

Selecting Home Appliances with Smart Glass based on Contextual Information

Quan Kong^{*}, Takuya Maekawa
 {kong.quan,maekawa}@ist.osaka-u.ac.jp
 Graduate School of Information Science and
 Technology, Osaka University

Taiki Miyanishi, Takayuki Suyama
 {miyanishi,suyama}@dent.osaka-u.ac.jp
 ATR Brain Information Communication
 Research Laboratory Group

ABSTRACT

We propose a method for selecting home appliances using a smart glass, which facilitates the control of network-connected appliances in a smart house. Our proposed method is image-based appliance selection and enables smart glass users to easily select a particular appliance by just looking at it. The main feature of our method is that it achieves high precision appliance selection using user contextual information such as position and activity, inferred from various sensor data in addition to camera images captured by the glass because such contextual information is greatly related in the home appliance that a user wants to control in her daily life. We design a state-of-the-art appliance selection method by fusing image features extracted by deep learning techniques and context information estimated by non-parametric Bayesian techniques within a framework of multiple kernel learning. Our experimental results, which use sensor data obtained in an actual house equipped with many network-connected appliances, show the effectiveness of our method.

ACM Classification Keywords: H.3.4 Information storage and retrieval: Systems and software.

Author Keywords: Wearable computers; smart glass; home appliances

INTRODUCTION

Various home appliances have been connected to home networks and enable the easy acquisition of such information about the appliances as their working status or energy consumption. In addition to acquiring information about appliances, we can control them by home networks using, for example, wearable or handheld computers. The following are the advantages of controlling home appliances with wearable computers: (1) appliances can be controlled even when their remote control devices are not nearby and (2) if we can achieve hands-free operations, e.g., with voice control, we can easily manage the appliances even when our hands are occupied. This hands-free approach is also effective for the disabled.

^{*}Quan Kong is now working for Hitachi, Ltd.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UbiComp '16, September 12–16, 2016, Heidelberg, Germany

© 2016 ACM. ISBN 978-1-4503-4461-6/16/09...\$15.00

DOI: <http://dx.doi.org/10.1145/2971648.2971651>

When a user wants to control a home appliance, she should first select one from the many appliances that exist in her house. In ubiquitous computing and HCI research areas, various methods for selecting home appliances have been developed. For example, methods based on voice, camera images, gestures, and pointing (beacon) [6, 31, 32] have been studied. A method based on pointing, for example, requires that an infrared receiver or beacon be attached to each appliance. Methods based on voice, camera images, and gestures suffer from a trade-off problem between a user's burden and scalability. For gesture-based methods, users should remember a wide variety of gestures when they want to identify an appliances from many appliances. For voice-based methods, it is difficult to identify an appliance using short descriptions such as "curtain." Therefore, users must often rely on long descriptions to accurately identify an appliance: "curtain on the kitchen's north wall." Because speech recognition accuracy for long descriptions is degraded when users speak fast, they must speak slowly. For image-based methods, while a user's burden is low, it is difficult to identify appliances with similar appearances. In this paper, we attempt to cope with the scalability problem related to the above methods by disambiguating a user's vague query using the user's context obtained from sensors.

Here we explain how the context information is utilized in the appliance selection methods. As for the gesture-based method, the user only has to remember a gesture for each appliance class. For example, when the user wants to turn on a TV in a bedroom, the user just has to perform a gesture corresponding to TV. Using the context information of the user, we can distinguish the TV in the bedroom from the other TVs. The context information helps the sound-based method in the similar way. By just saying "TV" in a bedroom, we can distinguish the TV in the bedroom from the other TVs. Specifically, this paper focuses on image-based appliance selection using wearable/portable devices such as smartphones and smart glasses because the recent rapid advances in the wearable technologies and deep learning techniques will soon make this approach feasible.

In this study, we focus on a method for selecting an appliance using a smart glass with a camera such as Google Glass. The method is image-based appliance selection that enables smart glass users to easily select an appliance by just looking at it (technically, just turning the head to it). By using the camera on the glass that captures the head direction, we can obtain an object located in the direction and estimate an appliance that the user wants to select. As mentioned above, image-based

approaches suffer from the scalability problem, including the difficulty of identifying appliances with similar appearances and appliances with few distinguishing image features. To cope with the problem, we exploit the user contextual information inferred by using sensors on the glass to supplement image-based appliance selection. Our method mainly employs the indoor positional information and activity information of users as contextual information that is greatly related to the home appliance that a user wants to control. For example, when a user is in a bedroom, she might want to control its lights and its television. When a user is cooking, she might want to control cooking-related appliances. Using such information, we disambiguate the results of image-based selection, e.g., distinguishing between a bedroom air conditioner and a kitchen air conditioner. Note that our method recognizes user positions and activities in an unsupervised manner, so users do not need to prepare labeled training data for estimating positions and activities.

Our appliance selection architecture is designed based on the state-of-the-art techniques attracting attentions in the machine learning and ubiquitous computing research areas. We extract image features from the camera image by using the deep neural network [16] that has received considerable attention in recent years. Our method also detects a user's activity and position with an unsupervised manner by the infinite Gaussian mixture model (IGMM) [28] with non-parametric Bayes approach. To deal well with the features extracted from various sensors by the different state-of-the-art methods, we design an appliance selection model based on multiple kernel learning (MKL), which is designed to handle data with different data distributions.

To the best of our knowledge, this is the first study that utilizes positional and activity information for appliance selection. The following are this paper's research contributions: (1) we design a state-of-the-art appliance selection method using contextual information by combining deep learning techniques and unsupervised learning techniques based on multiple kernel learning; (2) we investigate our method's performance in a smart house that is equipped with appliances that can be controlled by its home network; (3) our experiment in the smart house revealed that positional and activity information greatly improved appliance selection performance.

RELATED WORK

Appliance selection and control methods can be roughly divided into voice-, gesture-, beacon-, eyesight-, and vision-based approaches.

Voice

As mentioned in the introduction, voice-based methods have been developed [6, 27]. For example, Christensen et al. [6] developed a cloud-based voice control system for home appliances. However, users must often rely on long descriptions to accurately identify an appliance. Also the methods suffer from daily life noises.

Gesture and beacon

The gesture-based approach employs body-worn sensors for selecting and controlling appliances [31]. Body-worn cam-

eras or hand-worn inertial sensors are usually used to recognize hand gestures. In this approach, since one gesture is associated with each appliance, users have difficulty remembering the association between a gesture and an appliance when there are many appliances in their house.

The beacon-based approach utilizes infrared or LED beacons (or receivers) attached to home appliances for selecting appliances [26, 32]. After selecting an appliance, gesture- or voice-based methods are sometimes used to control it. For example, Neßelrath et al. [26] employed a Wii Remote's infrared sensor to capture the infrared signals emitted from the infrared beacons attached to appliances. However, this approach requires that end users install a beacon to each appliance.

