Robust Object Recognition in RGB-D Egocentric Videos based on Sparse Affine Hull Kernel

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

Abstract

In this paper, we propose a novel kernel function for recognizing objects in RGB-D egocentric videos. In order to effectively exploit the varied object appearance in a video, we take a set-based recognition approach and represent the target object using the set of frames contained in the video. Our kernel function measures the similarity of two sets by the minimum distance between the sparse affine hulls of the two sets. Our kernel function also allows convenient integration of heterogeneous data modalities beyond RGB and depth. We extensively evaluate the proposed method on three benchmark datasets, including two RGB-D object datasets and one thermal/visible face dataset. All the results clearly show that the proposed method outperforms state-of-the-art methods.

1. Introduction

Object recognition is a challenging problem with many real-life applications. Traditional object recognition has mainly focused on scenarios where the class of the object is predicted using only a single image [12, 22, 33]. On the other hand, due to the widespread use of wearable cameras, an increasing amount of research interest has been directed recognizing objects in egocentric videos [10, 11, 28].

In this paper, we consider the problem of recognizing objects in egocentric videos where both RGB and depth data is available. To this end, we introduce a new dataset composed of RGB-D egocentric videos capturing daily objects while they are being manipulated during human activities. See Figure 1 for examples from our dataset.

In contrast to previous datasets which consist of images or videos of static objects [8, 9, 24], the appearance of the objects in our dataset may change significantly due to varied pose, illumination, and hand occlusion. The objects may also undergo state changes during egocentric activities. For example, an apple can be cut into two halves in the "cutting an apple" activity.

Although object recognition in egocentric videos has been previously addressed [11, 28, 29], they all learned object models from individual frames, and little effort was made towards exploiting the appearance variations as displayed in the videos.

The major contribution of this work is a novel kernel function for object recognition in RGB-D egocentric videos. Instead of considering each frame independently, we model an object using the set of frames contained in the video. By measuring the distance between two sets as the minimum distance between their affine hulls under the sparsity constraint, our kernel function is less affected by object variations and significantly improves performance of object recognition in egocentric videos.

The proposed kernel function allows convenient integration of heterogeneous data modalities (RGB, depth, infrared, etc.) under the Multiple Kernel Learning (MKL) [13] framework. In particular, our algorithm is capable of learning object models that are highly adapted to object patterns such as texture, shape, and thermal radition by assigning appropriate weight to each modality. This proves to further improve the recognition performance.

As an important preprocessing step, we show that combining RGB and depth data is extremely useful for segmenting target objects in egocentric videos. Our object segmentation algorithm consists of foreground segmentation followed by skin removal. Compared to previous segmentation algorithms that rely on optical flow information [28], our algorithm is both accurate and efficient.

The rest of this paper is structured as follows. Section 2 briefly reviews related work on video-based object recognition. Section 3 presents our RGB-D egocentric object dataset. In Section 4, we describe in detail the object segmentation process and the feature representation. We give details on our object recognition algorithm in Section 5. Various experimental results are presented in Section 6, followed by the conclusion in Section 7.

