UNIVERSITY OF CALIFORNIA

Los Angeles

CamIoT: Recognizing and Interacting with Distant IoT Objects using a wrist-worn outward-facing camera

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Amirali Omidfar

2020

ABSTRACT OF THE THESIS

CamIoT: Recognizing and Interacting
with Distant IoT Objects using a
wrist-worn outward-facing camera

by

Amirali Omidfar
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2020
Professor Xiang Anthony Chen, Chair

CamIoT is an effort to add Artificial intelligence into the emerging technologies of Smart Home services. The Features embedded in the device also enable CamIoT to be used as an assistive technology for visually impaired people. CamIoT is the first wrist-worn platform that uses an outward-facing camera to recognize and interact with distant IoT objects. It provides a novel technique using the index finger's orientation to locate an IoT object in the camera view while supporting disambiguating selection in the presence of multiple objects. Our study ends with the full evaluation of the contributed methods and CamIoT's performance, highlighting the best case appliance recognition accuracy of **96%**.

The thesis of Amirali Omidfar is approved.

Ankur Mehta

Majid Sarrafzadeh

Xiang Anthony Chen, Committee Chair

University of California, Los Angeles

2020

To my mother . . .

whom I haven't seen in almost four years

but my love for her still grows

everyday

Table of Contents

List of Figures

Preface

The results discussed in this dissertation are the outcome of a constructive team work. So I would like to pay my special regards to Professor Chen, Yuan Liang, Nic and Simon for their help, support and leadership in different parts of CamIoT's project.

CHAPTER 1

Introduction

1.1 Motivation

1.1.1 Artificial intelligence in Smart Home

Artificial intelligence has played a significant role in enhancing Smart Home technologies lately. A Smart Home is associated with technologies containing sensors, actuators, wired and wireless networks, and intelligent systems. Smart Homes can monitor and control activities, provide comfort in users' interaction with appliances and save overall energy consumption [22]. Artificial intelligence (AI) depicts a device that perceives its environment and takes actions to maximize the chance of successfully achieving its goals [24]. The ideal state of artificial intelligence is thinking humanly, reasoning, acting like humans, and acting rationally [24]. AI technologies used in Smart Home products can be labeled in six core clusters of AI functions, i.e., activity recognition, data processing, voice recognition, image recognition, decision-making, and prediction-making [22]. Previous works have been more focused on the task-specific deployment of such AI systems in Smart Home technologies. In the activity recognition methods used in Hive Link and Essence Care@Home, for instance, smart home devices can identify human activity with the help of AI. It analyzes sensor data to track users' actions and announce the cases of undesired activities. In the aspect of voice recognition, AI works based on voice-based technologies allow people to interact with the device simply by having a conversation (Voice recognition is used in Amazon Alexa, Google Home, Ivee Sleek, Jibo, Athom Homey, Apple HomePod, Josh Micro, etc.) [22]. The aspect we further investigate in this work is a novel combination of **image recognition and**

prediction-making. Previous works in image recognition mostly used AI for facial and emotion recognition. The approach analyzed humans' behavior and physical aspects of the body's structure and form. It is used in Lighthouse, Nest Cam, Honeywell Smart Home Security System, Tend Secure Lynx Indoor Camera, Canary All-In-One, Netatmo Welcome Indoor Security Camera, etc. [7].

In prediction-making, embedded sensors are used to monitor the users while they perform daily routines. An AI agent then processes the data collected by a computer network and stored in a database to find useful knowledge such as patterns, predictions, and trends. [22]. Some examples of this application are Nest Thermostat, Olly and Viaroom home.
Our literature review showed a growing use of AI in Smart Home technologies. However, the use cases were mostly quite task-specific and focused on more narrowed down applications. Our intention, however, is to provide a platform that broadens and simplifies the use of AI in Smart Home technologies. The focus here is the interaction with smart home appliances. In this work, we assume an existing network of Internet of Things (IoT) objects (appliances) with a means of controlling them. Unlike conventional user interfaces, we introduce a novel interaction method based on image recognition. This simple interface can add any new appliance as long as it is part of the IoT network. Such a controller would then reduce the complexity of some of the task-specific smart home applications. We utilized a camera-based wrist-worn device as our prototype. Our other key motivation was to provide visual assistive technology for visually impaired people which is explained in the next section.

1.1.2 Visually Assistive Technologies (VAT)

Globally, an estimated 40 to 45 million people are blind, 135 million have low vision, and 314 million have some visual impairment [1]. The incidence and demographics of blindness vary significantly in different parts of the world. In most industrialized countries, approximately 0.4% of the population is blind, while in developing countries, it rises to 1%. It is estimated by the World Health Organization (WHO) that 87% of the world's blind live in developing

[1] https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment

countries. Over the last decades, visual impairment and blindness caused by infectious diseases have been significantly reduced (an indication of international public health action). However, there is still a visible increase in the number of blind or visually impaired people from conditions related to longer life expectancies. The great majority of visually impaired people are aged 65 years or older. It is estimated that there is a per-decade increase of up to 2 million people over 65 years with visual impairments. This group is growing faster than the overall population[27]. All the systems, services, devices, and appliances that are used by disabled people to help in their daily lives, make their activities more comfortable, and provide safe mobility are included under one umbrella term: **Assistive technology**.

Assistive technologies were introduced to help with the daily problems related to information transmission (such as personal care), navigation, and orientation aids, as part of mobility assistance [9]. As one of its subcategory, **Visual assistive technology (VAT)** is then divided into three categories: vision enhancement, vision substitution, and vision replacement. This technology became available for blind people through electronic devices that allow users to detect and localize the objects and offer those people a sense of the external environment utilizing sensors' functions. The vision replacement subgroup is more complicated than the other two as it deals with medical and technology issues. Vision replacement includes displaying information directly to the visual cortex of the brain or through an ocular nerve. However, vision enhancement and vision substitution are similar in concept; the difference is that in vision enhancement, the camera input is processed, and then the results will be visually displayed. Vision substitution is similar to vision enhancement, **yet the work constitutes a non-visual display, which can be vibration, auditory, or both based on the hearing and touch senses that can be easily controlled and felt by the blind user**.

