EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Coarse-to-fine online learning for hand segmentation in egocentric video

Ying Zhao[1,2*], Zhiwei Luo[2] and Changqin Quan[2]

Abstract

Hand segmentation is one of the most fundamental and crucial steps for egocentric human-computer interaction. The special egocentric view brings new challenges to hand segmentation tasks, such as the unpredictable environmental conditions. The performance of traditional hand segmentation methods depend on abundant manually labeled training data. However, these approaches do not appropriately capture the whole properties of egocentric human-computer interaction for neglecting the user-specific context. It is only necessary to build a personalized hand model of the active user. Based on this observation, we propose an online-learning hand segmentation approach without using manually labeled data for training. Our approach consists of top-down classifications and bottom-up optimizations. More specifically, we divide the segmentation task into three parts, a frame-level hand detection which detects the presence of the interactive hand using motion saliency and initializes hand masks for online learning, a superpixel-level hand classification which coarsely segments hand regions from which stable samples are selected for next level, and a pixel-level hand classification which produces a fine-grained hand segmentation. Based on the pixel-level classification result, we update the hand appearance model and optimize the upper layer classifier and detector. This online-learning strategy makes our approach robust to varying illumination conditions and hand appearances. Experimental results demonstrate the robustness of our approach.

Keywords: Hand detection, Hand segmentation, Egocentric, Unsupervised online learning

1 Introduction

Recently, the first-person camera embedded wearable computer, such as augmented reality headset and smart glasses, is growing vigorously and urgently requires suitable interaction patterns for egocentric vision. One feasible option is taking user's hand as the medium for human-computer interaction. The wearable computer interprets hand position, posture, and gesture into commands and produces appropriate responses to the user. These properties of hand are preceded by reliable hand detection and segmentation from the egocentric video. The egocentric view brings opportunities for hand detection and segmentation. Since the video is recorded from a first-person perspective, the occlusions are less likely to happen at the attention hand and the user prefers to concentrate on region in the center of view field. Meanwhile, the egocentric video also presents new challenges including rapid changes in illuminations, significant camera motion, and background clutter.

Great efforts have been made in detecting user's hand from the egocentric video especially in pixel-level detection [1–7]. Most of these methods are under an implicit assumption that the hand presents in the video all the time. But, the assumption fails in many situations in which the hand is not used, such as before or after the human-computer interaction. Subsequently, some cascade detection methods are put forwarded to get rid of the assumption by checking out hand presence before performing pixel-by-pixel classification [8–10]. However, these approaches rely on the existence of a large training set containing a broad variety of data which are collected from multiple users under diverse illumination conditions. Hand appearance varies greatly in diverse users and environmental conditions. Not only does the training set cost a lot of manual effort in data collection and labeling but also it does not guarantee to make the approach adapt to any hand appearance and environmental condition.

* Correspondence: ying.zhao@srcb.ricoh.com
[1] Ricoh Software Research Center (Beijing) Co., Ltd, Beijing, China
[2] Graduate School of System Informatics, Kobe University, Kobe, Japan

To address this issue, we propose a method for unsupervised hand detection and segmentation in egocentric video. In our approach, the frame-level hand presence or absence is observed based on motion saliency which is particular in the egocentric view. By combining motion and appearance property, we get unsupervised labeling results for the superpixel-level hand classification. Then, the pixel samples of hand are extracted according to confidences of the superpixels and used to train a pixel-level classifier which produces fine-grained hand segmentation. In order to be robust with varying environmental condition, we constantly update the classifier and detector by using a bottom-up optimization method. We test our method on challenging datasets, and the experimental results show that our method robustly produces precise segmentation, as illustrated in Fig. 1.

In summary, this paper makes three main contributions:

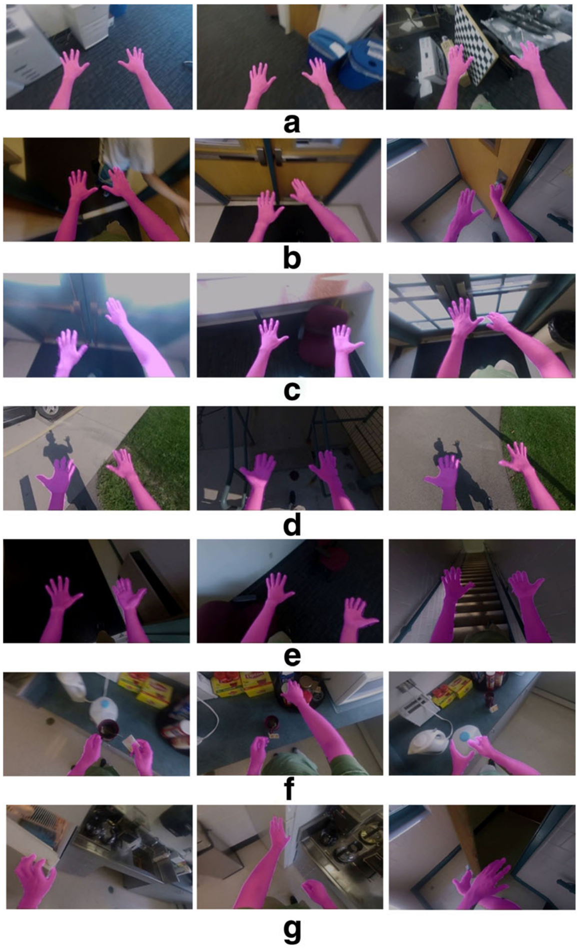

Fig. 1 Results of proposed method in challenge cases. From **a**–**g** are cases of hands are motion blur, background having skin-color, frames are overexposed, hands in contrast shadow, frames are underexposed, hands interacting with objects, and hands in varying poses

- We propose a frame-level hand presence detection method that utilizes hand motion saliency in the egocentric human-computer interaction, which reduces the false positive rate for the final target of pixel-level hand segmentation.
- We present a top-down cascaded classification method which segments hand hierarchically in levels of frame, superpixel, and pixel so as to reduce computational cost, in which the classifiers are trained on-the-fly so as to be robust to diverse users.
- We analyze and optimize the online trained classifiers by a bottom-up method which makes the hand segmentation robust to varying environmental conditions.

2 Related work

Egocentric vision is an emerging area in computer vision. According to survey of [11], the most commonly explored objective of egocentric vision is object recognition and tracking. Furthermore, hands are among the most common objects in the user's field of view, and a proper detection, localization, and tracking could be a main input for other objectives, such as gesture recognition, understanding hand-object interactions, and activity recognition [5, 12–20]. Recently, egocentric pixel-level hand detection has attracted more and more attention.

Most of the proposed methods are based on pre-training classifiers using abundant manually labeled data. Li and Kitani [1, 4] propose a pixel-level hand detection method using color- and texture-based features. Zhu et al. [2] propose a method which use local hand shape information in the training data and enforces shape constraints in the estimation. Serra et al. [3] integrate temporal and spatial consistency to complement the appearance features. Betancourt et al. [21] identify the left and right hands and models hand occlusions to improve the accuracy of hand segmentation. These methods improve the precision of pixel-level hand detection but still under the implicit assumption of hand presence in all frames. This assumption is not always true since the hand may be absence before or after the egocentric human-computer interaction.

