On-the-Fly Hand Detection Training with Application in Egocentric Action Recognition

Jayant Kumar*, Qun Li*, Survi Kyal, Edgar A. Bernal, and Raja Bala
PARC, A Xerox Company
800 Phillips Road, Webster, NY 14580
{Jayant.Kumar, Qun.Li, Survi.Kyal, Edgar.Bernal, Raja.Bala}@parc.com

Abstract

We propose a novel approach to segment hand regions in egocentric video that requires no manual labeling of training samples. The user wearing a head-mounted camera is prompted to perform a simple gesture during an initial calibration step. A combination of color and motion analysis that exploits knowledge of the expected gesture is applied on the calibration video frames to automatically label hand pixels in an unsupervised fashion. The hand pixels identified in this manner are used to train a statistical-model-based hand detector. Superpixel region growing is used to perform segmentation refinement and improve robustness to noise. Experiments show that our hand detection technique based on the proposed on-the-fly training approach significantly outperforms state-of-the-art techniques with respect to accuracy and robustness on a variety of challenging videos. This is due primarily to the fact that training samples are personalized to a specific user and environmental conditions. We also demonstrate the utility of our hand detection technique to inform an adaptive video sampling strategy that improves both computational speed and accuracy of egocentric action recognition algorithms. Finally, we offer an egocentric video dataset of an insulin self-injection procedure with action labels and hand masks that can serve towards future research on both hand detection and egocentric action recognition.

1. Introduction

The increasing abundance of low-cost cameras and sensors has prompted an explosion of wearable products such as Google Glass, Apple Watch, and the like. Meaningful analysis and interpretation of data sensed from wearable devices have therefore garnered much recent attention in the research community. In particular, wearable cameras provide an intimate first-person viewpoint –referred to as egocentric vision– and enjoy the benefit of continuously recording, monitoring, and assisting the user in his/her activities on-the-go [1]. Research in egocentric vision can be categorized into three broad related classes: human action and activity recognition, object and scene understanding, and event summarization [2, 3, 4, 5, 6, 7]. Applications are numerous, and include daily living assistance for sick, impaired or elderly citizens, assistance in complex assembly and repair tasks, remote/virtual training and automated compliance monitoring in specialized (*e.g.*, medical) procedures, assistance in law enforcement and emergency response, and event logging in consumer and professional settings [8, 9].

1.1. Motivation

In many of the aforementioned applications, the presence and patterns of motion of the user's hands may provide critical cues towards determining the nature of his/her actions, intentions, and focus of attention [3, 10, 11, 12]. In this paper we address the problem of reliably detecting the user's hands in egocentric video. This is a challenging problem since hand appearance varies widely across users, and even for one user can be significantly affected by environmental conditions (*e.g.*, lighting), hand motion relative to the camera, and camera parameters such as focus, exposure, and white-balance. Traditional, including state-of-the-art, hand detection approaches rely upon the existence of a large training set of videos captured *a priori* with multiple users under a variety of environmental conditions, accompanied by pixel-level manual labeling of hand masks within the video frames. The shortcomings of this approach are twofold: for one, the substantial effort expended in manual labeling does not lend itself to a scalable solution that continuously learns and adapts to new conditions. Secondly, our experiments show that even the most recent and sophisticated hand detection techniques do not generalize well to test conditions that deviate from training scenarios.

1.2. Contributions

We present three contributions in this paper. The first and primary one is a novel on-the-fly method to train a hand detector requiring no manual labeling of training samples.

*Equal Contribution

The user is prompted to perform and record a simple hand gesture just prior to performing the required activity, as illustrated in Figure 1. Motion and color analysis of the gesture enables automatic and unsupervised extraction of pixels from different regions of the hand, which are then used to train a hand detector. This dynamic method of training that is tailored for a specific user's hand, capture device, and environmental conditions is shown to produce superior detection performance than the standard offline method of training a detector with pooled features across multiple subjects. At the same time, a simple gesture obviates the impractical requirement of manually labeling training samples in a live application.

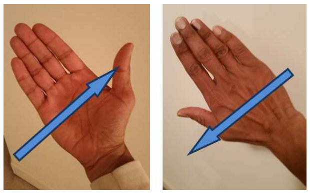


Figure 1. Illustration of calibration gesture performed just prior to the egocentric task (arrows indicate the direction of hand motion). Gesture analysis identifies hand pixels which are then used to train a hand detector on-the-fly.

As our second contribution, we propose an efficient video sampling scheme for egocentric activity recognition that is adapted based on the location of the detected hand regions. This idea is inspired by previous findings that indicate that hand regions provide important cues for user attention and activity on tasks involving substantial hand-eye coordination [3, 4, 10, 11]. The user's hands are located at a low frame rate (2 fps), and patches are sampled more finely in the immediate vicinity of the located user's hands, and more coarsely elsewhere. This approach enjoys the obvious advantage of reduced computational cost in feature extraction. More significantly, with the proper tuning of sampling parameters, the method also results in improved action recognition accuracy due to the fact that the feature descriptors computed from salient portions of the video are more discriminative across different actions, while being consistent across different users performing the same action. Experiments on the GTEA gaze dataset [4] and a new insulin self-injection (ISI) dataset show that computation times are reduced by approximately 66%, while mean average precision improves by 3-4%.

The third contribution is a dataset of 25 egocentric video clips of an insulin self-injection procedure performed by subjects under realistic environmental conditions. Each clip is segmented into seven actions, constituting a total of 175 action clips with labels and hand masks. The dataset supports a real-world application, namely medical procedure monitoring, and is intended to enable explorations in both hand detection and action recognition.