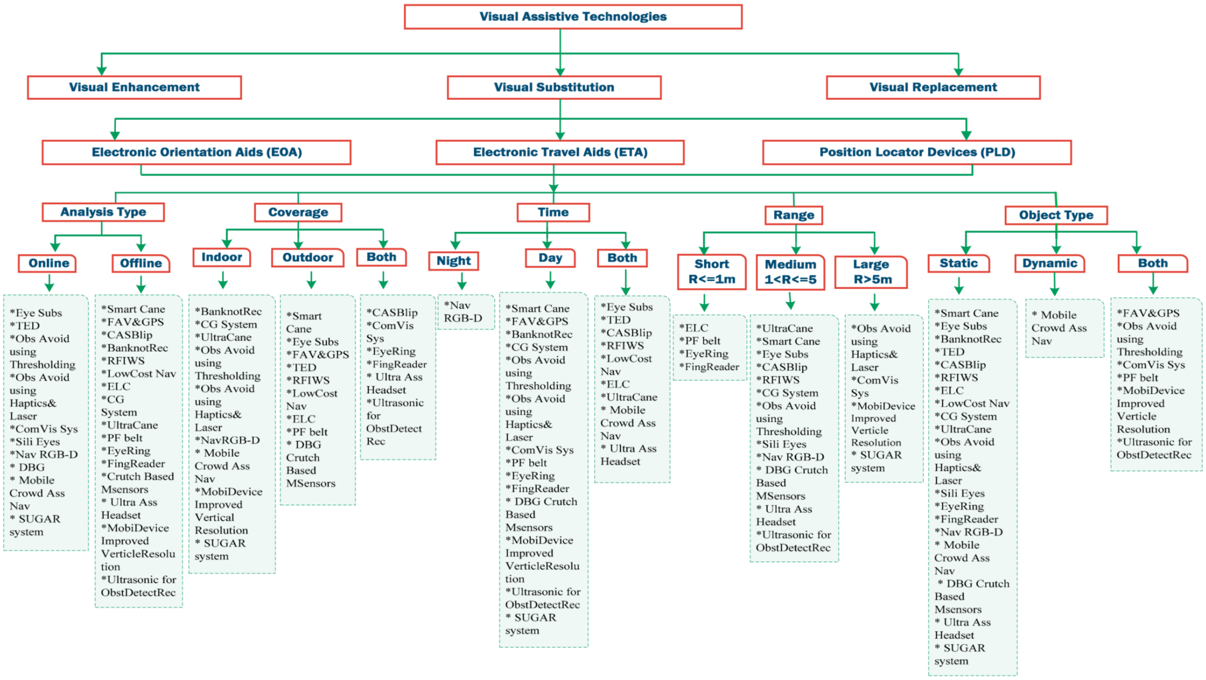

Visual Assistive Technologies
Visual Enhancement
Visual Substitution
Visual Replacement
Electronic Orientation Aids (EOA)
Electronic Travel Aids (ETA)
Position Locator Devices (PLD)
Analysis Type
Coverage
Time
Range
Object Type
Online
Offline
Indoor
Outdoor
Both
Night
Day
Both
Short R<=1m
Medium 1<R<=5
Large R>5m
Static
Dynamic
Both
*Eye Subs *TED *Obs Avoid using Thresholding *Obs Avoid using Haptics& Laser *ComVis Sys *Sili Eyes *Nav RGB-D * DBG * Mobile Crowd Ass Nav * SUGAR system
*Smart Cane *FAV&GPS *CASBlip *BanknotRec *RFIWS *LowCost Nav *ELC *CG System *UltraCane *PF belt *EyeRing *FingReader *Crutch Based Msensors * Ultra Ass Headset *MobiDevice Improved VerticleResolution *Ultrasonic for ObstDetectRec
*BanknotRec *CG System *UltraCane *Obs Avoid using Thresholding *Obs Avoid using Haptics& Laser *NavRGB-D * Mobile Crowd Ass Nav *MobiDevice Improved Vertical Resolution * SUGAR system
*Smart Cane *Eye Subs *FAV&GPS *TED *RFIWS *LowCost Nav *ELC *PF belt * DBG Crutch Based MSensors
*CASBlip *ComVis Sys *EyeRing *FingReader * Ultra Ass Headset *Ultrasonic for ObstDetect Rec
*Nav RGB-D
*Smart Cane *FAV&GPS *BanknotRec *CG System *Obs Avoid using Thresholding *Obs Avoid using Haptics& Laser *ComVis Sys *PF belt *EyeRing *FingReader * DBG Crutch Based Msensors *MobiDevice Improved VerticleResolution *Ultrasonic for ObstDetectRec * SUGAR system
*Eye Subs *TED *CASBlip *RFIWS *LowCost Nav *ELC *UltraCane * Mobile Crowd Ass Nav * Ultra Ass Headset
*ELC *PF belt *EyeRing *FingReader
*UltraCane *Smart Cane *Eye Subs *CASBlip *RFIWS *CG System *Obs Avoid using Thresholding *Sili Eyes *Nav RGB-D * DBG Crutch Based MSensors * Ultra Ass Headset *Ultrasonic for ObstDetectRec
*Obs Avoid using Haptics& Laser *ComVis Sys *MobiDevice Improved Verticle Resolution * SUGAR system
*Smart Cane *Eye Subs *BanknotRec *TED *CASBlip *RFIWS *ELC *LowCost Nav *CG System *UltraCane *Obs Avoid using Haptics& Laser *Sili Eyes *EyeRing *FingReader *Nav RGB-D * Mobile Crowd Ass Nav * DBG Crutch Based Msensors * Ultra Ass Headset * SUGAR system
* Mobile Crowd Ass Nav
*FAV&GPS *Obs Avoid using Thresholding *ComVis Sys *PF belt *MobiDevice Improved Verticle Resolution *Ultrasonic for ObstDetectRec

Figure 1.1: Classification of electronic devices for visually-impaired people.[9]

In CamIoT, as explained in the next sections, we embedded a speaker in the camera-based wrist-worn device. The voice feedback completes our novel AI based user interface design and also provides vision substitution for visually impaired users (VAT). From the perspective of assistive technologies, CamIoT helps the blind user perceive the environment by detecting and recognizing the surrounding objects while simplifying the overall system interface relies on its AI capabilities.

1.1.3 Contributions

According to our literature review, cameras are highly adopted, self-contained sensing modalities. However, there is yet relatively little work done to enable cameras for remote interaction by recognizing an interactive object from a distance. Our goal is to facilitate an always-available mechanism for directly pointing at and interacting with IoT objects from a distance without any instrumentation of IoT objects or the environment.

