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

ANONYMOUS AUTHOR(S)

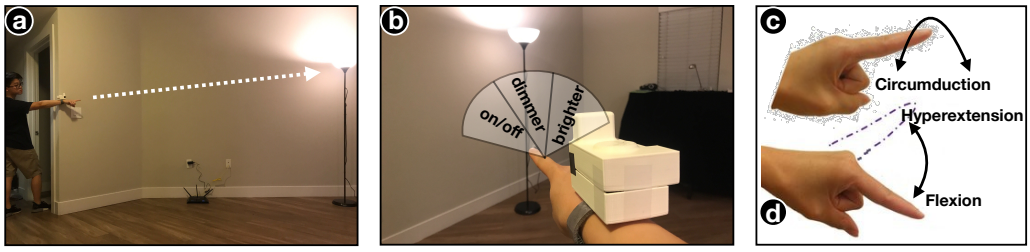


Fig. 1. Using *Camiot* to turn off a floor lamp before leaving his apartment: (a) a user points towards an IoT appliance, which enables the camera-based recognition of the appliance at a distance; (b) once an appliance is selected, the user interacts with the appliance using the same pointing finger to select a control shortcut with circumduction (c), and to confirm the selection with flexion (d).

An Internet of Things (IoT) is 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 to an IoT object. To achieve this goal, we develop *Camiot*—the first wrist-worn platform that uses an outward-facing camera to recognize and interact with IoT objects at a distance. *Camiot* is novel by leveraging the gesture of user's index finger: (*i*) it utilizes finger's pointing orientation for guiding the recognition of an IoT object in the camera view; and (*ii*) it detects finger's circumduction and flexion for the function selection of a selected IoT object. To validate *Camiot*, we experimentally measure its performance with household appliances pointed at from various distance/angles; further, a user study with eight participants assesses the feasibility of using *Camiot* to interface with household appliances.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**.

Additional Key Words and Phrases: Camera, an Internet of Things, gesture interaction, gesture recognition

ACM Reference Format:

Anonymous Author(s). 2018. Camiot: Recognizing & Interacting with Distant IoT Objects Using a Wrist-Worn Outward-Facing Camera. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 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

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

instrumenting the environment for voice or gesture based control. 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 user's index finger's orientation for locating and interacting with an IoT object in the camera view. *Camiot* complements existing IoT interaction by providing a small number of control shortcuts that are available as a user points at an appliance. Figure 1 illustrates an application scenario of *Camiot*: a user starts by raising the arm to point at an Internet-connected floor lamp, which is then located and recognized via the outward facing camera (Figure 1(a)); then, the user circumducts the index finger (Figure 1(c)) to select one of the three control shortcuts with audio feedback (Figure 1(b)); and finally the user flexes the finger (Figure 1(d)) to confirm the function selection (Figure 1(b)).

We validate *Camiot* with: (*i*) an experiment of finger circumduction/flexion to estimate the optimal parameters for dividing the camera-view space for function selection; (*ii*) a validation study of pointing at 10 appliances in a real-world household setting at various distances/directions, which measures the performance of our appliance recognition technique; *(iii)* a user study with 8 participants where *Camiot* on average took participants 7.45s to perform a pointing and finger selection task of an appliance's function and the accuracy was consistent with the validation results. Interviews with participants indicate that appliance selecting with pointing is intuitive, finger-based control option selection needs personally-calibrated angle thresholds, and the system is useful for many home scenarios.

Related Work. Prior work has explored using a handheld or wearable camera to interact with objects in the physical world. At a very close (*mm*) distance, miniaturized cameras can identify objects by their textures [16], convert visual information into haptic feedback [6], or recognize text on a book [13]. A little further away (*cm*), a wrist-mounted camera can detect handheld objects [9, 12], infer gestures from back-of-hand depth images [17], or reconstruct hand shapes via thermal imaging [7]. At $\sim 1m$ distance, a camera allows users to select a nearby object [1, 3, 4]. Finally, at distances over $1m$—most related to our interest on remote IoT interaction—SnapLink [2] 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 us to develop 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 smartphone camera, similar to Snap-to-It [3]; 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.

2 DESIGN

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 [8] to detect the motion of pointing, *i.e.*, raising one's hand and extending it outwards. Then the camera starts streaming its images for the Convolutional Neural Network (ConvNet) based appliance recognition, which is detailed later in this section. *Camiot* provides audio feedback by speaking out the name of the recognized appliance pointed by a user.