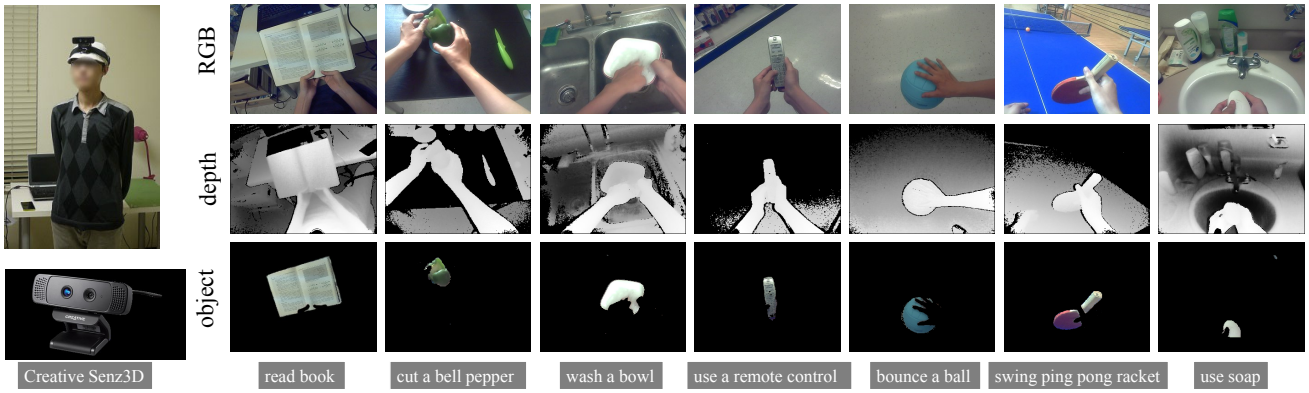

Figure 1. We mount the Creative Senz3D camera on the subject's head to record RGB-D egocentric videos of objects. Our dataset contains 40 object classes each of which has 16∼20 instances. The target object in each video sequence is automatically segmented based on RGB and depth cues.

2. Related Work

Recognizing objects of daily use can provide rich information about a user's daily activities. Several authors [27, 38] had been relatively successful in using RFID tags for detecting handled objects.

The problem of object recognition in egocentric videos is driven by the the emergence of wearable cameras. Mayol and Murray [23] first studied the detection of handled objects using a wearable camera by color histogram matching. Merler *et. al.* [24] addressed the problem of recognizing objects in videos collected in natural environment (*in situ*) with training images extracted from the web (*in vitro*). Ren and Gu [28] showed that figure-ground segmentation improved handled object recognition in egocentric video.

The recent advancement in sensing technologies, especially the introduction of Kinect-style depth sensors, has greatly facilitated the collection of depth data at relatively low cost. Several works [20, 21] demonstrated improved performance in handheld object recognition when combining RGB and depth cues.

Representing varied object appearance in a video using a set of frames has demonstrated superior recognition performance in unconstrained settings [17]. These methods can be broadly classified into two categories, parametric and non-parametric. Parametric methods seek to represent a set of frames by some parameterized distributions, *e.g.* single Gaussian [30] or Gaussian Mixture Model (GMM) [2], and then measure the similarity between two distributions in terms of the Kullback-Leibler Divergence (KLD).

Instead of assuming a parameterized distribution of a frame set, non-parametric methods usually represent a frame set using subspace/manifold. Convenient distance definitions between subspaces/manifolds (*e.g.*, principal angles) largely facilitate the application of off-the-shelf classifiers such as kNN, SVM [17, 35, 36].

objects in workplace activities	objects in recreational activities	objects in household activities
read a **book**	bounce a **ball**	set an **alarm clock**
use a **calculator**	play **billiards**	cut an **apple**
use a **cellphone**	use a **camera**	peel a **banana**
move a **chair**	pet a **cat**	drink from a **beer bottle**
use a **flashlight**	play **chess**	cut a **bell pepper**
use a **hammer**	swipe a **credit card**	wash a **bowl**
use a **hand sanitizer**	throw **darts**	open a **cereal box**
use a **highlighter**	play **flute**	cut a **cucumber**
speak to a **microphone**	swing a **ping pong racket**	use a **ladle**
use a pair of **pliers**	play with a **Rubik's cube**	use a **mug**
use a **stapler**	drink from a **water bottle**	pour water from a **pitcher**
use a **whiteboard cleaner**	lift **weight**	use a **soap**
		open a **soda can**
		use a **TV remote control**
		squeeze a **toothpaste**
		use a **towel**

Figure 2. A complete list of the object classes in our dataset. The objects are organized into three groups based on the corresponding activities.

Recently, [7] proposed to represent a frame set using the affine hull of the set. Since affine hull typically gives a rather loose approximation to the actual data distribution, several regularization schemes have been proposed to constrain the affine hull of a set [14, 25, 39].

To the best of our knowledge, this paper represents the first attempt to apply set-based classification to object recognition in egocentric videos. We propose a novel sparsity-regularized affine hull (SAH) kernel, where the distance between two sets can be efficiently solved by convex optimization. Compared to previous regularized affine hull methods [14, 25, 39], our formulation is more concise, requires less tuning, and admits a global optimum.

