PAPER

Hotspot Modeling of Hand-Machine Interaction Experiences from a Head-Mounted RGB-D Camera

Longfei CHEN[†a)], ***Nonmember***, **Yuichi NAKAMURA**[††b)], **Kazuaki KONDO**[††c)], ***Members***, ***and*** **Walterio MAYOL-CUEVAS**[†††d)], ***Nonmember***

SUMMARY This paper presents an approach to analyze and model tasks of machines being operated. The executions of the tasks were captured through egocentric vision. Each task was decomposed into a sequence of physical hand-machine interactions, which are described with touch-based hotspots and interaction patterns. Modeling the tasks was achieved by integrating the experiences of multiple experts and using a hidden Markov model (HMM). Here, we present the results of more than 70 recorded egocentric experiences of the operation of a sewing machine. Our methods show good potential for the detection of hand-machine interactions and modeling of machine operation tasks.

key words: *egocentric vision, machine operation experiences, hotspots, RGB-D, task modeling*

1. Introduction

In recent years, wearable consumer equipment has rapidly developed, becoming smaller and more powerful while enabling us to easily record various types of data in our daily lives as lifelogs. In addition to being personal memory aids, lifelogs have a variety of applications, such as sharing experiences within a group, providing knowledge for skill training, and analyzing the behaviors of patients. Wearable cameras have greatly enhanced the efficiency of recording experiences through human-centric vision, which is also known as egocentric vision or first-person vision/view (FPV). Without interrupting a person's daily activities, a small camera can continuously capture how the wearer sees and interacts with objects and other people.

One of the main challenges of analyzing records of egocentric experiences is summarization (i.e., how to distinguish and extract important portions from large amounts of data) [1]. In this study, we focused on machine operations and aimed at summarizations of experiences captured by egocentric vision. Machines such as printers, microwaves, and automobiles appear in our daily lives. Designing a user interface that does not require specific knowledge on the user's part is ideal, but in a variety of our daily situations, we still need to learn how to operate machines. If we could easily acquire knowledge from the experiences of other people, especially experts, the efforts and potential failures in the operating of unfamiliar machines would be greatly reduced.

We focused on the automatic extraction of essential hand-machine interactions and the summarizations of operational tasks in terms of the extractions, instead of the entire contents of lengthy videos. The summarizations contribute to (i) guidance by presenting knowledge and skills obtained from the experiences of mature operators, (ii) predictions of behavior using the obtained task models, and (iii) designs of artifacts based on how users operate a machine or perform tasks.

In this paper, we present a novel method for locating the areas where essential interactions occur and model the sequences of the interactions for machine operation tasks. We first mention related studies in Sect. 2 and then introduce the problems and ideas in Sect. 3. Then, the extraction of low-level features from recorded experiences is described in Sect. 4. The methods for detecting hotspots and classifications of interactions are introduced in Sect. 5. Probabilistic model acquisition based on hidden Markov model (HMM) is described in Sect. 6. Finally, we demonstrate the potential of our proposed method with more than 70 experiences for three different operational tasks and discuss potential future applications in Sect. 7.

2. Related Research

In order to analyze and summarize the massive contents of FPV videos, recognizing activities (i.e., identifying what the wearer is doing) is essential. Many state-of-the-art studies have explored human activities via egocentric vision, most of which focus on activities of daily living (ADLs) [2]–[5], such as making coffee in the office or baking bread in the kitchen. In these common daily activities, people tend to interact with a variety of objects and with a restricted interaction complexity in different scenes. The objects appearing in the scene are regarded as one of the most important clues to inferring the activity that is being carried out [4], [6], [7]. The study in [6] illustrates a video-summarizing method that focuses on the most important objects and people that the camera wearer interacts with. The study in [3] learns a hierarchical model of an activity by the joint properties of ob-

Manuscript received April 23, 2018.
Manuscript revised September 28, 2018.
Manuscript publicized November 12, 2018.

[†]The author is with the Graduate School of Engineering, Kyoto University, Kyoto-shi, 606–8501 Japan.
[††]The authors are with the Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.
[†††]The author is with the Department of Computer Science, University of Bristol, Woodland Road BS8 1UB, UK.
a) E-mail: chenlf@ccm.media.kyoto-u.ac.jp
b) E-mail: yuichi@media.kyoto-u.ac.jp
c) E-mail: kondo@ccm.meida.kyoto-u.ac.jp
d) E-mail: walterio.mayol-cuevas@bristol.ac.uk
DOI: 10.1587/transinf.2018EDP7146

jects, hands, and actions. The method in [4] classifies the detected objects into "active" and "passive" according to whether the user manipulates them or not and then suggests the related ADLs. For example, if "a TV," "a sofa," and "a remote" appear in a frame, this situation can be inferred to be "watching TV" [5]. In most of these studies, the types of existing objects must be known beforehand [8]. However, in circumstances of machine operation, the user mainly interacts with machine surfaces by manipulating important small areas, such as a button, a switch, a lever, etc. These areas could be not only small but not clearly isolated or not distinguishable from others in appearance, e.g., some parts on a machine's surface are textureless. Thus, the functions of what is being manipulated may not be as obvious as the isolated objects, we need a more powerful method to detect such small areas. Rogez et al. [9] suggested that hand poses, hand-object contact points, and contact force vectors would greatly contribute to understanding hand-object interaction (HOI) activities.

A multitude of studies on touch detection have reported taking advantage of depth devices, such as the use of stereo cameras [10] or the combination of fixed depth and thermal cameras [11]. Wilson [12] utilized a single fixed depth camera to sense touch on a tabletop, the similar environment has been implemented in [13]. However, these background modeling approaches do not work sufficiently well in our daily environments because of the rapid background changes caused by head motions. OmniTouch [14] extends touch sensing to wearable devices, whereas touch detection is limited to areas around the fingertips and is sensitive to the angle of approach.

Besides touch, the visual attention of the wearer can direct important portions of egocentric vision experiences [5]. There have been a considerable number of studies estimating the wearer's attention in egocentric vision, for example, approaches using an eye tracker [15]–[17], approaches combining egocentric vision with ego-motion and saliency maps [5], [18], or data-driven methods [19]–[21] that model the correlations among the head, eyeballs, hands, and gaze. These attention-based methods proved useful in discovering or inferring the user's intentions. Other studies have obtained good results using motion features for the segmentation or classification of tasks [22]–[24]. However, in our experimental environment, the scene/background of operating machines changes more slowly as compared to situations of ADLs or walking around, especially when the wearer is concentrating on manipulating objects. While the user's hands are continuously moving and touching small areas on a machine's surface, accurate detection of the target of attention is difficult by only using the direction of attention or the locations of the hands. Considering our environment, where areas on a machine's surface may not be salient, we chose the regression-model-based approach that was proposed in [19] to estimate the wearer's attention.

