Mining Discriminative States of Hands and Objects to Recognize Egocentric Actions with a Wearable RGBD Camera

Shaohua Wan J.K. Aggarwal
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
shaohuawan@utexas.edu aggarwaljk@utexas.edu

Abstract

Of increasing interest to the computer vision community is to recognize egocentric actions. Conceptually, an egocentric action is largely identifiable by the states of hands and objects. For example, "drinking soda" is essentially composed of two sequential states where one first "takes up the soda can", then "drinks from the soda can". While existing algorithms commonly use manually defined states to train action classifiers, we present a novel model that automatically mines discriminative states for recognizing egocentric actions. To mine discriminative states, we propose a novel kernel function and formulate a Multiple Kernel Learning based framework to learn adaptive weights for different states. Experiments on three benchmark datasets, i.e., RGBD-Ego, ADL, and GTEA, clearly show that our recognition algorithm outperforms state-of-the-art algorithms.

1. Introduction

Human action recognition has been an active area of research for the past several decades and has wide applications in surveilence and robotics. State-of-the-art recognition algorithms have achieved successful performance on realistic actions collected from movies [15, 21], web videos [28, 19], and TV shows [26]. On the other hand, due to the recent widespread use of wearable cameras, an increasing amount of research interest has been directed at egocentric actions that involve hand-object interactions.

Whereas traditional third-person action recognition commonly uses the bag-of-features model [5, 14] for capturing salient motion patterns of body parts or joints, representations that encode the states of hands and objects have been proven more effective for recognizing hand-object interactions in egocentric videos. For example, "drinking soda" is essentially composed of two sequential states where one first "takes up the soda can", then "drinks from the soda can". Several methods have been proposed that aim to represent egocentric actions with state-specific hand and object features (clenched hands and tilted soda can); notably, they have found that coupled with discriminative state-specific detectors, state-of-the-art action recognition performance can be achieved [23, 7, 6, 27].

However, all previous works train action classifiers based on manually defined states, and little is understood as to their optimality. For example, a two-state action classifier is learned in [7] where the object appearance changes in the starting and ending frames are used to recognize an action. An adaptive approach to defining states of hands and objects comes from the idea of spatial pyramids [16], where regular spatial grids of increasing granularity are used to pool local features. Similarly, temporal pyramids can provide a reasonable cover over the state space with variable scales.

Based on the temporal pyramids approach, we specifically aim to answer the following question: Which is the optimal weight of each state for egocentric action recognition? While temporal pyramids succeed in dividing an action into increasingly finer states, one can reasonably expect that some states are more discriminative than others and should be weighted more by the recognition algorithm.

Instead of manually tuning the state weights, we aim to explicitly learn the state weights for action recognition. Specifically, we adopt the idea of using kernel functions for measuring state similarity and show that the state weights can be efficiently optimized in the Multiple Kernel Learning [9] framework; in addition, by discarding those non-discriminative states, we significantly reduce the computational cost while maintaining high recognition accuracy in our experiments.

As a byproduct of our work, we contribute a large-scale RGBD egocentric action dataset, which features typical hand-object interactions in our daily life. By exploiting color and depth cues, we show that accurate hand and object masks can be computed. Beyond understanding algorithms for egocentric action recognition, our dataset can also serve to research fields of hand detection and object recognition.