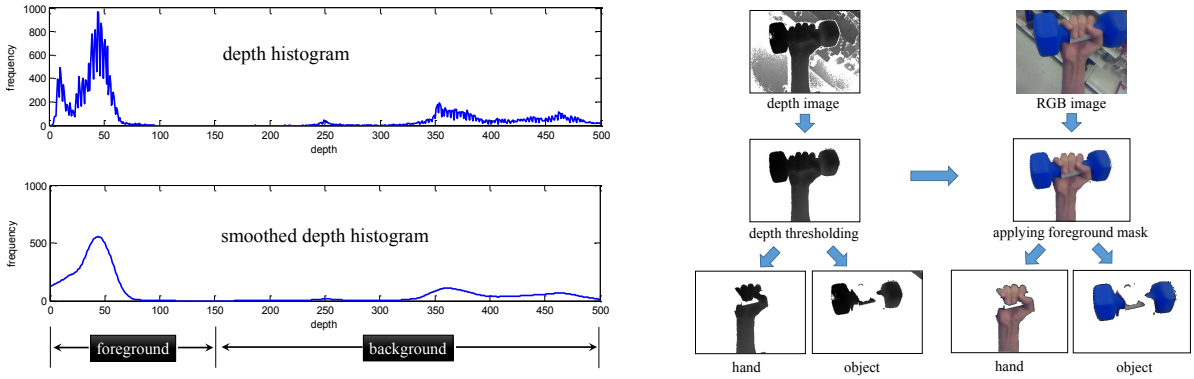

Figure 3. The depth value histogram of a depth frame (left) and the pipeline for segmenting the target object (right).

3. RGB-D Egocentric Object Dataset

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

To collect realistic videos of daily objects, we put together a list of human activities each of which involves an unique object class. A total number of 40 object classes are considered, see Figure 2. There are 16$\sim$20 instances for each object class. Using the camera, we record the subjects interacting with the objects without giving them detailed instructions. The objects' pose, appearance, and state may constantly change during the activity, thus giving rise to a challenging dataset for object recognition.

Depending on the complexity of the activity, the length of each video sequence varies between 150 and 500 frames, with an average of around 200 frames. Both the RGB and depth frames are calibrated using a set of checkerboard images in conjunction with the calibration tool of Burrus [6]. This also provides the homography between the two cameras, allowing us to obtain precise spatial alignment between the RGB and depth frames.

Following alignment with the RGB frames, the depth frames still contain numerous artifacts. Most notable of these is a depth "shadow" on the left edges of objects. These regions are visible from the depth camera, but not reached by the infrared projector pattern. Consequently their depth cannot be estimated, leaving shadow in the depth frame. A similar issue arises with specular and low albedo surfaces. The internal depth estimation algorithm also produces numerous fleeting noise artifacts, particularly near edges. Before extracting features for recognition, we filtered each depth frame using the cross-bilateral filter of Paris and Durand [26] to remove these artifacts.

4. Object Features

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

4.1. Object Segmentation

In order to extract features that are truly representative of objects, it is necessary to accurately segment the object in each video frame. Our segmentation pipeline consists of two steps, depth-based foreground segmentation and RGB-based skin removal, see Figure 3.

Foreground Segmentation In egocentric videos where the target object is manipulated by the subject, the foreground which consists of the hand(s) and target object is at a closer distance to the camera than to the background. This suggests that a thresholding operation on the depth frame can help segment each frame into foreground and background.

Figure 3 (left) shows the histogram of a depth frame from "lifting weight". Note the gap in the histogram that separates the frame into foreground and background. An extensive analysis of the egocentric videos in our dataset shows that the exact position of the separation gap may vary from video to video and there can be "deceptive" gaps due to the 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. This helps smooth the histogram and remove deceptive gaps. To identify the ideal threshold for segmenting the frame, we then seek the leftmost 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 Removal To further obtain the mask of the tar-

get object, we use a skin detector to detect and remove hand pixels from the foreground region. 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 to 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 [16].

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.*, banana). 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 [19].

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. Object Features

The appearance of an object may vary from frame to frame in a video. We extract HOG feature within the rectangular region containing the segmented object to characterize object appearance of a particular frame. 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. Instead of simply concatenating HOG features from RGB and depth frame, we treat them as two heterogeneous data channels. In the following section, we describe in detail how to learn a robust object model by integrating the object appearance from different frames and channels.