For modeling activities with the above features, Sundaram et al. [7] proposed a method to observe manipulations via a wearable camera and to classify activities with a three-level DBNs. Clarkson et al. [25] employed a simple ergodic HMM [26] trained with manually labeled events and used it for classifying new activities. Despite the promising results of these approaches, the main challenge in modeling FPV tasks has been elaborated by Betancourt et al. [1]: "the scalability to multiple users and multiple strategies to solve a similar task." In this study, an HMM was adopted for describing temporal operational steps. By taking advantage of the integration of the experiences of multiple experts, we expected to model all the essential operational steps while allowing for variations among different experiences (e.g., the order of steps can be changeable; steps can be substituted; personal minor mistakes can be removed).

3. Capturing Machine Operation Experiences through Egocentric Vision

3.1 Environments and Challenges

Of a wide variety of daily life situations in which we operate machines, we focused on situations of manipulating tabletop devices, such as printer, rice cooker, IH heater, DIY tools, or other devices with manipulation panels. Among them, we chose a sewing machine as a representative example, as sewing tasks demonstrate enough difficulty as well as a certain degree of freedom. Tasks such as stitching or embroidering fabrics, are composed of multiple steps that requiring certain knowledge and skills. The users may interact with different portions of a sewing machine and handle different objects, e.g., push buttons, seize a lever, rotate a knob, and guide the cloth. These interactions commonly appear for a wide variety of machines; therefore, the case of a sewing machine makes a good representative of everyday-machine operations.

We aimed to acquire models of such tasks from egocentric vision records. For this purpose, machine operations were recorded through portable head-mounted devices, such as the RGB-D camera, as illustrated in Fig. 1.

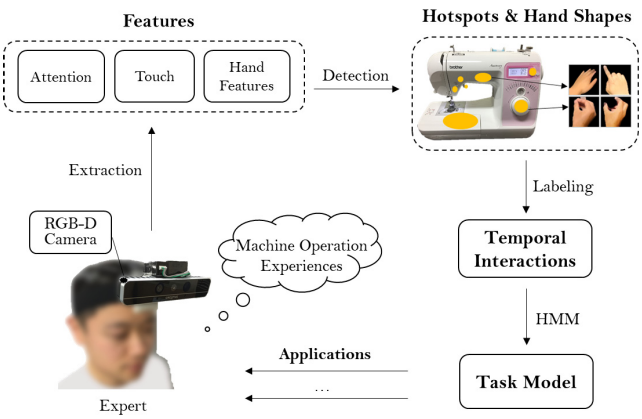


Fig. 1 The experiences of experts are recorded using a head-mounted RGB-D camera. Interactions are detected with hotspot locations and hand shapes. Multiple records by experts are integrated into a task model based on HMM. We regard the touch, attention, and hand features as the most important low-level features in detecting hotspots and interaction patterns.

The difficulties in analyzing and summarizing the records are as follows. First, hand-machine interactions often occurs at small portions such as buttons and switches, which are not well isolated from the other portions. This condition is not acceptable for most of the studies mentioned in Sect. 2. The meaning of each interaction (i.e., type, purpose, or usage) is not given by the user because making annotations for each interaction is tiresome and time-consuming. We need automatic classifications of such interactions. Another difficulty concerns the integration of samples through unsupervised learning. Although we employed experts who did not make serious mistakes for operating machines, there are still personal differences, minor mistakes, and noises. Hence, we need a mechanism for handling such variations.

3.2 Key Idea

We regard the critical areas of machines and hand-machine interactions as the most important tracks in experiences. In a similar spirit of discovering task-relevant objects, we hypothesize that such important areas, (e.g., a *button*, a *lever*, a *switch*, or a *handle*) can offer essential clues in understanding the activities (i.e., which area is manipulated and when) as well as how the task has been processed. We define these crucial areas as "*hotspots*," which are comparatively "hotter" than other areas. We use physical touches as primary clues for detecting hotspots. More specifically, we can use hand features in approaching or touching hotspots to discriminate hotspots and to categorize interactions.

The detected sequences of interactions are often insufficient to compose a complete machine operation process. During actual manipulations, essential interactions may be occluded, irregular and unnecessary touches may occur, and the sequences may differ among experts. For example, the *speed button* and the *pattern-choosing button* can be usually manipulated in an arbitrary order according to the user's preference. Given the above problem, we use probabilistic model learning with the adjustments of interactions. First, we supplement potential interactions that were possibly missed because of occlusions and remove unnecessary interactions, such as supportive touches, that only serve to assist other essential interactions. If most experts have their hands occluded in the same order and at the same location during an operation, the occluded interaction is regarded as an essential part of the operation. Similarly, unessential or accidental touches are excluded if they do not frequently occur in the same order. Next, we use probabilistic learning to create a model of the interactions obtained from the sequences. By this process, we expect to obtain task models with most of the essential interactions but few unnecessary interactions.

Based on the above idea, our approach assumes the following premises:
(a) Any essential step of a manipulation is caused by physical interactions between a hand and a machine. Nonmechanical interactions, such as voice interactions, or interactions with small moving objects that can be occluded easily by a hand, are left to future studies.
(b) A physical hand-machine interaction can be categorized according to the hand shapes made during the interaction.
(c) A sequence of interactions usually has dominant patterns, that is, a typical sequence of essential interactions. However, different approaches to accomplish the same goal are allowed; that is, the experts may act with a certain degree of freedom.

Our approach is organized as shown in Fig. 1. We first extract low-level features, the touch/attention locations and hand features. Then we create the global map for locating the features, which are used to detect hotspots and patterns of interaction, for the purpose of describing the essential physical-temporal interactions. Finally, a probabilistic model is obtained by learning from the hand-machine temporal sequences of interactions executed by multiple experts.

4. Hand and Touch Detection

4.1 Hand Area Detection

Depth, color, and size information are utilized to segment the hand areas. A chromatic histogram in the HSV space of the skin color of each user is formed beforehand from several frames at the beginning of the records. Hand size constraints are also considered; that is, a hand region should be observed in the common operating distance, which usually ranges from approximately 20 to 100 cm from the head-mounted camera[†].

For actual detection, areas with distance larger than the above common operation distance are eliminated as the background. The foreground F is identified as:

$$F(u,v) = \begin{cases} 1, & \text{if } 20 < d(u,v) < 100 \\ 0, & \text{else} \end{cases}, \tag{1}$$

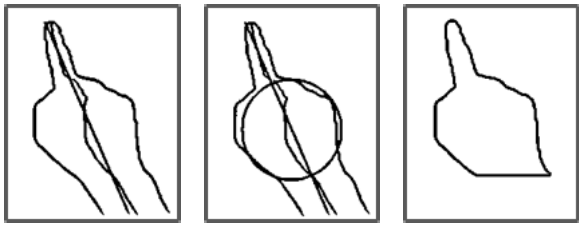
where u and v represent the location of a pixel on the image plane and d is the depth map.