The rest of this paper is structured as follows. Related

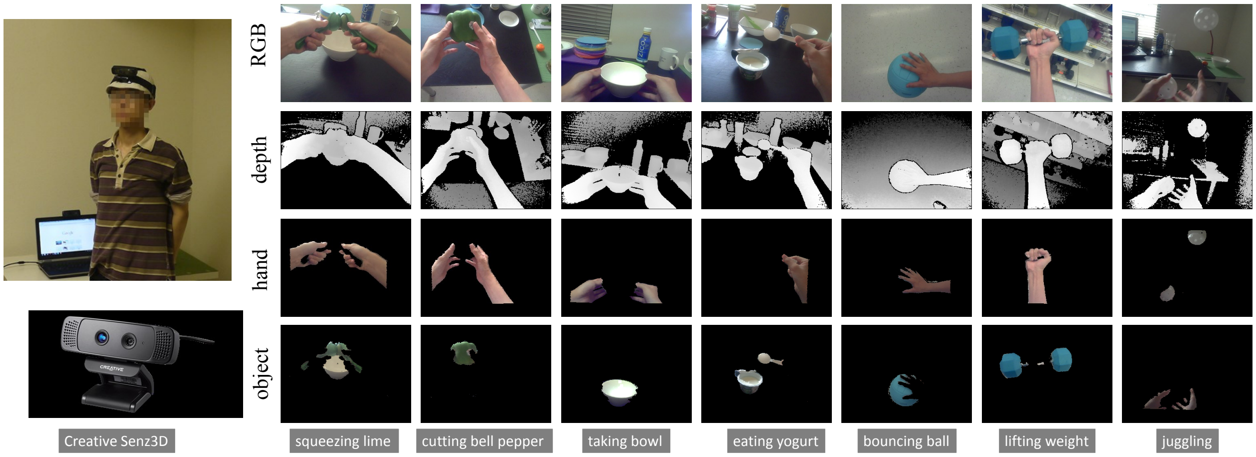


Figure 1. We mount the Creative Senz3D camera on a user's head to record RGBD egocentric actions. Our dataset contains 40 actions collected from 20 subjects. Pixel-level hand and object masks are available for each frame.

work is briefly reviewed in Section 2. In Section 3, we provide details on our egocentric action dataset. In Section 4, we describe how the hand and object features are extracted in this work. Our approach to mining discriminative states of hands and objects is described in Section 5. Various experimental results are presented in Section 6. The entire dataset and relevant code will be available online at http://xxxx.xxx.xxxxx.edu/xxxxxx/.

2. Related Work

Egocentric actions usually involve complex hand-object interactions. Wu *et al.* [32] demonstrated the effectiveness of object-centric representations for egocentric action recognition by recording the interacted objects using RFID tags attached to objects. [6, 10, 22] demonstrated improved recognition performance by jointly modeling hands and objects. In order to better discriminate between actions that involve the same object, *e.g.*, "opening jar" *v.s.* "closing jar", [25, 7, 27, 23] used state-specific hand/object detectors to acquire cause-and-effect relationship in egocentric actions. However, these approaches were largely impacted by inaccurate hand/object detection results due to the significantly varied appearance of hands and objects under occlusion and viewpoint changes.

Recent work have demonstrated robust recognition performance when aggregating visual information from a set of frames that cover different pose and occlusion [20, 13, 33, 17]. Inspired by this idea, we model each state of hands and objects using a set of consecutive frames within a video and propose a novel kernel function for comparing set similarity. We show that our state kernel is robust to intra-state variations due to occlusion and viewpoint changes.

The idea of using temporal pyramids to model the temporal structure of an action has been addressed in a number of works [15, 34, 23, 27], all using a predefined set of weights for individual temporal grids. We instead aim to learn the optimal weight for each temporal grid. Learning weights has the benefit of removing redundancy and increasing discriminativeness, thus improving the performance of the egocentric action recognition.

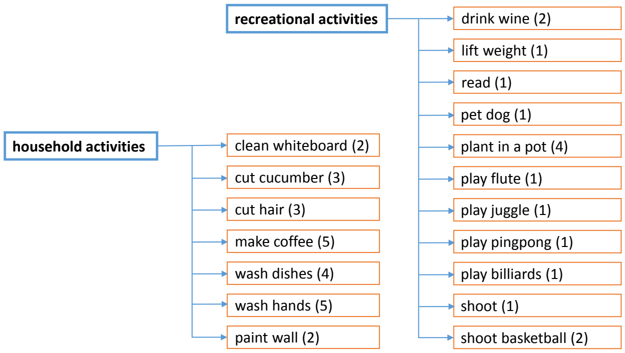


Figure 2. Activities categories in our dataset. The number in the parentheses denotes the number of actions making up the activity.

3. RGBD Egocentric Action Dataset

We used the Creative Senz3d camera to collect our RGBD egocentric action dataset. Creative Senz3d is a compact sized camera that records synchronized color and depth video at up to 30fps. The color video has a resolution of 640×480, and the depth video has a resolution of 320×240 with an effective range of 0.15 m to 0.99 m. We mount the camera on the user's head such that it covers the area in front of the user's eyes.

We put together a list of 18 daily activities, and asked 20 subjects to perform each activity twice in their own style in order to collect realistic and varied data. All the activities involve complex hand-object interactions, including 7 household activities (*e.g.*, washing dishes), and 11 recre-

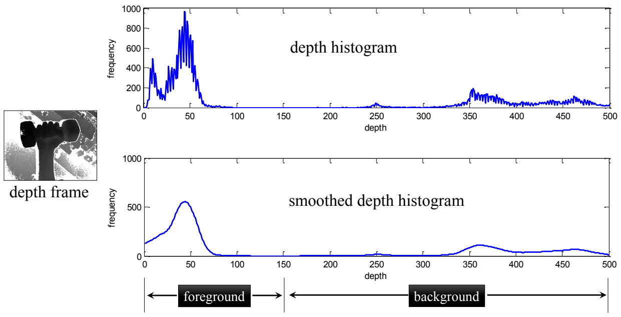


Figure 3. The histogram of a depth frame.

ational activities (*e.g.*, playing ping-pong), see Figure 2. Note that some activities may display a large amount of ego-motion in the recorded videos. For instance, in the case of "playing ping-pong", the field of view of the camera changes significantly as a result of the user hitting the ball.

