Camiot: Recognizing & Interacting with Distant IoT Objects Using a Wrist-Worn Outward-Facing Camera

Anonymous for review

ABSTRACT

An Internet of Things (IoT) are becoming increasingly ubiquitous, yet interacting with them often requires the extra effort of retrieving apps from personal devices or instrumenting the environment for voice or gesture control. Our goal is to enable always-available instrumentation-free interaction by simply pointing and gesturing at an IoT object. To achieve this goal, we develop Camiot—a wrist-worn platform that uses an outward-facing camera to recognize and interact with IoT objects at a distance. One novel aspect is leveraging the user's index finger, using its pointing orientation for locating an IoT object in the camera view and enabling interaction with a selected IoT object using finger circumduction and flexion. We report the performance of IoT and finger gesture recognition; a small-scale out-of-lab study of the integrated system validates the feasibility of using Camiot to interface with household appliances.

Author Keywords

Camera; an Internet of Things; gesture interaction; gesture recognition.

CCS Concepts

•Human-centered computing → Human computer interaction (HCI);

INTRODUCTION

The Internet of Things (IoT objects[1]) has become increasingly pervasive as more and more appliances are becoming Interent-connected. However, interacting with IoT objects often requires retrieving apps on personal devices or instrumenting the environment for voice or gesture based control. Our literature review suggests that camera is a highly adopted, self-contained sensing modality, yet relatively little work has been explored to enable camera for remote interaction by recognizing an interactive object from a distance.

[1]In this paper we use the terms 'IoT objects' and 'appliances' interchangeably.

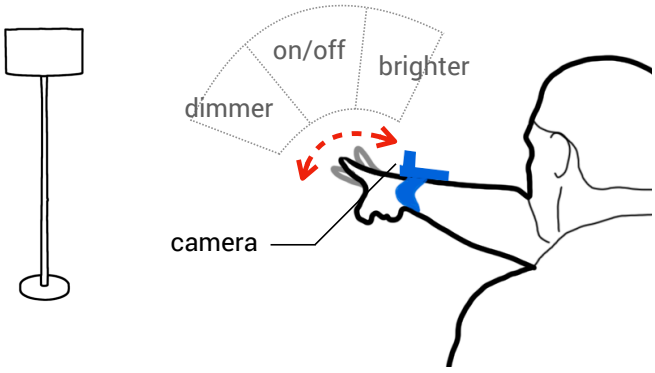

Figure 1. Camiot uses a wrist-worn outward-facing camera to recognize an IoT object as a user points at it, using finger flexion and circumduction gestures to interact with control shortcuts of a selected IoT.

Our goal is to enable 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. To achieve this, we develop Camiot——a hardware/software platform consisting of a wrist-worn camera that faces outwards and recognizes a distant IoT object a 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.

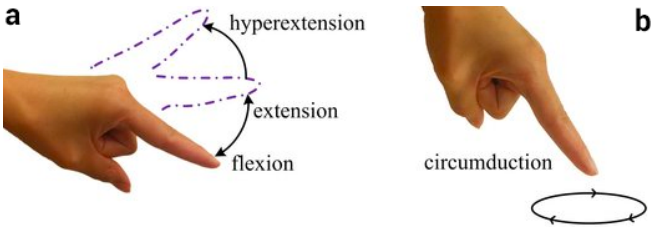

Figure 2. Camiot's outward-facing camera recognizes two anatomically-inspired finger gestures: circumducting (along a series of viertual sectors) to select an control option; flexing the finger to confirm a selection. Image credit: Wang *et al.*[31].

Figure 1 illustrates an application scenario of Camiot: a user points at an Internet-connected floor lamp, which is then located and recognized via the outward facing camera, allowing the user to circumduct (Figure 2a) the index finger to select one of the three control shortcuts, and flex the finger (Figure 2b) to

Submitted to a UIST 2020.

Table 1. A design space summarizing prior work on interacting with objects in the environment structured by *(i)* the user's distance to an object and *(ii)* the loci of sensors. Previous research on remote interaction tends to rely on instrumenting the environment. Meanwhile, camera emerged to be a highly adopted, self-contained sensing modality, yet little work has been explored to enable remote interaction by recognizing an IoT object.

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.

confirm a selection. Camiot can complement existing IoT interaction by providing a few control shortcuts that are available as a user points at an appliance.