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 [15], 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 1(c)), drawing a virtual semi-circle in the camera's view (because only the upper half

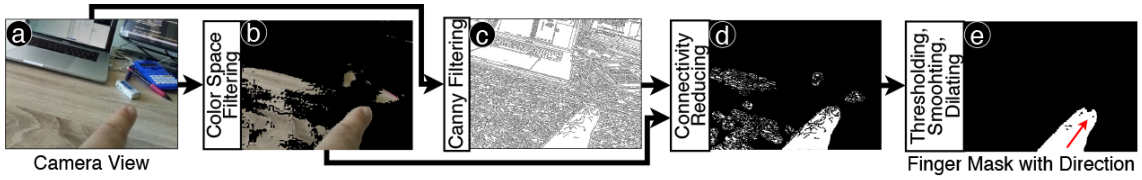


Fig. 2. Unsupervised finger segmentation from a camera view.

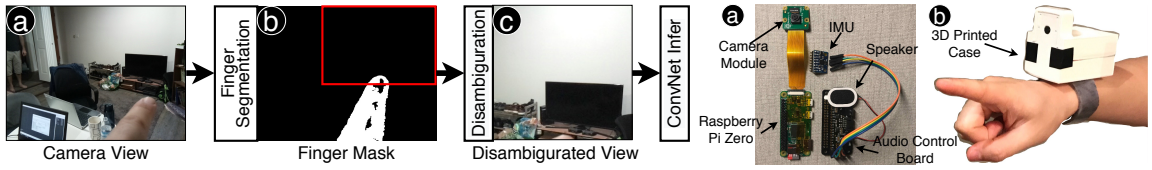


Fig. 3. (a): Finger-based disambiguation. (b) Hardware platform of *Camiot*.

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. When selecting different control options, audio feedback speaks out the name of the control option when the finger first enters the corresponding sector, assisted by which a user can adjust finger for a desired function.

(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 1d), similar to the 'airtap' gesture in Microsoft Hololens[2]. Once a user confirms a selection, the system will also acknowledge it with audio feedback.

3 IMPLEMENTATION

Unsupervised Finger Segmentation. Our method utilizes the characteristics of skin color and edge detections for the unsupervised finger segmentation. As shown in Figure 2, a rough finger mask is first generated by applying a skin color filter in YCbCr space [10] that is adaptive for general populations (Figure 2(a→b)). Because of background noises, multiple false positive regions can exist in this initial segmentation map. Thus, an edge mask is meanwhile generated with a Canny filter (Figure 2(a→c)), and is applied to the finger mask by rolling in each of up/down/left/right directions for one pixel, in order to cut off bordering regions with a connectivity less than four (Figure 2(d)). Among the masked segments, we take the largest isolated region with a size larger than a preset threshold and locates at the a lower portion of the image as the finger segment prediction, and further derive the finger direction by smoothing, dilating, and linear interpreting the row-wise midpoints of the segmentation (Figure 2(e)).

Appliance Recognition with Finger Disambiguation. *Camiot* utilizes a ConvNet of VGG-16 [14] optimized from few shot images of registered IoT objects for object recognition: in the setup stage, five images of each appliance are required to be collected as training data; while in the deployment stage, the trained model is downloaded for inference. Moreover, we utilize the pointing finger as a source of disambiguation, and feed the model with the indicated portion of an image for prediction as shown in Figure 3a. Specifically, guided by the finger orientation, an image patch that is 0.6 times the size of original image is cropped with the target appliance centered, and is fed into the model for

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

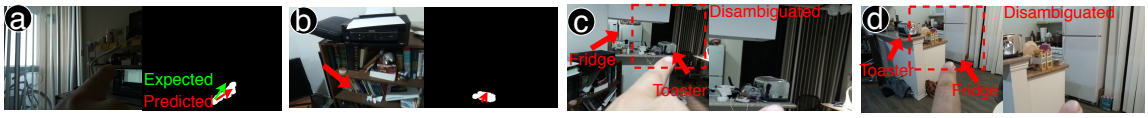


Fig. 4. (a) inconsistent finger direction estimation, (b) false negative detection of finger flexion. (c,d): finger-based disambiguation.

inference. The method can benefit the model performance because: *(i)* model attention can be focused on the appliances with reduced 'noises' from the background; *(ii)* the intended appliance would emerge to the foreground while other appliances in the same camera view will be cropped out. A more detailed description on the few-shot training of model can be found in Supplementary Materials A.1.