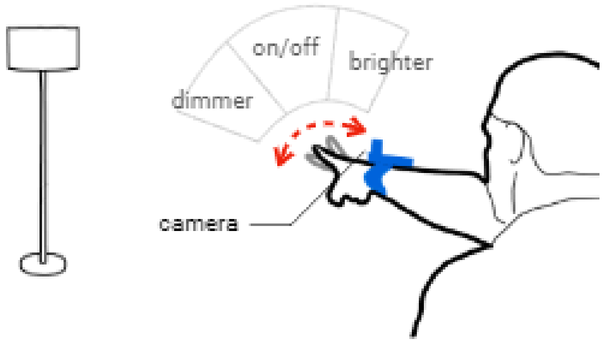


Figure 1.2: CamIoT uses a wrist-worn outward-facing camera to recognize an IoT object as a user points at it, using finger flexion and circumduction gestures to interact with control shortcuts of a selected IoT.

To achieve this goal, we develop CamIoT, a hardware/software platform consisting of a wrist-worn camera that faces outward and recognizes a distant IoT object the user points at, as well as the user's index finger's orientation for locating and interacting with an IoT object in the camera view.

Figure 1.2 illustrates an application scenario of CamIoT where a user points at an Internet-connected floor lamp. The lamp is then located and recognized via the outward-facing camera, allowing the user to circumduct (Figure 1.3 a) the index finger to select one of the three control shortcuts and flex the finger (Figure 1.3 b) to confirm a selection. CamIoT can complement existing IoT interactions by providing a few control shortcuts as the user points at an appliance.

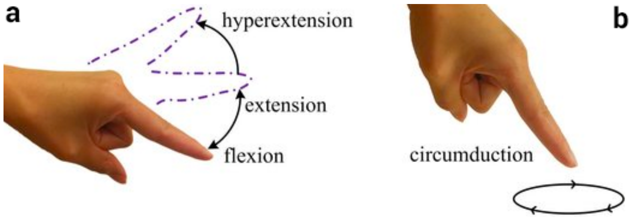


Figure 1.3: CamIoT's outward-facing camera recognizes two anatomically-inspired finger gestures: circumducting (along with a series of virtual sectors) to select a control option; flexing the finger to confirm a selection. image credit: [28]

Our preliminary evaluation that (i) takes a data-driven approach to find optimal parameters of finger circumduction gestures and measure the performance of the finger-based selection; (ii) reports performance of recognizing ten appliances from pointing in a real-world household setting at various distances; and further (iii) demonstrates the generalizability of appliance and finger gesture recognition by testing the integrated CamIoT system on unforeseen users.

The contributions focused on here are as follows:

- The first system that uses a wrist-worn outward-facing camera to recognize and interact with IoT objects at a distance;

- A anatomically-motivated gesture set based on index finger circumduction and flexion, which is amenable for capturing using a wrist-worn outward-facing camera;

- A novel technique that uses the index finger's orientation to locate an IoT object in the camera view that also supports disambiguating selection amongst multiple IoT objects.

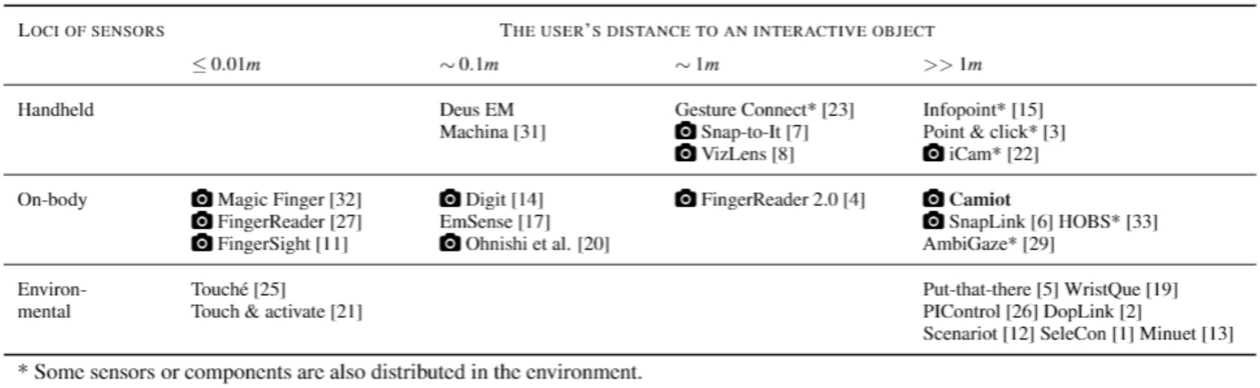
- An in-depth evaluation of the contributed methods and CamIoT's overall performance.

CHAPTER 2

Related Works

From our study of related work, there are two prominent dimensions to organize a design space. The first dimension is the distance to objects, which we discretize into four orders of magnitudes ($0.01m$, $0.1m$, $1m$, and $> 1m$). At about $0.01m$ distance, miniaturized wearable devices can identify objects by their textures [30]and convert visual information into haptic feedback [13]. At about $0.1m$ distance, electromagnetic waves can serve as unique signature of digital objects [29, 18]. At the $1m$ distance, both NFC [21] and camera [8, 4, 10] allow users to select nearby objects. Finally, at distances over $1m$—most related to our interest on remote IoT interaction—a myriad of sensing solutions have been explored (from prototypical infrared remote control [3], to tag-based augmented reality [16], Ultra-Wide Band radio [19, 1, 15], using patterns of audio [2] and light [25] signals).

The second dimension is the loci of sensors—handheld, on-body, or in the environment. For remote interaction, most approaches require instrumenting the environment [21, 16, 3, 20, 5, 19, 25, 2, 14, 1, 15]. In the meantime, all the self-contained solutions for $1m$ and beyond are camera-based [8, 10, 4, 6]; but, to the best of our knowledge, only Snaplink [6] can handle $> 1m$ interaction. SnapLink requires a 3D construction of the entire space for image localization, and its performance is unknown for residential apartments with more appliances in a smaller space. Such a gap motivates our work on developing a camera-based, self-contained (no instrumentation in the environment) device to enable remote interaction ($> 1m$) with IoT objects.