Our preliminary evaluation that *(i)* takes a data-driven approach to find optimal parameters of finger circumduction gestures and measure performance of the finger-based selection; *(ii)* reports performance of recognizing 10 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.

Our contributions 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.

RELATED WORK

From our study of related work, we discover two prominent dimensions that allow us to conceptually organize literature in a structured design space as shown in Table 1. Collectively, these samples of prior research all attempted to identify or recognize interactive objects in the environment.

The first dimension is the user's distance to an interactive object, 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 [32], convert visual information into haptic feedback [11], or recognize text on a book [27]. At about $0.1m$ distance, electromagnetic wave can serve as unique 'signature' of digital objects [31, 17] and wrist-mounted camera can detect handheld objects [14, 20]. At the $1m$ distance, both NFC [23] and camera [7, 4, 8] allow users to select a nearby object. 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 [15], to Ultra-Wide Band radio [19, 1, 13], to using patterns of audio [2] or light [26] signals, and to a fusion of different sensor data [5, 22, 6, 12].

The second dimension is the loci of sensors—handheld, on-body, or in the environment. As shown in Table 1, close-ranged interaction primarily relies on on-body sensors, *e.g.*, mounted on the finger [32, 11, 27] or worn on the wrist [14, 17, 20]. For remote interaction, the majority of approaches require instrumenting the environment [23, 15, 3, 22, 5, 19, 26, 2, 12, 1, 13], which limits the practicality and mobility of the interaction. In the meantime, all the self-contained solutions for $1m$ and beyond are camera-based [7, 8, 4, 6]; but, to the best of our knowledge, only Snaplink [6] is able to handle $> 1m$ interaction. However, 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 (*i.e.*, no instrumentation in the environment) device to enable remote interaction ($> 1m$) with IoT objects.

One alternative to Camiot is to use the pervasive smartphone camera, similar to Snap-to-It [7]; however, the main concern is 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 a user to point, shoot and control an IoT object.

SYSTEM DESIGN & IMPLEMENTATION

We first provide an overview of the Camiot system, following which we detailed its two key technical components.

System Overview

Hardware platform As shown in Figure 3, we built a proof-of-concept hardware platform for exploring finger pointing and gesturing to interact with IoT. Our platform consists of

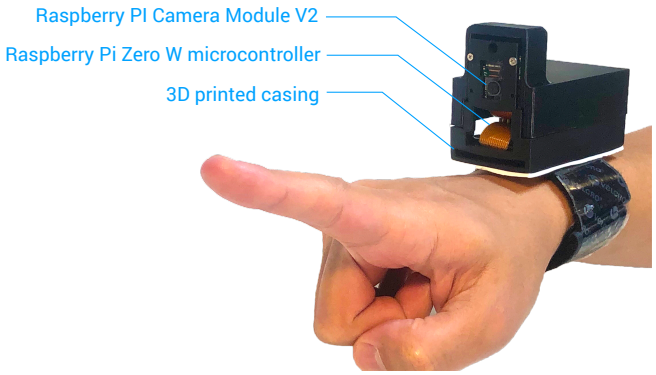

Figure 3. Our proof-of-concept hardware platform for exploring recognizing appliances and finger gestures from a wrist-worn outward-facing camera.

a Raspberry Pi Zero W as the controller, an MPU 6050 IMU sensor for providing accelerometer and gyroscope data, and a Raspberry PI Camera Module V2 for capturing IoT and the user's finger. The height of the camera (distance between its center and the user's wrist when worn) is about 4cm. The Pi 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), which was programmed in Python. For audio feedback, we use the speaker of the computer running the server program.

Pointing to select an appliance In order to enable camera-based recognition of IoT, we first need to know when a user points Camiot towards an appliance. We use the IMU module and implemented Kang *et al.*'s data-driven approach [13] to detect the motion of pointing, *i.e.*, raising one's hand and extending it outwards. Then the camera starts streaming its images to the server for appliance recognition, which is detailed later in this section.