Some activities in our dateset are characterized by a complex temporal composition of sub-activities. For example, "wash hands" can be divided into 5 sub-activities, *i.e.*, "wet hands under water", "use soap", "rub hands to make bubbles", "rinse hands", and "dry hands using towel". In this work, we use the term "actions" to refer to sub-activities that constitute the complex activities. We manually segment each activity into actions in time such that the first and last frame roughly correspond to the start and end of an action, resulting in 40 unique actions.

Both the RGB and depth frames were calibrated using a set of checkerboard images in conjunction with the calibration tool of Burrus [3]. This also provided the homography between the two cameras, allowing us to obtain precise spatial alignment between the RGB and depth frames.

4. Hand and Object Features

In this section, we first describe an effective RGBD-based hand and object segmentation algorithm (Section 4.1). Then we present the hand and object features that will be used for egocentric action recognition (Section 4.2).

4.1. Hand and Object Segmentation

In order to extract features that are truly representative of hands and objects, it is necessary to accurately segment hands and objects in each frame. Our segmentation pipeline consists of two steps, foreground segmentation and skin detection.

Foreground Segmentation We segment a scene into foreground and background based on the observation that hands and objects which constitute the foreground are at a closer distance to the first-person than to the background. This suggests that a thresholding operation on the depth frame can help segment the scene into foreground and background.

Figure 3 shows the histogram of a depth frame from "lifting weight". Note the gap in the histogram that separates the scene into foreground and background. An extensive analysis of the egocentric actions in our dataset shows that the exact position of the separation gap may vary from action to action and there can be "deceptive" gaps due to artifacts in depth frames.

In order to account for the varied statistics of depth frames, we first convert the histogram of each depth frame into a non-parametric probability density distribution using a Gaussian kernel function. This helps smooth the histogram and remove deceptive gaps. To identify the ideal threshold for segmenting the scene, we then seek the left-most minimum of the histogram curve. Finally, a foreground mask is obtained by thresholding the depth frame using the previously selected threshold. Empirically we find that a histogram of $k = 1000$ bins smoothed by a Gaussian kernel of variance $\sigma^2 = 5$ gives good segmentation results.

Skin Detection We perform skin detection to further segment the foreground into hands and objects. Our skin detector combines color and texture analysis. In color analysis, a bi-threshold classifier is used to label each pixel as skin given its RGB value. That is, pixels which are above the high threshold are classified as skin. Then, pixels which are above the low threshold are also classified as skin if they are spatial neighbors of a pixel above the high threshold (these thresholds are determined by cross-validation on groundtruth segmentation). The skin likelihood of a pixel given its RGB value is determined from a pre-trained lookup table [12].

Simply applying color analysis gives good skin detection results for many videos in the dataset, but is still problematic when the object has skin-like color (*e.g.*, an orange ping-pong ball). Therefore, we also perform texture analysis to improve the skin detection accuracy. In particular, we apply a Gabor feature based texture classifier on the output of color analysis in order to distinguish between genuine and fake skin pixels [18].

Combining color and texture analysis provides high-quality skin detection, and given the detected skin, all the remaining pixels in the foreground are classified as belonging to the object.

4.2. Hand and Object Features

Our hand and object features build on existing feature descriptors and go a step further by incorporating the depth information. The details are described as follows.

4.2.1 Hand Features

We extract dense optical flow from the hand region to characterize hand motion in the current frame. Utilizing depth data, we describe an effective way of computing 3D optical flow. In particular, we first compute the 2D dense optical flow field (u_t, v_t) in RGB frame t. Each point (x_t, y_t) in RGB frame t is tracked to the next frame $t+1$ using the 2D flow vector at (x_t, y_t)

$$(x_{t+1}, y_{t+1}) = (x_t, y_t) + (u_t, v_t)|_{(x_t, y_t)} \tag{1}$$

Assuming the RGB and depth frames have been accurately registered, the 3D flow vector at (x_t, y_t, z_t) can thus be written as $(u_t, v_t, w_t)|_{(x_t, y_t, z_t)}$, where $w_t|_{(x_t, y_t, z_t)} = z_{t+1} - z_t$, z_t is the depth value at (x_t, y_t) in depth frame t. As optical flow is subpixel accurate, (x_{t+1}, y_{t+1}) will usually end up between pixels. We thus use bilinear interpolation to infer z_{t+1}.