2. Related work

2.1. Hand detection

Many pixel-based hand segmentation methods have been proposed in the literature [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. The feature descriptor used to perform the detection is a critical factor. There have been efforts to model and detect hand skin regions based on low-dimensional color representations such as RGB, LAB, YUV, YCrCb, CMYK, *etc.* [19, 20, 21, 22] These often yield low accuracy due to the significant variation in hand color across different environmental and illumination conditions. Other techniques use motion features, and exploit temporal consistencies in hand motion patterns to aid segmentation and tracking tasks [15, 17, 24]. These techniques may be computationally prohibitive for real-time applications, a problem that is exacerbated by the limited computational resources of mobile and wearable devices. Researchers have also explored combining color with additional features such as texture [14, 16, 18, 23]). In [14], a generic pixel-level hand detector based on a combination of color, texture, and gradient histogram features is trained using over 600 manually labeled hand images (over 200 million labeled pixels) acquired under various illumination conditions and backgrounds, and has shown to outperform several baseline approaches. This method shows improvements in detection accuracy by including additional cues to color; however, once again, this benefit is gained at the expense of computational complexity.

In our approach, we seek a feature that is computationally efficient to compute, discriminative enough to separate hand from background regions, and robust to environmental variations, notably illumination. In recent work, color attributes exhibiting these properties have been employed in applications such as object recognition [25], object detection [26], action recognition [27], and visual tracking [28]. In particular, a linguistics study presented in [29] concluded that the English language contains eleven basic color names: black, blue, brown, grey, green, orange, pink, purple, red, white and yellow. In [30], a mapping from RGB values to this 11-dimensional named color representation was learned automatically using images retrieved through Google image search. The proposed high-dimensional color space has been shown to be suitable for real-world applications where increased robustness to illumination changes and discriminability are desired. In this work, we perform hand segmentation in this space.

The majority of hand detection approaches rely on the use of supervised classifiers which need to be trained with an extensive set of labeled data, particularly if the detection is performed at the pixel level [14, 16, 17, 19, 21]. Furthermore training is performed offline by pooling hand pixels from a variety of subjects and environmental conditions. In contrast, in this paper we propose a pixel-level hand detection technique that is trained on-the-fly for a given individual and environment, and whereby training samples are gathered automatically without need for manual labeling. Skin pixels are automatically segmented from a simple calibration gesture performed by the user, and a classifier is trained on the high-dimensional color representation of the segmented pixels. To our knowledge, this is the first attempt to use gestures to perform *in-situ* training of a hand detector. A related effort by Li and Luo [31] proposes using a Viola-Jones face detector to extract skin regions from the face, and trains a hand detector from these skin regions in third-person videos. At high level, our idea is similar, but our mechanism to extract skin color is different (*i.e.*, hand gesture). Clearly, face detection would not work for the egocentric setting. Also, our approach works when the user is wearing special apparel (*e.g.*, gloves) where Li's approach would fail.

A common approach to enhance segmentation is to use superpixel techniques to capture local redundancy and group pixels into perceptually meaningful regions. There are many approaches to generate superpixels, each with its own pros and cons [32, 33, 34, 35, 36]. In our work, we use a modification of a state-of-the-art algorithm termed simple linear iterative clustering (SLIC) [37]. This approach adapts k-means clustering to efficiently generate superpixels and has been shown to outperform previous methods in terms of adherence to boundary and computational efficiency. We modify SLIC in several ways to suit our problem, as will be described in Sec. 3.

2.2. Adaptive sampling for egocentric action recognition

Our second contribution is to apply hand detection to enhance the speed and accuracy of egocentric action recognition via adaptive video sampling. There are several threads of literature to cite. First, action recognition from first-person videos is itself a relatively recent but rapidly expanding area of exploration [2, 3, 7, 38, 39, 40, 41, 42, 43, 44]. As far as the authors know, all existing techniques employ uniform spatiotemporal sampling of the egocentric video. Researchers have explored adaptive sampling for the analysis of third-person images and video in order to reduce computation and/or improve accuracy for tasks such as object and action recognition [45, 46, 47, 48, 49]. In [45], saliency models are used as filters to the sampling process in a third-person action recognition pipeline so as to improve recognition accuracy. A variety of sampling schemes are proposed, including biologically inspired masks, an analytical mask based on a structure tensor, and an empirical mask based on eye-tracking data that reports the best performance in recognition accuracy. We employ a similar idea but tailored to the egocentric setting. We draw inspiration from previous studies that highlight the importance of hands as salient cues towards action and activity recognition [4, 7, 10, 38, 39, 40, 41, 42, 43, 44], and propose to directly modulate the density of the video sampling based on the detected hand regions. As in [45], we compare our approach with several masking schemes and feature extraction methods.

Also closely related to this paper is the work of [10] which presents a method to predict human gaze based on hand location, head motion, and a prior gaze model built from eye-tracking data. The authors demonstrate that action recognition accuracy can be improved by firing an action classifier only in the vicinity of the predicted gaze. While our proposal is similar at a high level, it is different in the following ways. First, our saliency model is simpler to compute in that it requires only hand region detection, and no head motion analysis or eye-tracking priors; and yet, as the results will show, only marginally compromises accuracy. This makes our approach more amenable to rapid training and recognition of a wide variety of multi-action procedures. Second, we apply saliency-based sampling in *both* the training and inference stages, so that our visual vocabulary and pooled feature descriptors are tuned specifically to descriptive regions of action in the video. Finally we offer an analysis of how the sampling budget affects the tradeoff between speed and accuracy.

3. On-the-fly training for hand detection

Just before performing a task wearing the egocentric vision system, the user trains the hand detector for his/her hands in the same environment where the task is to be performed. Training of the hand detector comprises three steps: i) prompt the user to perform a predetermined hand-gesture such as a wave or rolling motion; ii) capture egocentric video of the hand gesture with a head-worn camera; iii) use motion segmentation plus region growing to automatically extract hand pixels. Any hand detection algorithm can be trained with this data; in our work we train a Gaussian Mixture Model (GMM)-based hand detector. In the collected dataset, a waving gesture is used that exposes the front and back parts of the hand, as shown in Fig. 1. The user performs the calibration gesture shortly after initializing the hand detector application, which will be looking for salient hand motions based on thresholding the magnitude of the motion vectors.