Loci of sensors	The user's distance to an interactive object			
	$\leq 0.01m$	$\sim 0.1m$	$\sim 1m$	$>> 1m$
Handheld		Deus EM Machina [31]	Gesture Connect* [23] 📷 Snap-to-It [7] 📷 VizLens [8]	Infopoint* [15] Point & click* [3] 📷 iCam* [22]
On-body	📷 Magic Finger [32] 📷 FingerReader [27] 📷 FingerSight [11]	📷 Digit [14] EmSense [17] 📷 Ohnishi et al. [20]	📷 FingerReader 2.0 [4]	📷 **Camiot** 📷 SnapLink [6] HOBS* [33] AmbiGaze* [29]
Environ-mental	Touché [25] Touch & activate [21]			Put-that-there [5] WristQue [19] PIControl [26] DopLink [2] Scenariot [12] SeleCon [1] Minuet [13]

* Some sensors or components are also distributed in the environment.

Figure 2.1: Design space for summarizing prior works on interaction with objects based on: (i) Distance (ii) The loci of sensors.

Previous works on remote interaction rely more on instrumenting the environment and, little work has been done to enable remote interactions through recognizing IoT objects.

One alternative is to use pervasive smartphone cameras (similar to Snap-to-It [8]). However, the main concern is the acquisition time. We cannot expect a user to retrieve their smartphone every time they want to interact with an IoT object. Thus we chose to develop a custom-built, wearable camera that is always available and allows users to point, shoot, and control an IoT object.

CHAPTER 3

System Overview

Hardware platform As shown in Figure 3.1, we built a proof-of-concept hardware platform for exploring finger-pointing and gesturing to interact with IoT. Our platform consists of a Raspberry Pi Zero W as the controller, an MPU 6050 IMU sensor for providing accelerometer and gyroscope data, a mini (8 ohm 0.5 W) speaker and a Raspberry PI Camera Module V2 for capturing IoT and the user's finger. The camera height (distance between its center and the user's wrist when worn) is about 4cm. The Pi Camera V2 module captures and sends all the images via sockets across a local area network to a server program that performs image processing and classification (detailed below), programmed in Python. Finally, we used the speaker embedded in the device to provide voice feedback.

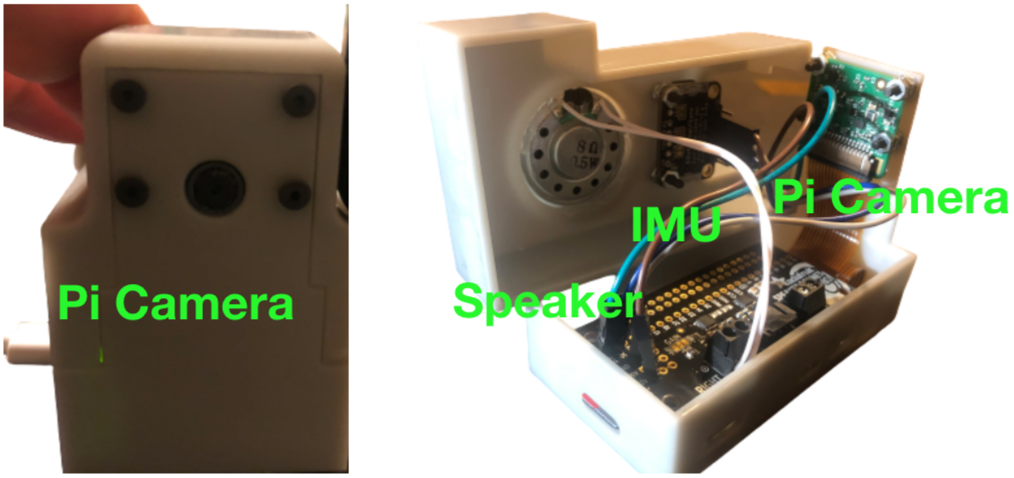


Figure 3.1: CamIoT hardware prototype

The process starts when the user raises his arm to take a picture. CamIoT senses such arm movements through the IMU data using [15]'s method. The image is then sent to the local server for classification. Once the result is communicated back to the device, the voice feedback announces the result. In this step, if the user decides to hold his/her finger in the camera view, the finger's direction helps to crop the image (we refer to this technique as disambiguation in later sections). After recognition, CamIoT solely tracks the index finger's position in the camera view (finger circumduction). The current setup matches the finger's location to either the left, middle, or right direction. The user confirms the finger's direction by dropping his/her finger from the camera view (flexion). The chosen direction is then mapped to a particular function on the IoT device (e.g., turn down TV Volume).

Finger gestures to select a control option Once an appliance is selected, CamIoT allows the user to interact with it using the index finger. Based on the index finger's anatomical and kinesthetic properties [28], we design two gestures to support such interactions:
(i) **Circumduction for selection** where the index finger first hyperextends and then rotates primarily around its Carpometacarpal joint (Figure 1.3 a). The finger motion covers a half-circle from the camera view. Thus we divided this hypothetical semi-circle in different ways so it would have N segments ($N \in 2, 3, 4, 5,$). Later in the evaluation section, we take a data-driven approach to compute the optimal thresholds for segmentation as well as the user's performance in placing the index finger into each sector given different numbers of divisions.
(ii) **Flexion for confirmation** happens to confirm a selection users have made. So they would confirm the selection by dropping (flexing) their index finger as shown in Figure 1.3 b (a similar to the 'airtap' gesture in Microsoft Hololens[1]).

We now focus on the two key technical components of CamIoT: The unsupervised finger gesture recognition pipeline and how we can leverage the finger's orientation to augment an appliance's recognition using a Convolutional Neural Network (ConvNet).

[1] https://docs.microsoft.com/en-us/windows/mixed-reality/interaction-fundamentals

1. Unsupervised Finger Gesture Recognition Pipeline: To recognize the finger direction, CamIoT performs automatic finger segmentation from the camera view. One popular method that achieves the state-of-the-art accuracy for object segmentation is using ConvNets [23, 11]. However, such methods heavily rely on supervised learning from large-scale data with detailed annotations, which can be very labor-consuming to label. In contrast, our method utilizes skin color and edge detection characteristics for compact but robust finger segmentation without any supervision.