From the remaining foreground area, we find all skin-color regions using a common YCrCb skin-color model as in [27]. Each region that has been found is regarded as a hand region if it has a color distribution similar to the user's skin color while its size is within the possible hand size range:

$$\boldsymbol{B}(h_i, h_m) < \lambda \ \text{ and } \ S_i > mn/k, \tag{2}$$

where $\boldsymbol{B}$ is the Bhattacharyya distance [28] between the hue histogram of the region i and the hue histogram of the user built beforehand. S_i is the area of the region and m and n represent the size of the image. We set the distance threshold λ to 0.2 and the area ratio k to 100 in our experiments.

[†]The range is determined by the possible distance from the camera to an operating hand. The minimum distance is the smallest distance between the camera and the surface of the machine being operated, whereas the maximum distance is the distance from the camera to the hand when the user fully stretches his/her arm.



(a) (b) (c)

Fig. 2 Palm detection by a morphologic property. (a) Find the midpoint of each row of a hand mask and fit with a line. (b) From the top to the bottom of this line, create circles and calculate the area ratio of hand in each circle. The circle with the maximum ratio is found. (c) The hand region inside or above the circle is regarded as the palm region.

(a) (b)

Fig. 3 Finger detection results (a) in a supportive interaction and (b) in an occluded interaction. The detected finger regions are shown in the red boxes. The blue boxes show the neighborhood regions of the fingers, which can be used to detect the existence of near-finger objects in occlusion detection.

The palm area is segmented out of the hand area according to the morphological property. As shown in Fig. 2, the arm area can be roughly regarded as a cylinder, whereas the palm (or fist) is more circular. The midpoint at each vertical position of the hand mask (i.e., a skeletal line of the hand region) is determined. A straight line is fit to the skeletal line. Then, at each pixel on this line, we overlay a circle with the point as its center and the width of the hand mask along the scan line as its diameter. The area ratio of the hand in each circle is calculated and the circle with the maximum area ratio is obtained as follows:

$$j = \arg\min_{j} \left(h_j / c_j \right), \tag{3}$$

where h_j is the area of the hand region in the circle for the jth scan line and c_j is the area of this circle. Finally, all the hand areas in and above the circle (including the fingers) are regarded as the palm region. The left and right palms are simply identified by their spatial locations. The estimated 3D location (x,y,z) of the palm is represented by its centroid, and its area is calculated by the number of pixels.

Due to the geometric property of egocentric vision taken from the user's head, the finger and thumb regions usually appear farther from the camera than does the wrist region. We detect the finger region by finding the area with a larger depth inside the palm region. Suppose that the depth of a palm region ranges from $d_{\min}$ to $d_{\max}$; the depth in the finger region d_{f} is found by

$$d_{\max} - \alpha \cdot (d_{\max} - d_{\min}) < d_{\mathrm{f}} < d_{\max}, \tag{4}$$

where α is a small constant. Figure 3 illustrates the actual results of attempts to detect the fingers.

4.2 Hand Features

In common machine operation situations, such as sliding a lever or rotating a dial, only one or two fingers are moving while the other fingers stay relatively still. We use such fast-moving finger areas as representative of the hand motion. In the case of the whole palm moving consistently or the fingers being occluded, the average velocity of the visible area is considered to be the hand motion. The hand motion V is represented by the motion in the image plane (Δx,Δy) and motion in depth (Δz). For motion (Δx,Δy) in the image plane, the background motion is subtracted to remove the ego-motion effects as follows:

$$V = \max\left(V_p - \overline{V_b}\right), \tag{5}$$

where V_p is the velocity of the area within the palm and $\overline{V_b}$ is the average velocity of the background, which is calculated as the average velocity of the nonhand area on the foreground F in Eq. (1). The motion in depth (Δz) is simply represented by the movement of the palm centroid in the depth direction.

We use the point-based hand shape description that was proposed by Shimada et al. [29]. The palm region is rotated according to its principal axis of the moment, and the palm contour is equally sampled from -30° to 210° (counterclockwise from the lower right to the lower left). These sampled points are featured in 241 dimensions with their distances to the palm centroid normalizing with the palm area.

4.3 Touch Detection

4.3.1 Basic Touch Detection

For touch detection, we simply use the depth difference between the hand area and the neighboring area, as illustrated in Fig. 4. Due to the fact that touches may happen at different locations of the palm, we consider small local windows along the palm contour. While the window slides along the contour, the average depth of the palm area is represented as d_p and the average depth outside the palm area d_n is calculated. If their difference is smaller than Δ, we regard that as a touch. Every detected touch area is indexed by its centroid (T_x,T_y) and radius (T_r). Since there are depth measurement errors, which often amount to several millimeters for the Intel RealSense SR300 camera [30], we set the distance threshold Δ to ± 7 mm in our experiments.

4.3.2 Eliminating Meaningless Touches

Some of the touches may be unnecessary for modeling manipulations. *Supportive touches*, in which a hand is placed

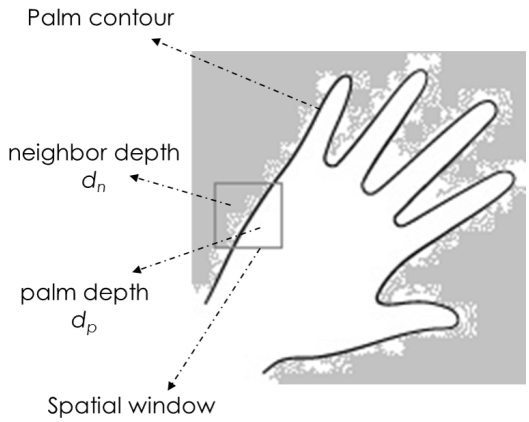


Fig. 4 Palm-oriented touch detection. Touches are detected by comparing the depth values around the palm contour.

somewhere with a minor role to support the other hand, are the most significant type and usually appear in actual operations when both hands are used. For example, one hand is resting on the machine's surface while the other is performing an actual manipulation, such as pushing a button. Both hands may be placed on a cloth while the cloth is automatically moving, as shown on Fig. 3 (a). Supportive touches can be distinguished by considering the hand motions and shapes; that is, the hands are stable. This condition can be represented as

$$|V| \leq \boldsymbol{C} \text{ and } p = \boldsymbol{p_2}, \tag{6}$$

where $\boldsymbol{C}$ is the velocity threshold of the palm motion, which is set to a small constant, and p is the interaction pattern label of the hand during an interaction as described in Sect. 5.4.

Casual touch is another type of an unimportant touch in which the user unconsciously makes contact with the machine's surface without the intention to manipulate it. The frequency of casual touch is low for experts because they usually avoid unnecessary motions. The positions of casual touches are divergent because they often occur accidentally. Consequently, we expect that most casual touches can be filtered out via the learning of a probabilistic model.

4.3.3 Occluded Touch Estimation

Important interactions sometimes occur behind objects, and so they are not visible to the camera; for example, we may push a button on the underside or handle a lever on the backside of a machine, as shown on Fig. 3 (b). In order to cope with this problem, we assume that frequent occurrences of hand occlusions at the same position are caused by touches to an occluded portion of a machine. Such occlusions are detected and considered as potential touches. In such a case, we assume that an occluding object exists near the finger region at a shorter distance from the camera, which satisfies

$$d_w < d_o < d_f, \tag{7}$$

where d_w, d_o, and d_f are the average depth values of the wrist, the object, and the finger area, respectively. Second, the occluded hand should be actively moving, which satisfies

$$V > \boldsymbol{C} \text{ and } p \neq \boldsymbol{p_2}, \tag{8}$$

where $\boldsymbol{C}$ and $\boldsymbol{p_2}$ are the same as in Eq. (6).

Third, the area of the palm decreases at the beginning of the occlusion but increases after it. The area can be described as follows:

$$A_o < A_{o-1} \text{ and } A_o < A_{o+1}, \tag{9}$$

where A_o is the average palm area when the occlusion occurs, whereas A_{o-1} and A_{o+1} are the palm areas before and after the occlusion period, respectively. The palm areas are represented by the average palm area of several frames before and after the occlusion period. However, if the period of a detected occlusion interaction overlaps with the period of an already-detected hotspot (explained in the next section), the occluded interaction is not counted as a new interaction. We assume that, at that moment, the visible interaction at the hotspot represents the operation better than the occluded interaction does.

5. Hotspots and Interactions

Hotspots are detected by integrating touches or attention locations using the global map, and then an interaction type is determined for each hotspot.

5.1 Global Map

We use a global map to integrate the detected touches or attention targets from different egocentric frames. For this purpose, we simply stitch several egocentric frames that have been recorded as the user looks over the machine before starting manipulation. Then, we adopt the SURF feature and RANSAC in order to find the corresponding points among the frames and calculate the Homography transformation matrix to stitch them together. Although there will be a small distortion if an image that includes a 3D surface is stitched, the distortion would not be a serious obstruction for unifying the locations of the features.

5.2 Attention

For comparison with visual touch detection, we attached an IMU to a headset to capture the wearer's head motion, and then we adopted a supervised learning approach, as mentioned in [19], to detect the wearer's attention existence from the angular velocity of the head, while the attention location estimator was trained by kernel regression with IMU data.

5.3 Hotspots Detection

We implemented the following two methods for hotspot detection and compared their performance. In the following cases, although we explain the methods by mentioning

Table 1 List of hotspots and interaction patterns.

No.	Patterns (p_i)	Hotspots (h_j)
1	push	pattern button
2	rest (relax)	speed button
3	slide	needle button
4	rotate	start/stop button
5	hold	cloth plate
6	draw (cloth)	lever (*occluded)
7		thread button

touches, attention locations can be used as an alternative.

Frequency-Based Approach. We simply detect hotspots by finding areas with high frequencies of touches on the global map, which is divided into blocks of size r, and then we calculate the accumulated frequency of touches on each block b_j (j =1,2... mn/r^2, where m and n are the height and the width of the global map, respectively). Blocks with frequencies over a threshold can be regarded as hotspots. However, some areas such as the cloth plate are touched many times during the whole process, whereas some portions like the needle button are touched fewer times with quick contact. The latter areas may be ignored by the frequency-based approach regardless of their importance.

Temporal Clustering (TC) Approach. Touches on the same portion which appear frequently within a short period of time can be considered good clues to specifying hotspots. We assume a small temporal sliding window ω with a step σ, and then we perform the clustering of all touch points within this small temporal window. If a certain area has been touched more than κ times in ω, the touches are regarded as valid. If a valid touch appears at a location, the location is added as a new hotspot. With this approach, we were able not only to accept essential spots with low global frequencies but also to filter out fake touches caused by depth estimation errors that last only one or two frames. The detected hotspots are represented by $\Theta(x, y, r, t_s, t_e)$, where x and y designate their locations on the global map and r is the area, while t_s and t_e are the starting and ending times, respectively. The TC approach is illustrated in algorithm 1.

5.4 Interaction Classification

For each detected hotspot, we classified the interaction using the hand shape at the hotspot and assigned the interaction type as a property of the hotspot. Table 1 shows the hotspots and interaction patterns included in our experiments of sewing machine operations. For the classification of the hand shapes, we used a Random Forest (RF) classifier [31]. We trained the RF classifier with 25,373 hand images manually labeled into six interaction categories: *push, rest (relax), rotate, slide, hold (lever),* and *draw (cloth)*.

6. Task Modeling

An observation of a machine operation experience E_j can be decomposed into a sequence of temporal interactions I_k, each of which is a combination of a hotspot and an interaction pattern:

Algorithm 1 Temporal Clustering (TC) approach

Input: temporal window size ω, frequency threshold κ, sliding step σ.
Output: hotspots Θ_n and hotspots number n.
Initialization: $n = 0$, reference frame R_0 = zeros(size of global map).
for i = 1 to *end* with step σ **do**
a) Gather all touch spots P_i in the window ω_i, create a touch binary index map M_i with all the touch spots.
b) For each connected-area C_j in M_i, cluster P_i to C_j, as P_{ij}; if the number of $P_{ij} < \kappa$, erase corresponding area C_j.
c) After checking all C_j, get new touch index map $M_i^{'}$.
if any area $C_j^{'}$ in $M_i^{'}$ has newly appeared to reference map R_{i-1} **then**
$n++; \Theta_n = C_j^{'}; t_s^n = i; R_i = M_i^{'}$,
else
if Θ_{n-1} disappeared compare to R_{i-1} **then**
$t_e^{n-1} = i, R_i = M_i^{'}$,
end if
else
update $R_i = (R_i + M_i^{'})/2$.
end if
end for

$$E_j = \{I_1, I_2, I_3 \ldots I_n, I_e\}; \quad I_k =< \boldsymbol{p_i}, \boldsymbol{h_j} >, \tag{10}$$

where n is the total number of interactions of the experience and I_e is the "end symbol" that is added to normalize the length of all samples for a task, which can be represented as $T = \{E_1, E_2, E_3 \ldots E_m\}$, where m is the total number of experiences. As mentioned in Sect. 3, actual tasks usually have a certain degree of freedom that allows order changes and alternative ways for doing the same thing while noises, such as accidental touches or detection errors, may be involved.

This study attempted to obtain a plausible model to explain all experiences, including their possible derivations. Hence, we applied the HMM to the above problem. Considering the temporal properties of an operational task, we chose a "left-to-right" (LR) HMM, which allows only the hidden states to remain as self-transition and forward-transition probabilities. With this structure, the order changes of interactions and allowable substitutions of certain interactions can be modeled by groups of hidden states.

For one task, multiple experiences are used to train an HMM with the Baum-Welch [26] algorithm. A task model needs to be carefully chosen to balance its goodness of fit with its complexity. We input the interaction sequences extracted from all of the experiences of a task to an HMM, training with a wide range of state numbers n_s from 1 to $2n_l$, where n_l is the length of the observation samples of the task after normalization. In order to evaluate each trained model, we calculate the sum of output log-likelihood ψ for all samples, through the Forward Algorithm:

$$\psi_{n_s} = \sum_{j=1}^{k} ln\,(P(E_j|n_s)), \tag{11}$$

where k is the total number of samples in a task.

To prevent overfitting, we do not choose the state number n_s that gives the maximum output probability ψ for all

samples, but rather the first number $\overline{n_s}$, whose output probability is greater than the ratio ρ to the maximum output probability ψ as

$$\overline{n_s} = min\{n_s : \psi_{n_s} \geq \rho \cdot max\{\psi_{n_s}\}\}, \quad (12)$$

where $n_s \subset \{1 \dots 2n_l\}$ and ρ is a small constant parameter between 0 and 1. We empirically set ρ to 0.95 in our experiments for all tasks because the data for modeling still contain noises that are caused by accidental touches, hotspot misdetections, and pattern classification errors. If we choose a larger state number, the trained model will overfit to these noises and some states would even be assigned to these noises.

7. Experimental Results

7.1 Experimental Environment of Machine Operation

In our experiment, 12 participants of both genders and different skin colors were asked to sit at a table and use a sewing machine. A total of 71 experiences were recorded for the three different tasks shown in Table 2.

According to the standard "Sewing Machine Operation Manual" that was provided by the manufacturer, Task A included five operations with a fixed order of operational steps and no degrees of freedom. Two operations consisted of touching the same spot continuously. Task B contained two methods to accomplish the task: one had six steps and the other had eight. The major steps of both methods were the same, but several steps were substitutable. There was one occluded step. Unnecessary or supportive touches were also possible. Task C had nine essential steps, of which contained two order-changeable operations. The occluded interactions appeared twice at the same location.

The recording devices used were Intel RealSense SR300, with a resolution of 640×480 for both color and depth sources at 30 fps (the actual fps dropped to about 23 fps for real-time recording), and an IMU (LP-WS1105 16G/1500 dps [32]) attached above the camera, which provided geomagnetism, acceleration, and angular velocity data at 50 Hz. The operational tasks usually lasted for 1–2 minutes. About 3,000 frames for each experience were gathered.

The process of hotspot detection was performed offline after each FPV video was recorded. It is mainly because the registration of each FPV frame to the global map is not tractable in real-time processing, and the detected touches need to be accumulated throughout the record.

7.2 Evaluation Metric and Parameters

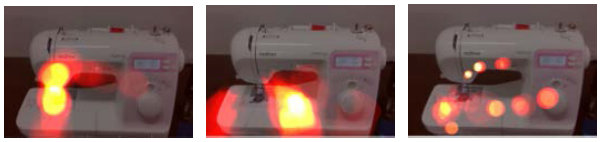
The ground truth of the temporal interactions $< \boldsymbol{p_i h_j} >$ and their orders for each task are manually made by consulting the standard operation manual, which gives the most common methods for achieving each task.

We labeled all hand shapes into six catalogs. The RF classifier was trained with 50 trees. To distinguish between the hands being supportive or active, we first smoothed the motion of the fastest moving area of the hand with a median filter and used the small velocity threshold of $\boldsymbol{C}$ = 2.4 cm/s (on average 1 pixel per frame) in the detection of an occluded interaction. As with removing the noise of a supportive hand, if the interaction pattern in a sequence was classified as $\boldsymbol{p_2}$ (*rest*) and the average motion of the hand was lower than $\boldsymbol{C}$, it is removed. Figure 11 shows the classification of the performances of the hand shapes of all hotspots. The classification was evaluated with a 10-fold cross-validation.

In order to detect hotspots with the TC approach, we found that the depth measurement noises usually flashed out quickly. However, the true touches usually lasted at least 0.5 s longer. Hence, the within-window frequency threshold κ (mentioned in Sect. 5.3) was set to 0.3 of fps in our experiment. Meanwhile, the detection results were not sensitive to temporal window size, because the windows were overlapping. We set ω to 1 second and the sliding step σ to 0.25ω. In order to filter out saccades in detecting attention, we empirically reduced the angular velocity threshold to 50°/s as compared to the walking-around situations in [19].

Table 2 The properties of the three operational tasks.

Properties	Task A	Task B	Task C
number of experience	17	26	28
operation steps (occluded)	5 (0)	6 or 8 (1)	9 (2)
hotspots (occluded)	4 (0)	4 (1)	6 (1)
interaction patterns	3	4	6
repeating touches	o	o	o
supportive touches (noises)	-	o	o
alternative operations	-	o	-
order-changeable operations	-	-	o

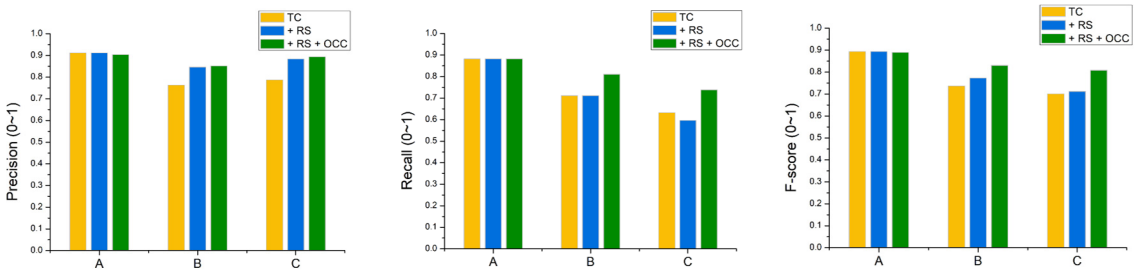


(a) (b) (c)

Fig. 5 Locations detected by (a) attention, (b) hand position, and (c) touch. These are visualized by the heat map (accumulated frequencies).

7.3 Hotspot Detection

We compared the performance of hotspot detection for three commonly used features: the hand location, the attention location, and the touch location. Figure 5 shows the accumulated frequencies of the three features on the global map. We noted that the touch location achieves the finest resolution as compared to the attention location and hand location. So, we applied only the touch-based method for further processing. We then compared the simple frequency-based (Fr) approach and the TC approach. Figure 6 above illustrates the results of the hotspot detections for three different operational tasks. The results of both methods match the crucial locations (i.e., the buttons or plate) on the machine's sur-



(a) (b) (c)

Fig. 7 (a) Precision, (b) Recall, and (c) F-score of the retrieval of the temporal interactions for three tasks.

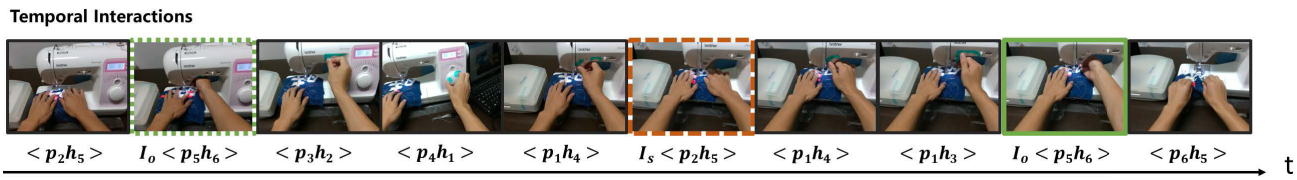


Fig. 8 Examples of the results of the detection of temporal interactions in Task C. In this experience, ten interactions are included: seven touch-based interactions (*black*), two occluded interactions (*green*), and one supportive interaction (*orange*). Each temporal interaction is automatically labeled with hotspot locations and its interaction patterns, where I_s is a supportive interaction and I_o is an occluded interaction.

Fig. 6 Results of hotspot detection with the Fr approach (upper row) and the TC approach (lower row) on the global map.

face. However, the TC approach shows a higher spatial resolution (connected areas) with smooth boundaries, whereas the block-based Fr approach contains some irregular adjacent areas and noises. In the following, we show the results of using touch with the TC approach.

7.4 Temporal Interaction Detection

We evaluated the performance of interaction detection compared to the ground truth. The results of the detection of temporal interactions before and after the detection of occlusions and noise removals are illustrated in Fig. 7. Higher F-scores for Task A than for other tasks is obtained both before and after the postprocessing due to the task's simplest interaction patterns. The increasing of steps and patterns for Task B and C, the detection errors, the supportive noises, and the missing occlusion steps were the main reasons for the degeneration of the score. The removal of supportive interactions (+RS) was the main reason for the significant improvements in precision for Tasks B and C. After the occluded interactions (+OCC) were added, the recall rate significantly improved. For the overall result, the accuracies of the detections of the supportive interactions and occlusion interactions were 79.2% and 68.3%, respectively. The detailed results of these processes are shown in Table 3.

Table 3 Detailed results of interaction detection (recall and precision).

Interaction Type		Regular	Supportive	Occlusion
Task A	*ratio*	100%	0%	0%
	R	0.88	0	0
	P	0.91	0	0
Task B	*ratio*	74.8%	12.6%	12.6%
	R	0.83	0.81	0.77
	P	0.87	0.96	0.95
Task C	*ratio*	71.5%	8%	20.4%
	R	0.81	0.77	0.64
	P	0.86	0.71	0.92
Average	*R*	0.83	0.79	0.68
	P	0.88	0.83	0.92

Fig. 8 shows some examples of detecting interactions. For examples, one occlusion interaction has been retrieved (*solid green*) while the other is still missing (*dotted green*). Meanwhile, one supportive interaction (*orange*) has been removed.

7.5 Task Modeling

We checked the performance of the obtained task models with the different numbers of hidden states. Figure 9 shows the sum of the log-likelihood ψ_{n_s} given by Eq. (11) and the chosen optimal numbers $\overline{n_s}$ (red dots) in Eq. (12). The sum

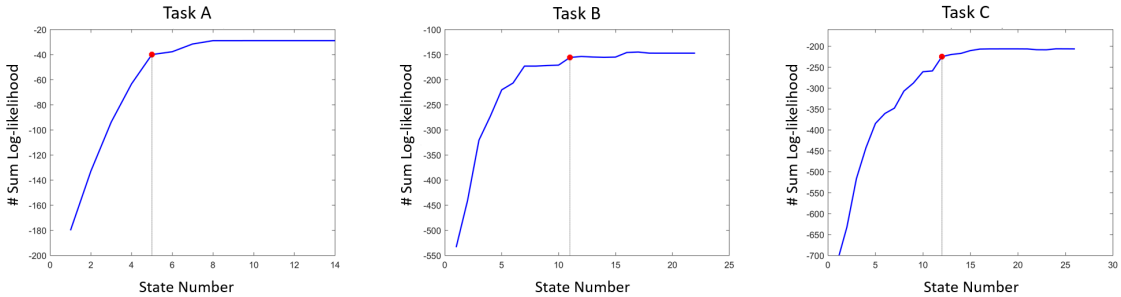


Fig. 9 Relationships between the number of states and the performance of the models. The optimal numbers for Tasks A, B, and C are 5, 11, and 12, respectively.

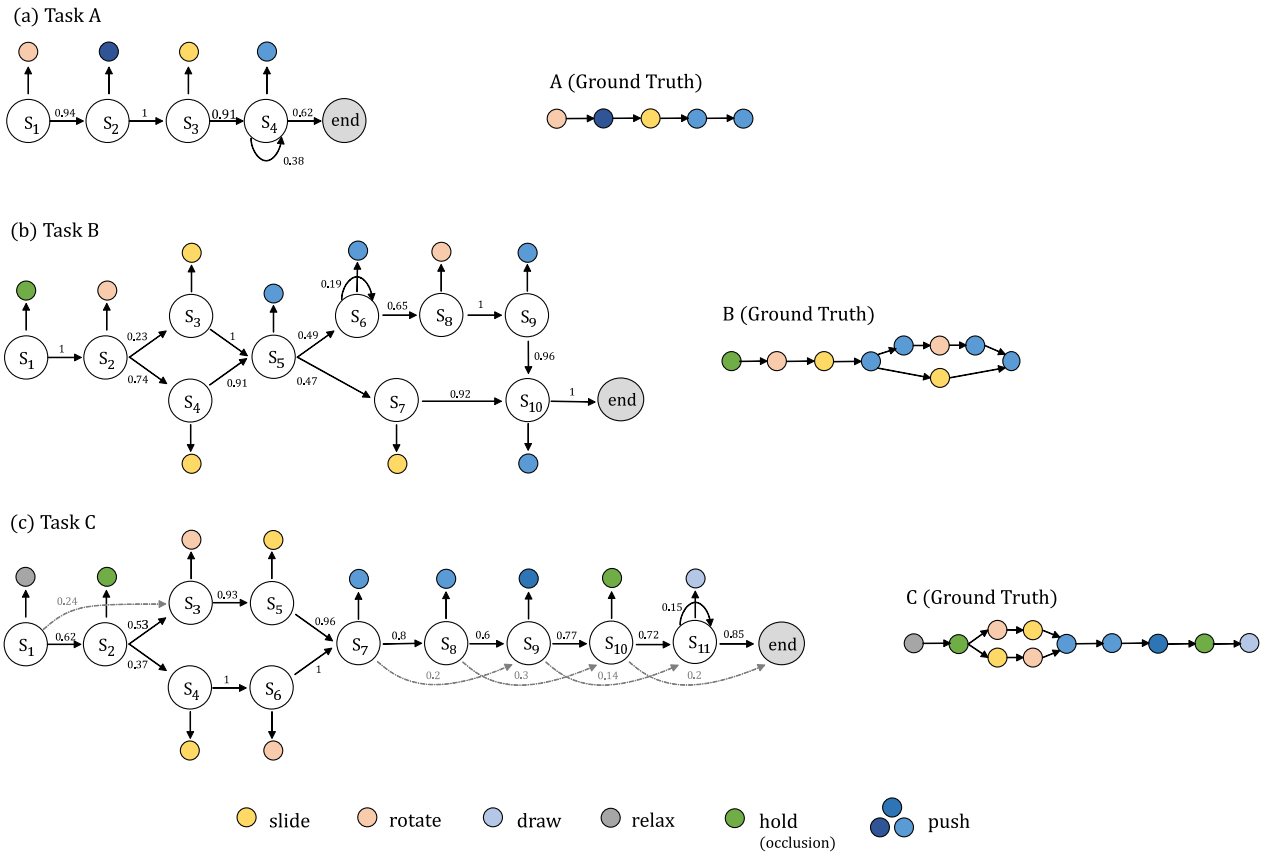


Fig. 10 Results of task modeling by HMM. The hidden states and their transition probabilities are shown. Observations (interactions) are represented by circles, to which colors are given on the basis of their interaction patterns. State transitions and observations with very low probabilities (less than 0.1) are omitted.

of the log-likelihood increases quickly with the increasing number of states, but it slows down after reaching the optimal number.