Finger gestures to select a control option Once an appliance is selected, Camiot allows the user to interact with the appliance by using the pointing finger to select a control option. Based on the index finger's anatomical and kinesthetic properties [30], we design two gestures to support such interaction:

(i) Circumduction for selection , where the index finger first hyperextends and then rotates primarily around its Carpometacarpal joint (Figure 2b), drawing a virtual semi-circle in the camera's view (because only the upper half of the circumduction is in the camera's field of view). We divide such a virtual semi-circular plane into N sectors ($N \in \{2,3,4,5\}$), each of which corresponding to a specific control option of an appliance. For appliances that have a large number of control options, we can use the sectors hierarchically. Later in the evaluation section, we take a data-driven approach to compute the optimal thresholds for division as well as the user's performance in circumducting the index finger into each sector given different numbers of divisions.

(ii) Flexion for confirmation : once a user circumducts their finger to a desired control option, they can confirm the selection by flexing the finger (Figure 2a), similar to the 'airtap' gesture in Microsoft Hololens[2].

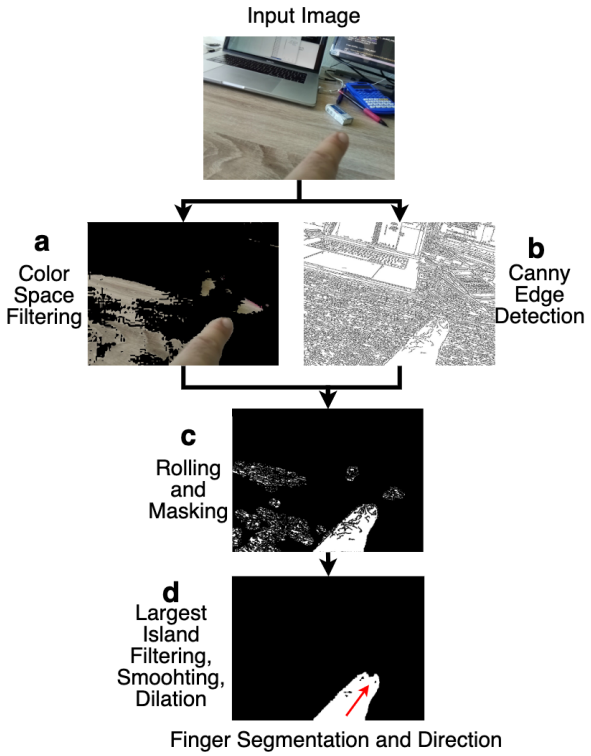

Figure 4. Compact and robust finger segmentation pipeline for finger interaction recognition.

Feedback Camiot provides audio feedback that announces the name of the recognized appliance at the end of a pointing action. When selecting different control options, a user can enable audio feedback that speaks out the name of the control option when the finger first enters the corresponding sector. Once a user confirms a selection, the system will also acknowledge it with audio feedback.

Below we focus on the two key technical components of Camiot: we first introduce an unsupervised finger gesture recognition pipeline, and how can leverage the finger's orientation to augment the recognition of an appliance, which we implement using a Convolutional Neural Network (ConvNet).

Unsupervised Finger Gesture Recognition Pipeline

In order to recognize the circumduction and flexion of the figure, Camiot performs automatic finger segmentation from the camera view. One popular method that achieves the state-of-the-art accuracy for object segmentation is by using ConvNets [24, 9]. However, such methods heavily relay on the supervised learning from large-scale data with detailed annotations, which can be very labor-consuming to label. In contrast, our method utilizes the characteristics of skin color and edge detections for compact but robust finger segmentation without any supervision.