Some of the proposed methods conquer the hand segmentation task sequentially. Betancourt [8, 22] proposes a sequential classifier consists of a hand-detector and a hand-segmentator. Betancourt et al. [9] extend SVM-based hand detector with a dynamic Bayesian network. These methods reduce false-positive rate of hand segmentation but also needs the offline training which requires manual labeled data. Kumar et al. [10] illustrate an on-the-fly hand detection training method which is initialized by a calibration gesture performed by the user. This simple preprocessing step saves a great deal of manual labeling but may not be friendly to the user. Zhu et al. [23] propose a two-stage detector which firstly generates bounding box proposals and secondly evaluates the proposals by a convolutional neural network. Moreover, all of these methods are still challenged with varying environment conditions since they do not have any model updating strategy.

In this paper, we are going to illustrate our fine-grained hand segmentation method which leverages unsupervised online learning pattern to robustly segment the hand in pixel-level from egocentric video.

3 Method

In this section, we discuss an unsupervised online learning method for fine-grained hand segmentation based on top-down classification and bottom-up optimization. By learning hand appearance and motion features on-the-fly, we segment out the hand with precise boundary from the egocentric video which is captured in varying illumination condition. From the point view of top-down strategy, we divide the classification task into three parts: frame-level detection and superpixel-level and pixel-level classifications. Before scanning pixel by pixel, we firstly estimate whether a frame contains a hand and whether a region of the frame contains hand pixels. By doing this, we reduce the false positive and initialize samples for further online training. After that, we learn feature from the labeled region and train two-level classifiers. To make sure the classifiers adapt to varying hand appearance, we update the hand appearance model and optimize the upper layer classifier and detector. Figure 2 shows the framework of our method.

3.1 Ego-saliency-based hand detection

Before scanning the frame pixel-by-pixel, the first task is detecting presence of hand from a frame-level perspective and then automatically initialize hand masks for subsequent classifications. Motion-based methods [24–26] are proposed for background subtraction for freely moving camera. In general, it is difficult to determine whether the hand is present or not without prior information about the environment or appearance of the hand. Fortunately, the egocentric interaction scenario provides many constraints that are suggestive of the hand's presence.

From the point view of an interaction cycle, the motion of hand in egocentric view has periodical specialty. In the interaction preparatory phase, the whole hand and part of the arm together gradually enter into the view field. During the interaction, the whole hand moves around the center of view field and the fingers are likely to make more vigorous motion than the palm and arm, such as making a gesture. When the interaction is finished, the whole hand and part of the arm together gradually move out of the view field. We observe that the preparatory phase is a natural bootstrap since the hand motion is more salient than other regions and the hand is hardly to enter into the view field from the top side.

Based on this observation, we define an ego-saliency metric E_f consists of spatial and temporal terms to estimate how likely the hand is present in the frame f. The higher the ego-saliency value, the more likely the hand is present.

$$E_f^{IN} = \sum_{i=1}^{W}\sum_{j=1}^{H}\left(\frac{1}{1+e^{\lambda*(h-j)}}-0.5\right)*M_f(i,j) + \sum_{t=f-n}^{f} \operatorname{sgn}(N_t - N_{t-1}) \tag{1}$$

where the first term is the spatial cue that restricts the hand motion should be salient and happened in the right position. The second term is the temporal cue that restricts the hand motion should be consequently increased. W and H denote width and height of the frame respectively. $M_f(i, j)$ is the motion saliency of a pixel at position (i, j) and calculated based on optical flow map

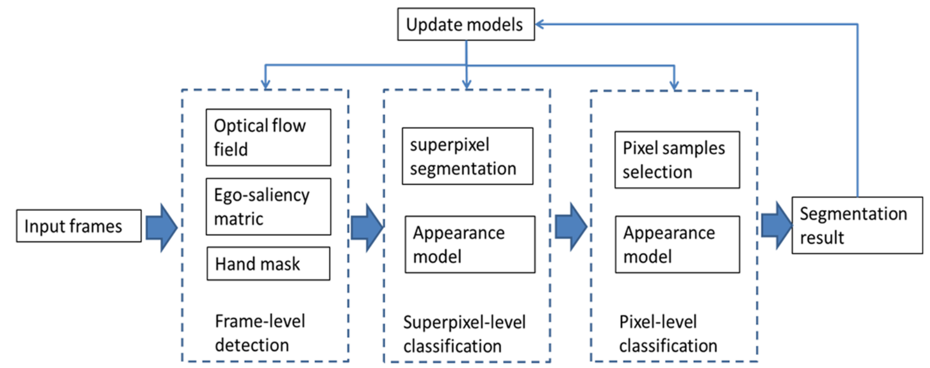

Fig. 2 Framework of proposed method

using method [27]. As shown in Fig. 3d, we set a non-interactive border with width W and height h from the top of the frame. We set h as one tenth of the frame height in experiments. And we use a distance-based exponential weight to restrict that hand motion should happen away from the non-interactive border. λ is the weight response control factor. The farther a pixel is away from the non-interactive border, the greater its weight is assigned. N_t is the number of non-zero values in the motion saliency map M_t. The consequent motion increment is observed by a sign function $sgn(\bullet)$ based on the number of pixels having salient motion in adjacent n frames.

After detecting the presence of hands, we initially segment moving hand regions based on motion and appearance clustering. By using dual TV-L1 optical flow [28], we extract dense motion flow fields and get a motion map. We cluster the motion map into k groups of regions using K-means and we set k as 10 in the experiments. The motion clustering naturally divides foreground and skin-colored background into different regions since they usually move differently. Figure 3d shows the regions got from motion clustering and the non-interactive border. With the help of non-interactive border, we easily select out a set of background regions $\{R_{BG}\}$ which intersect with the border. The rest unknown regions are further determined based on appearance clustering. According to Eq. (2), we calculate the likelihood $H_f(R_i)$ of an unknown region R_i belonging to hand region based on the similarity between the unknown region R_i and background regions $\{R_{BG}\}$. $S(\bullet,\bullet)$ is a function calculating color histograms similarity of two regions. Then, we find out the hand regions which have low color similarity with all the background regions. Figure 3 shows the initialized hand mask which is generated by using motion and color clustering.

$$H_f(R_i) = \sum_{j \in BG} S(R_i, R_j) \tag{2}$$

3.2 Online training two-level hand classifiers

With the ending of the interaction preparatory phase, hand motion is attenuating and may eventually become much less salient, such as only the fingers move to make a gesture while the palm holds still. Moreover, motion-based segmentation usually produces the result with blurry and noise boundaries around objects. Therefore, the appearance feature is more discriminative than motion cue for fine-grained hand segmentation during the interaction phase (Fig. 4).