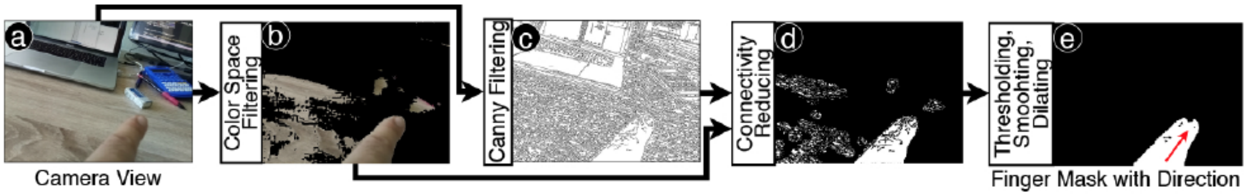


Figure 3.2: Finger Segmentation Pipeline

Figure 3.2 demonstrates the pipeline with one example image. We first derive a rough finger mask with the skin color model in YCbCr space [17] that is designed to be adaptive for most populations. Multiple false-positive regions can exist in the segmentation map, mainly because of background objects having similar colors to skins. Meanwhile, an edge mask is generated with the Canny filter, which is then applied to the finger mask with rolling in up/down/left/right directions for one pixel, to cut off bordering regions with connectivity less than four. We take the largest isolated region on the resulted finger mask that lies on the lower part of the image as the finger prediction and further derive the finger direction by linearly interpreting the row-wise midpoints of the segmentation. If no region has an area size larger than a preset threshold, the image is detected as no finger. The whole pipeline runs at 39.84 frames/second as measured on an Intel Core i5 processor.

2. Appliance Recognition with Finger tracking aid: In this work, we carry out the few-shot learning of a ConvNet for appliance recognition, as ConvNets are better for capturing small objects from images with features of large receptive fields [26, 12]. Moreover, we propose to utilize the finger for disambiguation and feed the model with the indicated portion of an image for prediction. Such a method can potentially benefit model performance because: Model attention can focus on the appliances with reduced background areas, thus reducing the probability of wrong classifications by eliminating other potential appliances from the background.

One limitation of such a method is that the object needs to occupy a larger space in the query image for reducing mismatches caused by background noises. Thus it is not optimal for our task of interacting with an IoT object from a distance, where appliances can take only a small portion of the camera view.

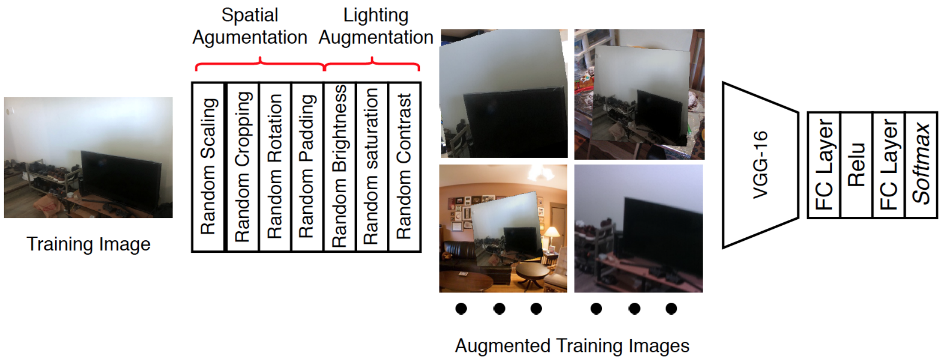


Figure 3.3: Appliance Recognition pipe line

In the setup stage, five images of each appliance are taken by CamIoT from various angles as training data. During the deployment, the finger segmentation is first derived automatically from the query image. Then, guided by the finger's direction, the image is cropped to 0.6 of its size and fed into the model for inference.

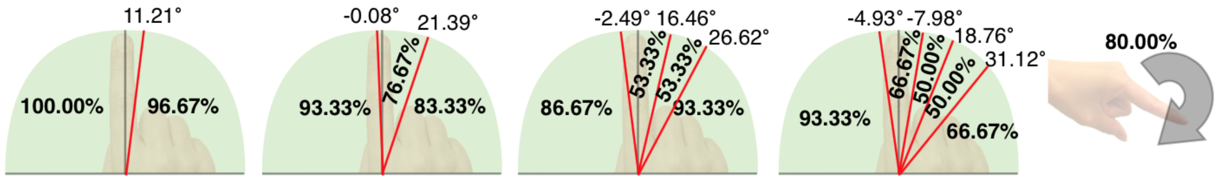
We now compare the performance of the ConvNet model with template matching methods and perform ablation tests to demonstrate the effectiveness of the finger-based disambiguation.

CHAPTER 4

Evaluation

Due to the COVID-19 pandemic, we had limited access to participants. We collected data from one participant (P1, male, aged 25) pointing and finger-gesturing at IoT objects to evaluate the index finger tracking and IoT objects recognition. Then we performed an integrated test of the whole CamIoT system on two other participants (both male, ages 27 and 30, living in the same household as (P1) to evaluate CamIoT's generalizability and usability.

We first evaluate the model accuracy on detecting the selection via finger circumduction on a virtual panel with different numbers of sectors, ranging from two to five. We then report the detection accuracy of the finger's flexing, which happens after each selection.



(a) 2-sector (b) 3-sector (c) 4-sector (d) 5-sector (e) Flexion

Figure 4.1: Finger interaction accuracy. Red lines show the determined angle thresholds for splitting the sectors. Bold numbers represent the detecting accuracy of finger pointing at the sectors and finger flexion.

1. Unsupervised Finger Gesture Recognition Pipe line: We first evaluate the model accuracy on detecting the selection via finger circumduction on a virtual panel with different numbers of sectors, ranging from two to five (we call each number of sectors a sector design). We also report the detection accuracy of the finger's flexing, used as the confirmation gesture.

To determine the most natural thresholds of angles for dividing the virtual panel, we carried out a pilot study asking one participant (P1) to point at each sector based on their estimation without any visual/audio reference or feedback. Specifically, we asked the user to perform three pointing tasks for each sector with their order randomized to avoid temporally-dependent behavior. We repeated this process for all the four sector designs and logged the pointing angles across all the trials. We then determined the optimal thresholds to be the angles that best split different sectors based on an exhaustive search.

Then, we informed P1 about the angle thresholds and measured how accurate P1 could point at each sector across all the four designs ($N \in 2, 3, 4, 5$). We randomly selected ten appliances from P1's household setting. With P1 pointing at the devices, we checked for the effect of background on the recognition accuracy. Specifically, P1 was asked to point at each sector and then perform a finger flexing gesture. Each sector design was performed three times randomly, and this process was repeated using the ten appliances as the background, which in total results in 10×(3+4+5+6)×3=540 images.