To quantize the orientation of a 3D optical flow vector, we use an icosahedron (*i.e.*, a regular polyhedron with 20 faces), where each face of the polyhedron corresponds to a histogram bin. We construct Histogram-of-Optical Flow (HOF) features from the 3D flow vectors. With an additional bin for zero flow, the resulting HOF feature has a dimension of 21. l_2-normalization is applied to the HOF feature.

4.2.2 Object Features

As object features we extract HOG within the rectangular region containing the segmented object. The rectangular region is first divided in 8×8 non-overlapping cells. For each cell, we accumulate a histogram of oriented gradients with 9 orientation bins. Finally, the histogram of each cell is normalized with respect to the gradient energy in a neighborhood around it. The HOG features from RGBD image pair are concatenated to describe the object appearance in the current RGBD image pair.

5. Modeling States of Hands and Objects

Modeling states of hands and objects is key to capturing the cause-and-effect relationships in egocentric actions, which is especially critical to discriminating between actions involving the same objects. Previous work mainly focused on improving the accuracy and scalability of state-specific object detectors [7, 27, 23], and do not generalize well to model gradual state transitions.

Inspired by the adaptivity of spatial pyramids [16], we propose to use temporal pyramids to combine states of arbitrary temporal scales. Without any prior knowledge of the intrinsic structure of action, a natural extension from spatial pyramids is to partition a full-length video into increasingly shorter segments and represent a state using the corresponding segment. In particular, given a video $\boldsymbol{X}^i$, we construct a temporal pyramid, where the top level $l = 0$ is the full-length video, the next level $l = 1$ contains two segments obtained by temporally splitting the segment on level $l = 0$ in two, and so on. Let $\boldsymbol{X}^i_{lk}$ denote the k^{th} segment on level l.

Assuming some appropriate kernel function $\kappa(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk})$ for measuring state similarity, a binary SVM classification scheme for recognizing a novel egocentric video $\boldsymbol{X}$ can be written as

$$f(\boldsymbol{X}) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i \mathcal{K}(\boldsymbol{X}^i, \boldsymbol{X}) + b) \tag{2}$$

where $y_i \in \{-1, 1\}$ is the label for $\boldsymbol{X}^i$, $\mathcal{K}(\boldsymbol{X}^i, \boldsymbol{X}) = \sum_{l=1}^{L} \sum_{k=1}^{K} \mu_{lk} \cdot \kappa(\boldsymbol{X}^i_{lk}, \boldsymbol{X}_{lk})$ is a compound kernel constructed from a weighted sum of $\kappa(\cdot, \cdot)$, $\mu_{lk} = 1/2^{L-l}$ assigns a small weight to state kernels on coarse temporal scales.

Eq. 2 provides a general definition that embraces existing egocentric action state models. For example, using a linear kernel and representing each state using a histogram of detected objects corresponds to the object-centric approaches in [23, 27].

5.1. A Novel State Kernel Function

While representing a state using the feature from a single frame is largely affected by intra-state variations due to viewpoint change and hand occlusion, we propose to aggregate features from a set of frames to cover potential variations within a state. In particular, we write $\boldsymbol{X}^i_{lk} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_p] \in \mathbb{R}^{d \times p}$, where $\boldsymbol{x}_m \in \mathbb{R}^d$, $1 \leq m \leq p$, is the feature capturing hand and object information in the m^{th} frame of $\boldsymbol{X}^i_{lk}$. Given two states, $\boldsymbol{X}^i_{lk} \in \mathbb{R}^{d \times p}$ and $\boldsymbol{X}^j_{lk} \in \mathbb{R}^{d \times q}$, we define our state kernel based on the notion of affine hull.

Mathematically, the affine hull of a set S is the set of all affine combinations of elements of S, *i.e.* $\text{aff}(S) = \{\sum_i \beta_i s_i | s_i \in S, \sum_i \beta_i = 1\}$. It provides a unified expression for "unseen" elements of S. Cevikalp *et al.* [4] proposed to define the distance between two sets as the minimum distance between elements from the affine hulls, *i.e.*,

$$\mathcal{D}(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk}) = \min_{\boldsymbol{\beta}^i, \boldsymbol{\beta}^j} \|\boldsymbol{X}^i_{lk}\boldsymbol{\beta}^i - \boldsymbol{X}^j_{lk}\boldsymbol{\beta}^j\|_2^2 \tag{3a}$$

$$\text{s.t.} \quad \sum_{m=1}^{p} \beta^i_m = 1 \;\; \text{and} \;\; \sum_{n=1}^{q} \beta^j_n = 1 \tag{3b}$$

where $\boldsymbol{\beta}^i \in \mathbb{R}^p$ and $\boldsymbol{\beta}^j \in \mathbb{R}^q$ are the affine coefficients for $\boldsymbol{X}^i_{lk}$ and $\boldsymbol{X}^j_{lk}$, respectively. However, the affine hull may turn out to be an overestimate of the extent of a set, especially when it comes to visual recognition problems[11].