Figure 10 illustrates the obtained task models. The interaction patterns and their orders, as given in the standard operation manual, are shown in the right column. The acquired models are shown in the left columns.

The result of Task A was almost perfect model of single routines in which each hidden state corresponded to an independent interaction, that is, four observations (interactions) to four states. In addition, S_4 shows a relatively high self-transition probability, which represents a self-repeating interaction of pushing the same button twice. The postprocessing does not affect the modeling because no supportive or occluded interactions had been included in this task (see Table 2).

Task B had two alternative ways of accomplishment, as depicted in the standard manual. The two transition paths in the obtained model, that is ($S_5 \rightarrow S_6, S_8, S_9 \rightarrow S_{10}$) and ($S_5 \rightarrow S_7 \rightarrow S_{10}$), support the alternative procedures. The repeating push interactions in one way (S_5, S_6 and S_9, S_{10}) while the single push interaction in the other way (S_5 and

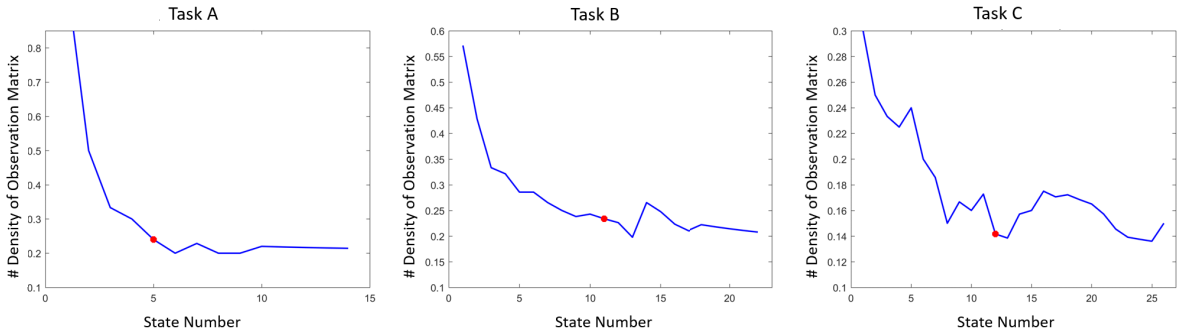


Fig. 12 The correlation between the Hidden State Numbers and the Density of the Observation Matrix. The *Density (1-Sparsity)* is calculated with k/mn, where k is the number of non-zero elements and m, n is the size of the observation matrix. The red dots show the state numbers chosen with our threshold-based method.

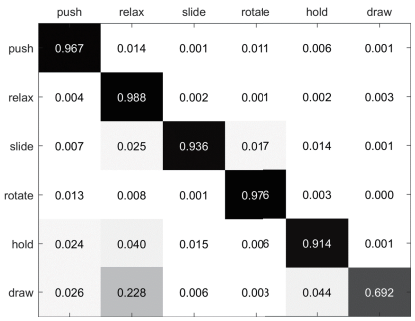


Fig. 11 Results of interaction pattern classification by the RF classifier, evaluated with 10-fold cross-validation.

S_{10}) are shown accurately. Furthermore, the occluded essential interactions (*green*) are supplemented to a relatively high probability to be observed, whereas the unnecessary noises (*gray*) are reduced to a low probability to disappear in the final model. Although S_3 and S_4 are split, they show the same observations and the same "in-and-out" states. This splitting may be caused by some noises appearing before or after the same observation among different experiences. Therefore, we can regard the two states as being a same state.

For Task C, the obtained model matches the standard operations well with all the noise interactions excluded and all the essential interactions included. Each state only observes a single pattern. However, there are some low-probability side transition paths that occurred because of the misdetections in the training samples.

7.6 Discussion

The experimental results show that the acquired task models represent the properties and variations of the experts' operations adequately. Although misdetections occur because of noises and occlusion, all essential operations are properly retrieved by integrating multiple experiences. However, learning with only positive expert samples may not result in a perfect model with clear boundaries between correct and incorrect samples. We need to introduce failures, such as beginner's experiences, in order to acquire more precise models in future studies.

We note that the observations in each hidden state in the obtained models are sparse; that is, in any hidden state, only a single interaction pattern has the dominant probability, as shown in Fig. 10. Alternative or order-changeable interactions are modeled on separate paths. Figure 12 shows the correlation between the hidden state numbers and the sparsity of the observation matrix. The chosen optimal state numbers (red dots) lie a little ahead of the sparsest point. Thus, the sparse solution of the observation matrix may also serve as a criterion to choose the state number for HMM.

This point can be an important factor for possible applications. We can consider applications such as (i) *automatic guidance generation*, in which an observation sequence with the highest probability is suggested as the most probable way of achieving a task at each step, and (ii) *behavior prediction* or *fault detection*, in which the most plausible future operational steps are predicted and used for avoiding failure. If the candidates for the interaction at each state are limited, guidance or prediction can be easier, because we can safely recommend or choose the appropriate interaction.

For future studies, the integration of a beginner's experience is an interesting topic. Beginners' experiences that may contain more variations or even faults as mentioned above could provide valuable information for (i) analyzing interaction difficulties or possible faults, (ii) inspiring experts to find new ways to execute tasks, (iii) acquiring better models for guiding beginners, (iv) user ability accessing, and so forth.

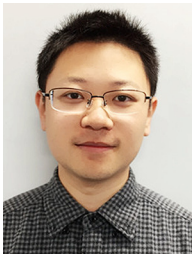
8. Conclusion

In this paper, we proposed a novel approach for analyzing and modeling recorded experiences, via egocentric vision, of the operating of machines. With our method, important regions of the machine and interaction patterns were detected as hotspots, after which HMM was applied to the modeling of the interaction sequences. The experimental results of 71 experts' experiences illustrate the method's effectiveness for accurately retrieving essential operation in-

teractions and modeling operational tasks. However, there were still misdetections of hotspots that may be harmful to obtaining good operational models, and there were difficulties in determining the number of hidden states in the HMM. These problems must be considered in conjunction with specific applications (e.g., what guidance is designed for whom and how). Experiences of other types of users, such as beginners, could be good resources for this purpose.