Here, we address a coarse-to-fine strategy-based hand segmentation method that learns appearance feature of hand and background on-the-fly. Based on the initial set of hand masks $\{B_t\}$ got from the frame-level detection, we firstly train a superpixel-level hand classifier so as to segment frames into superpixel regions from which the stable pixel samples are selected out. Then, we utilize the selected pixel samples to train a pixel-level classifier which produces fine-grained hand segmentations.

In the frame-level detection step, we obtain a coarse segmentation of the hands using motion and ego-saliency cues. It initially provides the ground truth labels of hand regions for superpixel-level training. We overly segment the recent n consecutive frames $\{F_t\}$ into superpixels by using a modification of a state-of-the-art algorithm termed simple linear iterative clustering (SLIC) [29].

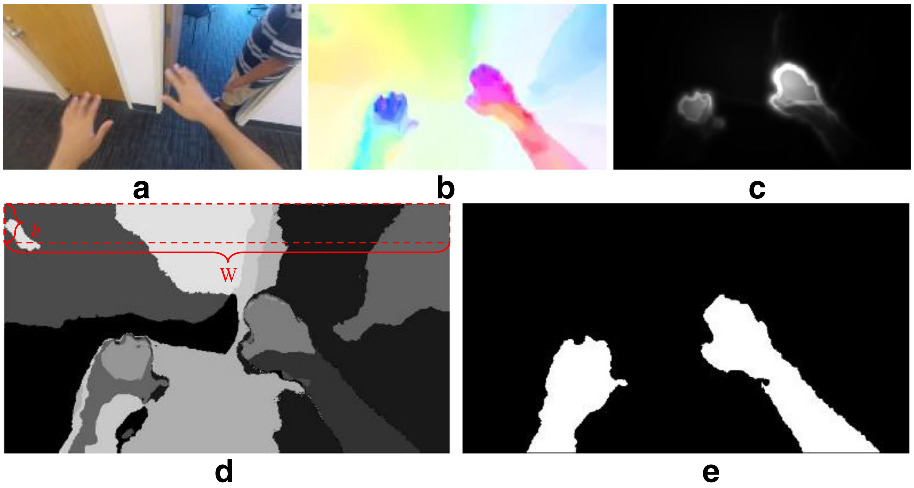

Fig. 3 Initialize hand label based on ego-saliency. **a**–**e** are original image, optical flow to the next frame, motion saliency of **b**, clustering result of **b** with non-interactive border labeled in red, and estimate hand regions

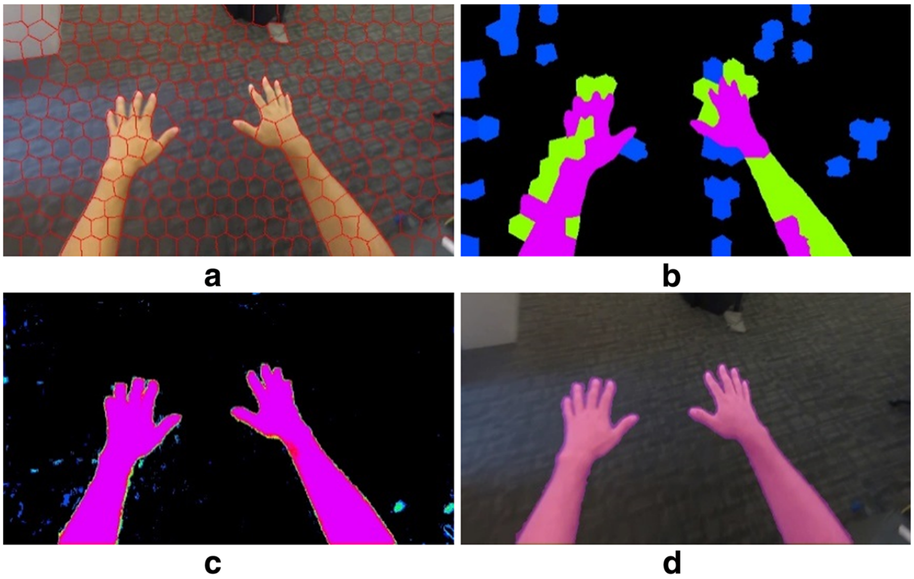

Fig. 4 Example of two-level classification. **a**–**d** are superpixel segmentation overlaid on original image, probability map of superpixel-level classification, probability map of pixel-level classification, and final segmentation overlaid on original image

The K-means clustering of motion map derives a binary segmentation separating the foreground from the background. However, the K-means segmentation has coarse boundaries which are sometimes inconsistent with the superpixels'. To select good samples for superpixel-level training, we initialize a label map based on the portion of positive pixels in each superpixel and refine it by energy optimization. Figure 5 illustrates the process of superpixel sample selection. Given a binary mask of the K-means segmentation, we assign the superpixels having 80% positive pixels as foreground candidates and their dilated superpixels as background candidates. The candidates are further selected based on confidence score calculation and energy optimization.

We define a confidence score to describe how much the superpixel is more similar to its homogeneous neighbors than the heterogeneous neighbors. For a candidate superpixel, we calculate its confidence score as Eq. (3).

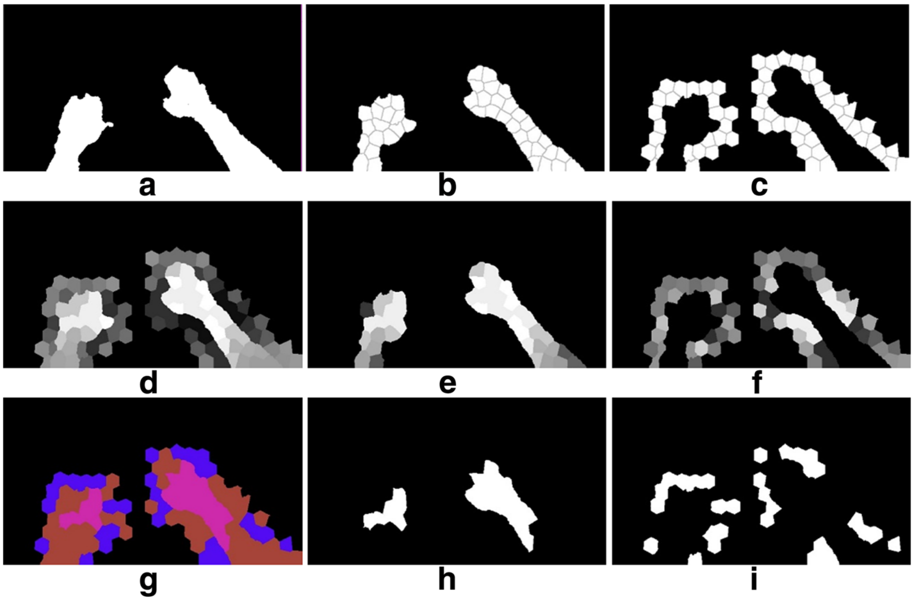

Fig. 5 Select superpixel samples based on energy optimization. **a**–**i** show the process of selection. Blue, pink, and brick in (**g**) illustrate the selected background, foreground, and abandoned samples respectively