Motivated by the recent success of sparse representation techniques [31], we introduce sparsity regularization terms on affine coefficients, *i.e.*,

$$\{\hat{\boldsymbol{\beta}}^i, \hat{\boldsymbol{\beta}}^j\} \leftarrow \arg\min_{\boldsymbol{\beta}^i, \boldsymbol{\beta}^j} \|\boldsymbol{X}^i_{lk}\boldsymbol{\beta}^i - \boldsymbol{X}^j_{lk}\boldsymbol{\beta}^j\|_2^2 + \lambda|\boldsymbol{\beta^i}|_1 + \lambda|\boldsymbol{\beta^j}|_1 \tag{4a}$$

$$\text{s.t.} \quad \sum_{m=1}^{p} \beta^i_m = 1 \ \text{ and } \ \sum_{n=1}^{q} \beta^j_n = 1 \tag{4b}$$

where $|\cdot|_1$ denotes the l_1-norm of a vector and is known for its sparsity-inducing properties. Under l_1-norm penalty, the unseen feature is restricted to be a weighted sum of a few existing features; this sparse representation is supported by the fact that the varied appearance of hand and object in a specific state lies in a low-dimensional subspace [1]. Eq. 4 is jointly convex with respect to $\boldsymbol{\beta}^i$ and $\boldsymbol{\beta}^j$, and the global solution can be efficiently solved by the Alternating Direction Method of Multipliers (ADMM) algorithm [2], see the supplementary material for details.

The state kernel function is defined as

$$\kappa(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk}) = \exp\big(-\frac{1}{\gamma}\mathcal{D}(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk})\big) \tag{5}$$

where $\mathcal{D}(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk}) = \|\boldsymbol{X}^i_{lk}\hat{\boldsymbol{\beta}}^i - \boldsymbol{X}^j_{lk}\hat{\boldsymbol{\beta}}^j\|_2^2$, and γ is the mean value of $\mathcal{D}(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^j_{lk})$ in the training examples.

5.2. Mining Discriminative States

One practical issue with the classifier defined in Eq. 2 is that, using a predefined set of weights $\{\mu_{lk}\}$ for different hand and object states may not be optimal, as various states contribute differently to classifying an action. To mine discriminative states, we adopt the idea of Multiple Kernel Learning (MKL) [9], and jointly learn the state weights and other SVM parameters by solving

$$\min_{\mu_{lk}, \boldsymbol{w}_{lk}, \xi_i, b} \frac{1}{2}\Big(\sum_{l=1}^{L}\sum_{k=1}^{K}\mu_{lk}\|\boldsymbol{w}_{lk}\|_2\Big)^2 + C\sum_{i=1}^{N}\xi_i \tag{6a}$$

$$\text{s.t.} \quad y_i\Big(\sum_{l=1}^{L}\sum_{k=1}^{K}\mu_{lk}\boldsymbol{w}_{lk}^T\phi(\boldsymbol{X}^i_{lk}) + b\Big) > 1 - \xi_i \tag{6b}$$

$$\sum_{l=1}^{L}\sum_{k=1}^{K}\mu_{lk} = 1 \ \text{ and } \ \mu_{lk} \geq 0, \ \forall k, l \tag{6c}$$

$$\xi_i \geq 0, \ \forall i \tag{6d}$$

where $\phi(\boldsymbol{X}^i_{lk})$ is the mapping function satisfying $\kappa(\boldsymbol{X}^i_{lk}, \boldsymbol{X}^i_{lk}) = \phi(\boldsymbol{X}^i_{lk})^T\phi(\boldsymbol{X}^i_{lk})$. The algorithm from [29] is used to optimize the parameters. To perform multi-class classification, we learn class-specific parameters $\{\mu^c_{lk}, \boldsymbol{w}^c_{lk}, \xi^c_i, b^c\}$ for the c-th action class using the one-versus-all approach.

	FG	*H/O-1*	*H/O-2*
precision	0.949	0.923	0.911
recall	0.978	0.953	0.944
F1 score	0.963	0.938	0.927
time (sec/frame)	0.025	0.274	0.282

Table 1. The performance of foreground segmentation and hand/object segmentation. *FG*: foreground segmentation. *H/O-1*: hand/object segmentation given the groundtruth foreground. *H/O-2*: hand/object segmentation given the foreground produced by *FG*.