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

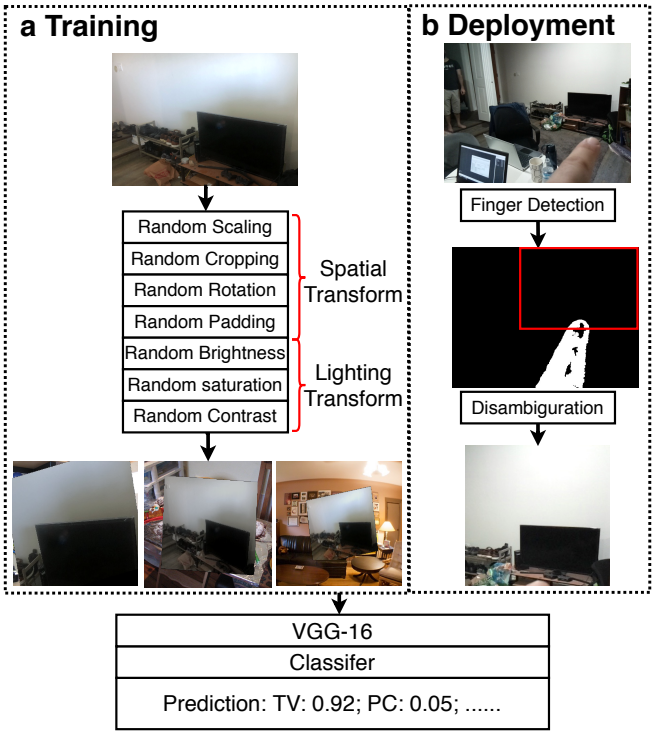

Figure 5. Appliance recognition model overview. Intensity augmentation and knowledge transfer are applied for few-shot learning from templates. Finger orientation is used to further locate an IoT object for disambiguation during deployment.

Figure 4 demonstrates the pipeline with one example image. We first derive a rough finger mask (Figure 4a) with the skin color model in YCbCr space [16] 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 (Figure 4b) is generated with Canny filter, which is then applied to the finger mask with rolling in up/down/left/right directions for one pixel, in order to cut off bordering regions with a connectivity less than four (Figure 4c). 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 linear interpreting the row-wise midpoints of the segmentation (Figure 4d). 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.

Appliance Recognition with Finger Disambiguation

Prior work on IoT device interaction, *e.g.*, Snap-to-it [7], has relied on template matching methods for object recognition: templates of each object are collected in the setup stage, and a query image is then matched to all the templates according to the pre-defined feature distances for recognition. One limitation of such method is that the object needs to occupy a larger space in the query image for reducing mismatches caused by background noises. As such, it is not optimal for our task of interacting with an IoT object from a distance, where appliances can take only a small portion in the camera view.

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 [28, 10]. Moreover, we propose to utilize the pointing finger as a source of disambiguation, and feed the model with the indicated portion of an image for prediction (Figure 7de). Such method can potentially benefit the model performance since: *(i)* model attention can be focused on the appliances with reduced areas of the background; *(ii)* the target appliance would emerge to the foreground while other appliances in the same camera view will be cropped out. As shown in Figure 5, the model consists of a shallow classifier of two fully connected layers, and a frozen VGG-16 that pre-trained on [28] for generating deep feature maps. In the setup stage, five images of each appliance are collected with Camiot taken from various angles as training data. During training (Figure 5a), intensive augmentations are applied for covering possible appearances of appliances in usage, which include: *(i)* random spatial transformations, *i.e.*, scaling, cropping, rotation, and padding with contextures extracted from the COCO datasets [18], and *(ii)* random lighting transformations, *i.e.*, shifts of contrast, saturation, and brightness. During deployment (5b), the finger segmentation is first derived automatically from the query image. Then, guided by the finger orientation, an image patch that is 0.6 times the size of as the original image is cropped with the target appliance centered, and is fed into the model for inference. We compare the performance of ConvNet model with template matching methods, and also perform ablation tests to demonstrate the effectiveness of the finger-based disambiguation.

PRELIMINARY 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.

Unsupervised Finger Gesture Recognition Pipeline

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 finger's flexing, which is used to confirm a selection.

In order to find out the most natural thresholds of angles for dividing the the virtual panel, we carried out a pilot study asking one participant (P1) to point at each sector based on their own estimation without any visual/audio reference or feedback. Specifically, we asked the user to perform three pointing task for each sector with their order randomized to avoid temporally-dependent behavior. We then repeated this process for all the four sector designs, while we logged the

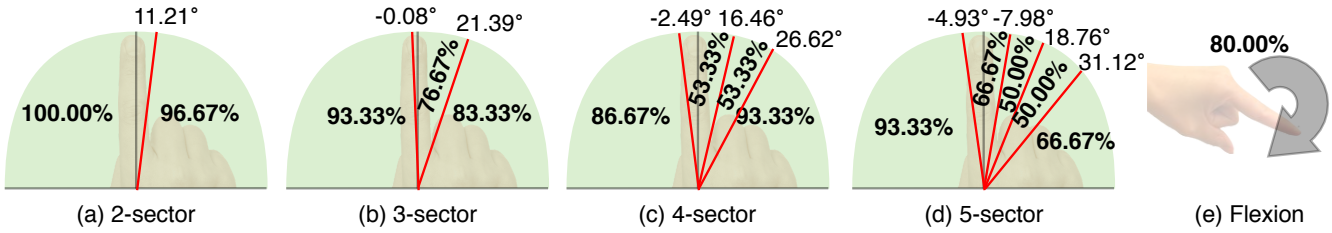

Figure 6. Finger interaction accuracy. Red lines show the determined angle thresholds for splitting the sectors. Bold numbers represent the detecting accuracy of finger pointing at the sectors and finger flexion.