After normalization, we get a score map as shown in Fig. 5d.

$$\mathrm{Score}_i = \frac{1}{Z}\sum_{j\in\Omega_i^-}\sum_{k\in\Omega_i^+}\frac{D\left(h^i,h^j\right)}{D\left(h^k,h^j\right)} \tag{3}$$

$$D\left(h^i,h^j\right) = \frac{1}{\sum_k c_k} c_1 D\left(h^i_{\mathrm{SIFT}},h^j_{\mathrm{SIFT}}\right) + c_2 D\left(h^i_{\mathrm{RGB}},h^j_{\mathrm{RGB}}\right) \tag{4}$$

where Ω_i^- and Ω_i^+ are sets containing samples collected from the neighborhood of superpixel *i*, the superscript “-” indicates that the samples have different class label with superpixel ***i*** while “+” stands for the contrary situation, and *Z* is a normalization factor. And, h_{SIFT} and h_{RGB} denote the SIFT and RGB histograms respectively, $D(h^i, h^j)$ is the Chi-square distance between the histograms h^i and h^j, and c_k is a constant to normalize the *k*-th descriptor.

We take the score as a label and optimize it for each superpixel by using Ising model [30]. The foreground and background candidates constitute a foreground system and a background system respectively. The energy of each system consists of the affinities and consistencies of superpixels to their neighborhood within the system. Color and texture are useful cues since foreground tends to have a difference appearance than the background behind it. Therefore, the affinity between a superpixel and its neighbor is computed as the Chi-square distance between their color and texture histograms. Higher affinity indicates stronger consistency for belonging to the same class. Therefore, we optimize the label based on an energy which encourages coherence in superpixels of similar appearance. For a superpixel, we inverse its label and calculate the energy change caused by the inversion. This label inversion is directly accepted if the system energy is increased. On the contrary, the process is further judged by an acceptance function. This routine is repeatedly executed until the system reaches equilibrium. Then, the superpixel labels are optimized.

Given a labeled region, we calculate the energy of each superpixel within it and accumulate them together to describe its system energy. For a superpixel, we first compute an affinity score and a label consistency score for each pair of adjacent superpixels. After normalizing the scores, we calculate their correspondence which is proportional to the superpixel energy. Based on the exponential correspondence, we obtain the superpixel energy. After that, we compute the system energy as

$$E = \sum_i \sum_{j\in\Omega_i^o} e^{-|S(i,j)-L(i,j)|} \tag{5}$$

where Ω_i^o is the neighborhood of superpixel *i* within the system, *S*(*i*, *j*) is the affinity, and *L*(*i*, *j*) is the label consistency between two adjacent superpixels.

To describe the appearance of a superpixel, we compute the histograms of SIFT features and RGB values from the image area of it occupies. Considering the appearance feature is prone to be coherence in a local region, we use the distance between two adjacent superpixels to restrict the contribution of the neighboring superpixel. Larger distance indicates smaller contribution. Moreover, the superpixels nearing to the system boundary are tend to be unstable. Hence, the distance from superpixel to boundary is also a term of the affinity score. Based on these four descriptors, the affinity score *S*(*i*, *j*) is defined for superpixel pair (*i*, *j*) as Eq. (6).

$$S(i,j) = 1-\frac{1}{\sum_k c_k}\left(c_1 D\left(h^i_{\mathrm{SIFT}},h^j_{\mathrm{SIFT}}\right) + c_2 D\left(h^i_{\mathrm{RGB}},h^j_{\mathrm{RGB}}\right) + c_3 A(i,j) - c_4 B(j)\right) \tag{6}$$

where *A*(*i*, *j*) is the Euclidean distance between the adjacent superpixel centers, and *B*(*j*) is the Euclidean distance from superpixel *j* to the system boundary.

We inverse the label of superpixel *i* and get its updated label consistency score *L*'(*i*, *j*) with the adjacent superpixel *j*. The energy of the system is renovated correspondingly. Then, we compute the increment $\triangle$E of the system energy. Based on the increment, we decide if it should accept the label inversion.

$$\mathrm{label}(i) = \begin{cases} \mathrm{label}(i), \Delta E \le 0 \text{ and } \exp(\text{-}\beta\Delta \mathrm{E}) < \mathrm{R}; \\ \mathrm{label}^{-1}(i), \text{otherwise}. \end{cases} \tag{7}$$

where $\mathrm{label}^{-1}(i)$ is the inversion value of label(*i*), $\boldsymbol{\beta}$ is a weight factor and *R* is a pseudo random number from uniform distribution. In a word, the label inversion will be accepted if it increases the correspondence between the superpixel's appearance similarity and classification type.

Given the notations of all superpixels, we initially train the superpixel-level classifier based on appearance features which consist of color and gradient statistic in each superpixel. The classifier is able to select out the superpixels belonging to hands with confidence values. Note that the motion cue may eventually become much less discriminative. Therefore, we apply the SLIC on color frame to get the superpixel segmentation in the subsequent online training. Because of benefiting from the relatively accurate boundaries produced by the SLIC, the segmentation in superpixel level is improved than the result of frame-level detection. However, the superpixel having low confidence value may partially contain hand region. It will cause misclassification if we take that kinds of superpixels as background and select negative pixel samples from them. Therefore, we proposed a sample selection strategy for pixel-level classifier training.

For pixel-level training, we select samples from the superpixels based on their classification confidence values. The negative samples are selected from the superpixel having confidence smaller than a threshold value T_U. The positive samples are selected from the candidate superpixels which have confidences greater than a threshold value T_L. The unstable superpixels having confidences between T_U and T_L are abandoned as unknown. Moreover, the higher the confidence of a superpixel belongs to the hand region, the more positive samples are extracted from it. Based on the property of superpixel generated by SLIC method, we suppose that pixels nearing to the center of the superpixel are more likely to be in the same class with the superpixel. Therefore, we divide the pixels of candidate superpixels into training and unknown groups based on the distance between the pixel and the superpixel's center. By combining the area A_{sp} and confidence W_{sp} of a superpixel, we define the distance threshold T_{sp} as Eq. (8). Then, the candidate superpixels are eroded based on the threshold T_{sp}. The pixels in the shrunk region are put into unknown group while the others are selected as positive training samples.

$$T_{sp} = A_{sp}{}^{*} W_{sp} \tag{8}$$

Following the previous pixel-level segmentation approach [1], we extract color features from RGB, HSV, and LAB color spaces and texture feature using HOG [31]. By using a pool of combination of features and random forest classifiers [32], we classify the unknown pixels and obtain fine-grained hand segmentations. After that, we also get a more precision description of the confidence of a superpixel belonging to the hand region. The confidence values of superpixels are updated with their portion of positive labeled pixels. Then, we re-train the superpixel-level classifier by using the superpixel having high confidence values. By doing this, we update the hand and background models on-the-fly which makes the method more robust to varying environment.