6. Experiments

In this section, we first experimentally verify the accuracy of the hand/object segmentation pipeline. We then extensively evaluate the proposed egocentric action recognition algorithm on 3 benchmark datasets. We show that significant performance improvement over the state-of-the-art algorithms is achieved by mining discriminative states of hands and objects.

6.1. Hand and Object Segmentation

Hand and object segmentation serves an important role in extracting features only from interest regions. In this section, we evaluate the efficiency and accuracy of the proposed hand/object segmentation method. To this end, we randomly select a set of RGBD images (10 RGB images and 10 depth images for each action class, 800 images in total) as our validation set. Groundtruth hand and object masks are obtained by means of manual annotation. We perform 3 independent experiments: 1) *FG*: foreground segmentation; 2) *H/O-1*: hand/object segmentation given the groundtruth foreground; 3) *H/O-2*: hand/object segmentation given the foreground produced by *FG*. All experiments are run on a standard PC with 3.40 GHz Intel Core I7 processors and 8 GB RAM.

Table 1 gives the results. For foreground segmentation, *FG* gives an F1 score of as high as 96.3% while being extremely efficient (0.025 sec/frame, or 40.0 frame/sec). As for hand/object segmentation, *H/O-2* performs approximately the same as *H/O-1*, indicating that skin detection is not affected much by the errors introduced in automatic foreground segmentation. A close look at the hand/object segmentation results reveals that illumination affects the skin detection more than any other factors. For example, our skin detector tends to give a low recall in environments such as a dark stairway. As part of the future work, we expect to improve the skin detection by explicitly modeling illumination changes.

		activity	action
(1)	principal angles [30]	0.578	0.469
(2)	KL-Divergence [24]	0.553	0.472
(3)	convex hull [4]	0.643	0.522
(4)	affine hull [4]	0.664	0.563
(5)	bag-of-active-objs [27]	0.547	0.423
(6)	sparse affine hull (ours)	**0.721**	**0.663**

Table 2. The activity and action recognition accuracy on the RGBD-Ego dataset using different state kernels.

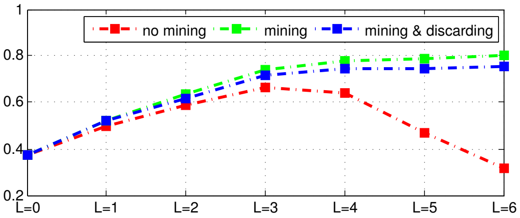
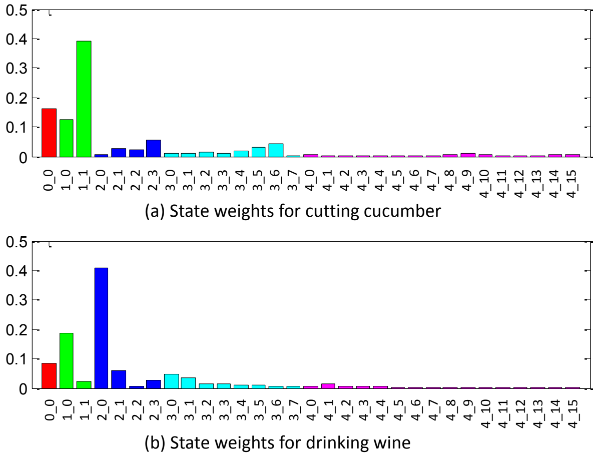


Figure 4. The action recognition accuracy on the RGBD-Ego dataset in 3 settings: (1) without mining states, (2) mining states, (3) mining states and discarding trivial states. The number of pyramid levels L is varied from 0 to 6.



(a) State weights for cutting cucumber

(b) State weights for drinking wine

Figure 5. The weights of states learned for two actions: (a) cutting cucumber and (b) drinking wine. The x-axis tick label l_k denotes the k-th state on the l-th level. States on the same level l are plotted in the same color.

6.2. Evaluating the Egocentric Action Recognition Algorithm

In this section, we present various experimental results on three egocentric video datasets, RGBD-Ego (our dataset), Activities of Daily Living (ADL) [27], and Georgia Tech Egocentric Activities (GTEA) [8].

6.2.1 Results on the RGBD-Ego Dataset

The RGBD-Ego dataset provides video annotation on two levels, (1) 18-class activities characterized by long-duration hand-object interations, and (2) 40-class actions characterized by short-duration hand-object interations. It is interesting to see how well our state model performs at the activity-level and the action-level. We thus perform two groups of experiments, that is, 18-class activity recognition and 40-class action recognition. For both groups, we use 10-fold cross-validation while ensuring that activities/actions performed by the same person do not appear across both training and testing data. The average recognition accuracy is computed by averaging the diagonal of the confusion matrix.

