Phone-Pointing Remote App: Using Smartphones as Pointers in Gesture-Based IoT Remote Controls

Ilan Kirsh[1][0000−0003−0130−8691] and Heinrich Ruser[2][0000−0001−6929−3276]

[1] The Academic College of Tel Aviv-Yaffo, Tel Aviv, Israel, kirsh@mta.ac.il
[2] Bundeswehr University Munich, Neubiberg, Germany, Heinrich.Ruser@unibw.de

Abstract. Remote control mobile applications for operating Internet of Things (IoT) devices using smartphones are commonly based on a touch user interface. The effort of using such apps is often disproportionate to the simplicity of carrying out the actions manually. For example, turning a light on or off via menus and forms of a standard remote app might not be very convenient. A voice user interface, while easier to use, gives rise to other issues, including user privacy and distracting others nearby. This paper proposes a new type of universal IoT remote control applications for smartphones: phone-pointing remote apps. Using a phone-pointing remote app, users can physically point their smartphones at IoT devices to select them, and operate them via movement gestures, without needing to turn on the phone screen or talk, and with no need for any additional hardware. This new approach provides a unique combination of advantages. It is simple, intuitive, fast, and voiceless. Instead of using the touchscreen or the microphone as the input source, phone-pointing remote apps will use a combination of standard smartphone sensors, including the GNSS sensor, Wi-Fi scanner, Bluetooth receiver, camera, barometer, magnetometer, g-force meter, accelerometer, and gyroscope. An analysis of the proposed model in light of relevant results from related studies provides positive preliminary indications regarding the feasibility of this novel approach.

Keywords: Universal Wireless Remote Control · Smartphone Sensors · Hand-Gestures · Internet of Things (IoT) · Smartphone Orientation · Azimuth · Pitch · Android · iPhone

1 Introduction

Wireless remote controlling has come a long way from the release of the first TV remote control, the Zenith Space Command [1], in 1956 to modern IoT remote controls [4]. Voice-controlled virtual assistants such as the Amazon Echo Alexa, Google Home Assistant, and Apple HomePod Siri are widely used nowadays to control IoT-enabled devices, such as lights, thermostats, and alarms [3]. Remote controls are also available as smartphone apps [2, 3]. Voice-controlling is associated with some potential issues regarding security and privacy [3, 11] and user

In Proceedings of the 23rd International Conference on Human-Computer Interaction (HCI International 2021), CCIS, Springer, 2021. The final authenticated version is available online at https://doi.org/10.1007/978-3-030-78642-7_3

acceptance [9]. Therefore, a voiceless operation based on a standard touchscreen user interface may be preferred in certain circumstances. However, touchscreen-based remote apps also have usability issues. For example, controlling the lights using a standard remote control app with a touch user interface often requires turning on the smartphone screen, launching the remote app, navigating through menus, and finally selecting the operation. This process may be considered overly complicated as the required user effort is disproportionate to the simplicity of such a routine operation.

We propose a new type of smartphone IoT remote control: phone-pointing remote apps. Unlike touch-based remote apps, phone-pointing remote apps can be used with the phone's screen turned off. As illustrated in Figure 1, users select devices by pointing their smartphone at them, and then use movement gestures, while holding the smartphone, to operate the devices. Unlike existing gesture-based pointing remotes that are based on dedicated hardware, such as a hand-held infrared projector and an appropriate receiver at every device [8], which increases costs and restricts their availability, the proposal made in this paper is based on built-in smartphone functionalities, and does not require additional hardware. Phone-pointing remote apps will use standard smartphone sensors for input, including (a) the GNSS (Global Navigation Satellite System, commonly referred to as GPS, based on the most widely used GNSS system) and the Wi-Fi receiver to evaluate the current user location, (b) the magnetometer and the g-force meter (or the accelerometer) to identify 3D pointing directions during device selection, and (c) the accelerometer and the gyroscope to recognize movement gestures while holding the smartphone.

In this paper, we present our vision for phone-pointing remote apps. We define a model based on Machine Learning (ML), which can be used to implement remote control apps of this type. Then we describe relevant technical details and discuss the opportunities as well as the challenges. Results from various related studies provide positive preliminary indications that this novel approach to wireless IoT remote controlling using smartphones is promising, as discussed in this paper.