References

[1] A. Betancourt, P. Morerio, C.S. Regazzoni, and M. Rauterberg, "The evolution of first person vision methods: A survey," IEEE Trans. Circuits Syst. Video Technol., vol.25, no.5, pp.744–760, 2015.
[2] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using RGB-D," Proc. 2012 ACM Conference on Ubiquitous Computing, pp.208–211, ACM, 2012.
[3] A. Fathi, A. Farhadi, and J.M. Rehg, "Understanding egocentric activities," 2011 International Conference on Computer Vision, pp.407–414, 2011.
[4] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.2847–2854, 2012.
[5] K. Matsuo, K. Yamada, S. Ueno, and S. Naito, "An attention-based activity recognition for egocentric video," 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.565–570, 2014.
[6] Y.J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.1346–1353, 2012.
[7] S. Sundaram and W.W.M. Cuevas, "High level activity recognition using low resolution wearable vision," 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp.25–32, 2009.
[8] A. Fathi, X. Ren, and J.M. Rehg, "Learning to recognize objects in egocentric activities," CVPR 2011, pp.3281–3288, 2011.
[9] G. Rogez, J.S. Supancic, and D. Ramanan, "Understanding everyday hands in action from RGB-D images," 2015 IEEE International Conference on Computer Vision (ICCV), pp.3889–3897, 2015.
[10] A. Agarwal, S. Izadi, M. Chandraker, and A. Blake, "High precision multi-touch sensing on surfaces using overhead cameras," Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07), pp.197–200, 2007.
[11] E.N. Saba, E.C. Larson, and S.N. Patel, "Dante vision: In-air and touch gesture sensing for natural surface interaction with combined depth and thermal cameras," 2012 IEEE International Conference on Emerging Signal Processing Applications, pp.167–170, 2012.
[12] A.D. Wilson, "Using a depth camera as a touch sensor," ACM international Conference on Interactive Tabletops and Surfaces, pp.69–72, ACM, 2010.
[13] S. Murugappan, R.K. Vinayak, N. Elmqvist, K. Ramani, "Extended multitouch: Recovering touch posture and differentiating users using a depth camera," Proc. 25th Annual ACM Symposium on User Interface Software and Technology, pp.487–496, ACM, 2012.
[14] C. Harrison, H. Benko, and A.D. Wilson, "OmniTouch: Wearable multitouch interaction everywhere," Proc. 24th Annual ACM Symposium on User Interface Software and Technology, pp.441–450, ACM, 2011.
[15] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W.W. Mayol-Cuevas, "You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video," BMVC, pp.30.1–30.13, 2014.
[16] I. Mitsugami, N. Ukita, and M. Kidode, "Estimation of 3D gazed position using view lines," 12th International Conference on Image Analysis and Processing, 2003.Proceedings., pp.466–471, 2003.
[17] V. Rantanen, T. Vanhala, O. Tuisku, P.H. Niemenlehto, J. Verho, V. Surakka, M. Juhola, and J. Lekkala, "A wearable, wireless gaze tracker with integrated selection command source for human-computer interaction," IEEE Trans. Inf. Technol. Biomed., vol.15, no.5, pp.795–801, 2011.
[18] K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki, "Attention prediction in egocentric video using motion and visual saliency," Pacific-Rim Symposium on Image and Video Technology, pp.277–288, Springer, 2011.
[19] T. Leelasawassuk, D. Damen, and W.W. Mayol-Cuevas, "Estimating visual attention from a head mounted IMU," Proc. 2015 ACM International Symposium on Wearable Computers, pp.147–150, ACM, 2015.
[20] Y. Okinaka, I. Mitsugami, and Y. Yagi, "Gaze estimation based on eyeball-head dynamics," Proc. IPSJ CVIM, vol.2016, 2016.
[21] Y. Li, A. Fathi, and J.M. Rehg, "Learning to predict gaze in egocentric video," 2013 IEEE International Conference on Computer Vision, pp.3216–3223, 2013.
[22] M. Okamoto and K. Yanai, "Summarization of egocentric moving videos for generating walking route guidance," Pacific-Rim Symposium on Image and Video Technology, pp.431–442, Springer, 2013.
[23] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp.2537–2544, 2014.
[24] K.M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," CVPR 2011, pp.3241–3248, 2011.
[25] B. Clarkson, K. Mase, and A. Pentland, "Recognizing user context via wearable sensors," Digest of Papers. Fourth International Symposium on Wearable Computers, pp.69–75, 2000.
[26] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," The Annals of Mathematical Statistics, vol.41, no.1, pp.164–171, 1970.
[27] J.A.M. Basilio, G.A. Torres, G.S. Pérez, L.K.T. Medina, and H.M.P. Meana, "Explicit image detection using ycbcr space color model as skin detection," Applications of Mathematics and Computer Engineering, pp.123–128, 2011.
[28] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," Sankhyā: the Indian Journal of Statistics, pp.401–406, 1946.
[29] N. Shimada, K. Kimura, and Y. Shirai, "Real-time 3D hand posture estimation based on 2d appearance retrieval using monocular camera," Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp.23–30, 2001.
[30] https://software.intel.com/en-us/realsense/sr300.
[31] L. Breiman, "Random forests," Machine learning, vol.45, no.1, pp.5–32, 2001.
[32] http://www.lp-d.co.jp/smallWiMs.html/.



Long-fei Chen is now a Ph.D student in the Department of Electrical Engineering, Kyoto University. He received B.E and M.E in electrical engineering from Sichuan Agricultural University and Sichuan University in 2011, 2014, respectively. His research interests are on computer vision and human-computer interaction.

Yuichi Nakamura received B.E, M.E, and Ph.D degrees in electrical en gineering from Kyoto University, in 1985, 1987, and 1992, respectively. From 1990 to 1993, he worked as an instructor at the Department of Electrical Engineer ing of Kyoto University. From 1993 to 2004, he worked for Institute of Information Sciences and Electronics of Univer sity of Tsukuba, Institute of Engineering Mechanics and Systems of University of Tsukuba, as an assistant professor and an associate professor, respectively. Since 2004, he has been a professor of Academic Center of Computing and Media Studies, Kyoto University. His research interests are on computer vision, multimedia, human- computer and human-human interaction including distance com munication, and multimedia contents production.

Kazuaki Kondo received his M.E. and Ph.D. degrees from Osaka Univer sity in Japan. He became a research as sociate at Osaka University in 2007, an assistant professor at Kyoto university in 2009, and a lecturer in 2015. He was awarded the Kusumoto award in 2002. His research interests are computer vision and intelligent support on human commu nications. He is a member of IEICE.

Walterio Mayol-Cuevas received his PhD in 2005 from The University of Oxford. And his BSc from The National University of Mexico (UNAM) in 1999. Since 2015 he is Professor at the Department of Computer Science, University of Bristol. His interests span Computer Vision, Robotics and Mobile Computing. Currently directs projects and dissertations on topics such as assistive systems, handheld robotics, automated learning from observation, visual mapping and novel visual sensors. Was general co-chair of BMVC 2013 and General Chair of IEEE ISMAR 2016.