The State Kernel Function

The proposed kernel function $\kappa(\cdot,\cdot)$ is defined based on calculating the distance between two sets using sparse affine hulls. There exist several other definitions of distance between two sets. We thus implement $\mathcal{D}(\cdot,\cdot)$ by different set-distance definitions: (1) principal angles [30], (2) KL-divergence [24], (3) convex hull [4], and (4) affine hull [4]. We also test the (5) bag-of-active-objs kernel proposed in [27]. For all tests, the number of pyramid levels L is set to 3. Results are given in Table 2.

As can be seen, the (5) bag-of-active-objs kernel gives a relatively low performance. This is mainly due to the limited accuracy of object detectors (with groundtruth object information, accuracy is improved to 0.674 and 0.589 for activity and action recognition, respectively). Other set-distance based kernels (1)$\sim$(4) and (6) give much higher accuracy. Our kernel gives the highest accuracy, indicating the effectiveness of l_1-norm regularization on the affine hull of a set.

Also note that the accuracy of activity recognition is consistently higher than action recognition across different kernels mainly because a) the number of activity classes is less than the number of action classes and b) similar objects tend to increase the confusion among different actions.

Mining Discriminative States

Using temporal pyramids to combine increasingly finer states of hands and objects, there are effectively two possible directions to improve the performance: to increase the number of pyramid levels L and to mine discriminative states. We argue that these two directions are complementary: the performance gain from mining discriminative states could not be simply replaced by increasing the number of pyramid levels. In fact, as the number of levels grows, the performance may drop due to feature variation and mismatching on fine temporal scales, while one can still obtain gains by mining discriminative states and discarding those that are not performance-enhancing.

To empirically justify this argument, we perform experiments by varying the number of pyramid levels L, and

BoW (HOF+HOG) [15]	0.235
Pirsiavash *et al.* [27]	0.369
McCandless *et al.* [23]	0.387
Mining States, Obj	**0.458**
Mining States, Hand+Obj	**0.513**

Table 3. The activity recognition accuracy on the ADL dataset.

	activity	action
BoW (HOF+HOG) [15]	0.571	0.213
Fathi *et al.* [6]	0.857	0.230
Fathi *et al.* [7]	n/a	0.397
Mining States, Obj	**0.857**	**0.459**
Mining States, Hand+Obj	**1.000**	**0.525**

Table 4. The recognition accuracy on the GTEA dataset.

comparing the recognition accuracy with and without mining states. The accuracy of action recognition is given in Figure 4. As can be seen, the performance without mining reaches the highest at $L = 3$ and significantly drops from beyond $L = 4$, while mining states always brings additional performance gains. The accuracy for activity recognition is interpreted similarly to action recogniton, and is omitted due to space limitations.

Furthermore, the importance of mining states lies in redundancy removal. Figure 5 plots the state weights of two actions learned from our algorithm when $L = 3$. As can be seen, many states have very small weights and are negligible to egocentric action recognition. In fact, we are able to maintain satisfactory performance when only using states whose weights are greater than $0.05 \cdot \max_{lk}\{\mu_{lk}\}$, see the blue curve in Figure 4. On average, this effectively removes 85% states from the temporal pyramid, and is particularly valuable to time-bounded applications.

6.2.2 Results on the ADL Dataset

ADL is an RGB egocentric video dataset of 18 different daily living activities, such as “making tea”, “washing dishes”, and “using computer”. These activities are each performed by 20 subjects in an uncontrolled manner. Note that no action-level video annotation is available for this dataset. For this dataset, only activity recognition is considered.

Pirsiavash *et al.* [27] demonstrated relatively successful recognition performance using the Bag-of-Active-Objects approach, where a state is represented by a histogram of interacted objects. McCandless *et al.* [23] achieved further performance improvement by adaptively learning spatial-temporal binning schemes. In this experiment, we include BoW (Bag-of-Words) as a baseline where HOF and HOG features are densely extracted and concatenated to represent egocentric actions [15].

For consistency, the same evaluation protocol as in [27, 23] is used. That is, we use leave-one-out cross-validation, where we ensure that activities of the same person does not appear across both training and testing data. We compute the overall recognition rate by averaging the diagonal of the confusion matrix. We obtain hand and object regions using the method in [8]. The pyramid level $L =$ is set to 3. Since no depth data is available, we simply extract 2D HOF and HOG features to characterize the state of hands and objects. Two versions of our algorithm are tested, MiningState-Obj, where only object features are used to ensure a fair comparison with object-centric approaches, and MiningState-Hand+Obj, where hand and object features are combined.