Figure 2 illustrates the process of automatic hand pixel labeling on one frame of the calibration gesture. In detail, Fig. 2(a) shows a sample frame from the gesture clip that exposes the back of the user's hand. Each training frame is mapped to the 11-dimensional color name space using the mapping derived in [30]. Figure 2(b) shows a pseudo-colored visualization of the mapping of the frame in Fig. 2(a) to the color name space. We observed that in the collected dataset, hand colors in the RGB space are mapped most frequently into one of three color names in the color name space (brown, green and red); we believe this is due to the various illumination conditions and skin colors present in the dataset. Figure 2(c) shows a mask where the foreground regions (in white) comprise pixels corresponding to one of the three hand color names. Next, motion analysis is performed based on prior knowledge of the gesture. Horn-Schunck optical flow [50] produces a motion vector field as shown in Fig. 2(d). Optical flow computation is followed by pruning of the resulting motion vector field based on motion vector magnitude. This step is effective in distinguishing salient motion patterns of the hand from apparent background motion caused by the user and camera movements. The pruned motion mask is shown in Fig. 2(e). Figure 2(f) shows pixels for which both color and motion are favorably indicative of hand presence, and computed as the intersection of the color mask 2(c) and the pruned motion mask 2(e). A seed for region growing is placed at the center of mass of the largest blob (highlighted in orange) of this intersection map (red cross in Fig. 2(g)). Figures 2(h) and 2(i) depict two visualizations of the automatically labeled hand pixels that result from the region growing process, the former as an overlay on the RGB image and the latter as a binary mask. In the final step, the labeled hand pixels, represented by their corresponding 11-dimensional color coordinates, are used to build a GMM-based detector. Once the detector is trained, it is used to perform hand pixel detection on subsequent video frames, which are usually captured while a task is being performed. As illustrated in the Sec. 4, we propose to use the location of the detected hand pixels as a cue to perform action recognition.

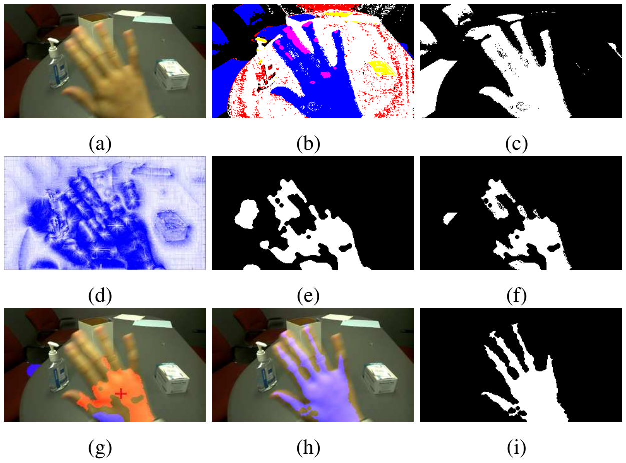


Figure 2. Illustration of the process for automatic labeling of hand pixels in the gesture-based training stage. See text for explanation of each step.

As an optional step, we investigate using superpixel segmentation to enhance the accuracy of both the hand labeling process in training and the hand detection process in testing. To further improve the efficacy of SLIC we implement a few modifications. First, instead of working in the CIELAB color space, as proposed by the authors of SLIC, the images are converted to the rg chromaticity space which reduces the dimensionality of the data, and thus aids computational efficiency, while still maintaining a degree of photometric invariance. Next, for added efficiency, SLIC is only applied to a neighborhood in the vicinity of the initially detected hand region, rather than to the whole image. The number of superpixels, which is a pre-determined input parameter in traditional SLIC implementations, is determined dynamically based on the size of the initial hand region. We refer to the modified SLIC as sped-up SLIC (sSLIC).

Figure 3 compares the performance of SLIC when applied in the LAB space (Fig. 3(a)) and the rg space (Fig. 3(b)). The images are 180×320 pixels in size, and the number of output superpixels was set to be 352. It took 33% longer for SLIC to converge in the LAB space relative to the convergence time in the rg space. From these timing figures and the segmentation results from Figures 3(a) and 3(b), it can be seen that SLIC is more efficient when it operates on a lower-dimensional color space such as rg, while also adhering to image boundaries satisfactorily.

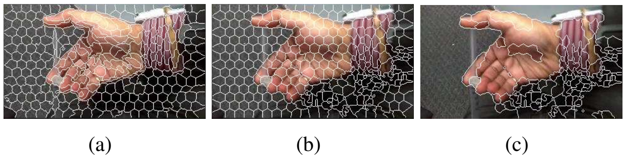


Figure 3. Examples of SLIC applied to an image in (a) LAB space, and (b) rg space, and (c) grouped superpixels from (b) using DBSCAN.

Lastly, as illustrated in Fig. 3(c), the superpixels are grouped using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [51], one of the most commonly cited clustering algorithms in the literature. The original hand mask is then used to decide which groups of superpixels to keep as part of the refined mask. To this end, the following criterion is adopted:

$$L(G_i) = \begin{cases} 1, & \text{if } |G_i \cap S| \geq \alpha |G_i| \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where S denotes the set of pixels in the original mask, G_i denotes the set of pixels associated with the i-th group of superpixels, $L(G_i)$ denotes the label of pixels belonging to G_i, α is the threshold which was set to 0.1 in our experiments, and $|\cdot|$ denotes the cardinality of a set. The pixels with label 1 then form the foreground region in the refined mask. Figure 4 gives examples of enhanced labeling and detection using the proposed method.