Hardware platform. Figure 3b shows our proof-of-concept hardware platform with a Raspberry Pi Zero as the controller, an MPU 6050 IMU sensor as accelerometer, a Raspberry PI Camera Module V2 for vision, and a mini speaker for audio feedback. With the 3D printed case, the height of the camera (distance between its center and the user's wrist when worn) is about 4cm. All the computations except for appliance recognition are done on *Camiot*, *i.e.*, arm motion detection and finger gesture recognition. For appliance recognition, captured images are sent to a PC server via a socket connection for ConvNet inference, mainly due to the limited computing resource of the Raspberry Pi Zero.

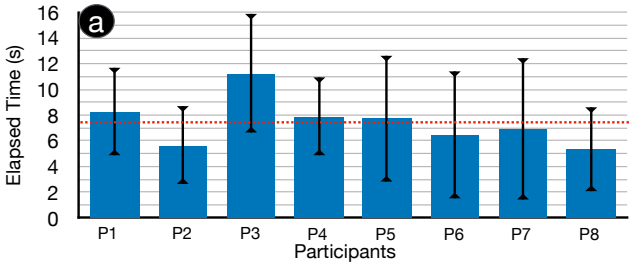
4 VALIDATING RECOGNITION PARAMETERS & MODELS

In this section, we validate the parameters and models for the two key techniques of figure gesture recognition and appliance recognition with one research member (RM) as an user.

4.1 Finger Gesture Recognition

Angle thresholds for dividing sectors. In order to find out the optimal number of sectors for the the virtual panel and the thresholds of angles for dividing the sectors, the RM was asked to point at each sector based on his own estimation via finger circumduction on a virtual panel with different numbers of sectors and without any visual/audio reference or feedback. Virtual panel designs with numbers of sectors from 2 to 5 were tested. The collected images were processed with the aforementioned figure gesture recognition algorithm to calculate angles. Based on this data, we set the optimal thresholds to be the angles that best split different sectors based on an exhaustive search. Details of the resultant thresholds for different sector designs are visualized in Figure 7 from Supplementary Materials A.2.

Pointing accuracy on different sector designs. We then evaluated the pointing accuracy on different sector designs under the determined angle thresholds, as well as the detection accuracy of finger flexion for function confirmation. The RM was first informed about the angle thresholds obtained earlier. Then, he was asked to point at each sector for different sector designs, confirmed by a finger flexing gesture. 10 appliances were selected as the background to account for its impact on the finger recognition accuracy. Detailed experiment setups can be found in Supplementary Materials A.2. The results show 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, and detection accuracy of finger flexion was 80.00%. Figure 7 from Supplementary Materials A.2 details the results. 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. By analyzing the figure segmentation results, we identify the following sources of error: *(i)* the sector range too small and the user's pointing falls out of the intended area; *(ii)* the finger segmentation's angular error (*e.g.*,



b

1: Strongly disagree — 7: Strongly agree									
P1	P2	P3	P4	P5	P6	P7	P8	Mean	Std
(S1) The procedure of appliance selection by pointing was demanding.									
2	2	1	1	1	2	2	1	1.50	0.53
(S2) The procedure of function selecting by finger gestures was demanding.									
3	1	3	3	4	4	7	4	3.63	1.69
(S3) I would use Camiot as an IoT object controller at home.									
6	5	3	6	5	3	2	6	4.50	1.60

Fig. 5. (a) Time cost for a trial. (b) Usability and preference of *Camiot*.

caused by over or under lighting causes finger direction discrepancy in Figure 4(a)); and *(iii)* objects of similar color as human skins mis-predicted as part of the finger (*e.g.*, environment noise triggers finger flexion in Figure 4).

4.2 Appliance Recognition

We evaluated the accuracy of appliance recognition from different distances and ablated the effect of finger disambiguation for accuracy. We built a dataset consisting of 10 appliances randomly chosen from the RM's household using *Camiot*. For each appliance, the RM pointed at it from controlled distances between the participant and the appliances of ~2m, ~4m and ~6m. Detailed experiment setups can be found in Supplementary Materials A.3.

Accuracy at distances. 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 model. The results show that our model achieves top 1 hit rates of 96.00%(2m), 77.33%(4m) and 60.67%(6m). Moreover, our model 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). We also compared our model with feature matching algorithms that have been widely applied in previous IoT object recognition work [2, 3], which shows our method outperforms both methods at all distance ranges. Detailed results can be found in Figure 8 in Supplementary Materials A.3.