5. Modeling Objects in RGB-D Egocentric Videos

Given a collection of videos $\{\boldsymbol{X}^i\}$, let $\boldsymbol{X}_k^i$ denote the k^{th} channel of video $\boldsymbol{X}^i$ (RGB, depth, infrared, etc.). Assuming some appropriate kernel function $\kappa(\boldsymbol{X}_k^i, \boldsymbol{X}_k^j)$ for measuring the similarity of the two videos in the k^{th} channel, a multi-channel SVM classifier for recognizing a novel video $\boldsymbol{X}$ can be written as

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

where $y_i \in \{-1, 1\}$ is the label for $\boldsymbol{X}^i$, $\mathcal{K}(\boldsymbol{X}^i, \boldsymbol{X}) = \sum_{k=1}^{K} \mu_k \cdot \kappa(\boldsymbol{X}_k^i, \boldsymbol{X}_k)$ is a compound kernel constructed from a weighted sum of $\kappa(\boldsymbol{X}_k^i, \boldsymbol{X}_k)$, μ_k is the weight for the k^{th} channel. Note that Eq. 1 only defines a binary classifier and can be extended to multi-class classification using the one-vs-all approach.

5.1. A Novel Kernel Function

While previous methods commonly use a single frame to represent an object, we propose to use the whole set of frames of a video to cover complex variations of an object. In particular, we write $\boldsymbol{X}_k^i = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_p] \in \mathbb{R}^{d \times p}$, where $\boldsymbol{x}_m \in \mathbb{R}^d$, $1 \leq m \leq p$, is the feature vector from the m^{th} frame of $\boldsymbol{X}_k^i$. Given two sets of features $\boldsymbol{X}_k^i \in \mathbb{R}^{d \times p}$ and $\boldsymbol{X}_k^j \in \mathbb{R}^{d \times q}$, we define our kernel function 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.* [7] proposed to define the distance between two sets as the minimum distance between their affine hulls, *i.e.*,

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

$$\text{s.t.} \quad \sum_{m=1}^{p} \beta_{km}^i = 1 \quad \text{and} \quad \sum_{n=1}^{q} \beta_{kn}^j = 1 \tag{2b}$$

where $\boldsymbol{\beta}_k^i \in \mathbb{R}^p$ and $\boldsymbol{\beta}_k^j \in \mathbb{R}^q$ are the affine coefficients for $\boldsymbol{X}_k^i$ and $\boldsymbol{X}_k^j$, respectively. However, the affine hull may turn out to be an overestimate of extent of a set, especially when it comes to visual recognition problems[15].

Motivated by the recent success of sparse representation techniques [37], we introduce sparsity regularization terms on affine coefficients, *i.e.*,

$$\{\hat{\boldsymbol{\beta}}_k^i, \hat{\boldsymbol{\beta}}_k^j\} \leftarrow \arg \min_{\boldsymbol{\beta}_k^i, \boldsymbol{\beta}_k^j} \|\boldsymbol{X}_k^i \boldsymbol{\beta}_k^i - \boldsymbol{X}_k^j \boldsymbol{\beta}_k^j\|_2^2 + \lambda |\boldsymbol{\beta}_k^i|_1 + \lambda |\boldsymbol{\beta}_k^j|_1 \tag{3a}$$

$$\text{s.t.} \quad \sum_{m=1}^{p} \beta_{km}^i = 1 \quad \text{and} \quad \sum_{n=1}^{q} \beta_{kn}^j = 1 \tag{3b}$$

where $|\cdot|_1$ denotes the l_1-norm of a vector and is known for its sparsity-inducing properties. Under l_1-norm regularization, the unseen feature is restricted to be a weighted sum of just a few existing features; this sparse representation is supported by the fact that the varied appearance of an object lies in a low-dimensional subspace [3]. Compared to previous regularized affine hull models (*e.g.*, [7, 14]), our model is more concise, requires less tuning, and is jointly convex with respect to $\boldsymbol{\beta}_k^i$ and $\boldsymbol{\beta}_k^j$. The global solution can be efficiently solved by the Alternating Direction Method

of Multipliers (ADMM) algorithm [5]. See the appendix for details.