4. Application of hand segmentation to adaptive sampling for action recognition

We explore the use of hand segmentation to improve the computational load associated with the performance of an egocentric action recognition pipeline. The goal is to select a small subset from the dense set of spatiotemporal video samples for feature extraction based on the determined hand location. We first compute the centroid pixel location **C**

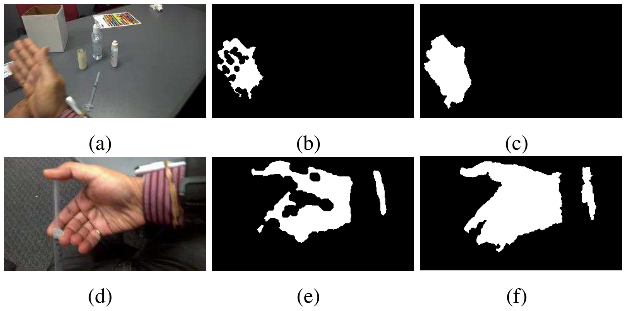


Figure 4. (a) Original training frame; (b) original, automatically labeled hand mask; (c) refined labeled hand mask; (d) original test frame; (e) original detected hand mask; and (f) refined hand mask.

of the hand mask obtained from the previously described hand detection module. Descriptors belonging to a region around $\mathbf{C}$ are considered for Bag-of-Words (BoW)-based feature computation. Frames that do not contain hand regions are not further processed in the pipeline. Figure 5 shows a high-level block diagram illustrating the pipeline for the proposed sped-up SLIC method.

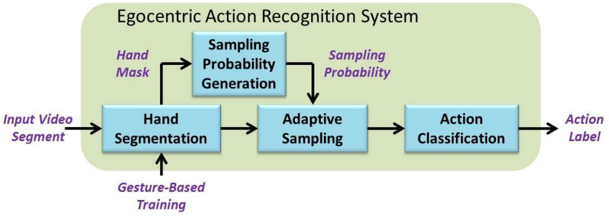


Figure 5. Block diagram of the proposed adaptive sampling pipeline for action recognition.

We propose a sampling scheme we term Hand-based Adaptive Sampling (HAS), and compare its performance in the context of activity recognition with other sampling schemes. A list of the considered sampling schemes along with a short description of the methods (including HAS) follows.

Hand-based adaptive sampling (HAS) — We define a mask $P(x, y)$ centered around $\mathbf{C}$ that defines a probability of selecting a feature at location (x, y). $P(x, y)$ may take on many functional forms; in our experiments we select a circularly symmetric 2D-Gaussian function with $\mathbf{C}$ as the mean. At each location of the dense sampling grid, $P(x, y)$ is compared with a random number r uniformly distributed in $[0, 1]$. For grid locations where P is greater than r, the sample is selected for feature extraction; otherwise it is eliminated. The optimal covariance matrix for P is estimated empirically using cross validation. With this scheme, sampling density decreases with increasing distance from $\mathbf{C}$. Figure 6 visually illustrates the effect of HAS when used in conjunction with the dense trajectories algorithm [52].

Image-center-based soft sampling (ICS) — We evaluate as a baseline a sampling scheme that concentrates a higher density of samples near the image center. The probability mask $P(x, y)$ is a Gaussian function (as above) that is fixed around the image center.

Random sampling (RS) — As a second baseline, we evaluate a sampling scheme which randomly selects locations across the video frame for feature extraction.

Gaze-based adaptive sampling (GAS) — For one of the datasets containing gaze annotations obtained from an eye-tracker, we evaluate an adaptive sampling scheme centered around the subject's gaze. A comparison of GAS and HAS provides insights on the efficacy of hand location as an approximate practical indicator of gaze and attention.

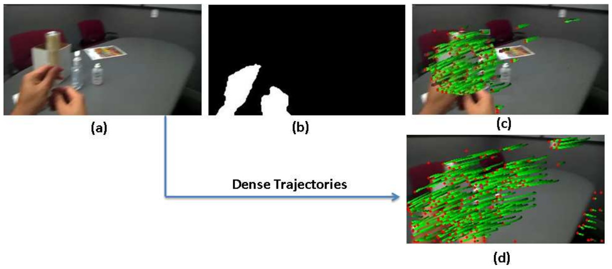


Figure 6. Illustration of proposed adaptive sampling scheme: (a) video frame showing an action from ISI dataset; (b) hand mask yielded by proposed method; (c) visualization of adaptively sampled dense trajectories; (d) visualization of the full set of dense trajectories. Observe in (d) the many spurious trajectories due to user head motion.

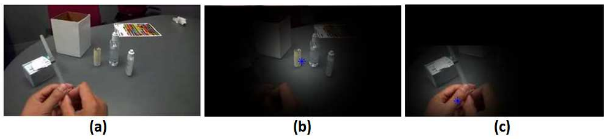


Figure 7. Illustration of sampling approaches (a) original video frame from ISI dataset (180×320 pixels); (b) ICS; and (c) HAS with $\sigma = 36$.

Figure 7 illustrates some of the sampling schemes. Each of the aforementioned sampling strategies is employed as a preprocessing filter in a standard action recognition pipeline [45, 53, 54]. The Dense Trajectory feature proposed in [52] is employed in our experiments as the state-of-the-art choice for action recognition. After feature extraction, a standard Bag-of-Words feature descriptor is computed followed by action classification using χ^2 SVM [54].