Using a head-mounted computing device is a direct way for people to interact with physical objects because their attention can be a strong indicator of their interest when they turn their heads to the physical objects. Zhang et al. [33] propose a glass type device by attaching an infrared (IR) emitter onto the glass to control the appliance. They also attach an IR receiver to each appliance for responding IR signal and communicating with the glass to receive commands from the user. While our solution also uses a glass type device, our method does not require any attachment to an appliance.

Vision and eyesight

Vision-based methods mainly use a wearable or smartphone camera for detection and specifying the target [24, 25, 29], and some of the systems require large or obtrusive tags. The eyesight-based approach employs a camera image that captures a user's eye gaze direction to select an appliance [7, 30]. These studies detect eye-gaze direction by actually sensing eye movements. The vision-based and eyesight-based approaches have several limitations, including the difficulty of identifying appliances with similar appearances and appliances with few distinguishing image features. Also, these approaches do not work well in the dark environment, and users cannot select an appliance outside of their sight. Our solution is also a vision-based approach but requires no tags and utilizes contextual information such as a user's position and activity in addition to the images. In this study, we attempt to improve the performance of the image-based appliance selection using the contextual information.

Although our method is based on the vision-based approach, we believe that our method, which enhances the appliance selection performance with a user's context data, is also applicable to other appliance selection approaches. For example, we can improve the voice recognition accuracy for voice-based appliance selection using the user's location as prior knowledge to distinguish between a bedroom TV and a lounge TV.

SYSTEM OVERVIEW

Assumed setting

In this study, we assume that a user wears a smart glass such as Google Glass. The smart glass is connected to a home network of their house and such home appliances as televisions, air conditioners, and lights are also connected to the

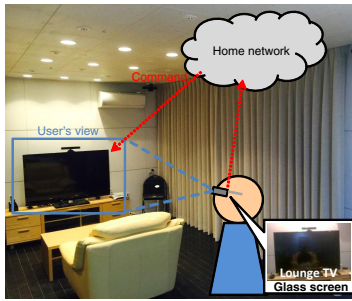


Figure 1. Our assumed setting in this study.

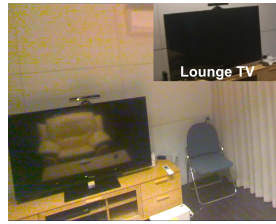


Figure 2. Example view of our application user. Upper right shows a card for a television shown on a smart glass's screen. Card has a photo of the appliance and its name.

home network. Users control the appliances using the glass by the network. When the glass is not equipped with a Wi-Fi module, we use a smartphone possessed by the user that is paired with the glass as a hub. The appliances are controlled by the glass via the smartphone. Figure 1 shows our assumed setting. In this example, because a user focuses on a television, the television is selected and this information is shown on the glass's screen. Also, Figure 2 shows an example view of the user that is created from an image captured by the glass during our experimental period.

Our glass application

Before explaining our method for automatically selecting appliances, we explain our basic system for manually selecting and controlling home appliances. In our glass application, a card consisting of an appliance's photo and its name is prepared for each appliance, as shown in the upper right of Figure 2. (Google Glass uses a card interface like Google Now which could be swiped left and right using the touch pad on the right of the frame.) The user first finds a card corresponding to the appliance that she wants to control by swiping a touch pad built into the side of the glass and selects the card by tapping the touch pad. After selecting the appliance (card), the user selects which appliance operation she wants to perform by using the touch pad, e.g., turning an air conditioner on or off. Because the house has many appliances, selecting just one target from many is laborious.

As mentioned in the introduction, we develop a method that enables smart glass users to select an appliance by just looking at it. The appliance is estimated by our method with no touch pad input but with sensor data, and its corresponding card is automatically shown on the glass's screen. If the estimation is correct, i.e., the estimated appliance is the one the user wants to operate, the user selects the estimated appliance by tapping the touch pad and then operates it. If the estimation is wrong, the user manually selects the correct one with the touch pad. With this approach, we can obtain labeled training data, i.e., appliance name and corresponding sensor data, from the user's daily life without placing large burdens on the user. Because we deal with the n -class classification problem, classification errors are unavoidable when n is large.

Although users manipulate appliances with a touch pad in our implemented system, we can also implement a function

that enables appliance control with voice input. For example, users can easily turn on an air conditioner by just looking at it and saying: "ON." Compared with a method that requires a user to say, "turn on the bedroom air conditioner," this method requires a very short phrase or word, "ON," and consequently the error rate of the speech recognition will be low.

System installation

Next we explain how our glass application is installed and set up in houses. Each appliance in the house has an identifier, such as a MAC address as well as a factory default name like Sony W800B LED HDTV. We assume that the tentative factory name or ID is initially registered to a card of our glass application. In the installation period, we assume that users name the appliances by changing the tentative factory name to, for example, *bedroom television* and take a photo of it using the glass. Our method learns an initial appliance selection model using the photo. After that, our method iteratively updates the model using the labeled sensor data obtained when the user routinely controls appliances every day. In contrast, context recognition does not require labels because it is based on unsupervised learning. Unlike appliance labels, which are automatically collected in the users' daily lives, preparing activity and location labels places an additional burden on the users. By doing so, we can reduce the burdens of collecting labeled training data.

APPLIANCE SELECTION METHOD

Overview of appliance selection

Figure 3 shows an overview of our method that estimates to which appliance the user pays attention using sensor data collected by the glass. Our method first detects the user's attention using orientation data. After that, it estimates the appliance at which the user looks.

We use a camera, a light sensor, and an orientation sensor on the glass and an acceleration sensor, a microphone, and a Wi-Fi module on a smartphone that the user carries as sensors. (Note that we can use an acceleration sensor, a microphone, and a Wi-Fi module on the glass. However, due to the limitation of the processing power of the glass, our current implementation uses these sensors on the smartphone.) The camera captures an image of the head (face) direction, and the orientation sensor captures the head direction. The acceleration sensor, the light sensor, and the microphone capture the user's activity information, and the Wi-Fi module captures her indoor position information by using the signal strengths from Wi-Fi access points located in or outside the environment of interest, i.e., Wi-Fi indoor positioning. Our method detects her activity and position with an unsupervised manner by the infinite Gaussian mixture model (IGMM) [28] without using any labeled training data because preparing labeled training data in a user environment is costly. Also, we extract image features from the camera image by using the deep neural network [16] that is reported to have the best performance in the object recognition task. As above, we extract features from camera and sensor data by using the state-of-the-art approaches that received considerable attention in the computer vision and ubiquitous computing research areas. Then the extracted image features and the detected activity and positional