Our algorithm achieves a sector-wise mean accuracy of 98.33%, 84.44%, 71.67%, and 65.33% for the 2-, 3-, 4- and 5-sector designs, respectively. The results clearly show the design trade-off between the number of sectors and recognition accuracy: while more sectors enable more control options, it also causes more errors.

The selection errors can be mainly caused by: (i) the sector range being too small so the user's pointing falls out of the intended area; or (ii) the finger segmentation not being good enough, leading to discrepancies between the real directions and the predicted directions. Figure 4.2 shows an example of accurate finger segmentation and its finger direction prediction. In contrast, Figure 4.2 b shows a typical imperfect segmentation caused by the

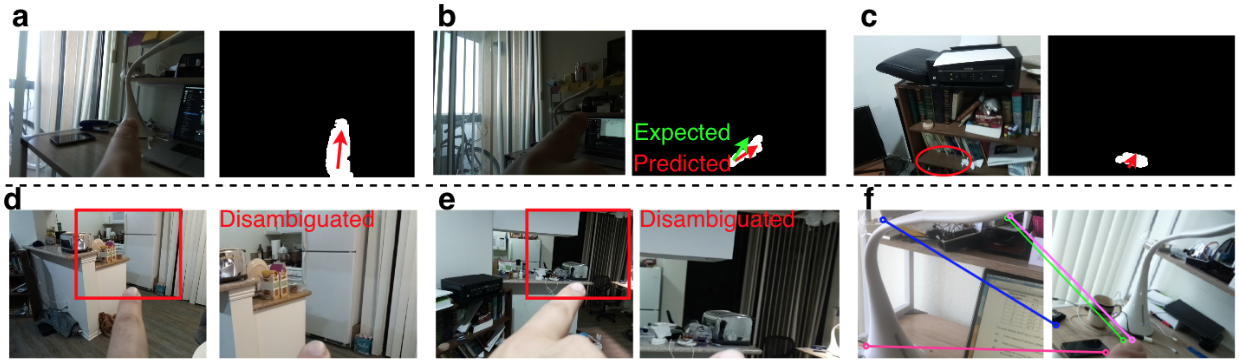


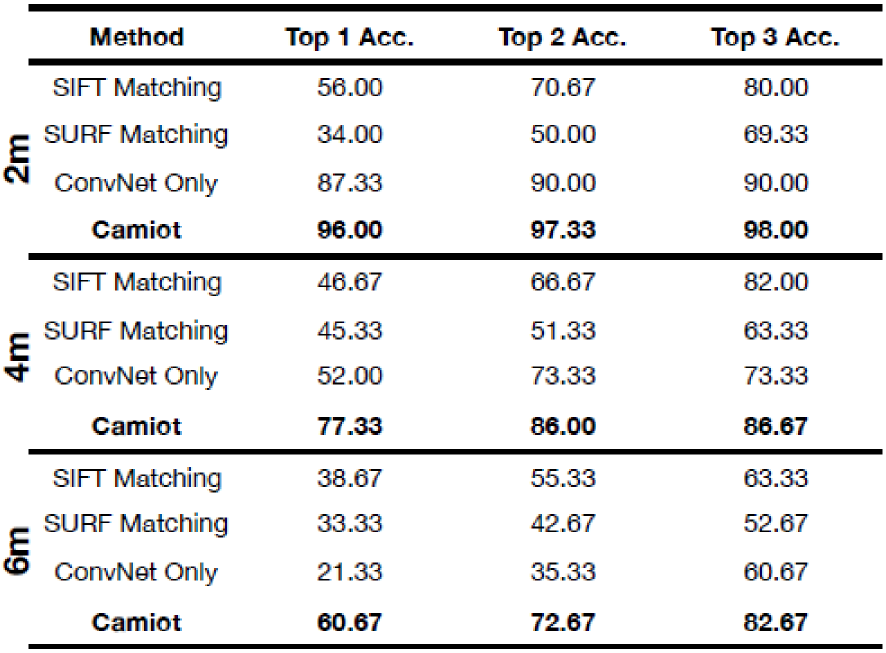
Figure 4.2: Case study for finger gesture recognition and appliance recognition. (a) An accurate finger segmentation with its derived direction. (b) An imperfect finger segmentation with direction discrepancy. (c) A failure case for finger flexion recognition. (d, e) Finger-based disambiguation localizes the desired appliance in the images where multiple appliances exist. (f) A typical failure case for other feature matching methods.

image's underexposure, such that a part of the finger shifting out of the predefined skin color distribution. The segmentation error can then lead to a discrepancy between the predicted finger direction and the user's expected direction, as shown in Figure 4.2 b.

For the recognition of finger flexion, our algorithm achieves a high detection accuracy of 80.00%. As demonstrated in Figure 4.2 c, flexion recognition failure would happen if some background object with similar skin color existed.

2. Appliance Recognition with Finger Disambiguation: To evaluate the accuracy of selecting appliances from a distance, we built a dataset consisting of 10 appliances randomly chosen from P1's household. First, we asked the participant (P1) to collect five templates for each appliance, all from about 0.5 meters away, to ensure the object's details could be captured (although our recognition can handle the much longer distance, as shown below). Moreover, an appliance's templates were taken roughly from the angles that evenly divided the outward surface of that appliance to profile its visual appearance comprehensively. Then we asked the participant to point at the appliances using CamIoT. For each appliance, the participant pointed at it 20 times from random positions. We controlled the distance between the participant and the appliances to study the robustness of our algorithm to distances: the participant pointed at each appliance at three ranges of distance: ∼2m, ∼4m and ∼6m. In total, we collected 10×15×3=450 images. which we used as our appliance recognition dataset.

We compared our algorithm with two template matching methods; one utilizes SIFT as the feature descriptor while the other uses SURF. Both methods were fed with cropped images based on finger disambiguation for a fair comparison. Moreover, we also perform an ablation test on our algorithm to investigate the impact of the figure disambiguation on recognition accuracy. To measure accuracy, we calculate the hit rate for the target appliance in the top 1, top 2, and top 3 of the ranked list of results returned by each method.