5. Experiments

5.1. Datasets and evaluation protocols

We have generated three new datasets. The first captures users performing a medical procedure –namely insulin self-injection– using a Google Glass^{TM} device. This dataset, which we denote ISI, was motivated by feedback from medical professionals that monitoring quality and correctness of medical procedures is an important problem and a valuable opportunity for wearables. Figure 7(a) shows a frame from a video clip in the dataset. Each subject was asked to perform the following seven steps required for self-insulin injection: (1) Hand Sanitization (2) Insulin Rolling (3) Pull air into syringe (4) Withdraw insulin (5) Clean injection site (6) Inject insulin (7) Dispose needle

A total of 8 subjects (4 female, 4 male) with different skin colors and ages performed this activity. Subjects were instructed only on the sequence of steps to be taken and were not coached on how to perform a given step. Prior experience varied widely. Three locations with different lighting and background conditions were used, and objects and their arrangement, as well as sitting geometry were allowed to vary freely. As evidence of inter-subject variability, durations of the video segments for a given action varied by a factor of 2 or more (*e.g.*, 5s vs. 10s, or 13s vs. 29s); furthermore, optical flow motion analysis reveals that mean motion magnitude varies on average by a factor of 4 and up to a factor of 6 across subjects for a given action. The dataset will be made publicly available by the authors, along with the corresponding action labels and hand masks for each action clip.

The second dataset called Dynamic Indoor (DI) was collected in an indoor environment with hand gestures being performed against a challenging background comprising people in motion. The third dataset named Dynamic Outdoor (DO) was collected in a moving vehicle with the outdoor street scene serving as a moving background. These datasets, with confounding human motion, extreme lighting variations, and low-contrast conditions, are intended to stress the robustness of hand detection in challenging environments. Figure 8 shows example frames from these datasets. We use all three datasets to evaluate the performance of the proposed hand detection method.

We also evaluate our adaptive sampling methods on the publicly available Georgia Tech Egocentric Activity (GTEA) gaze dataset [4]. This dataset comprises 17 activity sequences performed by fourteen subjects, with gaze tracked by Tobii eye-tracking glasses. The task was to make a recipe of the subject's choice in the kitchen. The beginning and ending time of the 25 action classes were annotated. We use both the GTEA and the ISI datasets to test action recognition performance. For the GTEA dataset, we use the same split of training (13 sequences) and testing (4 sequences) data as employed in [4]. For each of the seven actions in the ISI dataset, we randomly divide 25 video sequences into training and test sets, and report the mean performance across 100 iterations. At each iteration, 91 video samples are used for training and 84 videos for testing. The stride between samples in the dense sampling scheme was set to 10 pixels.

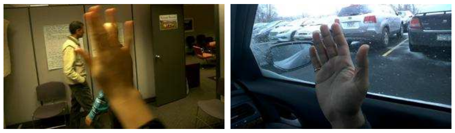

(a) (b)

Figure 8. Example frames from the (a) Dynamic Indoor (DI) and the (b) Dynamic Outdoor (DO) datasets.

5.2. Hand segmentation

The camera built into the Google Glass device was set to capture RGB video at a frame rate of 30 fps. The acquired videos were resized to a resolution of 180×320 pixels. We compare the performance of the proposed method with the simplified implementation* of the method of [14] made available by the authors. In this version, color histograms were used as features; at training, one global illumination model was extracted from each of the 442 training images. At the test stage, the 10 training models that best approximate the illumination conditions of the test image were averaged. Note that the detector in [14] is generic and was trained using over 442 manually labeled hand images (around 200 million manually labeled hand pixels) taken under various illumination conditions and different backgrounds. In contrast, the proposed hand detector is personalized for a specific user in a specific environment, and was trained using at most 10 frames per user, within which the hand pixels were labeled automatically via the use of the gesture-based training procedure described in Sec. 3. We set the number of components in GMM to 10 in the experiments. This number can be typically chosen based on the estimated diversity of hand color in the environment: the higher, the better the performance, at a cost of the computational resource.

We first report on the efficacy of the automatic gesture-based hand pixel labeling stage. Pixel-wise ground truth was established from the same set of gesture video frames used in the automatic training stage of the hand detector. The average pixel-level precision and recall achieved by the automatic hand-labeling stage were 0.866 and 0.267 respectively. Note that since the statistical model is built from the automatically labeled samples it is desirable to bias the process to minimize false positives, and these numbers reflect that bias.

Next, we evaluated the performance of the proposed hand detection method in comparison to standard approaches. To this end, we randomly selected 182 test frames from 12 videos in the ISI dataset. The test frames were extracted from videos acquired under a wide variety of illumination conditions ranging from outdoor sky lighting to dim indoor lighting. For ground-truth purposes, the test frames were manually labeled by creating binary masks indicating the location of hand pixels within each of the frames. Three different hand detection algorithms were evaluated as reported in Table 1: the proposed algorithm (row 1), the pixel-level hand detection method from [14] (row 2), and the detection stage of the proposed algorithm trained with manually labeled ground truth images (row 3). Mean and standard deviation of precision and recall across all testing images are reported.

*http://www.cs.cmu.edu/~kkitani/perpix/code_perpix/ver01/

Method	Precision	Recall
Proposed	0.947 +/- 0.053	0.613 +/- 0.170
[14]	0.829 +/- 0.271	0.472 +/- 0.333
Manual	0.902 +/- 0.106	0.547 +/- 0.231

Table 1. Hand detection performance comparison between tested algorithms.

It can be seen that the proposed method outperforms the method from [14], likely due to the fact that it is customized to the particular subject and illumination conditions. As explained above, the automatic hand pixel labeling algorithm used to train the model favored false negatives. A comparison of rows 1 and 3 in Table 1 indicates that this bias results in improved pixel-level hand detection performance at the test stage. We hypothesize this is due to the fact that boundary data in the manually labeled training set may be more significantly affected by compression artifacts and motion blur.

To evaluate the effect of superpixel-based enhancement, we tested three additional scenarios. Case 1 (sSLIC-te): sSLIC used only in testing as a post-processing step after the original detection. Case 2 (sSLIC-tr): sSLIC used only in training to improve the original labeling. Case 3 (sSLIC-both): sSLIC used in both training and testing. We tested the performance of the three variants of the algorithm on the ISI dataset. Table 2 shows the results.