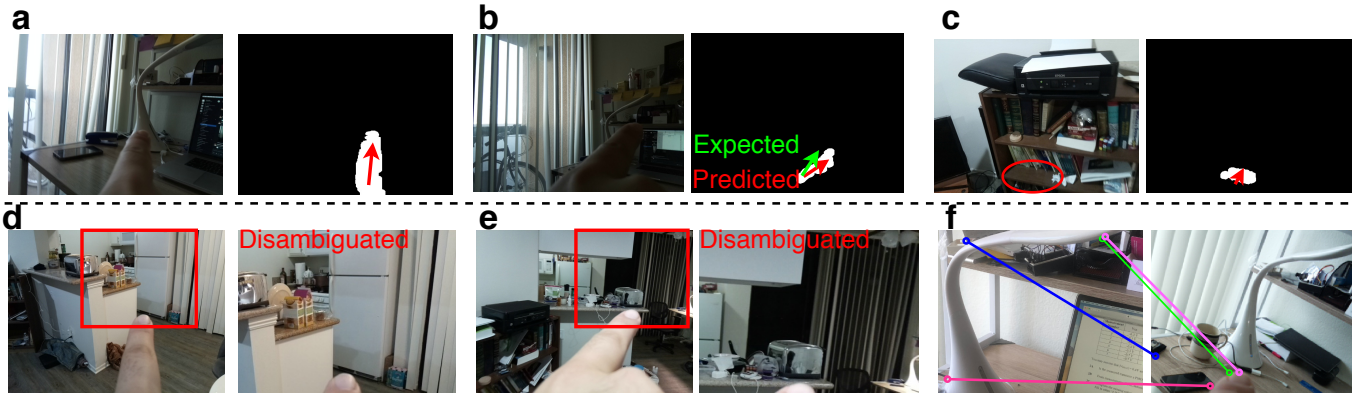

Figure 7. Case study for finger gesture recognition and appliance recognition. (a) An accurate finger segmentation with its derived direction. (b) An imperfect finger segmentation with direction discrepancy. (c) A failure case for finger flexion recognition. (d, e) Finger-based disambiguation localizes the desired appliance in the images where multiple appliances exist. (f) A typical failure case for other feature matching methods.

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. The resultant threshold are visualized as red lines in Figure 6.

Then, we informed P1 about the angle thresholds by showing him figures that similar to Figure 6, and measured how accurate P1 could point at each sector across all the four designs. We randomly selected 10 appliances from P1's household setting, and the experiment was carried out with P1 pointing at these 10 appliances in order to account for the impact of backgrounds on the finger recognition accuracy. Specifically, P1 was asked to point at each sector and then perform a finger flexing gesture. Each sector was gestured at three times in a randomized order and this process was repeated using the 10 appliances as the background, which in total results in $10\times(3+4+5+6)\times3=540$ images.

Figure 6 demonstrates the detection accuracy of selecting different sectors and flexing action as confirmation. 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. Even if the system provides a feedback loop (*e.g.*, using audio), it would take a user extra time to correct such errors and locate the correct sector.

The selection errors can mainly caused by: *(i)* a sector range is too small for the user's pointing to fall within the intended area; and *(ii)* the finger segmentation is not perfect, and there is discrepancies between the real directions and the predicted directions. Figure 7a shows an example of accurate finger segmentation and its finger direction prediction, while Figure 7b shows a typical imperfect segmentation caused by the underexposure of the image, 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 7b. For the recognition of finger flexion, our algorithm achieves a high detection accuracy of 80.00%. The flexion recognition failure can happen, as demonstrated in Figure 7c, typically when some objects in the background that have similar color as human skins lying in the lower part of the image, and being mis-predicted as a finger.