Ablation on finger-based disambiguation. We compared the ConvNet model accuracy with and without finger disambiguation. The results show that model performance was boosted by figure disambiguation for all distance ranges, and the boosting effect is more significant for larger distances (all for top 1: 87.33% →96.00%@2m, 52.00% →77.33%@4m, 21.33% →60.67%@6m). It can be explained as the desired appliance taking a smaller area in the image with larger distances, and might need finger's hint to be distinguished from environmental noises. For example, both Figure 4(c,d) contain two appliances in the camera view, *i.e.*, refrigerator and toaster, and our model achieved correct predictions with the aid of finger direction as disambiguation. Detailed results can be found in Supplementary Materials A.3.

5 USER STUDY

Participants & procedure. We recruited 8 participants from a local university with an age ranging from 21 to 23 (5 male and 3 female). The study took place in another research member's household different from the one in the validation experiment, and 10 appliances[3] from that household were selected as the target IoT objects. Based on the finger gesture recognition measured earlier, we chose a 3-sector design to strike a balance between the number of options and the accuracy of locating each sector and the optimized angle thresholds (Figure 7) were applied. Before

[3]All appliances were not Internet-connected, thus their control options were just proof-of-concept mock-ups that provide audio feed-back without real functionalities. The appliances from user study and appliance recognition validation had no overlap.

the study, each participant was randomly assigned 5 appliances to interact with, and for each appliance one of the three control options (left, middle, or right) was assigned. Since the appliances were not smart or Internet-connected, we did not actually implement any control options. Based on the appliance assignment, a separate recognition model was trained for each user. During the study, each participant was asked to point at each of the 5 appliances and follow audio prompts to perform index finger based selection and confirmation of control options. A participant could perform the task anywhere at least 6m away from the target appliance. Each participant only pointed at each appliance once regardless of the recognition correctness. In this way, we tried to test the system in the worst condition—one-shot trial at the largest distance we validated. In total, there were 8 (participants) × 5 (appliance, control option) = 40 trials.

Performance. We measured the elapsed time of each trail starting from the appliance recognition function being enabled (arm raised) to the successful selection of an appliance's control option, as shown in Figure 5(a). We note that there were cross-user variances in the time cost for performing one appliance control, which can be caused by: (*i*) the angle thresholds applied were determined based on a different user's data, which can not be optimal for all because of individuals' finger agility and perception of sectors are different; and (*ii*) how the device was worn could also contribute to the cross-participant variance. A total of 24 trials (60%) were successful for appliance recognition, which is consistent to the results of validation experiments with an appliance-user distance of ~6m (shown in Supplementary Material A.2); amongst the 24 successfully-recognized appliances, all of the sector selections were correctly recognized.

Usability & Preference. We interviewed each participant for the usability of *Camiot* with statements about the two key steps: (S1) selecting an appliance by pointing, and (S2) selecting a control option by figure gesturing. We also asked the preference of using *Camiot* as an IoT controller at home (S3). Figure 5(b) shows the results. All the participants agreed the appliance selection was simple and intuitive. For finger gesturing, some participants reported difficulty on getting the expected functions with his/her limited degree of finger movement (P5, P6, P7), which can be caused by improper angle threshold setting. Moreover, most participants prefer to use *Camiot* since it is intuitive and fun (P1), can be useful for the tired and disabled (P4, P5), and can be very helpful in situational impairment, *e.g.*, cooking (P8), while some concerned about the appliance recognition accuracy (P3, P7).

6 LIMITATIONS, DISCUSSION & FUTURE WORK

(i) **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, and can cause troubles for some participants during function selection. 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. *(ii)* **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 research member. 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. Another solution might be conducting a larger scale study in order to generalize the optimal threshold angles for the panel. *(iii)* **Appliance recognition accuracy.** our work shows the finger direction in camera view can be applied to help recognition appliances at distance. However, both the validation experiments and user study indicate there is still room for accuracy improvement. One promising solution is to increase the training dataset size in the device setup stage. It can be done by asking a user to collect more images of an appliances from different distances and angles. Moreover, pre-built image set, *e.g.*, collected images for a certain model of an appliance, can be used to augment the training process.

REFERENCES

[1] 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.

[2] 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. https://doi.org/10.1145/3161173

[3] 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* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177

[4] 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.

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

[6] 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.

[7] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 71 (June 2020), 24 pages. https://doi.org/10.1145/3397306

[8] 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.

[9] 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.

[10] 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).