Method	Precision	Recall
Proposed/sSLIC-te	0.901 +/- 0.074	0.756 +/- 0.154
Proposed/sSLIC-tr	0.905 +/- 0.095	0.678 +/- 0.214
Proposed/sSLIC-both	0.852 +/- 0.113	0.783 +/- 0.194

Table 2. Effect of sSLIC-based enhancement on hand detection.

Compared to row 1 in Table 1, sSLIC-based enhancement considerably increased recall, while only slightly decreasing precision. The decision as to whether to adopt such enhancement can be made based on the specific application. The computational per-frame overhead for the original SLIC algorithm was 2.77s (mean) +/- 0.08s (σ) and for the proposed sSLIC, 0.28s (mean) +/- 0.03s (σ), which indicates that sSLIC is about 10 times faster than SLIC. To set these figures in context, the per-frame processing time for the basic pixel-wise hand detection was measured as 0.15s (mean) +/- 0.003s (σ). Execution time was measured in seconds on a Windows 7 machine with 16GBytes of RAM and an Intel i7 2.80GHz processor. The implementation was done in Matlab R2013b.

A similar set of experiments was conducted on the DI and the DO datasets. Table 3 and 4 contain the results. It can be seen that even in the challenging environment where hand color appears similar to colors of both the static and dynamic backgrounds, the proposed method still exhibits robust performance. Naturally, almost all purely color-based hand detector would fail in the extreme case when foreground and background colors are the same; in this case, other features such as texture could be explored.

Method	Precision	Recall
[14]	0.30 +/- 0.30	0.33 +/- 0.36
Proposed	0.91 +/- 0.05	0.72 +/- 0.11
Proposed/sSLIC-te	0.86 +/- 0.05	0.88 +/- 0.05
Proposed/sSLIC-tr	0.89 +/- 0.05	0.87 +/- 0.05
Proposed/sSLIC-both	0.84 +/- 0.05	0.94 +/- 0.05

Table 3. Effect of sSLIC-based enhancement on hand detection performance on the DI dataset.

Method	Precision	Recall
[14]	failed	failed
Proposed	0.91 +/- 0.06	0.62 +/- 0.20
Proposed-/sSLIC-te	0.89 +/- 0.10	0.64 +/- 0.23
Proposed-/sSLIC-tr	0.82 +/- 0.13	0.78 +/- 0.18
Proposed-/sSLIC-both	0.78 +/- 0.17	0.84 +/- 0.20

Table 4. Effect of sSLIC-based enhancement on hand detection performance on the DO dataset.

As before, it can be seen that use of sSLIC-based enhancement increases recall and decreases precision. Note that in the particularly challenging outdoor dataset, the method from [14] completely failed to return any detections while the proposed method still performed reasonably well. We emphasize that the success of our on-the-fly approach hinges on the fact that training takes places in a given environment just before the user task is performed, and hence is not burdened (as traditional approaches are) with having to account for vast changes in ambient conditions. In this sense, it is difficult to make a completely fair comparison between our approach and that of [14] which was pre-trained across multiple subjects. Since our training approach can be employed with any hand detection algorithm, we expect competitive results when combining our training method with the detection algorithm of [14].

5.3. Action recognition

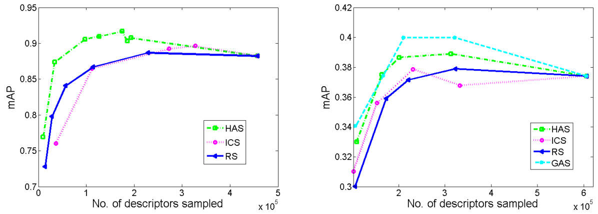
We now report the action recognition performance of the proposed adaptive sampling pipeline on videos taken from the ISI and GTEA datasets. Figure 9(a) is a plot of the mean average precision (mAP) of recognition as a function of sampling budget for each of the sampling methods used on the ISI dataset. Sampling budget was varied by adjusting relevant parameters for each sampling scheme. The point on the extreme right of each plot corresponds to the dense sampling (DS) scheme where no subsampling is performed. We observe that HAS achieves the best mAP of 0.92 using roughly one third of the total number of descriptors. Neither RS nor ICS achieve competitive accuracies.

Figure 9(b) shows a similar mAP plot for the GTEA gaze dataset. The authors of this dataset have provided gaze locations for each frame in the videos. Previous work has shown that human gaze provides an important clue in action recognition. We thus also evaluate the same action recognition pipeline with gaze-centered sampling where the same soft Gaussian probability mask used for HAS is centered instead at the gaze location in each frame. We denote this strategy GAS, and observe that it outperforms the other sampling methods. Note that while GAS serves as an upper bound in performance, it is difficult to execute in practice due to the need for eye-tracking instrumentation. We believe HAS serves as a practical alternative, in particular because it outperforms the remaining sampling schemes.

In order to assess the efficacy of adaptive sampling across different choices of features, we evaluated action recognition performance using SIFT3D [55] and the two-layer stacked convolutional independent subspace analysis (SC-ISA) network from [53]. Table 5 contains the action recognition performance across different spatio-temporal features achieved by the traditional dense sampling (DS) approach, the proposed HAS, as well as the two baseline approaches, ICS and RS, on the ISI dataset. Note that while the value of the σ parameter in the ICS and HAS schemes affects their performance, we only report the accuracy for the value of σ that yields the best performance. For RS, we randomly sampled 25% of the total descriptors. For SIFT3D features, HAS achieves a mAP of 0.89, while the DS scheme obtains a mAP of 0.87. This performance gain is achieved with only 28% of the total descriptors. For SC-ISA features, network weights were learned using a subset of videos from each dataset. HAS achieves an improvement of 0.04 in mAP over DS. For a comparable sampling budget, RS is ineffective across all features. In general, it can be seen that adaptive sampling is beneficial independently of the choice of features.