Note that the two-level classifiers select out the pixels that are most likely to be in the hands. The motion cue becomes salient and discriminative again when the interactive hand gradually moves out of the view field. Therefore, we still have to monitor the hand absence by aid of the egocentric saliency metric which is added a confidence term, as described in Eq. (9).

$$E_f^{\mathrm{OUT}} = \sum_{i=1}^{W} \sum_{j=1}^{H} \left(\frac{1}{1 + e^{\lambda * (h-j)}} - 0.5 \right) * M_f(i,j) + \sum_{t=f-n}^{f} \operatorname{sgn}(N_{t-1} - N_t) - \frac{1}{m} \sum_{k=1}^{m} W_{sp_k} \tag{9}$$

Where, the first term denotes the motion saliency, the second term observes the consequent motion decrement, the third term is the average superpixel confidence of the frame f, and m is the number of superpixels having confidence greater than 0.5.

4 Evaluation to update classifier

In evaluation stage, we use a bottom-up strategy. We evaluate bottom classifiers and feedback loss to the upper levels. The superpixel-level classifier is directly affected by precision of pixel-level classification since the confidence of superpixel is calculated based on pixel classification results. In the initialization step, we consider frames of a sequence equally to contribute to pixel-level classifier. Since background changes constantly, the appearance of hand varies a lot and becomes different from previous situation, such as hand enters into a shadow place. Therefore, the history frames contribute differently and we calculate weights W_t for n history frames of the training set $\{F_t\}$ to make their contributions more rational based on error of pixel-level classifier. The weight W_t consists of a local metric W_L^t and a global metric W_G^t.

$$W_t = 1 / \left(W_L^t + W_G^t \right) \tag{10}$$

Given a labeled training set $\{F_t\}$, we train a collection of classifiers $\{C_t\}$. By using the classifier C_t, we get the confidence value $W_{sp_k}^t$ of a superpixel SP_k belonging to the hand regions in current test frame f. The local metric W_L^t restricts that the result of classifier C_t has low variance with other classifiers of the set. Therefore, we calculate the loss of using training data from frame t based on the difference between classification results of test frame f produced by C_t and the average classifier $\overline{C}_{\{F_t\}}$.

$$W_L^t = \frac{1}{m} \sum_{k=1}^{m} \left(W_{sp_k}^t - \overline{C}_{\{F_t\}} \right) \tag{11}$$

$$\overline{C}_{\{F_t\}} = \frac{1}{n} \sum_{l \in \{F_t\}} W_{sp_k}^l \tag{12}$$

where m is the number of superpixels in current test frame f and n is the number of frames in the training set $\{F_t\}$.

From a global point of view, we estimate the loss of using training data from frame t based on the difference between the classification result of frame f produced by C_t and the classification result of frame *f-1* produced by the previous classifier $C_{\{Fp\}f-1}$ which is trained using data from $\{F_p\}$ under the constraint of weight W_p. Generally speaking, precise classification can segment hand region from background with clear boundary while smooth and flat inside the region. We calculate gradient map of the classification probability map and define three gradient-based constraints to evaluate the global loss. Firstly, the magnitude of the biggest contour in the gradient map should be large. Then, the gradient in the conjunction of two superpixels should be small. That is, the number of contours in the gradient map should be small. And last, the shapes of the biggest contours in current and previous gradient maps should be similar. Based on these three constraints, we calculate a global loss function having terms based on the average magnitude G_f of the biggest contour, the number N_f of contours, and the shape S_f of the biggest contour in the classification result of test frame f.

$$W_G^t = \left(G_f^t - G_{f-1}^{f-1}\right) + \left(N_f^t - N_{f-1}^{f-1}\right) + D\left(S_f^t, S_{f-1}^{f-1}\right) \tag{13}$$

where the right hand superscript denotes the classifier has been used, C_t or $C_{\{F_p\}}^{f-1}$. $D(\blacksquare,\blacksquare)$ is a function estimating the difference between two shapes.

By combining W_L^t and W_G^t, we evaluate the effectiveness of training samples from frame t not only in local superpixels but also in the global hand region. Based on the weight W_t, we optimize the pixel-level classification result which is used to update the superpixel-level classifiers. Note that the terms of the weight function will be normalized before combination.

5 Results and discussion

We evaluate our cascaded hand segmentation method on two types of egocentric data which correspond to different levels of human-computer interaction. The first type contains the both hands are exposed with little varying gesture and interacting with objects, such as holding a cup. The second type contains the hands performing gestures, such as virtual keyboard typing, without directly interacting with any object. We firstly compare our cascaded hand segmentation with the state-of-the-art methods and analyze the validity of our framework. Then, we illustrate that the egocentric human-computer interaction can benefit from our hand segmentation approach.

5.1 Evaluation on benchmark dataset

To compare with baseline methods, we first test our approach on the benchmark dataset CMU EDSH [1] which consists of egocentric videos containing diverse indoor and outdoor illumination and hand poses. The videos were collected by a subject wearing the head-mounted standard color camera and passing through scenes with varying illumination including the extreme cases of underexposed and overexposed at a resolution of 720p and a speed of 30 FPS. Besides the change of skin color, the hand pose also changes during the subject doing daily activities. The dataset contains 19,788 frames and 743 ground truth labels from three video clips, including EDSH1, EDSH2, and EDSHK. EDSH1 and EDSH2 involve data of bare hands with a few intentional gestures while EDSHK records hand interacting with objects in a kitchen. In order to match the scale of the ground truth, we downsample the resolution of the frame from 1280 × 720 to 640 × 480 pixels. We conduct quantitative and qualitative evaluation on the benchmark dataset to compare our detection performance with the prior arts.

In Table 1, we compare our method with the three state of the arts on F-score. Li and Kitani [1] predict hand pixel using color and gradient features based on Random Forest classifiers. Zhu et al. [2] extend the pixel-level method by introducing shape information of pixels based on structured forests. Baraldi et al. [5] utilize temporal and spatial coherence strategy to improve the hand segmentation of the pixel-level method. The state of the arts use video clip EDSH1 as the training data and test their approaches on the rest clips of EDSH2 and EDSHK. The corresponding F-scores are provided by their papers. Since our approach using online training strategy, we give out our F-scores on all the clips. As the F-scores shown in Table 1, our approach improves the detection precision in most experiments. We have implemented our algorithm and tested the non-optimized code on an Intel-based PC, with a i7-4500 U CPU that runs at 1.80 GHz. Most of time is spent on superpixel sample selection and online training. The time cost can be reduced by decreasing the number of samples used in all stages. In Table 2, we compare our method with of the three state of the arts on time.

Figures 6 and 7 show the visually comparison of test images overlapped by detection results provided by their papers and our method in the challenge cases of extreme lighting conditions and background color.