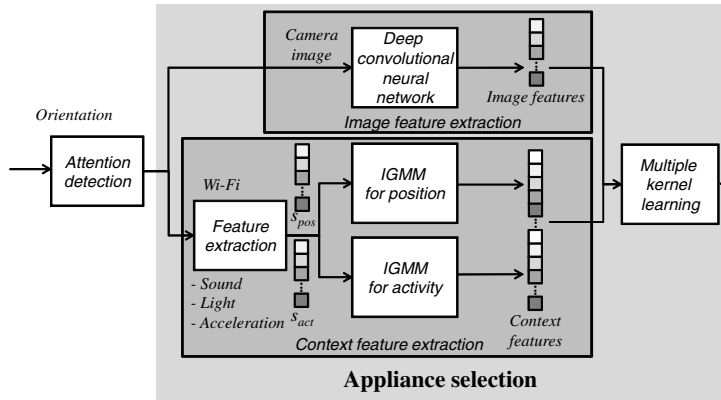


Figure 3. Overview of our proposed method.

information become inputs of an appliance selection model that is constructed based on multiple kernel learning (Figure 3), which can deal well with data that consist of features from different sources (sensors). We explain our method in detail below.

Detecting attention

In our system, a user looks at an appliance that she wants to operate. When the appliance is not in front of her face, she rotates her head, sets her face toward the appliance, and fixates her head. Figure 4 shows an example of time series orientation data obtained from the glass when an experimental participant moved to a lounge and turned on an air conditioner and a television after cooking. The red, green, and blue lines show x-, y-, and z-axis data, respectively. This figure also shows camera images captured during these activities. To focus on such appliances as air conditioners or televisions, the participant turned her head and fixated it. Therefore, we can detect an attention by finding a static sensor data segment, which corresponds to the fixation, that follows immediately after¹ a large change of the sensor data, which corresponds to the head rotation toward the appliance. Our method simply finds a segment with a large change and a static segment using the variance of sensor data within a sliding time window. We compute the variance value from three-axis data (x , y , and z) by the following equation:

$$v = \frac{1}{T} \sum_{i=t}^{t+T-1} (\bar{x} - x_i)^2 + \frac{1}{T} \sum_{i=t}^{t+T-1} (\bar{y} - y_i)^2 + \frac{1}{T} \sum_{i=t}^{t+T-1} (\bar{z} - z_i)^2,$$

where T is the window size and t is the index of the first sample in the window. When computed value v is lower than a threshold, the window (segment) is regarded as static. Otherwise, the segment is regarded to have a large change.

Here, for the attention detection performance, recall is more important than precision. A smart glass provides information to users on the periphery of their attention. For example, even when a user happens to turn her face toward an appliance that

¹within one second in our implementation

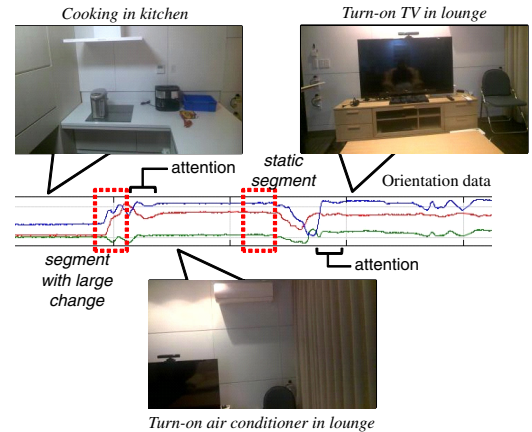


Figure 4. Example of time series orientation data obtained from smart glass.

she has no intention of operating and our method shows a card corresponding to the appliance on the glass's screen, she can easily ignore the information provided on the periphery of her attention.

Here, when a user wants to control an appliance that she is already focusing on, e.g., a user wants to turn off a TV that she is watching and a user wants to turn off a faucet when she is washing something, our method cannot detect the user's intention to control the appliance. We consider a possible solution to this problem. Google Glass has a function to detect the user's wink by an infrared proximity sensor. We can ask the user to express her intention to control an appliance by using this function. Also, by combining this function and the attention detection method, we consider that we can reduce the number of false alarms, i.e., incorrect detection, of the attention detection method. (Because the wink detection function is also not perfect, the function should be used to supplement the attention detection method.)

Image feature extraction with deep neural networks

After the above method detects that the user has looked at an appliance, we extract the features from the camera data. The extracted features will become the inputs of the appliance selection model. Assume that the user looks at time t_s . Our glass application captures an image just after t_s . Because we capture an image only when the user's attention is detected, we can save energy of the glass. In this study, we employ a deep learning approach to extract features from the image. We use the deep convolutional neural network (DCNN) architecture pre-trained on the ILSVRC-2012 dataset (over a million images in-the-wild) [16, 18], which achieves good recognition performance in general object recognition tasks, and its hidden layer activations as features. The input of the DCNN is a Google Glass's 1280x720 size of image. The DCNN used in this study consists of seven layers [16], and we take the activations of its sixth hidden layer as features (4096 dimensional features), which is the output of the DCNN shown in the upper portion of Figure 3 (image features). Since the later convolutional layers are likely to contain a richer semantic representation, we use the sixth layer [10].

Note that, because the feature extraction process requires high computational power, it is executed on a computer in the home network. The other extraction processes are executed on smartphones.

Feature extraction for IGMM input

In this subsection, for each sensor we explain how we extract the features that will be the inputs of the IGMMs. The extracted features are used to estimate user activities and locations. Assume that the user pays attention to an appliance at time t_s , which is the time of the detected static window, and the time of a detected window with a large change is t_l ($t_l < t_s$). (Remember that the attention detection method finds a static sensor data segment that follows immediately after a large change of the sensor data.)

Acceleration sensor on smart phone

Acceleration data, which show user movements, are used in many activity recognition studies [2, 22] to detect such activities as walking, standing, and sleeping. Note that acceleration signals in three orthogonal directions (x , y , and z) might be sensitive to smartphone placement, e.g., in pants or breast pockets. To cope with the problem, we use the previously proposed combined signal given by $R_i = \arcsin\left(\frac{z_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}\right)$ [12], where R_i is the i th combined signal. We compute the average and variance values from the combined data between times $t_l - w$ and t_l , where w shows the window size, and use them as features.

Microphone on smartphone

Sound data have also been used in many activity recognition studies [20, 21, 23] to detect such daily life related sounds as running water, speaking, and vacuuming. In one previous work [8], the Mel-Frequency Cepstral Coefficient (MFCC) was reported to be the best transformation scheme for environmental sound recognition. Another work [4] achieved highly accurate recognition of such bathroom activities as showering, flushing, and urination using MFCC. Thus, we decided to use the average MFCC components extracted from sound data between times $t_l - w$ to t_l as features.

Light sensor on glass

Light sensor data, which show whether a user is at a well-lit place, are strongly related to the use of lighting. We compute the average of the light data observed from times t_l to t_s and use it as a feature value.

Wi-Fi module on smart phone

Many researchers have attempted to construct indoor positioning systems utilizing Wi-Fi data [14, 17, 19]. We simply use the signal strength values observed at time t_s as features.

Unsupervised activity recognition and indoor positioning

We estimate user activities and indoor positions with unsupervised manner. The estimated activity and position are used as inputs to the appliance selection model. We employ acceleration, light, and microphone features as inputs to an activity recognition model. Also, we employ Wi-Fi features as inputs to a position estimation model. That is, we construct feature vector $s_{act,t}$ for an activity recognition model at time t by

concatenating the acceleration, light, and microphone feature values at time t as shown in Figure 3. Also, we construct feature vector $s_{pos,t}$ for an indoor positioning model at time t by concatenating the Wi-Fi feature values at time t . We use the same method to train the activity recognition and positioning models as follows.

Training phase

During the user's daily life, we can obtain unlabeled sensor data (feature vectors) for training the activity and indoor positioning models. We model the set of feature vectors using a mixture of Gaussians, and each Gaussian corresponds to a cluster of feature vectors. Because cluster members (feature vectors) of a cluster have similar features, each cluster shows an activity class or location usually found in the user's daily life. For activity recognition, a cluster shows the user's activity class, e.g., cooking or sleeping. For indoor positioning, a cluster shows the user's location, e.g., kitchen or bedroom. Note that each activity and location cluster does not actually have labels such as cooking and kitchen since we use an unsupervised method. Instead, each cluster has a simple identifier such as "activity cluster 01." Also, the numbers of location and activity clusters are unknown and depend on the user. Therefore, we obtain such clusters based on non-parametric Bayesian methods.

We achieve non-parametric unsupervised clustering based on an infinite Gaussian mixture model (IGMM), which is a Gaussian mixture model (GMM) with a Dirichlet process prior defined over mixture components [11]. For more details about IGMM, see [5].

Test phase

When a new test feature vector appears (i.e., $s_{act,t}$ or $s_{pos,t}$), we compute the distance between it and each cluster. For example, when the test vector is close to a cluster corresponding to *kitchen*, it is regarded as a member of the cluster. Because we have two IGMMs, i.e., IGMMs for activity recognition and indoor positioning, we compute the distance between $s_{act,t}$ and each cluster of the IGMM for activity recognition, and the distance between $s_{pos,t}$ and each cluster of the IGMM for indoor positioning. The computed distances correspond to the context features in Figure 3, and the features are the output of the IGMMs.

Appliance selection using MKL

Overview of classification

We find an appliance that a user wants to operate by using camera image features and context features. The inputs of the appliance selection model are image features output from the DCNN and the distance values (context features) output from the IGMMs. By using feature vectors consisting of the features, a discriminative classifier that classifies a test feature vector into an appropriate appliance class is trained. We use a multiple kernel learning (MKL) method in our discriminative selection model because it deals well with data that consist of features from different sources. Because MKL employs a linear combination of multiple base kernels while each kernel can describe a different property of the data, we prepare a kernel for each different data source (image data and context information in our case). In MKL, when a data source

is not useful for distinguishing appliances, for example, the weight of its corresponding kernel can be lowered. That is, the weights of the kernels are determined according to their usefulness.

Multiple kernels for context-aware appliance selection

A kernel function is used to compute the distance between instances to determine a linear decision function in the feature space. When we have N training instances $\{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^N$, the decision function, which is used to predict the estimation of unseen test instance \mathbf{x}_* , is written as

$$f(\mathbf{x}_*) = \mathbf{a}^T \mathbf{k}_* + b,$$

where \mathbf{a} and b are the vector of the weights assigned to each training instance and the bias. Also, $\mathbf{k}_* = [k(x_1, \mathbf{x}_*) \dots k(x_N, \mathbf{x}_*)]^T$, where $k(\cdot, \cdot)$ is a kernel function that calculates the distance (similarity) between two instances.

Based on MKL, which combines multiple base kernels, we employ the following linear combination of kernels as the decision function:

$$f(\mathbf{x}_*) = \mathbf{a}^T (e_{img} \mathbf{k}_{img,*} + e_{ctx} \mathbf{k}_{ctx,*}) + b, \quad (1)$$

where e_m is the weight of the m -th kernel and $\mathbf{k}_{m,*} = [k_m(x_1, \mathbf{x}_*) \dots k_m(x_N, \mathbf{x}_*)]^T$. Note that $m \in \{img, ctx\}$, and *img* and *ctx* show the image and context features. That is, we prepare kernels for the image and context features. In each kernel, we configure each kernel function and its hyperparameters to emphasize the corresponding features. For example, we use a polynomial kernel, which is usually used for image classification, for $\mathbf{k}_{img,*}$. As for $\mathbf{k}_{ctx,*}$, we use a radial basis function, which is usually used when there is no prior knowledge about the data. Also, because the image features and context features have different data distributions, we set the hyperparameters of the kernels so that each kernel focuses on its corresponding data distribution. For more detail about the setting of the hyperparameters, refer to [15]. As above, using different kernels enables us to represent that different features can have different similarity measures, and permits us to capture nonlinear relationships between features.

Parameter estimation and SVM training

Here we explain how we train our appliance selection model based on a one-vs.-rest SVM for each appliance class by using the above multiple kernels (Equation 1). We employ Bayesian efficient multiple kernel learning (BEMKL) [13] to estimate the parameters in Equation 1, i.e., \mathbf{a} , e , and b , where e_{img} and e_{ctx} are collectively represented by e .

Based on the above decision function (Equation 1), we train a one-vs.-rest SVM for each appliance class, which is used to select an appliance, i.e., $y = \text{sign}(f(x_*))$. Note that, when we train an initial appliance selection model based on MKL, we employ only image features extracted from appliance photos that were taken during the system installation period. After that, we re-train the model by using the labeled sensor data consisting of image and context sensor data obtained when the user routinely controls appliances every day.

Classification with SVMs

By using the above one-vs.-rest SVM for each appliance class, we choose the class that classifies a test instance with the greatest margin as the final estimation result: the selected appliance.

Coping with small amount of training data

We assume that our method learns an initial appliance selection model using photos that are taken during the system installation period. After that, our method iteratively updates the model using labeled sensor data obtained when the user routinely controls appliances every day. Therefore, the amount of training data just after the system install is small, and this may degrade the appliance selection performance.

Here we attempt to improve the appliance selection performance when we have scant training data. We discuss two approaches to improve the performance by using additional training data; reusing other users' training data and utilizing an online image database.

Reusing other users' training data

We can easily increase the amount of training data by reusing labeled training data collected by other users living in the same environment. That is, an initial appliance selection model is trained on the labeled training data of other users. By doing so, we can achieve good appliance selection performance with scant training data collected by the user of interest. Note that activity sensor data are reported to be slightly different for each user [1], and this may degrade the appliance selection performance. Therefore, we iteratively update the model using labeled sensor data from the user of interest obtained when the user routinely controls appliances every day.

Utilizing online image database

We attempt to utilize appliance images on the Internet to cope with the scant training data problem. Therefore, we train an initial appliance selection model on the online images in addition to the appliance photos taken during the installation period to increase the amount of training data. We use ImageNet [9], which is an image database with labeled images, as the appliance image data source on the Internet. Because ImageNet has images with labels of appliances such as "television," we can find online images corresponding to each house appliance in the environment using the labels. We then find appropriate online images for the model training, i.e., similar images to the appliance photos. We first extract image features from each appliance photo taken during the installation period by using the same DCNN architecture used in Figure 3, and construct a image feature vector by concatenating the extracted feature values. We then collect online images for each category of appliance in the environment from ImageNet, and extract a feature vector from each online image by using the DCNN. We find top- k similar online images for each appliance in the environment based on the Euclidean distance computed using the extracted feature vectors and train the initial appliance selection model using the selected online images.

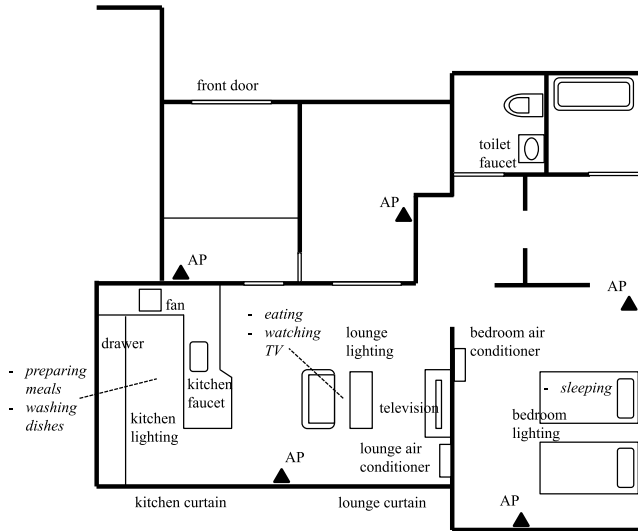


Figure 5. Floor plan of experimental environment.

Table 1. Appliances installed in experimental environment.

lounge air conditioner	bedroom air conditioner	bedroom lighting
front door	drawer	kitchen faucet
fan	kitchen lighting	lounge curtain
kitchen curtain	lounge lighting	toilet faucet
television		

EVALUATION

Data set

We collected a data set from a house built by our laboratory for research purposes because several appliances and furnishings in the house are connected to a home network through which we can control them. Figure 5 shows a floor plan of the experimental environment. Table 1 lists the appliances installed in the environment, and Figure 5 shows their positions. Figure 5 also shows locations of Wi-Fi APs and locations where activities were performed. Our experiment used signals only from APs in our environment. In the environment, three participants collected sensor data with a Google Glass and a Google Nexus 5 smartphone, which was inserted into a pants pocket.

We collected sensor data using a semi-naturalistic collection protocol [1] that permits greater variability in participant behavior than laboratory data. In the protocol, the participants followed written instructions and performed a random sequence of activities. They were granted much freedom regarding how they performed each activity because the instructions are relatively vague: “go to the toilet” or “watch TV.” During the activities, the participants controlled the home appliances listed in Table 1. To obtain labeled sensor data, we asked the participants to look at the appliance they want to control and then manually select and control it using our basic system explained in the *Assumed environment* section. The glass recorded a label that included the name of the controlled appliance and a timestamp, which is used as both ground-truth and training data.

During the experimental period, the participants completed the data collection sessions that included the random sequence of activities. A data collection session started when a participant came home, i.e., opening the front door. After that the participant performed the following random sequence of activities that included appliance use: *preparing meals*, *eating*, *washing dishes*, *watching TV*, *going to the toilet*, and *sleeping*. The data collection session ended when the participant left the house. Each participant completed ten sessions in our experimental environment. Before the experiment, the participants registered their appliances in our system by photographing them.

Evaluation methodology

Since we used 13 appliances in our experiment, we have a thirteen-class classification problem. We evaluated the performance of our method based on the precision, recall, and F-measure ($\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$). To investigate its effectiveness, we tested the following six methods:

- *SVM w/ cam*: simply uses camera features as inputs to an SVM. (We used LIBSVM [3]. We use a linear kernel function instead of MKL.)
- *SVM all*: employs activity and position related sensor data features in addition to camera features.
- *Proposed w/ cam*: This is our proposed method that only uses camera features.
- *Proposed*: This is our proposed method.
- *Proposed w/o act*: This is our proposed method that does not use activity related sensor data.
- *Proposed w/o pos*: This is our proposed method that does not use position related sensor data.

First, each method trains an initial classification model using only the appliance photos prepared in the installation phase. (In our experiment, we used a short movie for each appliance.) After each session, the labeled sensor data obtained during the session are added to the training data for the classification model. The classifier is updated after each session using the session’s data.

Results of attention detection

Before evaluating the classification performance, we briefly evaluate our attention detection method. When our method detects an attention using our collected data and a participant actually controlled an appliance after the detected attention, we assume that the detection is correct. When our method detects a new attention after the first detected attention before the participant controlled an appliance, we assume that the first detected attention is incorrect (false alarm). Also, we used a value of threshold v that yielded good performance in our preliminary experiment. The precision and recall computed with the data set were 70.4% and 94.6%, respectively. Our method achieved very high recall over 94%. As mentioned in the *Detecting attention* section, recall is crucial for our glass application. Also, we can easily reduce the number of false alarms of the attention detection by using the wink detection function of the glass.

Table 2. Classification accuracies for six methods.

	Precision (%)	Recall (%)	F-measure (%)
<i>SVM all</i>	76.6	70.0	73.2
<i>SVM w/ cam</i>	74.5	71.3	72.9
<i>SVM cam+acc</i>	75.3	70.9	73.0
<i>SVM cam+WiFi</i>	75.8	70.6	73.1
<i>SVM cam+sound</i>	74.7	71.0	72.8
<i>SVM cam+light</i>	75.7	70.3	72.9
<i>Proposed</i>	85.8	78.1	81.8
<i>Proposed w/ cam</i>	83.5	72.0	77.3
<i>Proposed w/o act</i>	85.6	74.9	79.9
<i>Proposed w/o pos</i>	84.3	74.9	79.3
<i>Proposed w/o IGMM</i>	84.0	73.8	78.6

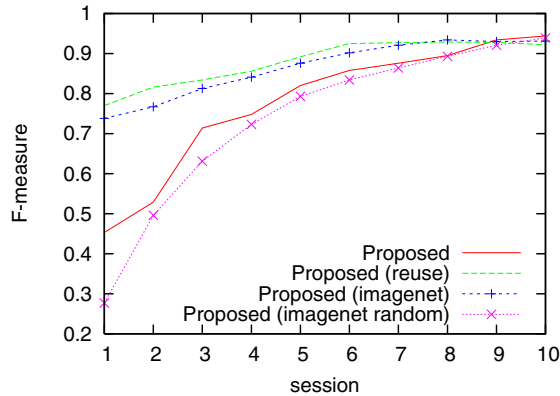


Figure 6. Transitions of average F-measures for *Proposed*, *Proposed (reuse)*, and *Proposed (imagenet)*. The methods iteratively update their models after each session.

Results of appliance selection

Effect of context information

Table 2 shows the average precisions, recalls, and F-measures of the six methods. They were computed using all the ten-session data. We first focus on the *Proposed w/ cam* result. We achieved good accuracies by solely using the camera images. The average F-measure exceeded 75%.

Based on the *Proposed w/o act* and *Proposed w/o pos* results in Table 2, the contributions of the positional and activity information are significant ($p < .05$). By using the positional information, the F-measures related to “lighting” and “air conditioner” improved from the *Proposed w/ cam* results. The improvement was about 5% on average. The positional information helped distinguish these appliances with few distinguishing image features. As for the activity information, we improved the accuracies for “kitchen lighting” and “fan.” The improvement was about 7% on average. The “preparing meals” activity captured by the microphone and acceleration sensor contributed to recognizing these appliances. By using the positional and activity information, *Proposed* achieved 81.8% accuracy (F-measure). Table 2 also shows results when we did not use IGMM (*Proposed w/o IGMM*), i.e., extracted features are simply used as inputs of MKL. From this result, we could confirm the effectiveness of IGMM (3.2% improvement).

Our proposed method outperformed *SVM all*, which simply uses extracted sensor data features. Table 2 also shows results of SVM-based methods, which use features from each indi-

Table 3. Classification results when we reuse other users’ training data.

	Precision (%)	Recall (%)	F-measure (%)
<i>SVM all</i>	82.7	81.5	82.1
<i>SVM w/ cam</i>	81.0	80.3	80.6
<i>Proposed</i>	93.5	90.2	91.8
<i>Proposed w/ cam</i>	90.8	86.4	88.5
<i>Proposed w/o act</i>	92.4	88.1	89.2
<i>Proposed w/o pos</i>	92.7	88.5	89.6



Figure 7. Photos taken during the installation period and their similar images obtained from ImageNet. The Euclidean distance between photo and online image is associated with online image.

vidual sensor (*SVM cam+acc* - *SVM cam+light*). As shown in the results, the accelerometer and Wi-Fi module, which capture activities and locations, contributed.

Amount of training data

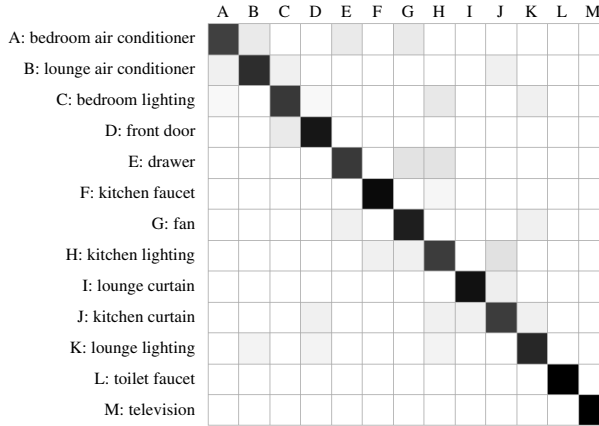
The above result includes the results of sessions with scant training data, e.g., the 1st and 2nd sessions. After each session, our method updated the appliance selection model by adding the sensor data obtained during it as training data. Figure 6 shows the average F-measure for each session (*Proposed* line). For the first session, the method used only images of the appliances obtained in the installation period as training data. As the amount of training data increases, the F-measure also increases and reached about 95%.

Reusing other users’ training data

As above, the F-measures of sessions with scant training data were poor. To cope with the problem, we attempt to reuse labeled training data collected by other users in the same environment. That is, we use two other participants’ training data (twenty-session data in total) in addition to training data collected by a test participant. Table 3 shows the results. Also, the *Proposed (reuse)* line in Figure 6 shows the F-measure for each session. As for the first session, for example, because we could use twenty-session training data collected by other participants, we could achieve about 80% accuracy.

Effect of online images

Here we investigate the effectiveness online images obtained from ImageNet. We collect online images for each category

Figure 8. Visual confusion matrix of *Proposed w/ cam* result.

of appliance in the environment from ImageNet, and then find top- k similar online images for each appliance in the environment ($k = 100$, which yielded good performance in our preliminary experiment). The similar online images are used to train the appliance selection model in addition to training data collected by a test participant.

We first show example similar online images selected by our method. Figure 7 shows example photos taken during the installation period and their top-4 similar online images. As shown in the figure, we could find online images similar to the photos. The figure also shows the distance between the photo and the online image. Because the distance between photos that capture the same appliance taken during the test sessions was about 200 - 500, we believe that our method could find similar online image to photos that capture appliances in the environment.

The *Proposed (imagenet)* line in Figure 6 shows the F-measure for each session when we used the online images. For the first session, we could achieve 73.8% accuracy that is about 30% higher than *Proposed* and only about 3% lower than *Proposed (reuse)*. Figure 6 also shows the accuracies when we use randomly selected online images as training data (*Proposed (imagenet random)* line). From these results, we could confirm the effectiveness of the similar image selection. As above, by using the online images, we could achieve the accurate appliance selection without placing any additional burdens on users.

Leave one session out cross validation

We investigate the performance of our proposed method when we have enough training data. In this evaluation, we conducted a leave one session out cross validation evaluation and tested one session using a classifier trained on nine other sessions. Table 4 shows the results. When we had nine-session training data, our proposed method achieved very good accuracy: about 95%. By employing both positional and activity information, we improved the accuracy by about 10%.