	DS	HAS	ICS	RS
DT	0.88	**0.92**	0.89	0.86
SIFT3D	0.87	**0.89**	0.87	0.85
SC-ISA	0.81	**0.85**	0.83	0.80

Table 5. Performance comparison between DS and HAS across different types of features on the ISI dataset.



(a) (b)

Figure 9. Mean average precision (mAP) for action classification on (a) ISI dataset and (b) GTEA gaze dataset using DT features

	HAS	ICS	RS	DS [52]
Time	0.0483	0.0394	0.0374	0.1217

Table 6. Execution times in seconds (per frame) for different sampling schemes.

In order to quantify the computational gain brought about by the proposed sampling methods, we timed the execution of the standard dense trajectory approach from [52] and the proposed approach on a 30 fps, 8 second video. In the timing experiment, the number of descriptors for the subsampling schemes (i.e., HAS, ICS, and RS) was fixed to 18103. Table 6 contains the results.

5.4. Conclusions

We have proposed a novel on-the-fly method to train a personalized pixel-level hand detector based on analysis of a simple known user gesture. This form of *in-situ* training can effectively address the challenging variations brought about by hand appearance differences across users and environmental conditions such as lighting, shadows and motion. Results show that bringing a “human in the loop” effectively simplifies and automates the training process by benefiting from the specific context of egocentric vision. The fact that our customized detector significantly outperforms generic, state-of-the-art techniques indicates that it is difficult to capture the full range of variability of factors present in egocentric scenarios with currently available statistical models trained offline on pooled data. We have also investigated the use of superpixel region growing to perform segmentation refinement and improve robustness to noise in both the training and testing phases. A significant benefit of the proposed approach is that the painstaking process of manually labeling large amounts of pixels in training images is eliminated. In order to economize on mobile computational processing, we train only on color features; incorporation of additional attributes, *e.g.*, texture, will surely enhance detection performance.

We have also shown that adaptive sampling schemes for egocentric action recognition that guide feature extraction towards hand regions in the video can improve both computational performance and recognition accuracy. As the number of samples is reduced from the dense to adaptive scheme with increasing selectivity, recognition performance initially improves as features from irrelevant regions are filtered out, and then eventually deteriorates due to paucity of features from even the salient regions. There is an optimal operating point at which recognition accuracy outperforms that of dense sampling by about 3-4%, while computation times are reduced by approximately 66%. If computational cost is the prime consideration, HAS permits a reduction in the sampling budget by a factor of about 10 while maintaining an equivalent level of accuracy. This trend holds true across a variety of feature descriptors, and marks significant progress towards making real-time action recognition practical. We have also shown that HAS incurs only a modest compromise in performance compared to gaze-based sampling, while avoiding the use of costly eye-trackers. Finally the ISI dataset is made publicly available to serve further research in egocentric action recognition and hand detection.

References

[1] T. Kanade and M. Hebert, “First-person vision,” *Proceedings of the IEEE*, vol. 100, pp. 2442–2453, Aug 2012. 1

[2] M. Ryoo and L. Matthies, “First-person activity recognition: What are they doing to me?,” in *CVPR 2013*, 2013. 1, 3

[3] A. Fathi, A. Farhadi, and J. Rehg, “Understanding egocentric activities,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 407–414, Nov 2011. 1, 2, 3

[4] A. Fathi, Y. Li, and J. M. Rehg, “Learning to recognize daily actions using gaze,” in *Proceedings of the 12th European Conference on Computer Vision - Volume Part I*, ECCV’12, pp. 314–327, 2012. 1, 2, 3, 6

[5] B. Xiong and K. Grauman, “Detecting snap points in egocentric video with a web photo prior,” in *Proceedings of the 14th European Conference on Computer Vision*, vol. 8693 of *ECCV’14*, pp. 282–298, 2014. 1

[6] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in *CVPR 2013*, pp. 2714–2721, 2013. 1

[7] E. Spriggs, F. De la Torre, and M. Hebert, “Temporal segmentation and activity classification from first-person sensing,” in *CVPRW 2009*, pp. 17–24, 2009. 1, 3

[8] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *CVPR 2012*, pp. 2847–2854, 2012. 1

[9] A. Behera, D. Hogg, and A. Cohn, “Egocentric activity monitoring and recovery,” in *Asian Conference on Computer Vision*, 2012. 1

[10] Y. Li, A. Fathi, and J. Rehg, “Learning to predict gaze in egocentric video,” in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013. 1, 2, 3

[11] A. Fathi, X. Ren, and J. Rehg, “Learning to recognize objects in egocentric activities,” in *CVPR 2011*, 2011. 1, 2

[12] A. Fathi and J. Rehg, “Modeling actions through state changes,” in *CVPR 2013*, pp. 2579–2586, 2013. 1

[13] G. Serra, M. Camurri, L. Baraldi, M. Benedetti, and R. Cucchiara, “Hand segmentation for gesture recognition in egovision,” in *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile &#38; Portable Devices*, IMMPD ’13, pp. 31–36, 2013. 2

[14] C. Li and K. Kitani, “Pixel-level hand detection in egocentric videos,” in *CVPR 2013*, 2013. 2, 6, 7

[15] P. Morerio, L. Marcenaro, and C. Regazzoni, “Hand detection in first person vision,” in *Information Fusion (FUSION), 2013 16th International Conference on*, pp. 1502–1507, 2013. 2

[16] A. Betancourt, M. Lopez, C. Regazzoni, and M. Rauterberg, “A sequential classifier for hand detection in the framework of egocentric vision,” in *CVPRW 2014*, 2014. 2

[17] S. Lee, S. Bambach, D. Crandall, J. Franchak, and C. Yu, “This hand is my hand: A probabilistic approach to hand disambiguation in egocentric video,” in *CVPRW 2014*, 2014. 2

