming that a user would control home automation software mostly inside the building, we have chosen this CV-based approach in order to build a *perceptual interface* of this home automation application. The term perceptual interface denotes the concept of an interface that tries to "perceive what the user is doing using Computer Vision..." [24] or gestures to improve interaction between the interface and the user. The perceptual interface doesn't rely solely on conventional user input like typing or mouse movements, but tries to exploit more ways such as visual information, gestures or audio impressions to communicate with users and their surroundings.

A conceivable use case scenario would be one in which the user wants to control a lamp in the room. Instead of walking to the light switch and manually flipping it, the user points the handheld device towards the lamp. Via its built-in camera the device recognizes the lamp and displays control elements on the screen of the handheld that let the user turn the light on, off or dim it.

In order to explore the possibilities and constraints of such an AR user interface in a home automation application, a prototype program was created that enables the user to control a room's lighting system that has been equipped with a *digitalSTROM* home automation system (see section 4.1 for

Advantages and disadvantages of Computer Vision tracking indoors	
Advantages	*Disadvantages*
The room doesn't need to be equipped with additional hardware (e.g., ultrasound beacons)	The room has to be sufficiently lit in order to recognize the pattern of an object
Localization accuracy is within one millimeter after pattern was detected [35]	Patterns have to be known beforehand (exception: markerless tracking like SLAM [11]
Independent from the physical properties of the surrounding walls that would cause echoes or attenuation	Susceptible to occlusion of the object to be detected
CV software can be run on off-the-shelf mobile devices [41]	The object has to be in view to be localized
Invariant to distortions in the magnetic field	Objects with the same optical pattern can't be discerned

Table 1.1: Comparison of advantages and disadvantages of CV-based indoor tracking.

more information on digitalSTROM. The interface of this application has an AR view that displays control elements like a power switch button for TV sets or dim buttons for lights. These overlaid elements can be used to control lamps, provided they are digitalSTROM ready. Besides lamps digitalSTROM plans to allow controlling jalousies, the burglar alarm and other parts of the house [2].

This paper gives attention to the deliberations of building and the design of such an AR interface that uses Computer Vision in order to recognize devices that can be controlled via the digitalSTROM bus system or —in possible future versions of the prototype— a similar technology like KNX or LonWorks.

1.1 Motivation

A home automation bus system like digitalSTROM will enable the user to program the devices in their household according to personal preferences as well as corresponding to natural circumstances like the course of the sun or severe weather. In the latter case the system can be programmed to adjust the sunblinds or close the windows automatically. Furthermore the support for the elderly and their caregivers is a growing issue that can be effectively addressed using appropriate technology as Haigh and Yanco argue [16]. In terms of security a home automation system can for example not only offer

[2] For further information see http://www.digitalstrom.org/

an easier way to control the burglar alarm but is also able to provide indoor light simulations that evoke the impressions that the house is inhabited for situations where its residents are in fact on vacation. Equipping a home with such functions can therefore increase the well-being and security of the people living in it.

In addition a bus system allows saving electricity by letting the user shut down unnecessary devices when the household is vacant or detecting devices that use a disproportionate amount of power. The more sophisticated and complex the possibilities in home automation become, the more intuitive the way of managing those features should be. An AR interface has the appeal of a certain aim-and-shoot interaction it offers to the user: By pointing the handheld controller at a device the selection step is as intuitive as it is with a common remote control. The virtual control elements displayed on top of the real device can be in the form of standard icons like the on/off button or a volume gauge making again the actual control step as habitual as using a remote control. This way of interacting with a device, has in comparison with conventional list-based interfaces the advantage of being able to limit the available control options to those, which are relevant to the currently selected device. Another strong benefit that comes with the CV-based approach of the prototype is the ability to run on commercial off-the-shelf devices which do not require purchase or installation of additional hardware like sensory beacons (as discussed in 1.1). The perceptual interface of such a program would turn a customary smartphone into a seeing remote control.

1.2 Goals

The goal of this paper is the examination of the topics of Augmented Reality, Computer Vision and home automation and the synergies that can be achieved when combining the technologies of these fields. It shall be explored if and how well an AR interface can enrich a home automation application and whether Computer Vision is a suitable approach to locate household objects that will be controllable via the home automation application.

To showcase and survey the practical implications of such an application, a prototype program that combines home automation with Augmented Reality was developed for the HTC Desire running Android 2.2. The home automation system utilized in this prototype software was digitalSTROM, using the dSK20 demo trunk including three lamps and one adapter for testing. For the Computer Vision and Augmented Reality part of the prototype the Qualcomm Augmented Reality SDK 0.10.0 (*QCAR*) was employed. See chapter 3 for a closer study of the SDK and fig. 1.1 for the setup used for the testing of the software prototype.

1.3 Challenges

The first question that arose in the planning phase of developing the AR interface, was which technology should be employed to determine at which object the user is pointing. As mentioned in chapter 1 multiple options were discussed before deciding to use Computer Vision for its accuracy and the fact that no additional hardware (e.g., ultrasound emitters for indoor positioning) is required. After settling on the type of localization mechanism, finding an appropriate framework for Computer Vision and Augmented Reality on mobile devices was the next task, as developing an own framework was out of scope of this bachelor's thesis. Experimenting with different frameworks like OpenCV [3], DOT by Stefan Hinterstoisser [4] and Studierstube by the University of Graz [5] yielded unsatisfactory results as they either were only capable of recognizing QR-like markers or because of their hardware requirements, making their use on a mobile device impossible. Eventually the freely available QCAR SDK met the requisites as it could recognize natural features [6] on a planar object, rendering virtual content at the position of the object and being able to do this on a mobile phone with Android 2.2 as the operating system.

Learning to correctly employ the QCAR SDK and making use of its sample applications was the next step. Especially the beta stage of the QCAR SDK and mastering the intricacies of the Java Native Interface were challenges that could eventually be overcome by doing research and sharing experience with the Qualcomm forum employees.

1.4 Outline

This paper describes the fundamental technologies that power the prototype software in chapter 2, namely Augmented Reality and Computer Vision with respect to the definition of these fields and their area of application. After discussing vision-based AR from a theoretical perspective, the QCAR SDK is explored in chapter 3 which embodies the practical implementation of this technology.

Chapter 4 is dedicated to discussing the ideas behind home automation and presents the chosen home automation bus system, digitalSTROM. Furthermore the method of communicating with digitalSTROM is explained in section 4.2 of the same chapter.

[3] OpenCV's website can be found here: http://opencv.willowgarage.com/wiki/

[4] More detail on DOT: http://campar.in.tum.de/personal/hinterst/index/downloads.html

[5] The Studierstube software: http://studierstube.icg.tugraz.at/

[6] Natural features are machine-recognizable optical traits of a picture. As opposed to QR codes, these natural features do not have to be created artificially but can be found in any picture that fulfills the criteria described in section 3.1.

Figure 1.1: The setup used for testing the prototype seen running on the HTC Desire. The scarf in the front was used to control the TV set and the tapestry in the back served to trigger light scenes.

The prototype software developed as the practical implementation for this bachelor thesis is explained in more detail, including the system design, in chapter 5. The paper closes with discussing the advantages and disadvantages of vision-based AR interfaces in home automation and gives an outlook on possible future developments in chapter 6.

Chapter 2

Augmented Reality & Computer Vision

As the implementation of the prototype software this paper is concerned with, relies heavily on the two technologies of Augmented Reality combined with Computer Vision, both scientific fields shall be outlined and discussed in this chapter.

To realize an AR application like the prototype software, Computer Vision is employed to optically detect objects and calculate their relative position to the user. This information is then used to augment the object by inserting virtual content (this can be any virtual object like a simple disk, a teapot, a fire-breathing dragon, etc.) at the position of the detected object. This position information is constantly updated, causing the virtual content to be fixed onto the real object, even when the object or the user moves. The aligning of the virtual object with the real object creates the illusion, that the virtual content is actually part of the real world, which is by definition [1] Augmented Reality.

The following chapter defines the terms Computer Vision and Augmented Reality and demonstrated their respective purposes.

2.1 Definition of Computer Vision

Computer vision aims to make a machine recognize and process visual input in order to "[c]onstruct meaningful descriptions of physical objects from images" as defined by Ballard and Brown [2]. CV arose as a scientific field in the 1970s and has strong ties to mathematics, geometry, and computer science as well as—to a lesser extent—to psychology, neuroscience, physics and Artificial Intelligence [18]. It is employed in a variety of domains including robotics, medical science and astronomy. Compared to the small frequency band of electromagnetic waves perceptible to the human eye, Computer Vision is able to use a significantly broader spectrum of electromagnetic radiation like infrared or x-rays and is therefore not confined to visible light but can use other radiation besides that to improve for instance the camera's image capturing process [20, Cha. 5].

In the case of our home automation prototype, recognizing an object is achieved by comparing one or more known pictures with the live camera footage of the mobile device. Assessing the frames of the camera input, the prototype determines whether one of the pictures learned a priori is currently in view and where the detected picture is in relation to the camera. Such image comparisons require the reference picture to have enough distinguishable optical features (see section 3.1 for details on the requirements of a detectable picture)

2.2 Purpose of Computer Vision

Computer vision can be used in a variety of fields to solve complex and arduous tasks. The human vision system has excellent capabilities of recognizing human faces indicating that the brain processes faces in a neurologically different way than other objects [6]. One of the application areas of Computer Vision aims to automatically recognize faces as a means of biometrical identification, people counting or for marketing/entertainment purposes. [1] In astronomy Computer Vision is for example employed to determine the distance and chemical consistency of stars by analyzing their emitted light using a spectrometer. Industrial processes use Computer Vision for quality inspection by visually examining every processed article for defects providing automatic feedback to the staff. Such automated inspections are especially valuable when the task of controlling the product would be dangerous or tedious for a human being to do [36].

Another intended purpose for Computer Vision is optical character recognition in order to transform written or printed text into digital form. Character recognition can for instance be used to automatically recognize and solve a Sudoku puzzle as shown by Google Goggles [2].

2.3 Definition of Augmented Reality

The term Augmented Reality was coined by Caudell and Mizell in 1992 and means adding virtual content to the physical world in real time [7]. An exemplary use case of AR illustrated by Wagner [39] is to have a town map lying on a table (the physical world) that is augmented by placing a landmark of the town like the model of a cathedral (the virtual content) on the map when looking through a mobile device.

[1] As demonstrated by Total Immersion: www.weareautobots.com
[2] www.google.com/mobile/goggles/

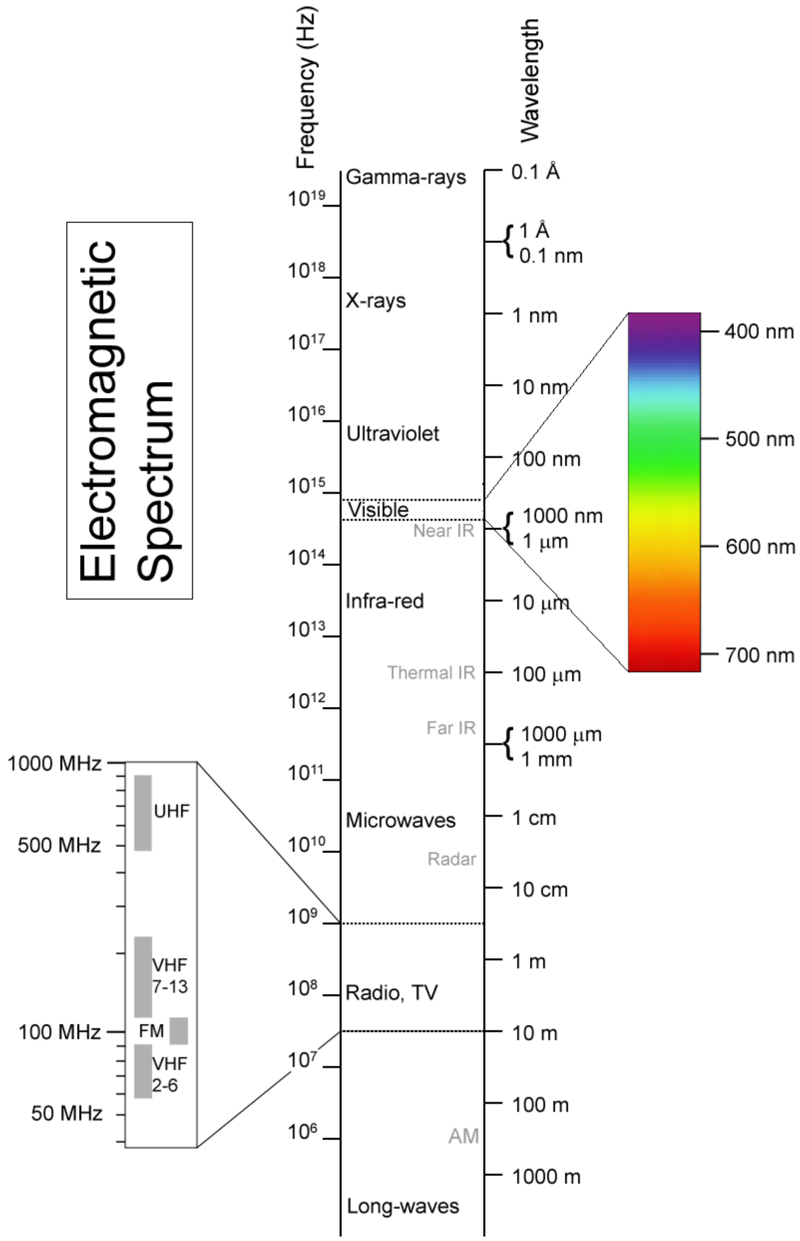


Figure 2.1: The fraction of electromagnetic radiation observable to the human vision system is narrow compared to the overall spectrum. Computer vision can employ a much wider portion. Courtesy of wikimedia.org

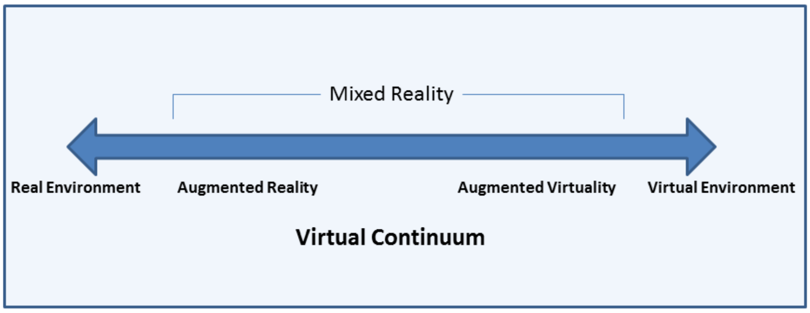


Figure 2.2: The Virtuality Continuum according to Milgram. He places AR closer to reality than virtuality as it supplements the real world rather then masking it. Figure is based on a diagram in Milgram' and Kishino's *Taxonomy of Mixed Reality visual displays* [31]

Figure 2.2 shows where AR lies in context of Milgram's Virtuality Continuum [31]. It distinguishes itself from Augmented Virtuality by placing virtual objects into the real world and not vice versa as Virtual Reality aims to do. AR seeks to make data presentation and human-computer interaction more natural and intuitive by integrating computer-generated information into the real world.

The concept of Augmented Reality was first envisioned by Sutherland [37] as early as in 1968 where he describes a head-mounted display that uses the signal of ultrasonic beacons and a mechanical tracker as means of localizing the user(See fig. 2.3 for a photo of this setup). Commonly visual objects are overlaid onto reality but supplementing audio content is feasible as well and has already been explored [3]. Recently Narumi et al. have shown that augmenting reality with taste sensations is possible using optical stimuli and a gustatory simulator that produces scented air that evokes certain flavors with the user [32].

According to Azuma [1] AR not only "[c]ombines real and virtual" objects, but is also interactive and the virtual content is "[r]egistered in 3-D" space, meaning that the virtual content seems to be fixed to the object in reality. The postulate of interactivity separates AR from non-interactive movies that embed pre-rendered objects.

Video gaming on the other hand is not AR per se because mainstream video games have not included live footage of the real world. Nevertheless, considering the change in user interaction regarding console gaming in the form of three dimensional movement detection and built-in cameras, AR games on consoles and computers are conceivable as well [3].

[3]The AR games on the handheld Nintendo 3DS are an example of emerging AR in entertainment applications.

Figure 2.3: Sutherland with his head-mounted display and the mechanical tracker hanging from the ceiling in 1968. Courtesy of wikimedia.org

2.4 Purpose of Augmented Reality

A feat of AR and the reason for being used as an interface in our home automation prototype is its ability to present computer-generated content in a natural and straightforward way to the user. By embedding the virtual content into reality users not only experience a novel way of interacting with a computer, it has also been shown that acceptance towards an AR application is high and that it requires little to no training at all from new users, regardless of their demographic [38, Cha. 7]. Modern smartphones are on the strength of their hardware and their rapid growth on worldwide markets [4] an apt host for commercial AR-based software and are said to more likely become the standard platform for future AR applications than any other device [38, Cha. 1].

AR interfaces have been used in a wide array of fields including pedestrian navigation [12] [22] [42], architecture [43], aviation [29] and surgery [27] [14] [4].

In the context of a home automation environment the smartphone and the AR application combined can be thought of as a universal remote control for the household. Moreover the magic lens metaphor [5] evoked by AR applications conveys the ease of control AR can achieve by displaying only the control elements of those household devices that are visible to the user, therefore filtering out some of the complexity a home automation system implies (Mendez et al. have explained how to use magic lenses to unclutter an interface [30]). As for these reasons an AR based application running on a smartphone can serve the purpose of facilitating the handling of a home automation system.

To our best knowledge, the research that was done in combining home automation (see chapter 4 for a closer look on home automation and smart homes) and hand-held Augmented Reality is relatively sparse compared to the effort that was put in exploring the two fields separately. In 1996 Hammond et al. [17] have built an AR interface for home automation that used a heavy-weight HMD that was located via magnetic radiation and employed infrared to communicate with household objects. Hammond et al. came to the conclusion that home automation and AR are "mutually supportive technologies" [17, Cha. 1.2].

Nijholt et al. have investigated the possibilities of creating a virtual meeting room in a smart home environment by tracking gestures of the meeting participants [33]. Yet Nijholt's project employs Virtual Reality and not Augmented Reality to achieve its goals.

[4] According to the IDC Worldwide Quarterly Mobile Phone Tracker of November 4^{th} 2010

Chapter 3

Qualcomm Augmented Reality SDK

The Qualcomm Augmented Reality Software Development Kit was released in October 2010 [1]. It is designed as a software framework for mobile developers enabling them to create their own Augmented Reality application for an Android device. In contrast to other AR efforts like Layar or Wikitude which rely upon GPS, Accelerometer and Magnetometer sensor data, the QCAR SDK employs computer Vision to determine the position of the user in relation to the focused *point of interest*. In the case of the QCAR SDK the point of interest is the trackable (see 3.1 for a definition of trackables). The two main feats of the QCAR SDK a developer can leverage are computer Vision meaning that the SDK is capable of optically detecting patterns learned beforehand and the ability to render virtual content onto these recognized trackables.

The rendering of such virtual content (e.g., a three-dimensional object) is done in respect to the relative position of the trackable. This means that virtual content appears as if it were fixed upon or above the real trackable, therefore fulfilling the requirement of Augmented Reality according to Azuma [1]. As the constant scanning and tracking of features within a camera frame is conceivably hardware demanding as well as the simultaneous rendering of multiple virtual objects, these tasks are implemented with C and C++ libraries of the QCAR SDK. These native functions can be accessed via the Java Native Interface from the Java portions of the application. Refer to chapter 5 for an illustration of an exemplary software architecture.

[1]For further information about the QCAR SDK consider its dedicated website https://ar.qualcomm.com/qdevnet/

3.1 Trackables

The terms trackable and target in the context of the QCAR SDK denote a planar [2] object that is optically detectable and trackable by the Computer Vision algorithms of the QCAR SDK. What sets the QCAR SDK apart from other software dealing with Computer Vision like the ARTag SDK or ARToolKit is its capability of recognizing natural features on a trackable — while running on a mobile platform— as opposed to the plain black and white patterns of QR-codes and fiducial markers described by Fiala in 2005 [13]. Picture 3.2 is an example of a trackable and its natural features which lie predominantly in areas of high contrast.

Qualcomm has established a web service on the SDK's website allowing developers to let their application recognize the trackables of their choice. After the developer has uploaded a well-lit, frontal picture of a trackable (one picture per trackable), the web service generates a file containing the machine-readable information necessary for the software to detect and track the depicted object. This file has to be compiled along with the rest of the source code as the current state of the employed QCAR SDK (0.10.0) does not allow learning new patterns on the fly. All the patterns that can be seen in the setup (fig. 1.1) have been processed to the above mentioned file in order to be recognized be the prototype application discussed in chapter 5.

For the object to be detectable and trackable it must bear a pattern with a sufficient number and a steady distribution of machine-recognizable *optical features*. Such optical features (also called keypoints) are found in high-contrast patterns with sharp transitions and great detail. A robust trackable has an even distribution of those features yet with few or no repetitions of patterns within the trackable [3]. The retrieval of optical features can be done by first applying a Gaussian filter to smooth the image and to reduce noise in it, as Lowe describes in his paper on SIFT [28].

Afterwards Lowe suggests to apply the Difference of Gaussian (DoG) algorithm to localize scale-invariant keypoints in the picture (Wagner et al. deemed the DoG algorithm to be too computationally expensive for mobile devices, so he replaced it with the FAST corner detector [40]). The gradient orientation around those keypoints are then calculated and a histogram of their orientations is formed [41]. So when the live video stream is analyzed and sufficient optical features match the keypoints learned beforehand (these keypoints are looked up using multiple spill trees [41, Cha. 4] which are similar to k-dimensional trees [26]), the pose of the pattern relative to the

[2]As can be seen in the setup of lampshades in fig. 1.1, those planar objects can be curved to some degree. Albeit the optical distortion of the trackable, the QCAR SDK is still capable of detecting it.

[3]A checkerboard pattern for example would make a terrible trackable: Although it has a lot of sharp transitions in respect to contrast, the repetitive nature of the pattern makes tracking impossible for the QCAR SDK.

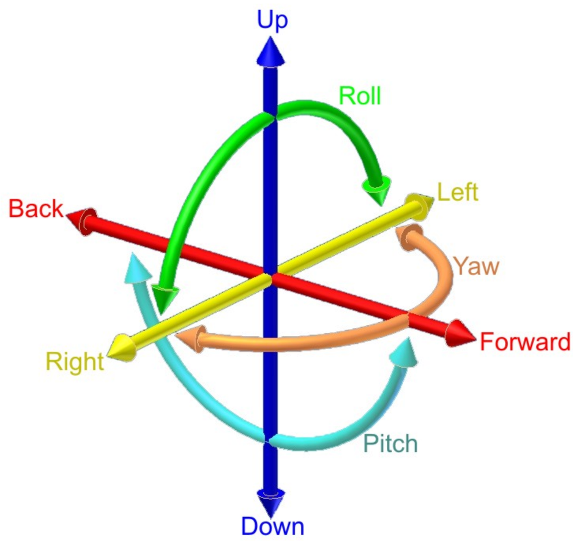


Figure 3.1: The six degrees of freedom. The pose of an object is defined by its translation and rotation relative to the observer. Courtesy of wikimedia.org.

observing camera can be estimated.

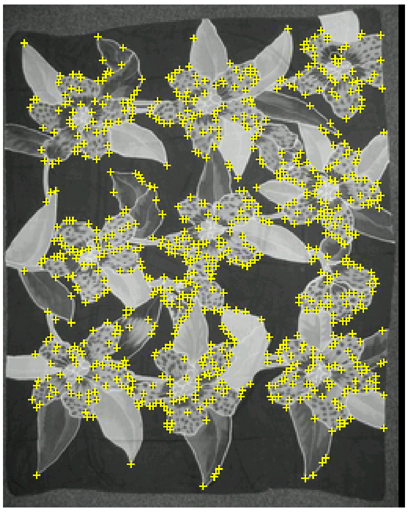
The variety of different colors or color transitions found in the picture are not taken into account as only the luminance is considered by the web service and the SDK. However, the exact method employed to generate and match this set of optical features is kept confidential by Qualcomm.

The SDK is capable of tracking a target in *six degrees of freedom* (6DOF) meaning that not only the translation of an object is registered but the rotation (i.e., roll, pitch, and yaw movement) as well. As the SDK is a 6DOF natural feature tracker it can determine by analyzing the optical features of a trackable the relative position (the pose) of the target to the mobile device. It does so by comparing the optical features of the patterns learned a priori to the current video frame. Illustration 3.1 helps visualizing the movements in six degrees of freedom.

3.2 Application of Computer Vision in the Prototype Application

In the prototype software Computer Vision is used to detect and track patterns like that of a lamp shade or that of a tapestry (see fig. 1.1 for an example). The QCAR SDK is hereby responsible for the task of computationally recognizing and tracking the position of those patterns relative to the observing camera. The maximum distance of detection between the mo-

(a) Pattern as seen by humans.



(b) Pattern as seen by the QCAR SDK.

Figure 3.2: Using Qualcomm's web service a file is generated from a PNG or JPG file that is used by the SDK to detect the picture. (a) The pattern in its natural form, (b) the generated data set of optical features the software will look for.

bile phone running the prototype software and the pattern to be detected depends on the light conditions and the size of the pattern. Yet tests with the software prototype running on the HTC Desire have shown that the QCAR SDK's Computer Vision algorithms can recognize patterns with a diameter of thirty centimeters at a distance of approximately two meters under ideal light, with no or minimal occlusion. The *tracking* of the recognized pattern is however more resistant to distance: A pattern with the same diameter can be tracked up to four meters under the same circumstances once it has been detected.

Chapter 4

Home Automation

Home automation systems aim to facilitate every-day life by connecting the devices and services used in a home via a bus system thus creating a “smart home”. By establishing communication between a set of diverse devices like the air conditioning system, the hi-fi unit or the light installation the potential of easy to operate central control and monitoring is created, among other benefits like reduced energy consumption and improved security for human residents as well as property.

As Kastner et al. [23] argue, the greatest benefit of home automation — particularly in terms of financial return— is achieved when a maximum of devices and systems are connected through a bus system. This includes the integration of sensors like air gauges or thermometers, enabling the home automation system to react to weather changes or changing light conditions.

Smart home projects like the MavHome [8] (Managing An Intelligent Versatile Home) aim for predicting the activities of its inhabitants by recording their behavior with video cameras. After analyzing the behavior of the inhabitants, the smart home system automates actions that would have to be done manually, like ordering groceries, filling the bathtub with water, or control the lights. The automation done by MavHome is transparent to the user; the user is provided no interface to control the smart home.

Davidoff et al. [10] point out that such a catering approach has serious downsides: Families living in these smart homes suffer from a sense of “losing control” and may experience the home automation system as a factor that introduces additional complexity instead of facilitating everyday life. Davidoff et al. therefore propose [10, Cha. 6] a form of home automation, that gives the user control over specific devices and services.

As for the reasons a home automation system like the Swiss digitalSTROM (described in section 4.1) that allows the user to control a single appliance as well as groups of devices, is a viable way to ensure that users remain in control of their home. Furthermore the ease of integrating the digitalSTROM communication layer between the handheld device and the home appliance (discussed in section 4.2) into the AR interface made it an attractive candidate as the employed home automation system.

There exists a variety of prototypes and implementations for home automation interfaces that can be considered related work, like the gesture-based approach of Jake et al. [21]. Corcoran and Desbonnet [9] proposed a browser-style interface running on Java that enables the user to select and control home appliances via list-like menus. Gill et al. [15] successfully experimented with a remote control that used an LCD display in a smart home environment. However, most mainstream home automation systems rely on a panel built into the home's walls or on a portable version of the same interface [1].

4.1 digitalSTROM

The home automation system digitalSTROM [2] is developed and propagated by aizo, a home automation company founded 2004 with offices in Switzerland and Germany. The concept for the digitalSTROM bus standard was conceived in 2001 and implied the connection and synchronization of household appliances in order to reduce costs caused by devices in stand-by mode. In 2007 the digitalSTROM alliance was founded as a non-profit organization with the purpose to spread the digitalSTROM system and to establish it as a home automation standard worldwide. The two organizations share team members with the president of the digitalSTROM alliance being concurrently the vice president of aizo. The digitalSTROM standard currently only offers control of lights and has no support for environment sensors (e.g., temperature, wind speed, brightness).

Although only supporting lamps at the moment [3], an additional digitalSTROM adapter allows the limited control of other electronic devices that use electrical sockets for their power supply. By interposing the adapter between the electrical socket and the plug of the device, the device's power supply can be turned off and on alternatively using the digitalSTROM standard. Using the adapter, devices can be powered on or off that are otherwise not controllable by the digitalSTROM standard like TV sets or computer monitors.

An implementation of the digitalSTROM home automation system contains the following components [4]:

[1] For example the LonWorks system by Echelon or the Gira HomeServer

[2] Information about the digitalSTROM alliance can be found on their website: http://www.digitalstrom.org/

[3] digitalSTROM's website names controlling jalousies, TV sets and hi-fi units as future possibilities.

[4] This list is firstly the result of experience gained while working with the digitalSTROM system and secondly information from aizo's dedicated homepage: http://www.aizo.com/en/products/products-overview.php

- **digitalSTROM server (dSS):** The Linux-based server provides the web interface for remotely controlling all connected digitalSTROM devices and provides information about their total power consumption in watt. It may serve as a gateway for mobile applications trying to access a digitalSTROM system.
- **digitalSTROM meter (dSM):** This unit is connected to the digitalSTROM devices using the existing 230 volt in-house power network. Its purpose is to send commands to the devices connected to it and to register the power consumption of all those devices.
- **digitalSTROM luster terminal:** A digitalSTROM luster terminal is integrated as an actuator into the power line in front of a power consuming device (e.g., a lamp). It receives commands from the meter using a high voltage chip with a unique ID (dSID) and controls the power supply to the device.
- **digitalSTROM switch terminal:** Switch terminals are designed to be built into a common 230 light switch. They relay press commands by the user to the digitalSTROM meter.

The method of how digitalSTROM utilizes the existing power network is only known to members of the digitalSTROM alliance and therefore not open to the general public. The developer team mentioned in an interview that the signaling method does not rely on frequency modulation but rather on modifying the base band near the zero crossing [5].

4.2 Communicating with digitalSTROM

The digitalSTROM server offers a built in web-interface that can be accessed via the server-specific URL. The web-interface lists all devices connected to the digitalSTROM server and displays their total power consumption. In order to control digitalSTROM devices remotely, HTTP-requests are sent to the server which relays the commands to the meter which again sends the instruction to the individual devices using the unique chip ID of the digitalSTROM luster terminals. Fig. 4.1 shows a schematic of a standard digitalSTROM setup that reflects the above mentioned order of communication.

To send a HTTP-request from a mobile device to the server, the mobile device has to share the same IP subnet as the digitalSTROM server. To ensure this, a router with configured IP masks may be used. See fig. 4.2 for a connection diagram of the prototype setup.

[5] The interview in German from 2007 can be found here: http://heise.de/-280047

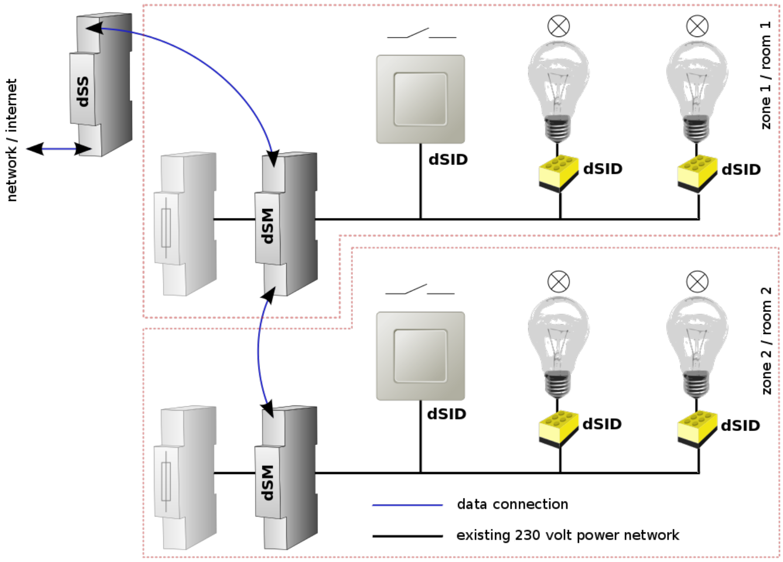


Figure 4.1: A schematic illustration depicting a typical digitalSTROM setup used in a home. Based on a schematic on http://de.wikipedia.org/w/index.php?title=Datei:DigitalSTROM.svg

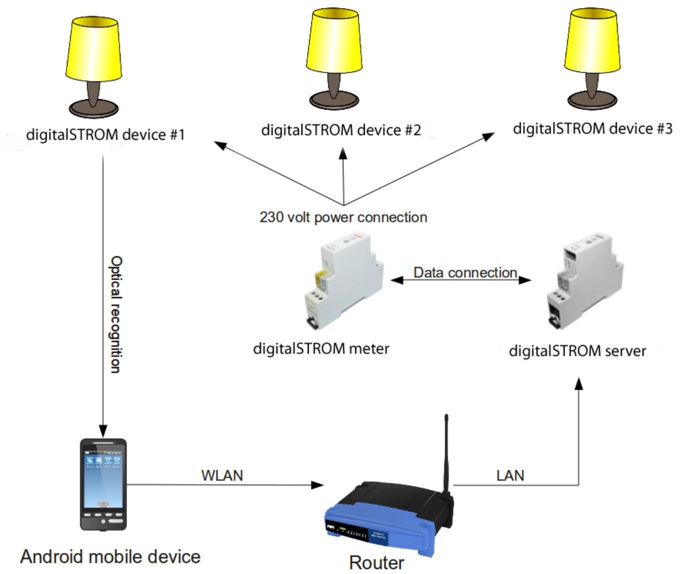


Figure 4.2: This diagram shows the components used in the prototype setup along with their corresponding connections.

The HTTP-request to control a lamp contains its unique device ID followed by a parameter that represents the action (i.e., on, off, brighter, darker). The structure for nearly all (see exception below) commands is **server URL + basic command + device ID + action**. The following Java code excerpt from the prototype software shall demonstrate how to power on a digitalSTROM-ready lamp.

```
// The digitalSTROM server URL
String SERVER_URL = "http://192.168.0.10/";

// The basic command rudiment that is followed
// by the device ID and the action parameters
String BASIC_CMD = "api/basic?class=5&function=17&selector=";

// The ID of the lamp
String LAMP_ID = "3504175fe000000000000a5c";

// The parameter for powering on the light
String LIGHT_ON = "&parameter=14";

String cmd = SERVER_URL + BASIC_CMD + LAMP_ID + LIGHT_ON;

// Send the HTTP request using an Android WebView
new WebView(applicationContext).loadUrl(cmd);
```

As an exception to the above shown command structure and an example on how to get information from the server, this HTTP-request returns the total power consumption of all the devices connected to a server:

```
http://192.168.0.10/api/basic?class=0\&function=1.
```

The IP address in the command is the address of the digitalSTROM server that receives the command and returns the currently consumed power in watt.

Chapter 5

The Prototype

To explore and demonstrate the synergies that can result from combining the two technologies of home automation and Augmented Reality, a software prototype was developed and tested using lamps seen in the setup depicted in fig. 1.1. Furthermore the adapter described in 4.1 was tested by controlling a PC monitor and a TV set respectively.

The prototype combines the control features of a mobile home automation application with a vision-based Augmented Reality interface. The QCAR SDK enables the perceptual GUI (as outlined by Landay et al. [24]) by letting the mobile device recognize patterns learned a priori one at a time. Those patterns are applied as lampshades to the digitalSTROM-ready lamps.

Once a known pattern is in focus, the lampshade is marked with a green disk and the user is presented control elements that correspond to the type of device associated with the recognized pattern (see below for a list of the different types of devices). By pressing such a button an HTTP-request is sent to the digitalSTROM server that relays the command to the digitalSTROM device as explained in section 4.2.

For the prototype application three different types of control sets have been implemented:

- **Lamp control:** When the user focuses the mobile phone's camera on a pattern (such as the lampshades in fig. 1.1) that has been associated with a digitalSTROM-ready lamp, three buttons are displayed. One of these buttons serves as the light switch. The shape of this button is a light bulb and its color represents the state of the real lamp it is associated with: The button shines bright when the lamp is powered on, and is translucent when the corresponding lamp is turned off.

  As the digitalSTROM bus system offers no backchannel for individual devices, the on/off state is not queried by the digitalSTROM server but rather managed internally by the prototype application. Furthermore the lamp control GUI provides two dimming buttons for increasing

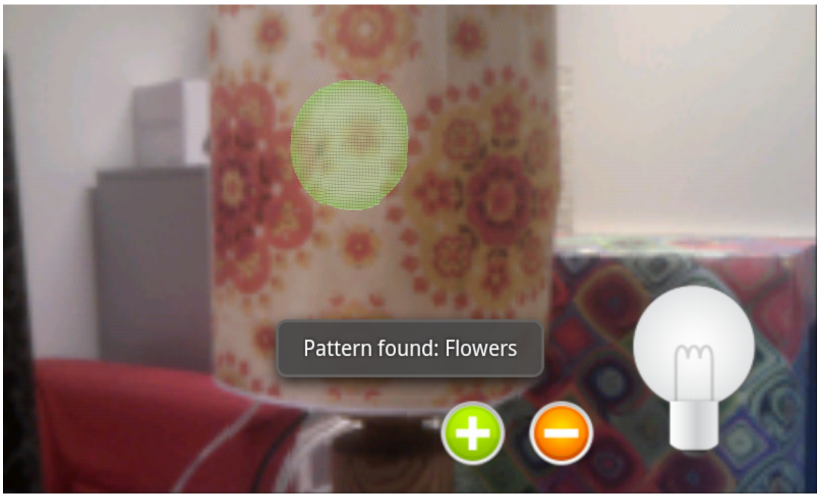


Figure 5.1: The currently focused object is marked with a transparent, green disc. A set of buttons dependent on the type of the object (in the example shown it's a lamp) is presented to the user alongside a short message (acoustical as well as optical) with the name of the recognized object.

and decreasing the brightness of a lamp. See fig. 5.1 for a screenshot depicting a detected lamp including the green marker.

- **Scene control:** This control type is shown when the user focuses on a pattern that is linked to a light scene. A single button appears on the phone's screen that lets the user activate the light scene associated with the pattern. A light scene is defined as a number of lamps being powered on with a given level of brightness for specific situations or activities in a room (e.g., reading, watching TV).
- **Adapter control:** The adapter control consists of a power on/off button that represents the state of the device connected to the adapter. The adapter control is shown when focusing a pattern associated with the digitalSTROM adapter described in section 4.1.

All these different types of control elements provide haptic feedback to the user when pressing a button and are responsive to the physical orientation of the mobile device, permitting usage both in landscape and portrait mode.

5.1 Features

Apart from letting the user control digitalSTROM-ready devices using a perceptual Augmented Reality GUI, the prototype application sports a set of complementary features. The common purpose of the features is facilitating

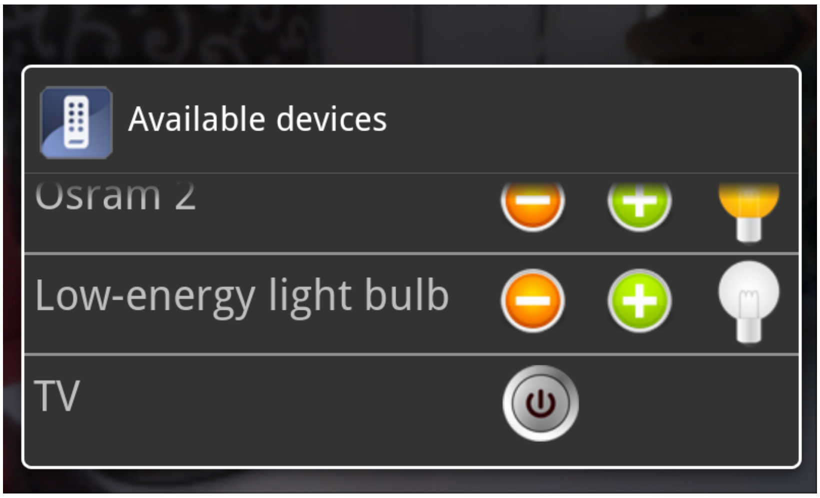


Figure 5.2: The list view provides the user a means to control the connected digitalSTROM devices without relying on an optical connection between phone and the controllable device.

the usage of an AR-based home automation interface. This section shall give a description of each of those features.

5.1.1 List View

By pressing the MENU button of the Android phone the user can select the list view. It provides a list of all controllable digitalSTROM devices along with their corresponding buttons and serves as an alternate user interface. As the AR interface is vision-based, bad lighting and object occlusion can hinder or prevent the prototype application from recognizing the device the user wants to control. In such a case the simple list view has the benefit of being a reliable way to control the devices. A similar reason to use the list view instead of the AR view is when there is no line of sight from the phone to the device because the user is in another room, making an optical recognition impossible. The advantage of the list view is that users do not have to establish an optical connection between the phone that is running the prototype application and the device they want to control. As a downside the list view gets cluttered when the number of controllable devices rises and lacks furthermore the intuitive way of the AR interface to control a device.

5.1.2 Power Statistics

The prototype software collects information about the power consumption of all the devices connected to the digitalSTROM server from the moment the application starts. When selecting the Statistics option in the prototype's

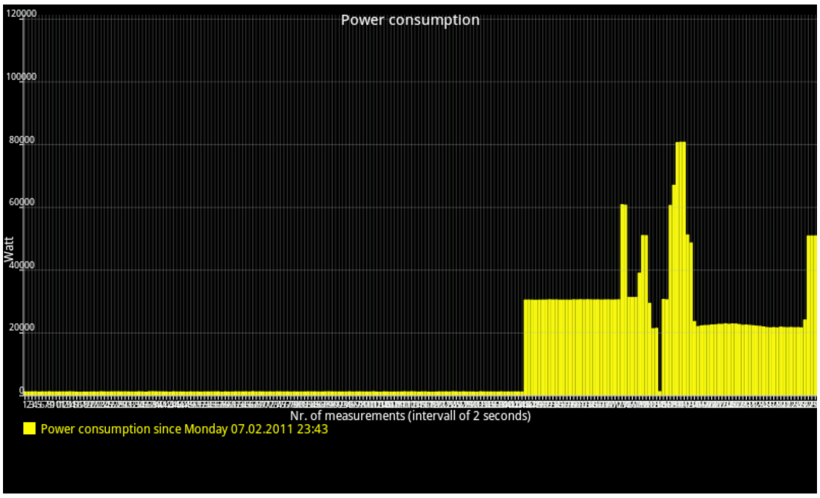


Figure 5.3: The statistics feature gives information about the power consumption of all connected digitalSTROM devices over time.

option menu a bar graph is presented to the user displaying the power consumption in watt over time. This way the user can for example identify the amount of power consumed when all devices are on standby or how the wattage has changed over time. The chart engine used by the prototype application for displaying the graph is AChartEngine by 4ViewSoft.

5.1.3 Voice

When a trackable is detected by the prototype application, the name of the trackable is read out loud using the Android Text-to-Speech API. The prototype software saves the last detected trackable internally to avoid the phone repeating the name of the same trackable when redetecting it. The volume of the voice reading the names can be controlled by clicking the hardware plus and minus buttons on the phone. The feature can be toggled using the option menu. This feature serves the purpose of strengthening the user's experience of having a seeing remote control that is in addition able to speak the names of the patterns it recognizes.

5.1.4 Shaking Shutdown

Shaking of the device causes the prototype software to send the command to turn off all connected digitalSTROM devices. This feature demonstrates that gestures can be a viable and convenient way to control a home automation system (as Jake et al. [21] have shown before). Moreover it works as a reset mechanism when the state of a device (on or off) and its GUI representation do not match anymore due to the missing backchannel of the digitalSTROM server as described in chapter 5.

5.2 System Design

The system design of the prototype software was partially predetermined by making use of the QCAR SDK's sample application *ImageTargets* and *Dominoes*. The computing intense tasks of detection and tracking of a trackable as well as rendering the augmentation onto the trackable is realized in C++ code employing the Native Development Kit provided by Android.

To allow communication between the native classes and the Java portions of the code, the prototype software makes use of the *Java Native Interface* (JNI). The following description and the figure 5.4 below shall exemplarily illustrate the program flow from initially detecting a trackable to eventually controlling a digitalSTROM device.

When a trackable is detected it is first registered in the native ImageTargets class which —using the JNI— invokes a method of the Java object ImageTargetsRenderer sending the name of the trackable as a parameter. After checking with which sort of digitalSTROM device this trackable is associated, the ImageTargetsRenderer then relays a thread-safe message to the GUI manager object that is designed after the singleton pattern [1].

The GUI manager chooses the appropriate control elements depending on the device class and displays them to the user (as described in chapter 5). It reacts to user input and calls the methods of the different digitalSTROM device objects such as on/off or dimming. Ultimately these device objects send the corresponding HTTP request to the digitalSTROM server.

Because all types of digitalSTROM devices (i.e., lamps, adapters, scenes) share certain attributes like the need to send HTTP requests, all specific digitalSTROM devices inherit from the abstract digitalSTROM device class. These classes have been grouped in their own package for better organization.

[1] The singleton design pattern applied in the system design (also called Initialization On Demand Holder idiom) and the inventor's deliberations behind it are described here: http://www.cs.umd.edu/~pugh/java/memoryModel/jsr-133-faq.html

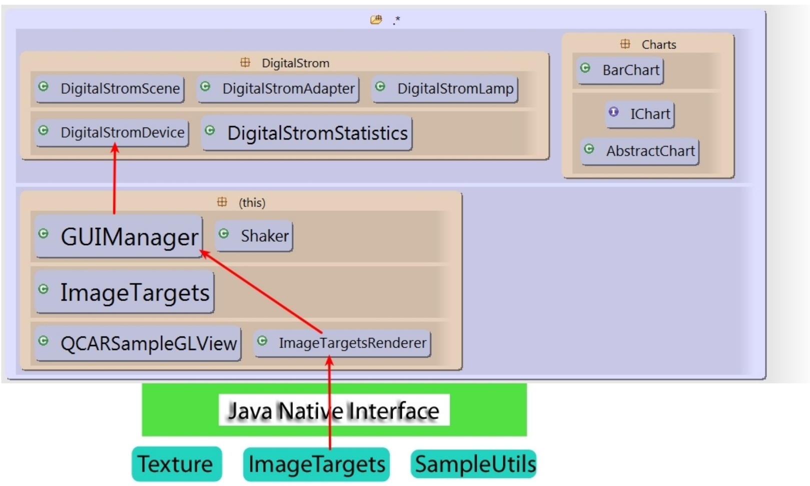


Figure 5.4: The program flow from detecting a trackable to controlling a digitalSTROM device. The size of the blocks representing the classes is proportional to their number of member variables.

Chapter 6

Conclusions and Outlook

The software prototype presented in this paper has shown that vision-based Augmented Reality on smartphones has the potential to serve as the technological foundation for perceptual interfaces for home automation. The statements and opinions gathered at the university's open house day indicated firstly that a strong interest in intuitive interfaces and interaction methods exist in the field of home automation and secondly that AR on smartphones can be a viable way to build this kind of interfaces.

The users' familiarity with smartphones facilitated learning the usage of the prototype software while the rather unconventional area of application positively astonished users. The pattern detection capabilities combined with the home automation feature of the 'seeing remote control' mark certainly a unique feature, but the synergy stemming from other characteristics like the Voice feature 5.1.3 and the Shaking Shutdown 5.1.4 shows the smartphone as a multi-purpose device to good advantage. It allows the user to discover control elements in a natural and vision-based way by using the smartphone as a 'magic lens' [5].

It provides not only visual feedback when finding a controllable device but also acoustically identifies the detected pattern by reading its name out loud. Furthermore the mobility of smartphones in general and Android's APIs in specific allow the use of simple and easy to learn gestures like the aforementioned shaking to control the devices. Should these unconventional interface methods fail, there is always the option of falling back to the list of controllable devices. All these features that interact with the user on multiple levels of perception improve the quality of a home automation interface compared with the current standard of list-based user interfaces.

Such a method of optical recognition for a perceptual user interface has its limitations of course. The application assumes that every device the user wants to control looks significantly different than any other device i.e. no duplicates of a device exist in the household. Such a constraint may be tolerable in a household up to family size but renders the application unusable in environments where many identical devices exist e.g. a company office.

It is worth mentioning that in scenarios like the latter an approach of mixed technologies can be successful: Assuming that the company has a wireless *LAN* in use (or even more than one) the application can use *Fingerprinting* [25] to give an approximate position which limits the number of possible devices to be identified. Consequently the objects can be detected through optical recognition. However, this hybrid strategy is mentioned here so that it might serve as inspiration for thought but it is not discussed further on.

What has yet to be achieved is a way to let the users define what object to associate with which control functions (e.g., dim buttons, power switch). Ideally all electronic devices in a household are connected via a universal bus system. The user would train the interface to recognize household devices by filming them from multiple angles, distances, and under varying light conditions. After the software has gathered sufficient optical features of the object, the user would associate it with a set of control functions appropriate to the type of device.

Davison et al. have already shown that reliable feature extraction of three-dimensional objects using solely one camera is possible, yet on computers that exceed the current performance of smartphone processors [11]. To improve the optical recognition efficiency the application may continue to collecting visual information about an object after it has already been detected.

Once these hardware requirements are met and given that the used home automation system allows dynamically adding controllable devices, the seeing remote control will outgrow its prototype stage.

Bibliography

[1] Azuma, R. T.: *A survey of augmented reality*. Presence: Teleoperators and Virtual Environments, 6(4):355–385, Aug. 1997.

[2] Ballard, D. and C. Brown: *Computer Vision*. Prentice Hall, 1982.

[3] Bederson, B. B.: *Audio augmented reality: a prototype automated tour guide*. In *Conference companion on Human factors in computing systems*, CHI '95, pp. 210–211, New York, NY, USA, 1995. ACM.

[4] Bichlmeier, C., S. M. Heining, M. Feuerstein and N. Navab: *The Virtual Mirror: A New Interaction Paradigm for Augmented Reality Environments*. IEEE Transactions on Medical Imaging, 28:1498–1510, 2009.

[5] Bier, E. A., M. Stone, K. Pier, W. Buxton and T. D. Derose: *Toolglass and magic lenses: The see-through interface*. In *Annual Conference on Computer Graphics*, vol. 27, pp. 73–80, 1993.

[6] Bigun, J.: *Vision with Direction. A Systematic Introduction to Image Processing and Computer Vision*. Springer, 2006.

[7] Caudell, T. P. and D. W. Mizell: *Augmented reality: an application of heads-up display technology to manual manufacturing processes*. Proceedings of the TwentyFifth Hawaii International Conference on System Sciences, pp. 659–669, 1992.

[8] Cook, D. J., M. Youngblood, E. O. Heierman, III, K. Gopalratnam, S. Rao, A. Litvin and F. Khawaja: *MavHome: An Agent-Based Smart Home*. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, PERCOM '03, pp. 521–, Washington, DC, USA, 2003. IEEE Computer Society.

[9] Corcoran, P. M. and J. Desbonnet: *Browser-style interfaces to a home automation network*. IEEE Transactions on Consumer Electronics, 43(4):1063–1069, 1997.

[10] Davidoff, S., M. K. Lee, C. Yiu, J. Zimmerman and A. K. Dey: *Principles of smart home control*. In *Proceedings of Ubicomp 2006*. Springer, 2006.

[11] Davison, A. J., I. D. Reid, N. D. Molton and O. Stasse: *MonoSLAM: Real-Time Single Camera SLAM*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):1052–1067, 2007.

[12] Feiner, S., B. Macintyre, T. Höllerer and A. Webster: *A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment*. In *International Symposium on Wearable Computers*, vol. 1, pp. 74–81, Washington, DC, USA, 1997. IEEE Computer Society.

[13] Fiala, M.: *ARTag, a Fiducial Marker System Using Digital Techniques*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2:590–596, 2005.

[14] Fuchs, H., M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake and A. A. Meyer: *Augmented Reality Visualization for Laparoscopic Surgery*. In *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '98, pp. 934–943, London, UK, 1998. Springer-Verlag.

[15] Gill, K., S.-H. Yang, F. Yao and X. Lu: *A ZigBee-based home automation system*. pp. 422–430. IEEE Computer Society Press, 2009.

[16] Haigh, K. Z. and H. Yanco: *Automation as Caregiver: A Survey of Issues and Technologies*. In *Proceedings of the AAAI-02 Workshop "Automation as Caregiver"*, pp. 39–53, 2002. AAAI Technical Report WS-02-02.

[17] Hammond, J. C., P. M. Sharkey and G. T. Foster: *Integrating Augmented Reality with Home Systems*. In *Proc. 1st International conference on disability, virtual reality and associated technologies ECDVRAT '96*, pp. 57–66, 1996.

[18] Hartley, R. and A. Zisserman: *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 ed., April 2004.

[19] Hinterstoisser, S., V. Lepetit, S. Ilic, P. Fua and N. Navab: *Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects*. 2010.

[20] Jahne, B. and H. Haussecker: *Computer vision and applications: a guide for students and practitioners*. Academic Press, Inc., Orlando, FL, USA, 2000.

[21] Jake, T. S., J. Auxier and D. Ashbrook: *The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring*. In *International Symposium on Wearable Computing*, pp. 87–94, 2000.

[22] Kalkusch, M., T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann and D. Schmalstieg: *Structured Visual Markers for Indoor Pathfinding*. In *Proceedings of the First IEEE International Workshop on ARToolKit*. IEEE, 2002.

[23] Kastner, W., G. Neugschwandtner, S. Soucek and H. M. Newman: *Communication Systems for Building Automation and Control*. Proceedings of the IEEE, 93(6):1178–1203, June 2005.

[24] Landay, J. A., J. Hong, S. Klemmer, J. Lin and M. Newman: *Informal PUIs: No Recognition Required*. pp. 86–90. AAAI Press, 2002.

[25] Li, B., Y. Wang, H. K. Lee, A. Dempster, C. Rizos, B. Li, Y. Wang and A. Dempster: *A New Method for Yielding a Database of Location Abstract Fingerprints in WLAN*, 2005.

[26] Liu, T., A. W. Moore, E. Gray and K. Yang: *An investigation of practical approximate nearest neighbor algorithms*. pp. 825–832. MIT Press, 2004.

[27] Livingston, M. A., W. F. Garrett, G. Hirota, M. C. Whitton, E. D. Pisano and H. Fuchs: *Technologies for augmented reality systems: realizing ultrasound-guided needle biopsies*. In *Annual Conference on Computer Graphics*, pp. 439–446.

[28] Lowe, D. G.: *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, 60:91–110, November 2004.

[29] McCarty, W. D., S. Sheasby, P. Amburn, M. R. Stytz and C. Switzer: *A Virtual Cockpit for a Distributed Interactive Simulation*. IEEE Computer Graphics and Applications, 14:49–54, January 1994.

[30] Mendez, E., D. Kalkofen and D. Schmalstieg: *Interactive context-driven visualization tools for augmented reality*. 2006 IEEEACM International Symposium on Mixed and Augmented Reality, pp. 209–218, 2006.

[31] Milgram, P. and F. Kishino: *A Taxonomy of Mixed Reality Visual Displays*. IEICE Transactions on Information Systems, E77-D(12), Dec. 1994.

[32] Narumi, T., S. Nishizaka, T. Kajinami, T. Tanikawa and M. Hirose: *Augmented reality flavors: gustatory display based on edible marker and cross-modal interaction*. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pp. 93–102, New York, NY, USA, 2011. ACM.

[33] Nijholt, A., J. Zwiers and J. Peciva: *Mixed reality participants in smart meeting rooms and smart home environments*. Personal and Ubiquitous Computing, 13(1):85–94, January 2009.

[34] Reitmayr, G. and T. W. Drummond: *Going out: Robust Tracking for Outdoor Augmented Reality*. In *Proc. ISMAR 2006*, pp. 109–118, Santa Barbara, CA, USA, October 22–25 2006. IEEE and ACM, IEEE CS.

[35] Rolland, J. P., Y. Baillot and A. A. Goon: *A Survey of Tracking Technology for Virtual Environments*, 2001.

[36] Shapiro, L. G., G. C. Stockman, L. G. Shapiro and G. Stockman: *Computer Vision*. Prentice Hall, Jan. 2001.

[37] Sutherland, I. E.: *A head-mounted three dimensional display*. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), pp. 757–764, New York, NY, USA, 1968. ACM.

[38] Wagner, D.: *Handheld Augmented Reality*. PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, October 2007.

[39] Wagner, D., T. Langlotz and D. Schmalstieg: *Robust and unobtrusive marker tracking on mobile phones*. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pp. 121–124, Washington, DC, USA, 2008. IEEE Computer Society.

[40] Wagner, D., G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg: *Pose tracking from natural features on mobile phones*. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pp. 125–134, Washington, DC, USA, 2008. IEEE Computer Society.

[41] Wagner, D., G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg: *Real-Time Detection and Tracking for Augmented Reality on Mobile Phones*. IEEE Transactions on Visualization and Computer Graphics, 16:355–368, 2010.

[42] Walther-Franks, B. and R. Malaka: *Evaluation of an Augmented Photograph-Based Pedestrian Navigation System*. In *Proceedings of the 9th international symposium on Smart Graphics*, SG '08, pp. 94–105, Berlin, Heidelberg, 2008. Springer-Verlag.

[43] Webster, A., S. Feiner, B. Macintyre, W. Massie and T. Krueger: *Augmented Reality in Architectural Construction, Inspection, and Renovation*. In *In Proc. ASCE Third Congress on Computing in Civil Engineering*, pp. 913–919, 1996.