Figure 8 shows the visual confusion matrix of the *Proposed w/ cam* result. As shown in the matrix, the classification accuracies related to “air conditioner” and “lighting” were

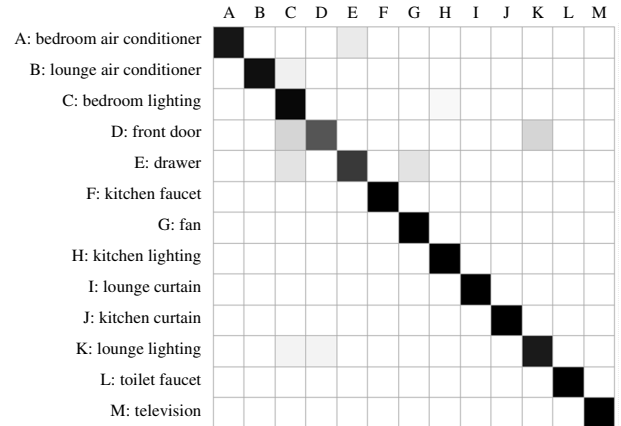
Figure 9. Visual confusion matrix of *Proposed* result.

Table 4. Classification results of leave one session out cross validation.

	Precision (%)	Recall (%)	F-measure (%)
<i>SVM all</i>	84.5	84.4	84.4
<i>SVM w/ cam</i>	81.3	81.2	81.2
<i>Proposed</i>	95.5	93.6	94.5
<i>Proposed w/ cam</i>	85.7	86.2	85.9
<i>Proposed w/o act</i>	92.8	89.7	91.2
<i>Proposed w/o pos</i>	89.4	87.8	88.6

relatively poor. *Proposed w/ cam* could not distinguish between “kitchen lighting” and “bedroom lighting.” Because *Proposed w/ cam* used only camera images, it could not correctly identify appliances with similar appearances. In addition, the classification accuracy related to “drawer” was poor. This might be because the drawer looks simple and does not have any distinguishing image features.

Figure 9 shows the visual confusion matrix of the *Proposed* result. As for the above two appliance types (“air conditioner” and “lighting”), the F-measure increased about 14% on average. Also, as shown in Table 4, *Proposed* greatly outperformed *Proposed w/ cam* by employing contextual information, and the F-measure improved by about 10%. On the other hand, for the standard SVM-based methods, the F-measure improved only about 3% by employing contextual information, perhaps because our MKL-based methods can deal well with data that consist of features from different sources.

Time latency

Because we assume that a user focuses on an appliance simultaneously with performs a voice command such as “Volume up,” time latency of our method should not be much longer than that of speech recognition (1.5 sec for Echo). Since feature extraction can be run in parallel, the computation time is greatly affected by the image feature extraction, which is executed on a computer. The image transmission time and image processing time are 0.33 seconds and 0.08 seconds, respectively (using a computer with GeForce GTX TITAN Z). Since the classification time by SVM is 0.01 seconds, the total computation time of our selection method is about 0.43 seconds.

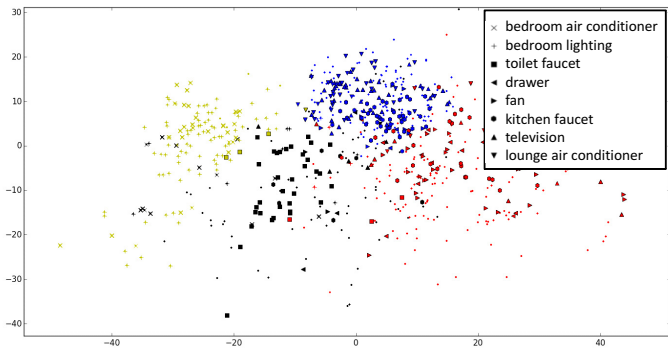


Figure 10. Clustering result of IGMM for positional information.

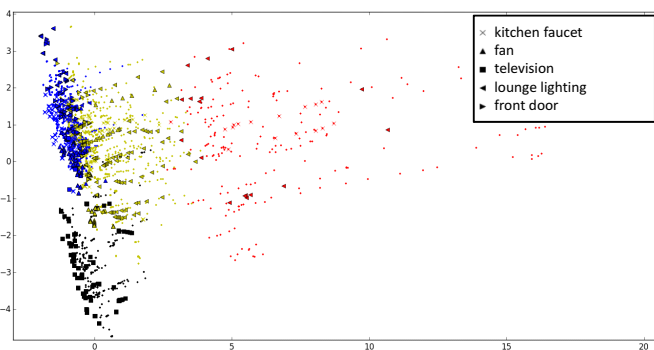


Figure 11. Clustering result of IGMM for activity information.

IGMM results

Finally, we show the results of the IGMMs, which provide non-parametric clustering of indoor positions and activities, to confirm whether or not the IGMMs can differentiate indoor positions and activities using sensor data in a non-parametric manner. Figs. 10 and 11 show the results visualized in a 2-dimensional space by using Principal Component Analysis (PCA). For the limitation of the figure space, we show only the result of typical appliances operated in our experiment. The small dots in the figures show the unlabeled sensor data points (without activity and position label) obtained during the experiment, i.e., obtained when the participants did not select and control appliances. The other points show data points corresponding to appliance selection. (We show only selected appliances for visibility.) The color of a point shows its belonging cluster. As shown in Figure 10, many data points corresponding to appliances used at the same places were grouped into the same clusters, e.g., bedroom appliances (yellow cluster) and kitchen appliances (red cluster). As for Figure 11, the data points corresponding to kitchen appliances formed a cluster (blue cluster). Also, the data points corresponding to low activity levels (television) are located far from the data points corresponding to the kitchen activity.

DISCUSSION

Advantages and disadvantages of our system

Voice-control devices such as Amazon Echo and Google Home have been attracting attention. Here we discuss advantages and disadvantages of our approach over these products.

[Advantages of our approach]

(1) Since a user can select an appliance by just looking at the appliance, it takes a shorter time to select it than the voice-based approaches. (2) Because our method uses a wearable device, the user can select an appliance anywhere in the house. (3) A mute user can easily select an appliance.

[Disadvantages of our approach]

(1) Our method requires training data collected in a user's environment. However, our evaluation revealed that using online images as training data reduces the cost of the data collection. (2) Our method cannot uniquely identify an appliance when a captured image includes multiple appliances. There are two possible solutions to this problem. The first is to detect the gaze direction using an eye tracker in order to detect the image region that the user is focusing on. The second solution is to show a list of appliances estimated to be included in the image to the user. This approach permits the user to select an appliance from the list by using, for example, head gestures.

[Advanced functions of modern voice-based devices]

(1) The modern voice-based devices allow high-level control of devices. For example, the user can change lightings in a room to a "theater mode." However, this kind of function can be easily implemented in our system by just associating an appliance (or multiple appliances) with high-level commands. (2) Since the modern voice-based devices also allow customized appliance names, e.g., "mom's room's curtain." We believe that this naming function is suitable for remote control by a fixed device such as Amazon Echo. In the case of wearable scenario, disambiguating the user's vague command such as "curtain" based on her location is more useful.

Simultaneous use

When there are multiple users in a house, the users are likely to simultaneously access the same device. In such case, we should control their access rights by, for example, limiting access to a single user at a time. When some user initiates control of the device they would lock out access by other users. Other users' operations would be refused, while notifying them that the appliance is currently in use.

Portable appliances

Our method employs indoor location information as classification features. When dealing with portable appliances that can be moved throughout the house (e.g., electric fans), the location features will vary greatly in the training data and will therefore be generally ignored by the classifier.

Rarely used devices

Since we assume that additional training data are collected during a user's daily life, the recognition accuracies for rarely used appliances will be low and the accuracies may follow Figure 6. When training the classifier, we should address the class imbalance problem, since the amount of added training data depends on the frequency of use.

Acknowledgment

This study is partially supported by JST CREST, JSPS KAK-ENHI Grant Number 26730047, a contract with the Ministry of Internal Affairs and Communications entitled, 'Novel and innovative R&D making use of brain structures,' and IMPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

REFERENCES

1. Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive 2004*. 1–17.
2. Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda Rahel Schmidtke, and Michael Beigl. 2010. ActiServ: Activity recognition service for mobile phones. In *International Symposium on Wearable Computers (ISWC 2010)*. 1–8.
3. Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Issue 3.
4. Jianfeng Chen, Alvin Harvey Kam, Jianmin Zhang, Ning Liu, and Louis Shue. 2005. Bathroom activity monitoring based on sound. In *Pervasive 2005*. 47–61.
5. Tao Chen, Julian Morris, and Elaine Martin. 2006. Probability density estimation via an infinite Gaussian mixture model: application to statistical process monitoring. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 55, 5 (2006), 699–715.
6. Heidi Christensen, Iñigo Casanueva, Stuart Cunningham, Phil Green, and Thomas Hain. 2013. homeService: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition. In *4th Workshop on Speech and Language Processing for Assistive Technologies*. 29–34.
7. Fulvio Corno, Alastair Gale, Päivi Majaranta, and Kari-Jouko Riih . 2010. Eye-based Direct Interaction for Environmental Control in Heterogeneous Smart Environments. *Handbook of Ambient Intelligence and Smart Environments* (2010), 1117–1138.
8. Michael Cowling. 2004. *Non-speech environmental sound recognition system for autonomous surveillance*. Ph.D. Dissertation. Griffith University.
9. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR 2009*. 248–255.
10. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. DeCAF: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531* (2013).
11. Thomas S Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* (1973), 209–230.
12. Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Sondrol. 2006. Biometric gait authentication using accelerometer sensor. *Journal of computers* 1, 7 (2006), 51–59.
13. Mehmet Gonen. 2012. Bayesian Efficient Multiple Kernel Learning. In *ICML 2012*.
14. Michael Hardegger, Gerhard Tr ster, and Daniel Roggen. 2013. Improved ActionSLAM for long-term indoor tracking with wearable motion sensors. In *International Symposium on Wearable Computers (ISWC2013)*. 1–8.
15. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Kernel Smoothing Methods. In *The Elements of Statistical Learning*. Springer, 191–218.
16. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia 2014*. 675–678.
17. Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. 2012. ARIEL: Automatic Wi-Fi based room fingerprinting for indoor localization. In *UbiComp 2012*. 441–450.
18. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS 2012*. 1097–1105.
19. Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, and others. 2005. Place lab: Device positioning using radio beacons in the wild. In *Pervasive 2005*. 116–133.
20. Nicholas D. Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T. Campbell, and Feng Zhao. 2011. Enabling large-scale human activity inference on smartphones using community similarity networks (CSN). In *UbiComp 2011*. 355–364.
21. Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. 2009. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *MobiSys 2009*. 165–178.
22. Takuya Maekawa and Shinji Watanabe. 2011. Unsupervised Activity Recognition with User's Physical Characteristics Data. In *International Symposium on Wearable Computers (ISWC 2011)*. 89–96.
23. Takuya Maekawa, Yutaka Yanagisawa, Yasue Kishino, Katsuhiko Ishiguro, Koji Kamei, Yasushi Sakurai, and Takeshi Okadome. 2010. Object-based activity recognition with heterogeneous sensors on wrist. In *Pervasive 2010*. 246–264.

24. Ankit Mohan, Grace Woo, Shinsaku Hiura, Quinn Smithwick, and Ramesh Raskar. 2009. Bokode: Imperceptible Visual Tags for Camera Based Interaction from a Distance. *ACM Trans. Graph.* 28, 3, Article 98 (July 2009), 98:1–98:8 pages.
25. Thomas P. Moran, Eric Saund, William van Melle, Anuj Gujar, Kenneth P. Fishkin, and Beverly L. Harrison. 1999. Design and Technology for Collaborative: Collaborative Collages of Information on Physical Walls.. In *ACM Symposium on User Interface Software and Technology (UIST 1999)* (2002-12-16). 197–206.
26. Robert NeBelrath, Chensheng Lu, Christian H Schulz, Jochen Frey, and Jan Alexandersson. 2011. A Gesture Based System for Context–Sensitive Interaction with Smart Homes. In *Ambient Assisted Living Congress 2011*. 209–219.
27. Ilyas Potamitis, Kallirroi Georgila, Nikos Fakotakis, and George K Kokkinakis. 2003. An integrated system for smart-home control of appliances based on remote speech interaction.. In *Interspeech 2003*.
28. Carl Edward Rasmussen. 1999. The infinite Gaussian mixture model. In *NIPS 1999*, Vol. 12. 554–560.
29. Jun Rekimoto and Yuji Ayatsuka. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In *Designing Augmented Reality Environments (DARE 2000) (DARE '00)*. 1–10.
30. Fangmin Shi, Alastair Gale, and Kevin Purdy. 2006. Helping People with ICT Device Control by Eye Gaze. *Computers Helping People with Special Needs* (2006), 480–487.
31. Utpal V Solanki and Nilesh H Desai. 2011. Hand gesture based remote control for home appliances: Handmote. In *World Congress on Information and Communication Technologies (WICT 2011)*. 419–423.
32. Koji Tsukada and Michiaki Yasumura. 2001. Ubi-finger: Gesture input device for mobile use. In *UbiComp 2001*. 11.
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 *the 2nd ACM Symposium on Spatial User Interaction (SUI '14)*. 17–25.