[18] X. Zhu, X. Jia, and K.-Y. K. Wong, “Pixel-level hand detection with shape-aware structured forests,” in *Proc. Asian Conference on Computer Vision*, (Singapore), Nov 2014. 2

[19] M. Jones and J. Rehg, “Statistical color models with application to skin detection,” in *CVPR 1999*, vol. 1, 1999. 2

[20] V. Vezhnevets, V. Sazonov, and A. Andreeva, “A survey on pixel-based skin color detection techniques,” in *GraphiCon*, pp. 85–92, 2003. 2

[21] L. Sigal, S. Sclaroff, and V. Athitsos, “Skin color-based video segmentation under time-varying illumination,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, pp. 862–877, July 2004. 2

[22] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, “A survey of skin-color modeling and detection methods,” *Pattern Recogn.*, vol. 40, pp. 1106–1122, Mar. 2007. 2

[23] S. M. G. Tofighi and N. Ghasem-Aghaee, “Rapid hand posture recognition using adaptive histogram template of skin and hand edge contour,” in *Iranian Machine Vision and Image Processing Conference*, 2010. 2

[24] M. Kolsch and M. Turk, “Hand tracking with flocks of features,” in *CVPR 2005*, vol. 2, p. 1187, June 2005. 2

[25] F. S. Khan, J. Weijer, and M. Vanrell, “Modulating shape features by color attention for object recognition,” *International Journal of Computer Vision (IJCV)*, vol. 98, no. 1, 2012. 2

[26] F. Shahbaz Khan, R. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, and A. Lopez, “Color attributes for object detection,” in *CVPR 2012*, pp. 3306–3313, 2012. 2

[27] F. S. Khan, M. A. Rao, J. van de Weijer, A. D. Bagdanov, A. Lopez, and M. Felsberg, “Coloring action recognition in still images,” *International Journal of Computer Vision (IJCV)*, vol. 105, pp. 205–221, dec 2013. 2

[28] M. Danelljan, F. Khan, M. Felsberg, and J. van de Weijer, “Adaptive color attributes for real-time visual tracking,” in *CVPR 2014*, pp. 1090–1097, 2014. 2

[29] B. Berlin and P. Kay, *Basic Color Terms: their Universality and Evolution*. Berkeley and Los Angeles: University of California Press, 1969. 2

[30] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, “Learning color names for real-world applications,” *Image Processing, IEEE Transactions on*, vol. 18, July 2009. 2, 3

[31] Y. Li and J. Luo, “Task-relevant object detection and tracking,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 3900–3904, 2013. 3

[32] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision*, vol. 59, pp. 167–181, Sept. 2004. 3

[33] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, “Turbopixels: Fast superpixels using geometric flows,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 2290–2297, Dec 2009. 3

[34] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 888–905, Aug 2000. 3

[35] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking.,” in *ECCV (4)*, vol. 5305, 2008. 3

[36] O. Veksler, Y. Boykov, and P. Mehrani, “Superpixels and supervoxels in an energy optimization framework,” in *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV’10, pp. 211–224, 2010. 3

[37] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, Nov 2012. 3

[38] K. Ogaki, K. Kitani, Y. Sugano, and Y. Sato, “Coupling eye-motion and ego-motion features for first-person activity recognition,” in *CVPRW 2012*, pp. 1–7, 2012. 3

[39] S. Sundaram and W. Cuevas, “High level activity recognition using low resolution wearable vision,” in *CVPRW 2009*, pp. 25–32, 2009. 3

[40] W. W. Mayol and D. W. Murray, “Wearable hand activity recognition for event summarization,” in *ISWC ’05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, (Washington, DC, USA), pp. 122–129, Dec. 2005. 3

[41] M. Hanheide, N. Hofemann, and G. Sagerer, “Action recognition in a wearable assistance system,” in *Pattern Recognition, 2006. 18th International Conference on*, vol. 2, pp. 1254–1258, 2006. 3

[42] L. Sun, U. Klank, and M. Beetz, “Eyewatchme - 3d hand and object tracking for inside out activity analysis,” in *CVPRW 2009*, pp. 9–16, 2009. 3

[43] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *CVPR 2012*, pp. 2847–2854, 2012. 3

[44] K. Matsuo, K. Yamada, S. Ueno, and S. Naito, “An attention-based activity recognition for egocentric video,” in *CVPRW 2014*, pp. 565–570, 2014. 3

[45] E. Vig, M. Dorr, and D. Cox, “Space-variant descriptor sampling for action recognition based on saliency and eye movements,” in *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV’12, 2012. 3, 5

[46] F. Shi, E. Petriu, and R. Laganiere, “Sampling strategies for real-time action recognition,” in *CVPR 2013*, 2013. 3

[47] X. Peng, Y. Qiao, Q. Peng, and X. Qi, “Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition,” in *British Machine Vision Conference*, 2013. 3

[48] E. Nowak, F. Jurie, and B. Triggs, “Sampling strategies for bag-of-features image classification,” in *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*, ECCV’06, pp. 490–503, 2006. 3

[49] J. Yang, N. Zheng, J. Yang, M. Chen, and H. Chen, “A biased sampling strategy for object categorization,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1141–1148, 2009. 3

[50] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *ARTIFICAL INTELLIGENCE*, vol. 17, pp. 185–203, 1981. 4

[51] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” pp. 226–231, AAAI Press, 1996. 4

[52] H. Wang, A. Kläser, C. Schmid, and C. Liu, “Dense trajectories and motion boundary descriptors for action recognition,” *International Journal of Computer Vision*, vol. 103, pp. 60–79, May 2013. 5, 8

[53] Q. Le, W. Zou, S. Yeung, and A. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” in *CVPR 2011*, 2011. 5, 8

[54] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *British Machine Vision Conference*, 2010. 5

[55] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th International Conference on Multimedia*, pp. 357–360, 2007. 8