Given the minimum distance between sparse affine hulls, the kernel function is defined as

$$\kappa(\boldsymbol{X}_k^i, \boldsymbol{X}_k^j) = \exp\big(-\frac{1}{\gamma}\mathcal{D}(\boldsymbol{X}_k^i, \boldsymbol{X}_k^j)\big) \tag{4}$$

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

5.2. Integration of Heterogeneous Data Modalities

While it is possible to determine appropriate weights μ_k for different data modalities via cross-validation, this approach quickly becomes infeasible as the number of object classes increases. Inspired by the idea of Multiple Kernel Learning (MKL) [13], we thus propose to jointly learn μ_k and other SVM classifier parameters by solving

$$\min_{\mu_k, \boldsymbol{w}_k, \xi_i, b} \frac{1}{2}(\sum_{k=1}^{K}\mu_k\|\boldsymbol{w}_k\|_2)^2 + C\sum_{i=1}^{N}\xi_i \tag{5a}$$

$$\text{s.t.} \quad y_i(\sum_{k=1}^{K}\mu_k\boldsymbol{w}_k^T\phi(\boldsymbol{X}_k^i) + b) > 1 - \xi_i \tag{5b}$$

$$\sum_{k=1}^{K}\mu_k = 1 \ \text{ and } \ \mu_k \geq 0, \ \forall k \tag{5c}$$

$$\xi_i \geq 0, \ \forall i \tag{5d}$$

where $\phi(\boldsymbol{X}_k^i)$ is the mapping function satisfying $\kappa(\boldsymbol{X}_k^i, \boldsymbol{X}_k^i) = \phi(\boldsymbol{X}_k^i)^T\phi(\boldsymbol{X}_k^i)$. The algorithm from [31] is used to optimize the parameters. To perform multi-class classification, we learn class-specific parameters $\{\mu_k^c, \boldsymbol{w}_k^c, \xi_i^c, b^c\}$ for each object class c using the one-versus-all approach.

6. Experiment

In this section, we first verify the accuracy of the object segmentation algorithm. We then evaluate the proposed egocentric object recognition algorithm on three benchmark datasets.

6.1. Object Segmentation

Object segmentation serves an important role in extracting features only from target object. In this section, we evaluate the accuracy and efficiency of the proposed object segmentation algorithm. To this end, we randomly select a set of RGB-D frames from our dataset (10 RGB-D frames for each object class, 400 RGB-D frames in total) as our validation set. Groundtruth hand and object masks are obtained by means of manual annotation. We perform three 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.

	FG	*H/O-1*	*H/O-2*
precision	0.955	0.927	0.919
recall	0.981	0.959	0.942
F1 score	0.968	0.943	0.930
time (sec/frame)	0.028	0.272	0.281

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

Table 1 gives the results. For foreground segmentation, *FG* gives an F1 score of as high as 96.8% while being extremely efficient (0.028 sec/frame, or 35.7 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 skin detection more than any other factor. For example, our skin detector tends to give a low recall in environments such as a dark stairway. As part of future work, we expect to improve skin detection by explicitly modeling illumination changes.

6.2. Evaluating the Object Recognition Algorithm

In this section, we present various experimental results of object recognition. We compare the proposed algorithm to two single-frame based classification algorithms (1 and 2), and four set-based classification algorithms (3 $\sim$ 6):

1. Locality-constrained Linear Coding (LLC) [34];
2. Hierarchical Matching Pursuit (HMP) [4];
3. Discriminant Canonical Correlation (DCC) [17];
4. Manifold Discriminant Analysis (MDA) [35];
5. Affine Hull based Image Set Distance (AHISD) [7];
6. Sparse Approximated Nearest Points (SANP) [14];
7. Sparse Affine Hull Kernel (SAH).

The implementation of all these algorithms are available in the authors' website. For LLC, we followed the setup in [34], that is, we trained a codebooks with 4096 bases, and used 4×4, 2×2 and 1×1 sub-regions for SPM. For HMP, we follow the parameter settings as specified in [4]. Specifically, two-layer hierarchical matching purpuit is used. We set the number of the filters to be 3 times the filter size in the first layer and to be 1000 in the second layer. We use batch orthogonal matching pursuit to compute sparse codes. We set the sparsity level K in the two layers to be 5 and 10,

Table 2. The object recognition accuracy on the REO dataset. *set size* denotes the number of frames sampled from each video sequence. SAH-equal and SAH-learned denotes using equal weights and learned weights to integrate the RGB and depth data channels, respectively.