	Method	Top 1 Acc.	Top 2 Acc.	Top 3 Acc.
2m	SIFT Matching	56.00	70.67	80.00
	SURF Matching	34.00	50.00	69.33
	ConvNet Only	87.33	90.00	90.00
	Camiot	**96.00**	**97.33**	**98.00**
4m	SIFT Matching	46.67	66.67	82.00
	SURF Matching	45.33	51.33	63.33
	ConvNet Only	52.00	73.33	73.33
	Camiot	**77.33**	**86.00**	**86.67**
6m	SIFT Matching	38.67	55.33	63.33
	SURF Matching	33.33	42.67	52.67
	ConvNet Only	21.33	35.33	60.67
	Camiot	**60.67**	**72.67**	**82.67**

Figure 4.3: Accuracy comparison between different methods for appliance recognition at different distances. All values are in percentage.

Figure 4.3 shows that our method achieves top 1 recognition accuracy of 96.00%, 77.33% and 60.67 for 2m, 4m, and 6m distances, respectively, which are the highest among all the methods. Note that CamIoT also achieves a high top 3 accuracy of 98.00% (2m), 86.67% (4m), and 82.67% (6m), which suggests that the user can select the desired appliance with two extra steps (via a wrist rotation gesture that selects the next best in the result list).

3. Informal User Testing on the Integrated System: Finally, we conducted an informal user testing with P2 and P3 on the integrated CamIoT system. Given the COVID-19 pandemic, we did not intend this study to replace a full user evaluation; rather, our goal was to provide preliminary performance results of CamIoT to investigate whether our appliance and finger gesture recognition techniques (trained on P1) can generalize (for P2 and P3).

The main task was to use CamIoT to interact with a new set of five appliances[1]: pointing at each of the five appliances and the following audio prompts to perform index finger-based selection and confirmation of appliance-specific control options. Based on the performance measured earlier, we chose a three-sector design to balance the number of options and the accuracy of locating each sector.

Each participant was asked to interact with each appliance five times in a randomized order. Participants were standing the entire time. Each time for each appliance, we randomly changed the participant's position (while maintaining a line of sight of the appliance) to vary the angle and distance from which CamIoT captured and recognized the appliance. The distance between the participant and an appliance was always between three to five meters.

For each trial, if an appliance was misrecognized, we asked the participant to abort and restart a new trial. For finger gesture recognition, to maintain consistency and comparability with the earlier P1's testing, participants performed the circumduction and flexion gestures without any feedback.

In total, the participants performed $2 \times 5 \times 5 = 50$ trials of finger pointing + gesturing interaction with appliances. The results are reported as follows.

[1] We used TV, toaster, lamp, printer and coffee maker. Except for the TV, all appliances were not Internet-connected. Thus their control options were just proof-of-concept mock-ups that provide audio feedback (described in the System section) without real functionalities.

Accuracy. Participants made a total of 58 attempts for selecting appliances, resulting in a recognition accuracy of 86.21%. Amongst the 50 trials, for 44 times, an appliance was correctly recognized in one shot (88.00%), three appliances took two attempts, and the other three took three tries. Note that the performance numbers here are higher than those in Figure **??** because the number of appliances was smaller (five compared to ten).

The overall accuracy of using finger circumduction to select a sector was 76.00%; in comparison, the aforementioned P1 testing accuracy was 84.44%. Such a drop in performance was expected, as the optimal thresholds were determined based on P1's data. Multiple cross-user variances (finger appearance, finger agility, perception of different sectors, how the device was worn) could have contributed to the discrepancy in circumduction gesture recognition performance. On the other hand, all finger flexion gestures were recognized correctly.

Best-case response time. We profiled each trial in two phases: (i) the appliance selection phase starting from the prompt and ending when the system correctly recognizes an appliance; (ii) the control option phase starting after the system correctly recognizes an appliance and ending when the user selects the correct control option via a finger gesture. Across all trials, the application selection phase took an average of 3.0s and the control option phase 3.5s, resulting in a total of 6.5s per interaction. Note that this result of 6.5s only indicates the best-case response time where both the appliance and the control option are selected correctly in one shot. If a feedback loop (using audio) is provided, the response time will be longer. As the user can continuously adjust their arm and finger until the intended appliance or control option is selected. At present, our best-case response time is mainly bottlenecked by latency due to the video capturing routines and suboptimal networking speed in a residential household setting.

CHAPTER 5

Limitations

Based on our preliminary findings, we summarize the limitations of CamIoT and discuss future works in this section.

Latency. Our integrated system currently experienced latency issues (running at ∼3 FPS) due to a combination of video capturing on an embedded device and suboptimal networking speed.

Lighting Condition: Lighting condition is one of the most common factors that can affect the performance of a vision-based system. In this work, we did not intentionally control the lighting when carrying out the experiments to study its impact formally. However, we have noticed the finger segmentation pipeline does get affected by lighting changes. Specifically, a lack of lighting or colored interior lighting reflections can change the tone of the finger's color, possibly affecting our skin-color-based finger segmentation algorithm.

Virtual Panel Design: Due to our limited access to participants during the COVID-19 pandemic, we designed the sector thresholds for the virtual panel by referring to the finger data from one user. Such thresholds might not represent most users; thus the next step would be to develop a per-user calibration mechanism, which involves a user performing finger circumduction with CamIoT to determine the best thresholds for each individual automatically. Long-term future work should conduct a larger-scale study to generalize the optimal threshold angles for the panel.

Control options: Currently, we employ an absolute mapping from finger orientation to the selection of a control option. One challenge of such design is the scalability to handle many control options since the experiments show the pointing error rate increases when having more sectors in the virtual panel. One alternative solution is to use relative mapping by tracking sequences of index finger actions, e.g. moving the finger clockwise or towards some directions, to act as arrow keys that navigate a list of control options. Based on our real-time finger segmentation pipeline, the recognition algorithm for finger actions can be further developed in the future for the purpose.

Variation in wearing the device. We found that the device position/orientation/tightness is different each time it was put on a user's wrist during our studies. Due to the limited number of participants, we did not formally study how such variation can impact the performance of CamIoT's interactions. We will address this issue in future work by having more people wear the device and test our system's robustness against such variation.

CHAPTER 6

Conclusion

This work was an effort in the space of utilizing AI in smart home technologies. The idea was to simplify the current existing smart home technologies by unifying the underlying controller used for different home appliances (a universal controller). Using AI's recognition and prediction-making capabilities, CamIoT proposed a novel method for interaction with home appliances. Assuming the appliances are IoT objects in the same network, we showed new devices could be added by just taking five close-up shots of them with minimal training time (we used the transfer learning model in this work). Our method also provided a novel finger tracking approach, which helped us both in appliance recognition and interaction with them.