Appliance Recognition with Finger Disambiguation

In order to evaluate the accuracy of selecting appliances from distance, we build a dataset consisting of 10 appliances randomly chosen from P1's household. First, we required the participant (P1) to collect 5 templates for each appliance, all from about 0.5 meters away in order to ensure the details of the object could be captured (although our recognition can handler much longer distance, as shown below). Moreover, the templates of an appliance were taken roughly from the

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

angles that evenly divide the outward surface of that appliance to fully profile its visual appearance. Then we asked the participant to point at the appliances using Camiot. For each appliance, the participant pointed at it for 20 times from random positions. We controlled the distance between the participant and the appliances in order to study the robustness of our algorithm to distances: the participant pointed at each appliance at three ranges of distance: $\sim$2m, $\sim$4m and $\sim$6m. In total, we collected $10\times15\times3$=450 images. which we used as our appliance recognition dataset.

We compare 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 for the 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. Figure 8 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 (*e.g.*, via a wrist rotation gesture that selects the next best in the result list). Further analysis shows that our model outperforms feature matching methods in capturing features of large receptive regions: Figure 7f demonstrates a typical failure case of SIFT matching method where the object (lamp) lacks salient key points for feature matching, while our method successfully obtains the correct prediction.

Finger-based disambiguation (Figure 5) narrows down and feeds the most relevant part of the image patch into the ConvNet for inference. An ablation test shows that with such disambiguation, Camiot boost the performance of the standalone ConvNet (*i.e.*, no finger disambiguation) significantly by +8.67%, +25.33%, and +39.34%, as shown in Figure 8. As an example, both Figure 7d and e contain the same two appliances, *i.e.*, refrigerator and toaster. Guided by the finger's orientation, we can locate the desired appliances intended by the user's pointing.

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 is to provide preliminary performance results of Camiot to investigate whether our appliance and finger gesture recognition techniques (trained on P1) can generalize (to P2 and P3).

The main task was to use Camiot to interact with a new set of five appliances[3]: pointing at each of the five appliances and 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 strike a balance between 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, in order to maintain consistency and comparability with the earlier P1 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.

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 appliance took two attempts and the other three took three. Note that the performance numbers here are higher than those in Figure 8 because the number of appliances is smaller (five compared to ten).

The overall accuracy of using finger circumduction to select a sector was 76.00%; in comparison, the accuracy in the aforementioned P1 testing was 84.44%. Such a drop of performance was expected, as the optimal thresholds were determined based on P1's data. Multiple cross-user variances (*e.g.*, finger appearance, finger agility, perception of different sectors, how the

[3]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. The appliances do not overlap with the aforementioned appliance recognition dataset.

device was worn) could have contributed to the discrepancy in circumduction gesture recognition performance. On the other hand, all finger flexion gestures were correctly recognized.

Best-case response time. We profile each trial as two phases: *(i)* the appliance selection phase starts from the prompt and ends when an appliance is correctly recognized by the system; *(ii)* the control option phase starts from the system correctly recognizes an appliance and ends when the user selects the correct control option using 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 (*e.g.*, using audio) is provided, the response time will be longer as a 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, which we discussed further in the next section.

LIMITATIONS, DISCUSSION & FUTURE WORK

Based on our preliminary findings, we summarize limitations of the current system and discuss future work to address the existing issues.

Latency. Currently our integrated system experienced latency issues (running at $\sim$3 FPS), due to a combination of video capturing on an embedded device and suboptimal networking speed. To ensure that Camiot provides the benefit to not having to retrieve a remote control or an app on a personal device, our future work needs to reduce the latency by experimenting with faster cameras, more efficient video capturing routines and processing images on the edge without networking.

Lighting Condition. Lighting condition is one of the most common factors that can effect the performance of a vision-based system. In this work, we did not intentionally control the lighting when carrying out the experiments to formally study its impact. However, we have noticed the finger segmentation pipeline does get affected by lighting changes. Specifically, a lack of lighting or reflections of colored interior lighting can change the tone of the finger's color, possibly affecting our skin-color-based finger segmentation algorithm. To overcome this issue, future work could explore applying cameras that more robust against lighting changes, *e.g.*, thermal camera and depth camera, as the supplements of RGB imaging to increase the robustness of the system.

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 only referring to the finger data from one user. Such thresholds might not be representative of most users; thus our immediate next step is to develop a per-user calibration mechanism, which involves a user performing finger circumduction with Camiot to automatically determine the best thresholds for each individuals. Long-term future work should conduct a larger scale study in order 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 a large number of 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, recognition algorithm for finger actions can be further developed in the future for the purpose.

Combining with voice input. Voice has been an important modality for interacting with an IoT device. Although only the finger-based interaction was considered in Camiot, we envision that applying voice as another input modality can further improve the efficiency of the system. For example, while simple functions, *e.g.*, volume up and turn off, can be easily encoded and rapidly expressed with finger actions, more complex commands, *e.g.*, setup a countdown for two minutes, would be much easier to be specified using voice. Future research will explore how to fuse both input sources to enhance the expressiveness of Camiot.

Variation in wearing the device. During our studies, we found that the device position/orientation/tightness is different each time it was put on a user's wrist. 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 the robustness of our system against such variation.

Privacy conerns. Like any other camera-based system, privacy is a potential issue for Camiot. Possible mitigation solutions for future work include processing all images on the edge without saving the data, allowing a user to see what an image of an appliance looks like before deciding to use it as a tempalte, providing feedback (*e.g.*, vibration) to alert a user in case the camera takes a picture due to a false positive trigger.

REFERENCES

1. Amr Alanwar, Moustafa Alzantot, Bo-Jhang Ho, Paul Martin, and Mani Srivastava. 2017. 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)*. ACM, New York, NY, USA, 47–58. DOI: http://dx.doi.org/10.1145/3054977.3054981

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

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

4. Roger Boldu, Alexandru Dancu, Denys JC Matthies, Thisum Buddhika, Shamane Siriwardhana, and Suranga Nanayakkara. 2018. 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 (2018), 94.

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

6. Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 129 (Jan. 2018), 27 pages. DOI: http://dx.doi.org/10.1145/3161173

7. Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. 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)*. ACM, New York, NY, USA, 5909–5920. DOI: http://dx.doi.org/10.1145/2858036.2858177

8. Anhong Guo, Xiang 'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P Bigham. 2016. 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*. ACM, 651–664.

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

10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). (2015).

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

12. Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018. Scenariot: Spatially Mapping Smart Things Within Augmented Reality Scenes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 219, 13 pages. DOI:http://dx.doi.org/10.1145/3173574.3173793

13. Runchang Kang, Anhong Guo, Gierad Laput, Yang Li, and Xiang 'Anthony' Chen. 2019. Minuet: Multimodal Interaction with an Internet of Things. In *To Appear at the ACM symposium on Spatial user interaction*. ACM.

14. David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 167–176.

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

16. S Kolkur, D Kalbande, P Shimpi, C Bapat, and J Jatakia. 2017. Human skin detection using RGB, HSV and YCbCr color models. *arXiv preprint arXiv:1708.02694* (2017).

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

18. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

19. B. D. Mayton, N. Zhao, M. Aldrich, N. Gillian, and J. A. Paradiso. 2013. WristQue: A personal sensor wristband. In *2013 IEEE International Conference on Body Sensor Networks*. 1–6. DOI: http://dx.doi.org/10.1109/BSN.2013.6575483

20. Katsunori Ohnishi, Atsushi Kanehira, Asako Kanezaki, and Tatsuya Harada. 2016. Recognizing activities of daily living with a wrist-mounted camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3103–3111.

21. Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 31–40.

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

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

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

25. Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 483–492.

26. Dominik Schmidt, David Molyneaux, and Xiang Cao. 2012. 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*. ACM, 379–388.

27. Roy Shilkrot, Jochen Huber, Wong Meng Ee, Pattie Maes, and Suranga Chandima Nanayakkara. 2015. FingerReader: a wearable device to explore printed text on the go. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2363–2372.

28. Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

29. Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 812–817. DOI: http://dx.doi.org/10.1145/2901790.2901867

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

31. Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Deus EM Machina: on-touch contextual functionality for smart IoT appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4000–4008.

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

33. Ben Zhang, Yu-Hsiang Chen, Claire Tuna, Achal Dave, Yang Li, Edward Lee, and Björn Hartmann. 2014. HOBS: head orientation-based selection in physical spaces. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*. ACM, 17–25.