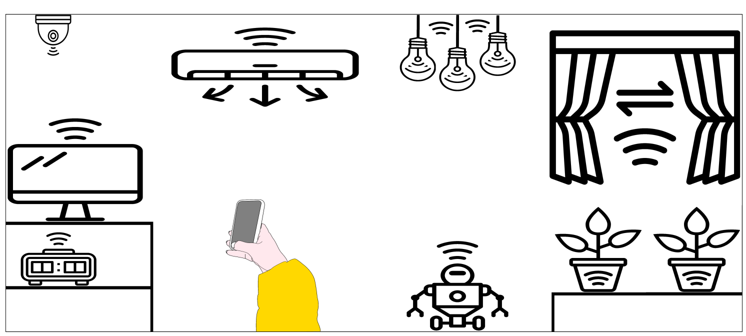

Fig. 1. Controlling an air conditioner using a phone-pointing IoT remote control app

2 Toward Phone-Pointing Remote Apps

Modern remote apps use IoT APIs, such as Google Assistant API, Apple HomeKit, Amazon Alexa Home Skills API, and others, to communicate with compatible IoT devices, as illustrated in Figure 2. The arrows in Figure 2 indicate the main directions of action. To operate IoT devices, users select devices and operations by using either touch or voice user interfaces. Selections are translated by the remote app into instructions that are sent to the devices through the IoT APIs, usually over the local network via Wi-Fi. Information flows also in the opposite direction, where IoT APIs are used to collect information about available devices, which is provided to the user by the app. The main responsibility of an IoT remote app is to provide a user interface that bridges the gap between the users and the relevant IoT APIs and devices.

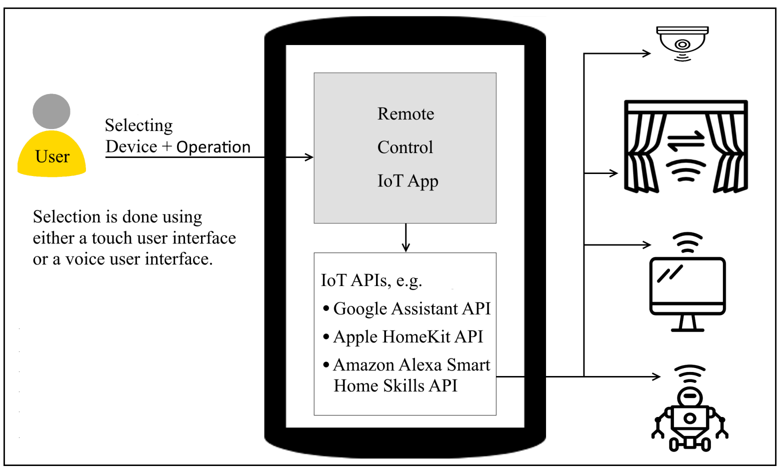

Fig. 2. General structure of IoT remote control apps

For the sake of simplicity, we define the required input to operate a device to be of the form $(device, operation)$. For example, to increase the television volume, the device is a specific television, and the operation is *volume-up*. Both selections, the device and the operation, can be chosen from menus and forms in a touch-based user interface or by speaking to a virtual assistant in a voice-based user interface. A phone-pointing remote app, on the other hand, should be able to function with the smartphone's screen and microphone turned off, as other sensors are used to collect the input that is needed for inferring the user's $(device, operation)$ selection.

Recognition of the user selection can be split into two distinct subproblems: recognition of the selected device and recognition of the selected operation. Assuming prior knowledge about the location of the IoT devices in a specific environment, an app may be able to identify which device is selected based on the current location of the smartphone (and the user) and its pointing direction. Therefore, the subproblem of recognizing the selected device can be further split into two more fundamental problems (recognition of current location and recognition of pointing direction). The division of our problem into three basic subproblems is illustrated at the top half of Figure 3 (labeled 'Required Input'). The bottom half of Figure 3 (labeled 'Phone Sensors') lists standard smartphone sensors that can be used in solving each one of these three basic subproblems.

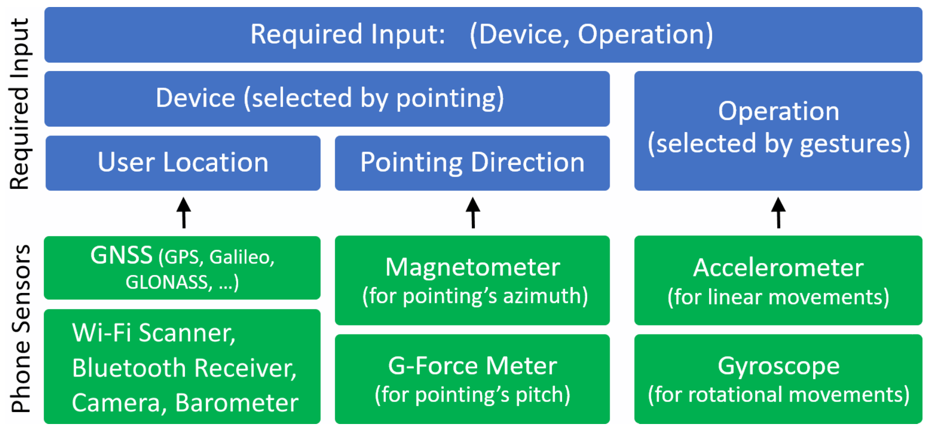

Fig. 3. Using sensors to collect input in phone-pointing remote control apps

GNSS sensors provide the main built-in ability to localize the smartphone. However, to improve accuracy, especially inside buildings, data from other sensors such as the Wi-Fi scanner [10], the Bluetooth receiver [6], and the camera [13] can be used. Barometer values (if available) can help in establishing the 3D location, as barometric pressure values can indicate height (expected to be useful in multi-floor buildings).

The two main sensors for evaluating the 3D pointing direction are the magnetometer and the g-force meter (or the accelerator). Data from the magnetometer can be used to determine the horizontal pointing direction, i.e. the azimuth angle. Data from the g-force meter can be used to determine the vertical pointing direction, i.e. the pitch angle. The g-force (or gravity) meter is often a virtual software sensor, implemented using data from physical hardware sensors, such as the accelerometer. By combining the data from these standard smartphone sensors, the app can establish a 3D pointing direction [7].

After selecting a device by pointing at it with their smartphone, users can specify an operation using arm movement gestures while still holding the smartphone. For example, an upward forearm movement can signal *turn-on*, whereas a downward movement can signal *turn-off*. Similarly, a clockwise rotation can signal *volume-up*, whereas a rotation in the opposite direction, anticlockwise, can signal *volume-down*. The exact gestures can vary from one implementation to another, and even in the same application could be customized by the user. Movement gestures can be identified mainly using data from the linear accelerometer and the gyroscope sensors.

In standard remote apps, the exact physical location of the devices does not play a role (as long as the connection to the network is stable). A phone-pointing remote app, on the other hand, has to be aware of the layout of the devices in a specific environment. We can use Machine Learning (ML) for this purpose, as illustrated in Figure 4. The arrows in Figure 4 indicate the main direction of data flow (although data streams in both directions).

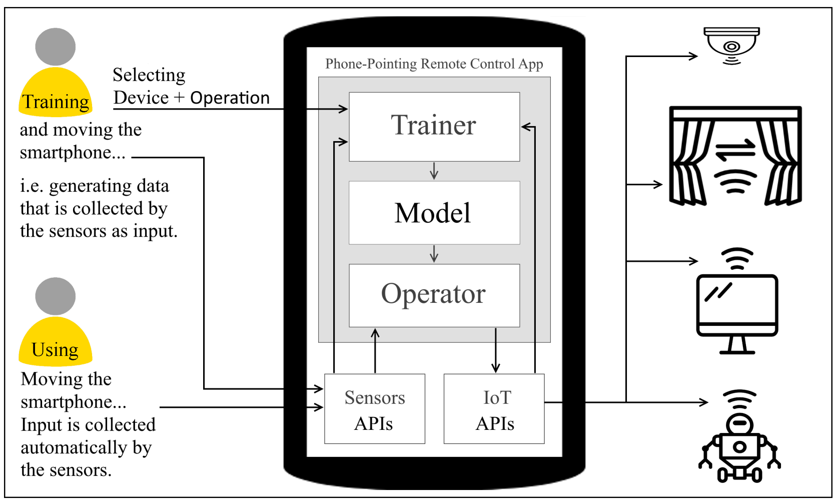

Fig. 4. Structure of phone-pointing IoT remote control apps

During the training phase, the app should learn the environment. This can be achieved by guiding users through on-screen or voice menus and instructions. Users should demonstrate selections of devices by pointing at them from different relevant locations and demonstrate selections of operations by movement gestures. The Trainer component (see Figure 4) collects data from the relevant sensors and relates the data to devices and operations. An ML-model is built based on this information and when it has trained sufficiently the app is ready for use. Using the trained ML-model, the Operator component can convert sensor data from users (which do not use the touchscreen or voice) into $(device, operation)$ values that are then sent to the IoT APIs and processed as commands to the relevant devices.