Employing the user's index finger, we created a more reliable appliance recognition method, which outperformed SIFT Matching, Surf Matching, and solo ConvNet models (Top 1 accuracy of 96%). Although the training images were collected from a close distance ($\sim 0.5m$), to capture more image features, our results remained promising with the proper processing techniques. So we claimed CamIoT as an appropriate method for interacting with distant IoT objects.

This characteristic, coupled with CamIoT's on board voice feedback system, placed us closer to our second motivation in terms of visual assistive technologies (VAT). As explained in Chapter 1, our focus in terms of VAT in this work was vision substitution. Although CamIoT was not solely designed towards visually impaired people's needs, relying on the camera feed, we demonstrated accurate and viable vision substitution via auditory. Most of the works in VAT domain focus on navigation needs and obstacle avoidance, which would require further work (e.g., sensor implementation) on the current design of CamIoT. An exciting approach would be to treat obstacles as other objects and again utilizes CamIoT's object recognition and computer vision features. Our observations showed how the audio could help the users to get a better sense of their environment. Thus we see great potentials in taking this approach given proper analysis (e.g., asking visually impaired individuals for user studies, etc.).

These promises, however, should not distract us from the concerns provided in the Limitations chapter. Both our appliance recognition and finger tracking methods can be improved; there were scenarios requiring optimizations for both methods. In appliance recognition, for instance, we faced test cases where the pointing angle did not have enough coverage of the object leading to a wrong classification. Similarly, the current finger tracking approach is quite sensitive to the background color, which caused wrong predictions in cases with a too bright background.

In conclusion, CamIoT presented an innovative approach for interaction with smart home appliances using AI capabilities and computer vision. The two main novelties introduced here were: **Unsupervised Finger Gesture Recognition** and **Appliance Recognition with Finger Disambiguation**. Finally we evaluated and analyzed our methods in detail to show CamIoT'S potentials and provide future works directions.

Bibliography

[1] Amr Alanwar, Moustafa Alzantot, Bo-Jhang Ho, Paul Martin, and Mani Srivastava. Selecon: Scalable iot device selection and control using hand gestures. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, IoTDI '17, pages 47–58, New York, NY, USA, 2017. ACM.

[2] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. Doplink: using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 583–586. ACM, 2013.

[3] Michael Beigl. Point & click-interaction in smart environments. In *International symposium on handheld and ubiquitous computing*, pages 311–313. Springer, 1999.

[4] Roger Boldu, Alexandru Dancu, Denys JC Matthies, Thisum Buddhika, Shamane Siriwardhana, and Suranga Nanayakkara. Fingerreader2. 0: Designing and evaluating a wearable finger-worn camera to assist people with visual impairments while shopping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):94, 2018.

[5] Richard A Bolt. *"Put-that-there": Voice and gesture at the graphics interface*, volume 14. ACM, 1980.

[6] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. Snaplink: Fast and accurate vision-based appliance control in large commercial buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(4):129:1–129:27, January 2018.

[7] Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. Casas: A smart home in a box. *Computer*, 46(7):10.1109/MC.2012.328, Jul 2013. 24415794[pmid].

[8] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. Snap-to-it: A user-inspired platform for opportunistic device interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5909–5920, New York, NY, USA, 2016. ACM.

[9] Wafa Elmannai and Khaled Elleithy. Sensor-based assistive devices for visually-impaired people: Current status, challenges, and future directions. *Sensors (Basel, Switzerland)*, 17(3):565, Mar 2017. 28287451[pmid].

[10] Anhong Guo, Xiang 'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P Bigham. Vizlens: A robust and interactive screen reader for interfaces in the real world. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 651–664. ACM, 2016.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[13] Samantha Horvath, John Galeotti, Bing Wu, Roberta Klatzky, Mel Siegel, and George Stetten. Fingersight: Fingertip haptic sensing of the visual environment. *IEEE journal of translational engineering in health and medicine*, 2:1–9, 2014.

[14] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. Scenariot: Spatially mapping smart things within augmented reality scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 219:1–219:13, New York, NY, USA, 2018. ACM.

[15] Runchang Kang, Anhong Guo, Gierad Laput, Yang Li, and Xiang 'Anthony' Chen. Minuet: Multimodal interaction with an internet of things. In *To Appear at the ACM symposium on Spatial user interaction*. ACM, 2019.

[16] Naohiko Kohtake, Jun Rekimoto, and Yuichiro Anzai. Infopoint: A device that provides a uniform user interface to allow appliances to work together over a network. *Personal and Ubiquitous Computing*, 5(4):264–274, 2001.

[17] S Kolkur, D Kalbande, P Shimpi, C Bapat, and J Jatakia. Human skin detection using rgb, hsv and ycbcr color models. *arXiv preprint arXiv:1708.02694*, 2017.

[18] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 157–166. ACM, 2015.

[19] B. D. Mayton, N. Zhao, M. Aldrich, N. Gillian, and J. A. Paradiso. Wristque: A personal sensor wristband. In *2013 IEEE International Conference on Body Sensor Networks*, pages 1–6, May 2013.

[20] Shwetak N Patel, Jun Rekimoto, and Gregory D Abowd. icam: Precise at-a-distance interaction in the physical environment. In *International Conference on Pervasive Computing*, pages 272–287. Springer, 2006.

[21] Trevor Pering, Yaw Anokwa, and Roy Want. Gesture connect: facilitating tangible interaction with a flick of the wrist. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 259–262. ACM, 2007.

[22] Biljana L. Risteska Stojkoska and Kire V. Trivodaliev. A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454 – 1464, 2017.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[24] Ashish Sabharwal and Bart Selman. S. russell, p. norvig, artificial intelligence: A modern approach, third edition. *Artif. Intell.*, 175:935–937, 04 2011.

[25] Dominik Schmidt, David Molyneaux, and Xiang Cao. Picontrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 379–388. ACM, 2012.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Ramiro Velázquez. Wearable assistive devices for the blind. *Lecture Notes in Electrical Engineering*, page 331–349, 2010.

[28] Lefan Wang, Turgut Meydan, and Paul Ieuan Williams. A two-axis goniometric sensor for tracking finger motion. *Sensors*, 17(4):770, 2017.

[29] Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. Deus em machina: on-touch contextual functionality for smart iot appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4000–4008. ACM, 2017.

[30] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 147–156. ACM, 2012.