Table 3 lists the recognition accuracy of various methods. BoW, being unable to encode the states of hands and objects, achieves the lowest accuracy among all compared methods. While Pirsiavash *et al.* [27] and McCandless *et al.* [23] represent videos using detected objects and achieve significantly higher accuracy than BoW, they are inherently limited by the spurious output of object detectors. Our algorithm achieves the highest accuracy (0.458 and 0.513), demonstrating the effectiveness of the proposed kernel function and the state mining algorithm.

6.2.3 Results on the GTEA Dataset

The GTEA dataset [8] is an RGB egocentric video dataset that contains 7 food/drink preparation activities performed by 4 subjects, such as “making cheese sandwich” and “making coffee”. These 7 activities are further segmented into a total number of 61 actions. Pixel-wise hand and object masks are provided with the dataset.

Fathi *et al.* [6] built a graphical model for jointly learning activities, actions, hands, and objects, and demonstrated promising performance. Fathi *et al.* [7] improved the performance of action recognition using state-specific object detectors.

We perform activity and action recognition using our state model. As in [6, 7], we use the videos by subjects 1, 3 and 4 for training, and use the videos by subject 2 for testing. We compute the overall recognition rate by averaging the diagonal of the confusion matrix. The number of pyramid levels L is set to 3. 2D HOF and HOG features are extracted to characterize the states of hands and objects, respectively. The results are given in Table 4.

Our state model significantly outperforms previous methods for both activity and action recognition. It is worth noting that our model gives 100% accuracy for activity recognition on the GTEA dataset when combining states of hands and objects.

7. Conclusion

Modeling the state of hands and objects is critical to recognizing egocentric actions. In this paper, we presented a novel approach to mining discriminative states of different action classes. Our results showed that the significance of each state is vastly different across the action classes, and optimizing their respective weights is capable of achieving dramatically improved accuracy on 3 benchmark datasets. We also proposed a novel kernel function for calculating the similarity between two states. Our kernel function is capable of covering complex hand and object variations during a hand-object interaction task, thus greatly improving the robustness of our recognition algorithm.

In our current work, the use of temporal pyramid restricts each state to be temporally aligned with the grid of the pyramid. To allow more flexible state definitions, our future work consists of mining discriminative states that may be of arbitrary time-shift and length.

References

[1] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *TPAMI*, 25(2):218–233, Feb. 2003. 5

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011. 5

[3] N. Burrus. Kinect rgb demo v0.4.0. *http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV2*. 3

[4] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, 2010. 4, 6

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCCN*, 2005. 1

[6] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011. 1, 2, 7

[7] A. Fathi and J. Rehg. Modeling actions through state changes. In *CVPR*, 2013. 1, 2, 4, 7

[8] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011. 6, 7

[9] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *JMLR*, 12:2211–2268, July 2011. 1, 5

[10] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 31:1775–1789, 2009. 2

[11] Y. Hu, A. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011. 4

[12] M. Jones and J. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, Jan. 2002. 3

[13] T.-K. Kim, J. Kittler, and R. Cipolla. Incremental learning of locally orthogonal subspaces for set-based object recognition. In *BMVC*, 2006. 2

[14] I. Laptev. On space-time interest points. *IJCV*, 2005. 1

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 1, 2, 7

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 4

[17] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *CVPR*, 2003. 2

[18] C. Li and K. Kitani. Pixel-level hand detection in ego-centric videos. In *CVPR*, 2013. 3

[19] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009. 1

[20] Y. Liu, Y. Jang, W. Woo, and T.-K. Kim. Video-based object recognition using novel set-of-sets representations. In *CVPR Workshops*, 2014. 2

[21] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 1

[22] W. Mayol and D. Murray. Wearable hand activity recognition for event summarization. In *ISWC*, 2005. 2

[23] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, 2013. 1, 2, 4, 7

[24] P. J. Moreno, P. P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *NIPS*. 2004. 6

[25] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi. Head, eye, and hand patterns for driver activity recognition. In *ICPR*, 2014. 2

[26] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in tv shows. *TPAMI*, 34(12):2441–2453, Dec. 2012. 1

[27] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 1, 2, 4, 6, 7

[28] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *MVA*, 24(5):971–981, 2013. 1

[29] S. Sonnenburg, G. Rätsch, and C. Schäfer. A General and Efficient Multiple Kernel Learning Algorithm. In *NIPS*, 2006. 5

[30] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *JMLR*, 4:913–931, 2003. 6

[31] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2):210–227, Feb. 2009. 4

[32] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, 2007. 2

[33] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *AFGR*, 1998. 2

[34] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In *ECCV*, 2012. 2