As shown in Figure 3, each one of the three basic recognition subproblems (user location, pointing direction, operation by gestures) relates to a different set of sensors, and therefore, can be managed as a separate ML task. Using sub-ML-models for the three subproblems may have some advantages, such as simplicity and the ability to choose the most appropriate ML method for each subproblem separately. On the other hand, one combined ML-model for the complete recognition problem of (*device*, *operation*) may have some other significant advantages, such as the ability to fill gaps in data for one subproblem by using data available for the other subproblems. For example, if the location information is uncertain, a neural network model for the entire problem can identify a (*device*, *operation*) pair using only the pointing direction and the movement gestures if they are unique in that environment for a particular (*device*, *operation*) pair.

3 Discussion

To the best of our knowledge, there are no implementations of phone-pointing remote apps yet, so it is impossible to provide a precise evaluation of the feasibility, usability, and acceptability of this model in this early work.

Regarding usability and acceptability, it is reasonable to expect that phone-pointing remote apps, if they work as expected (i.e. provide a proper level of accuracy) could be usable and acceptable in various situations due to their unique combination of potential advantages as being simple, intuitive, fast, and voiceless (i.e. fewer concerns regarding privacy and distractions compared to voice assistants). Adding phone-pointing capabilities to regular remote apps with conventional user interfaces, rather than providing a separate pointing-only user interface, may improve the overall usability and acceptability, as such hybrid apps could provide the best of both worlds. Definitive answers regarding the usability and the acceptability of phone-pointing remote apps would naturally require implementations of such solutions and field studies.

The key question is whether this model is feasible, i.e. whether a proper level of accuracy can be achieved. Although we cannot provide a definitive answer to this question at this point, we can try to evaluate the possibilities and the challenges of this model in light of relevant results from related studies. It may help, in the context of this discussion, to evaluate each of the three basic recognition subproblems (see Figure 3) separately.

Out of the three subproblems, the most challenging one might be finding the smartphone location. Although localization is one of the most commonly used capabilities of smartphones nowadays (for navigation and many other uses), pointing remote apps may need higher levels of accuracy and are expected to be used mainly inside buildings, where GNSS sensors are less accurate. Nevertheless, there are several reasons why we can expect this challenge to be manageable. First, there are some successful implementations of indoor localization (including using the camera [13], Wi-Fi signals [10], and Bluetooth signals [6]). Second, remote apps may be used mainly from very specific locations, e.g. while sitting near a desk or on a sofa, lying in bed, etc. In other words, we do not necessarily need a continuous recognition of every point in space, but instead, we mainly want to be able to recognize specific locations, and that may be more tolerant to errors. Third, many circumstantial data, such as the day and time, the way and direction that the user puts the smartphone, and residual movements of the smartphone until it was put to rest, can all help in inferring the location (as well as pointing and gesture operations, i.e. using data related to the two other subproblems as additional hints). Fourth, a pointing remote app with no localization capabilities at all could still be useful, when used from a single location, e.g. from a hospital bed by a patient, or when the current location can be selected by other means.

Regarding the second subproblem (recognizing the pointing direction), recent studies show that the direction of pointing (or the orientation of the smartphone) can be measured with high accuracy using the smartphone's sensors. Although the accuracy varies across different smartphones, the mean inaccuracy per device is up to 2.1 degrees for the pitch orientation [5]. Even when adding up inaccuracies due to deviation of pointing operations by the user, we can expect the total error margin of the orientation measurements to be up to approximately 5 degrees, which should usually be sufficient to select different devices located in a room by pointing at them with the smartphone [7].

The third subproblem, identifying movement gestures based on smartphone sensors, was investigated in recent studies with overall promising results [12, 14].

The complete problem of recognizing the $(device, operation)$ input cannot be more difficult than the combination of the three underlying subproblems. As discussed above, it might be even easier, because each of the subproblems contributes data that may assist in solving the other subproblems, by using elimination techniques. For example, a certain pointing direction or a certain gesture could be unique in a particular installation, and therefore sufficient to recognize the $(device, operation)$ selection despite missing data.

4 Conclusions and Further Work

This paper presents a new approach to IoT remote control apps. Using phone-pointing remote apps, users would be able to select devices by pointing at them with the smartphone and operate the devices by movement gestures, detected by the smartphone's built-in sensors. Instead of the usual input sources, such as the touch screen in regular remote apps and the microphone in voice assistants, in the proposed phone-pointing remote apps input is collected from a combination of standard smartphone sensors, including the GNSS sensor, the Wi-Fi scanner, the magnetometer, the g-force (gravity) meter, the accelerometer, and the gyroscope.

The main challenge in implementing phone-pointing remote apps is recognizing $(device, operation)$ inputs that users generate by pointing and gesture operations. Our analysis, based on results from related recent studies, provides preliminary positive indications regarding the feasibility of this novel approach.

Further work should involve the development of phone-pointing remote apps, based on this proposed model, and the evaluation of the feasibility, usability, and acceptability of this approach in field studies.

Acknowledgment. The icons of IoT devices in Figures 1, 2, and 4 (Air Conditioner, Alarm Clock, Computer, Curtain, Lightbulb, Nature, Robot, and Webcam) are provided by Mavadee from thenounproject.com.

References

1. Acebron, J., Spigler, R.: The remote control and beyond: The legacy of Robert Adler. SIAM News **40**(5), 1–2 (2007)
2. Bai, Y., Su, H., Hsu, W.: Indoor and remote controls and management of home appliances by a smartphone with a four-quadrant user interface. In: IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA. pp. 319–320 (2017). https://doi.org/10.1109/ICCE.2017.7889336
3. Hoy, M.B.: Alexa, Siri, Cortana, and more: An introduction to voice assistants. Medical Reference Services Quarterly **37**(1), 81–88 (2018). https://doi.org/10.1080/02763869.2018.1404391
4. Khan, A., Al-Zahrani, A., Al-Harbi, S., Al-Nashri, S., Khan, I.A.: Design of an IoT smart home system. In: 15th Learning and Technology Conference (L&T). pp. 1–5 (2018). https://doi.org/10.1109/LT.2018.8368484
5. Kuhlmann, T., Garaizar, P., Reips, U.D.: Smartphone sensor accuracy varies from device to device in mobile research: The case of spatial orientation. Behavior Research Methods **53**, 22–33 (2021). https://doi.org/10.3758/s13428-020-01404-5
6. Murata, M., Ahmetovic, D., Sato, D., Takagi, H., Kitani, K.M., Asakawa, C.: Smartphone-based indoor localization for blind navigation across building complexes. In: 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 1–10. IEEE (2018)
7. Ruser, H., Kirsh, I.: "Point at it with your smartphone": Assessing the applicability of orientation sensing of smartphones to operate IoT devices. In: Proceedings of the 23rd HCI International Conference (HCII). Springer, Cham (2021)
8. Ruser, H., Vorwerg, S., Eicher, C.: Making the home accessible - experiments with an infrared handheld gesture-based remote control. In: Proceedings of the 22nd HCI International Conference (HCII). pp. 89–97. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50732-9_13
9. Sayago, S., Neves, B.B., Cowan, B.R.: Voice assistants and older people: Some open issues. In: Proceedings of the 1st International Conference on Conversational User Interfaces (CUI). pp. 1–3. Association for Computing Machinery, New York, USA (2019). https://doi.org/10.1145/3342775.3342803
10. Shin, B.J., Lee, K.W., Choi, S.H., Kim, J.Y., Lee, W.J., Kim, H.S.: Indoor wifi positioning system for android-based smartphone. In: International Conference on Information and Communication Technology Convergence (ICTC). pp. 319–320 (2010). https://doi.org/10.1109/ICTC.2010.5674691
11. Terzopoulos, G., Satratzemi, M.: Voice assistants and smart speakers in everyday life and in education. Informatics in Education pp. 473–490 (2020). https://doi.org/10.15388/infedu.2020.21
12. Wang, X., Tarrío, P., Metola, E., Barbolla, A., Casar, J.: Gesture recognition using mobile phone's inertial sensors. In: Distributed Computing and Artificial Intelligence (AISC). vol. 151, pp. 173–184. Springer, Cham (2012). https://doi.org/10.1007/978-3-642-28765-7_21
13. Werner, M., Kessel, M., Marouane, C.: Indoor positioning using smartphone camera. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN). pp. 1–6 (2011). https://doi.org/10.1109/IPIN.2011.6071954
14. Xie, C., Luan, S., Wang, H., Zhang, B.: Gesture recognition benchmark based on mobile phone. In: 11th Chinese Conference on Biometric Recognition (CCBR), Chengdu, China. pp. 432–440. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46654-5_48