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 963–971. https://doi.org/10.1145/3332165.3347867

A SUPPLEMENTARY MATERIAL

A.1 Appliance Recognition ConvNet Model

Prior work on IoT device interaction, *e.g.*, Snap-to-it [3], 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 image features of different scales with parameter optimizations [5, 14]. As shown in Figure 6, the model consists of a shallow classifier of two fully connected layers, and a frozen VGG-16 that pre-trained on

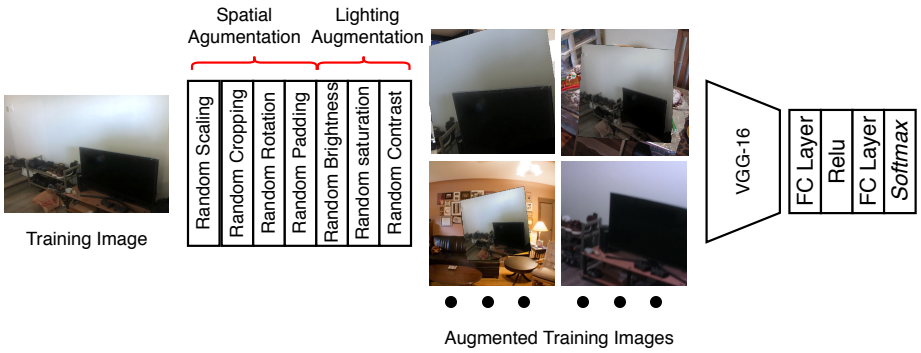


Fig. 6. Model architecture diagram and training data augmentation.

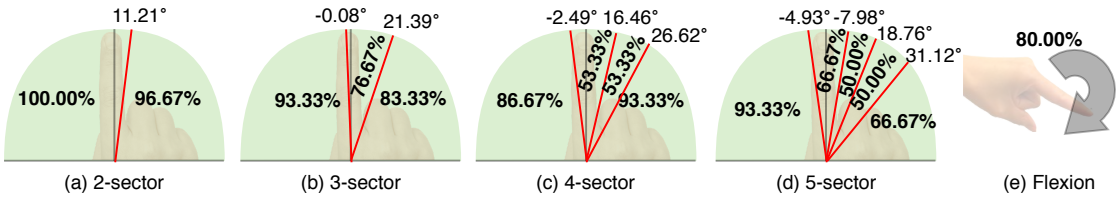


Fig. 7. Finger interaction accuracy. Red lines show the determined angle thresholds for splitting the sectors. Bold numbers represent the detecting accuracy of finger pointing at the sectors and finger flexion.

[14] for generating deep feature maps. The pre-trained model is applied and frozen during the training process to reduce the model over-fitting caused by the limited training data. In the setup stage, five images of each appliance are collected with *Camiot* taken from various angles as training data. During training (Figure 6a), 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 contexture extracted from the COCO datasets [11], and *(ii)* random lighting transformations, *i.e.*, shifts of contrast, saturation, and brightness. Such augmentation is applied in order to increase the model generalization of dealing with input images from different directions/distance and with various lighting conditions.

A.2 Pilot Study on Finger Gesture Recognition

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 research member to point at each sector based on his own 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 then repeated this process for all the four sector designs (sector number $N \in \{2, 3, 4, 5\}$) 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. The resultant threshold are visualized as red lines in Figure 7.

Then, to evaluate the model accuracy on detecting the selection via finger circumduction and flexion, we informed the member about the angle thresholds, and measured how accurate he could point at each sector across all the four

	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**

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

designs. We randomly selected 10 appliances from the member's household setting, and the experiment was carried out with pointing at these 10 appliances in order to account for the impact of backgrounds on the finger recognition accuracy. Specifically, the member 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×(3+4+5+6)×3=540 images. Figure 7 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. For the recognition of finger flexion, our algorithm achieves a high detection accuracy of 80.00%.

A.3 Pilot Study on Appliance Recognition

In order to evaluate the accuracy of selecting appliances from distance, we build a dataset consisting of 10 appliances randomly chosen from the research member's household. First, we required the member 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 handle much longer distance). Moreover, the templates of an appliance were taken roughly from the angles that evenly divide the outward surface of that appliance to fully profile its visual appearance. Then we asked the member to point at the appliances using *Camiot*. For each appliance, the member pointed at it for 20 times from random positions. We controlled the distance between the member and the appliances in order to study the robustness of our algorithm to distances: the member 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 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. 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. Moreover, the ablation of finger-based disambiguation shows that with such disambiguation, Camiot boosts 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.