In Fig. 6, the test frames of EDSH2 were taken under extreme lighting conditions of overexposed, underexposed and high contrast shadows. Parts of the hand are blended into background by the strong or insufficient light while the color and texture of the other parts are faded inordinately. Li and Kitani [1] fail to give good prediction in these cases. In contrast, our approach has much higher detection precision. Our continuously online training strategy makes the classifiers robust to varying illumination even in the extreme conditions.

Figure 7 shows the case of background sharing similar color and texture with hand. Both methods of Li and

Table 1 Comparison with state of the art on *F*-score

Data	Li and Kitani's [1]	Zhu et al.'s [2]	Baraldi et al.'s [5]	Ours
EDSH1	Training	Training	Training	0.8667
EDSH2	0.835	0.8353	0.852	0.8995
EDSHK	0.840	0.9352	0.901	0.9130

Table 2 Comparison with state of the art on time

	Li and Kitani's [1]	Zhu et al.'s [2]	Baraldi et al.'s [5]	Ours
Time(ms)	2138	3277	1092	3497

Kitani [1] and Zhu et al. [2] fail to distinguish the hand from the textureless and skin-colored background. In contrast, our approach gives more correctly prediction in this case. By using online learning, our method gradually updates the hand and background models so that the classifiers are more robust to varying scene.

5.2 Evaluation on egocentric application

Fingertip position is one of the most practical information for egocentric vision-based human-computer interaction, such as the user inputs command via a virtual keyboard. Fingertip detection can directly benefit or suffer from the precision of the hand segmentation. Therefore, we use a simple fingertip detection method to further evaluate our hand segmentation method from the practical point of view.

As shown in Fig. 8, we evaluate the applicability of hand segmentation by an application of virtual keyboard interaction. The ready gesture of index finger up triggers the virtual keyboard to show up. Then, the egocentric view field is divided into girds each of which corresponds to a key. In the experiment, we divide the view field into 5 × 7 grids which provide relative comfortable interaction scale for the user. The duration of fingertip activates the key input and the corresponding position will light up. We extract tip position of the index finger from the hand segmentation result by convex hull analysis. The video was recorded by a subject wearing the head-mounted Logitech camera in the indoor scene at a resolution of 640 × 480 and a speed of 30 FPS. The test video totally contains 1439 frames consist of the whole interaction procedure including hand moving into the view field, ready gesture showing up, fingertip hovering and moving through keys, and hand moving out of the view field. Figure 8a–c illustrates the robustness of our hand segmentation-based fingertip detection. Figure 8d shows the failure case caused by the noise of the segmentation which could be removed by extra post-process.

Figure 9 shows the performance of our hand segmentation method in the virtual keyboard interaction application. The red and blue dots are the detected fingertip interaction frames. We can see that the detected fingertip position and the ground truth respectively in the keyboard position is stable and with little

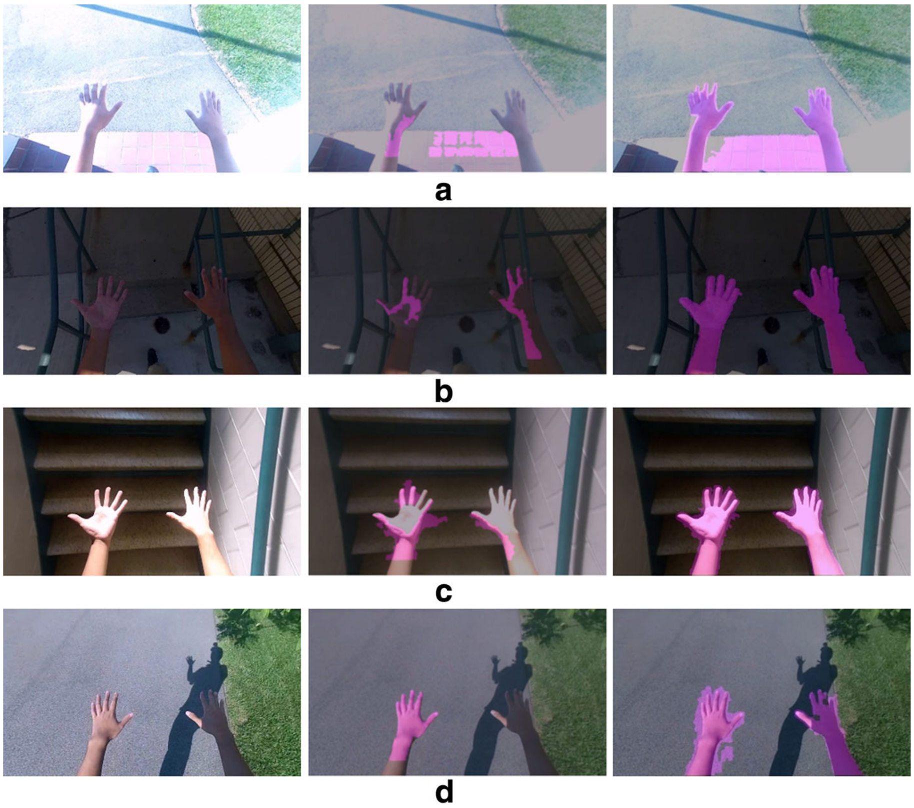

Fig. 6 a–**d** Challenge illumination cases comparison. From top to bottom are cases of overexposed, underexposed, overexposed, and high contrast shadows. The first column shows the original images. The middle column shows results of Li and Kitani [1] and the last column shows our results

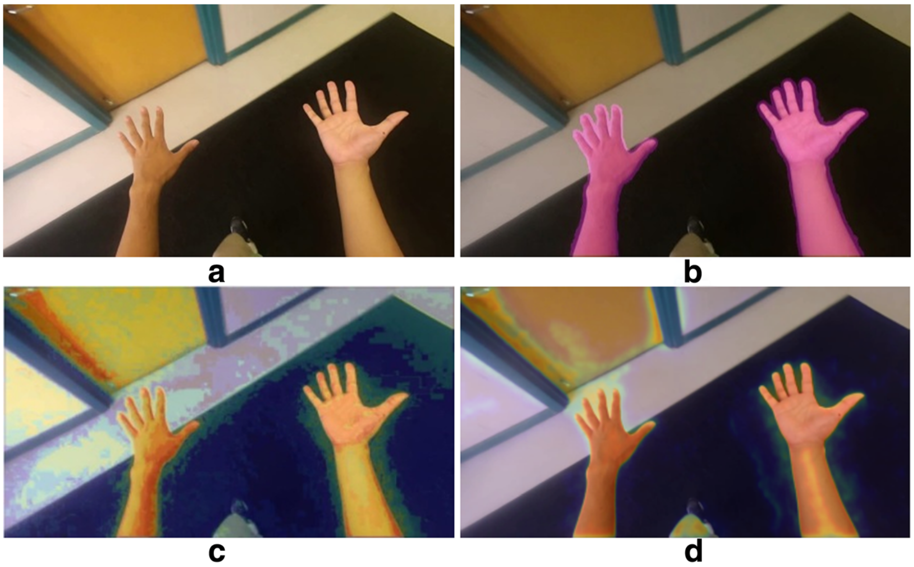

Fig. 7 **a** The original image has similar color in background and hand regions. From (**b**–**d**) are corresponding results of ours, Li and Kitani [1] and Zhu et al. [2]