set size	LLC	HMP	DCC	MDA	AHISD	SANP	SAH - equal	SAH - learned
50	0.632	0.639	0.707	0.752	0.742	0.763	0.798	**0.811**
100	0.663	0.648	0.734	0.772	0.758	0.782	0.813	**0.846**
all	0.699	0.689	0.773	0.814	0.769	0.794	0.821	**0.859**

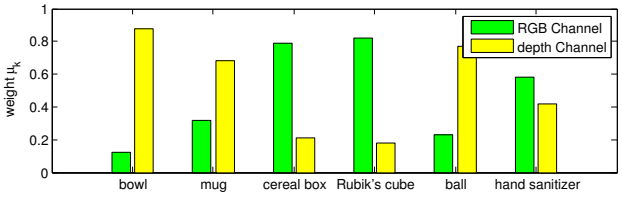

Figure 4. The RGB and depth channel weights learned for different object classes.

respectively. For DCC, the embedding dimension and subspace dimension is set to 100 and 10, respectively. 10 maximum canonical correlations are used to calculate set similarity. For MDA, the parameters are tuned for each dataset as specified in [35]. For AHISD, linear kernels are used and the subspace dimension is set by retaining enough leading eigenvectors to account for 98% data variance as in [7]. For SANP, the reconstruction weight parameter λ_1 is fixed as 0.1, and the sparsity weight parameters λ_2 and λ_3 is adaptively determined. For SAH, the sparsity penalty parameter λ is set to 10.

For algorithms $1 \sim 6$, we concatenate the features from the RGB and depth channel to represent object appearance. For single-frame based classification algorithms 1 and 2, the accuracy is calculated as the proportion of correctly classified frames in the whole testing set.

We test the above algorithms on two RGB-D object datasets, our RGB-D Egocentric Objects (REO) dataset and the University of Washington RGB-D Objects (UWRO) dataset [18]. We also perform experiments on a thermal/visible face dataset, that is, the IRIS Thermal/Visible Face (ITVF) dataset [1].

6.2.1 Results on the REO Dataset

We perform our experiments using 10-fold cross-validation. Since the performance of set-based recognition depends critically on the set size, we thus vary the set size by randomly sampling frames from each video sequence and see how recognition accuracy changes (3 different set sizes are used, *i.e.*, 50, 100, all). As for single-frame based recognition, the set size determines how many frames are extracted from each video for training/testing. To verify the importance of proper integration of heterogeneous data modalities, we test two versions of the proposed algorithm, SAH-equal, which uses equal weights for the RGB and depth channel, and SAH-learned, which uses learned weights for the RGB and depth channel. Table 2 lists the results.

As can be seen, set-based recognition algorithms outperform single-frame based recognition algorithms, and all algorithms have improved performance with increased set size. This is expected because the more frames used for modeling the object appearance, the more robust the classifier is to the object appearance variations.

Both SAH-equal and SAH-learned achieves higher recognition accuracy than other algorithms. It is also worth noting that with a set size of just 100 frames, SAH-learned is capable of achieving higher accuracy than all other algorithms with a set size of all. This can be attributed to the benefit of learning proper weights for heterogeneous data integration. Figure 4 plots the learned weights of the RGB and depth channel for a subset of object classes. As expected, for objects with rich textures (*e.g.*, cereal box and Rubik's cube), the RGB channel plays a more important role, whereas textureless objects (*e.g.*, bowl and mug) rely more on the depth channel.

6.2.2 Results on the UWRO Dataset

UWRO contains 300 daily objects organized into 51 categories. For each object, it is placed on a turntable which revolves at a constant speed. Then, RGB-D videos are recorded at 20 fps with three cameras mounted at three different angles relative to the turntable (30°, 45°, 60°). Each video contains around 250 frames, giving a total number of 250,000 RGB-D frames in the dataset. Compared to objects in REO, objects in UWRO have consistent illumination and are not occluded or modified by hands.

Following the experimental settings in [18], we perform object recognition at two levels, category-level and instance-level. Category-level recognition is to determine the category of objects (*e.g.*, mug *v.s.* apple). Instance-level recognition is to recognize whether an object is physically the same object that has previously been seen (*e.g.*, Alice's mug *v.s.* