Abstract
This work presents a system that makes use of the Microsoft Kinect to enable Point&Click interaction for the control of appliances in smart environments. A backend server determines through collision detection which device the user is pointing at and sends the respective control interface to the user’s smartphone. Any commands the user issues are then sent back to the server which in turn controls the appliance. New devices can either be registered manually or using markers such as QR codes to identify them and get their position at the same time. The video demonstrates the interaction concept and our technical implementation.

Author Keywords
Ambient Control, Smart Environments, Assistive Technology, Remote Control, Kinect, Universal Usability, Wireless Computing, Home Automation

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms
Design, Experimentation, Human Factors
Introduction and Related Work
As we have more and more devices in our work environments and homes today, controlling them all can become increasingly overwhelming. There are existing solutions for this problem, e.g. using programmable universal remotes or smartphones to control different devices. However, such approaches usually require that the user somehow selects the interface of the device that he wants to control, e.g. from a list. With an increasing number of devices, the efficiency of this explicit interaction decreases. This work presents a way of re-introducing the paradigm of pointing at an appliance and clicking a button to control it.

This Point&Click mode of interaction – especially for devices that don’t possess a user interface of their own and/or are in spatial distance – was already proposed for the universal control in smart environments before smartphones existed [1]. Simultaneously, efforts towards developing a standard for a Universal Remote Console, “a personal device that can be used to control any electronic and information technology device”, were started [8]. The authors of [2] present an interaction model specifically designed for mobile devices to act as mediators between users and smart objects. A user study [5] has shown that using phones to control smart devices can greatly improve usability, especially with highly complex tasks.

Similar concepts have been presented in the past. MagicPhone is “a mobile phone that can sense what you are pointing to and can act as a physical ubiquitous interaction device in real world” [7]. However, the system solely uses the accelerometer and magnetometer to get phone orientation. To infer what the user is pointing at, accurate phone location is needed which is hard to come by in indoor environments. Furthermore, changes in the environment may distort the recognition. Overall, the publication is rather vague on implementation details and puts more emphasis on possible interactions.

The UbiControl system [4] uses laser-pointers to select devices and control them with PDAs. All controllable devices need to be equipped with sensor nodes that carry solar cells which are used to recognize the pulse sequences emitted by the laser pointer. The devices then communicate to the server that they have been targeted. PIControl [6] uses handheld pico projectors in order to control physical devices. In this approach, the projection simultaneously serves as presentation of the user interface and as communication channel to send commands. While the system does not require any infrastructure, the devices that are to be controlled need to be augmented with – albeit inexpensive – sensor units carrying one or several photo sensors. Instead of the device, the user has to select the appropriate interface that he wishes to use. Through the projection it is also possible that multiple smart devices in close physical proximity are involuntarily covered and thus controlled at the same time.

Similar to our own approach, the Multimodal Interface for Smart Objects (MISO) [3] makes use of the Kinect for pointing recognition. Appliances are controlled by looking at them and pointing simultaneously while performing one out of six gestures. The advantage of MISO is that no remote is required at all, which makes control very natural. On the other hand the authors point out, that in order to reduce the number of false positives, either training data from a large number of individuals or training on the specific user is required. Furthermore, the gestures limit the control possibilities to a fairly small number of commands and the system cannot resolve ambiguity, e.g. in case two devices are in the same line of sight.
**Point & Control Concept**

We propose to use a server located in the same smart environment that the user wants to control. It is reasonable to assume that this server can store information about connected devices, their control interface, their position and is able to locate an user and track his or her pointing direction. With this information it provides two services: 1) Deliver the remote control of the appliance that the user is pointing at, and 2) abstract the different control interfaces with generic commands. Theses services can be accessed via different personal devices e.g. a smartphone. With this architecture (Figure 3), no prior knowledge about the appliances in the smart environment is needed to interact with them but it is still possible to access all functionality. It is also not necessary to augment the devices.

**Implementation**

The **Kinect** depth camera is used to locate and track the users’ individual body parts and posture. To associate a tracked person with the smartphone in his hand we use a dedicated registration step, where a given pose is matched with the camera data. Currently the user simply raises her hand and touches the screen of the phone (Figure 1). In this way multiple persons can instantly and independently start controlling registered appliances in the same space. These devices do not necessarily have to be in the **Kinect**’s field of view, only the controlling user has to be. Whenever the user points to a registered object and clicks (first click, Figure 2), the interface for the particular device is sent to the phone in **HTML** format. With this remote control the user then issues commands (second click) to the server, which relays them accordingly. A pointing vector is calculated from the position of the user’s hand and elbow joints (Figure 4). Using collision detection, the current target appliance is determined.

If the pointing selection is ambiguous, the system returns a small list with the subset of devices for clarification. This is necessary when multiple objects are positioned close to each other or in the same line of sight.

Currently, anything that has or can be extended with network control functionality can be controlled. The prerequisite is, that an appropriate interface description is present and that the device’s position is registered. The user interface and the control logic is implemented in the backend system via plug-ins which can be added and changed at runtime. The registration of a device can happen in one of two ways: Either its position is entered manually, or if the device is in the **Kinect**’s field of view, the user can use the application to create a new target at the position of her hand. With the help of markers attached to the devices – e.g. QR codes – the identity and location can be acquired in a single step.
The server uses a Kinect to continuously track the users' skeletons and calculate their aiming direction.

Conclusion and Future Work

We have shown a novel way to allow users to take control of networked devices located in a smart environment. The known Point&Click paradigm was extended to move all services needed for intuitive interaction to a server in the same space. In future work we plan to overcome some of the usability limitations that we could identify in first evaluations. One of them is the explicit registration step required before the interaction. Through the use of accelerometer data, it should be possible to correlate smartphone movement and Kinect data in a more intuitive way. The sensors could also be used to enable more natural gesture based interaction with the controlled appliances. The second issue is the (re-)localization of appliances in the smart space. We are currently experimenting to use the same Point&Click interaction to infer device locations. In theory the user should be able to specify the position of a new or moved device by targeting it from two to three different places, even if it is not in the Kinect's field of view. Other possible technical improvements of the system are porting the application to a platform-independent implementation relying on HTML5 and adding support for other control modalities, such as IR or Bluetooth, through the server. Lastly, employing multiple Kinects would allow for a larger coverage of the smart environment, reducing blind spots.

References