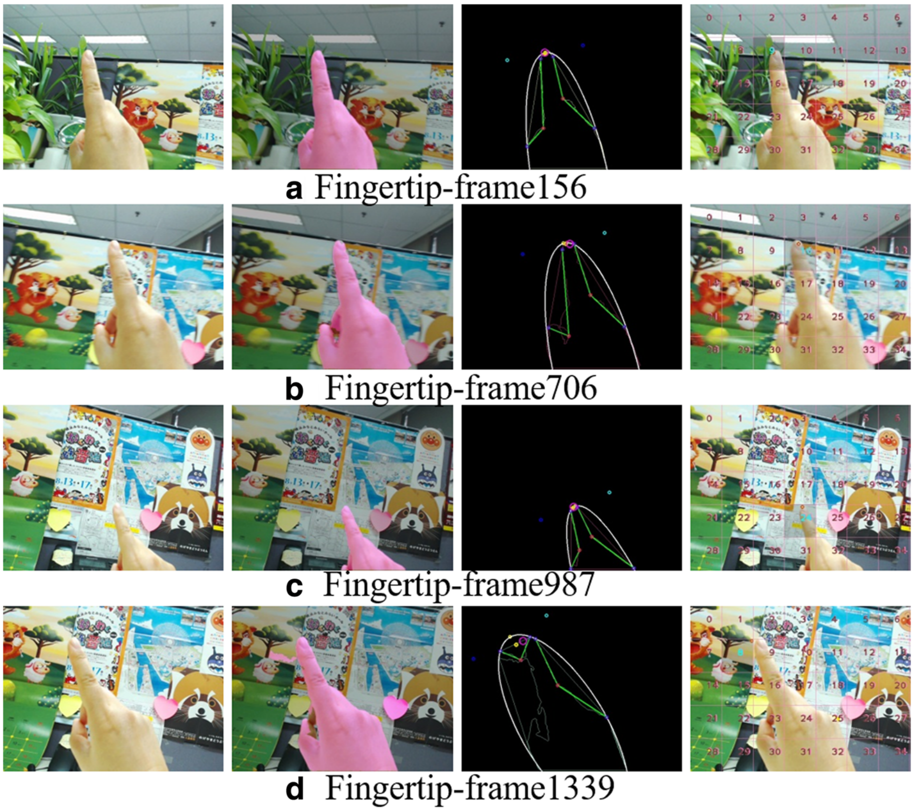

Fig. 8 Virtual keyboard interaction. **a**–**d** illustrate examples of fingertip detection for virtual keyboard interaction. Columns from left to right are original images, hand segmentation results overlapping on original images, fingertip detection by analyzing convex of hand segmentation result, and virtual keyboard interaction

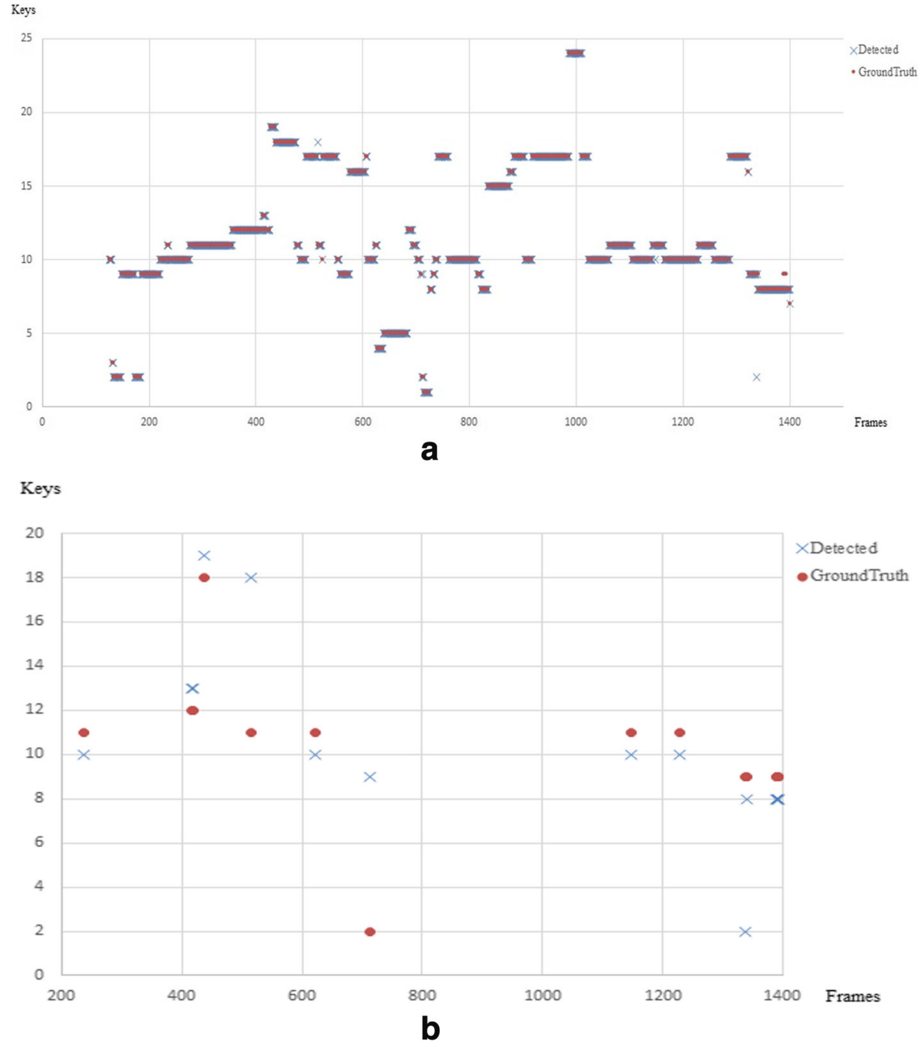

Fig. 9 Performance of hand segmentation in virtual keyboard interaction. **a** Fingertip detection results compared with ground truth. **b** Close-up of errors in **a**

jitter. And the fingertip detection accuracy rate is 0.9867 over the test video. Figure 9b shows the total 17 failure cases over the 1277 interactive frames. It proves that our hand segmentation approach is reliable and prone to be used in egocentric vision based human-computer interaction.

6 Conclusions

In this paper, we presented an unsupervised on-the-fly hand segmentation method which consists of top-down classification and bottom-up optimization. From the point of view of egocentric interaction loop, an unsupervised frame-level hand detector is proposed for the purpose of reducing the false positive caused by hand absence. We implement the frame-level detection by setting a non-interactive border based on an assumption that the hand is hardly to enter into the view field from the top side for egocentric interaction. Based on the frame-level detection result, the superpixel-level and pixel-level classifiers are trained on-the-fly sequentially aimed at improving reliability of hand segmentation. To get stable samples for superpixel-level training, we select the candidates based on steps of confidence score calculation and energy optimization. In order to be robust to vary environmental conditions, the classifiers are updated from the bottom up based on the proposed performance evaluation method. Experiments carried on public datasets validate the generality of the proposed approach. This paper shows the potential of unsupervised method for pixel-level hand segmentation in egocentric interaction. We believe that it can be transferred to the pixel-level object segmentation by combining with gaze analysis and contributing to activity recognition.

Acknowledgements
Not applicable

Funding
Not applicable

Availability of data and materials
The web links to the sources of the data (namely, images) used for our experiments and comparisons in this work have been provided in this article.

Authors' contributions
YZ generated this idea, implemented the core algorithm, designed all the experiments, addressed the resulting data, and drafted the manuscript. ZL and CQ are responsible for the final proofreading along with the technical support. All authors have read and approved the final manuscript.

Authors' information
Ying Zhao is currently pursuing the Ph.D. degree in Graduate School of System Informatics at Kobe University, Kobe, Japan. She also works for Ricoh Software Research Center (Beijing) Co., Ltd., Beijing, China. Zhiwei Luo is a professor and Changqin Quan is an associate professor in Graduate School of System Informatics at Kobe University, Kobe, Japan.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 27 December 2017 Accepted: 19 March 2018
Published online: 03 April 2018

References

1. C. Li and K. M. Kitani, "Pixel-Level Hand Detection in Ego-Centric Videos," 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, 2013, pp. 3570-3577.
2. X. Zhu, X. Jia, and K. Y. K. Wong, "Pixel-Level Hand Detection with Shape-Aware Structured Forests," Asian Conference on Computer Vision (ACCV), Springer International Publishing, 2014, pp. 64-78.
3. G. Serra, M. Camurri, L. Baraldi, M. Benedetti, R. Cucchiara, "Hand Segmentation for Gesture Recognition in EGO-Vision," ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices, 2013, Vol.24, pp.31-36.
4. C. Li and K. M. Kitani, "Model Recommendation with Virtual Probes for Egocentric Hand Detection," 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, 2013, pp. 2624-2631.
5. L Baraldi, F Paci, G Serra, L Benini, R Cucchiara, Gesture recognition using wearable vision sensors to enhance visitors' museum experiences. IEEE Sensors J. **15**(5), 2705–2714 (2015)
6. X. Ren and C. Gu, "Figure-Ground Segmentation Improves Handled Object Recognition in Egocentric Video," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, 2010, pp. 3137-3144.
7. P. Morerio, L. Marcenaro and C. S. Regazzoni, "Hand Detection in First Person Vision," Proceedings of the 16th International Conference on Information Fusion, Istanbul, 2013, pp. 1502-1507.
8. A. Betancourt, "A Sequential Classifier for Hand Detection in the Framework of Egocentric Vision," 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, 2014, pp. 600-605.
9. A. Betancourt, P. Morerio, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni, "A Dynamic Approach and a New Dataset for Hand-Detection in First Person Vision," in International Conference on Computer. Analysis of Images and Patterns, Malta, 2015.
10. J. Kumar, Q. Li, S. Kyal, E. A. Bernal and R. Bala, "On-The-Fly Hand Detection Training with Application in Egocentric Action Recognition," 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, 2015, pp. 18-27.
11. A Betancourt, P Morerio, CS Regazzoni, M Rauterberg, The evolution of first person vision methods: a survey. IEEE Trans. Circuits Syst. Video Technol. **25**(5), 744–760 (2015).
12. T. Ishihara, K. M. Kitani, W. C. Ma, H. Takagi and C. Asakawa, "Recognizing Hand-Object Interactions in Wearable Camera Videos," 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 1349-1353.
13. S. Bambach, S. Lee, D. J. Crandall and C. Yu, "Lending a Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1949-1957.
14. A. Fathi, X. Ren and J. M. Rehg, "Learning to Recognize Objects in Egocentric Activities," 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, 2011, pp. 3281-3288.
15. H. Pirsiavash and D. Ramanan, "Detecting Activities of Daily Living in First-Person Camera Views," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 2847-2854.
16. Z. Lu and K. Grauman, "Story-Driven Summarization for Egocentric Video," 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, 2013, pp. 2714-2721.
17. C Yan, H Xie, D Yang, et al., Supervised hash coding with deep neural network for environment perception of intelligent vehicles [J]. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–12 (2017).
18. C Yan, H Xie, S Liu, et al., Effective Uyghur language text detection in complex background images for traffic prompt identification [J]. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–10 (2017).
19. C Yan, Y Zhang, J Xu, et al., A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors [J]. IEEE Signal Process Lett. **21**(5), 573–576 (2014).
20. C Yan, Y Zhang, J Xu, et al., Efficient parallel framework for HEVC motion estimation on many-core processors [J]. IEEE Trans. Circuits Syst. Video Technol. **24**(12), 2077–2089 (2014).
21. A. Betancourt, P. Morerio, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni, "Left/Right Hand Segmentation in Egocentric Videos," Computer Vision and Image Under-Standing, 2016.
22. Betancourt, A., Morerio, P., Marcenaro, L., Barakova, E., Rauterberg, M., & Regazzoni, C. (2015). Towards a Unified Framework for Hand-Based Methods in First Person Vision. In Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on (pp. 1-6). IEEE.
23. X. Zhu, W. Liu, X. Jia and K. Y. K. Wong, "A two-Stage Detector for Hand Detection in Ego-Centric Videos," 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, 2016, pp. 1-8.
24. Y. Sheikh, O. Javed and T. Kanade, "Background Subtraction for Freely Moving Cameras," 2009 IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, 2009, pp. 1219-1225.
25. M. Narayana, A. Hanson and E. Learned-Miller, "Coherent Motion Segmentation in Moving Camera Videos Using Optical Flow Orientations," 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, 2013, pp. 1577-1584.
26. X. Liang, C. Zhang, and T. Matsuyama, "Inlier Estimation for Moving Camera Motion Segmentation," Asian Conference on Computer Vision (ACCV), Springer International Publishing, 2014, pp. 352-367.
27. R. Margolin, A. Tal and L. Zelnik-Manor, "What Makes a Patch Distinct?," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 1139-1146.
28. JS Pérez, TV-L1 optical flow estimation. Image Proces. Line **2**(4), 137–150 (2013).
29. R Achanta, A Shaji, K Smith, A Lucchi, P Fua, S Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012).
30. E Ising, Beitrag zur theorie des ferromagnetismus [J]. Z. Phys. **31**(1), 253–258 (1925).
31. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.
32. L Breiman, Random forests. Mach. Learn. **45**(1), 5–32 (2001).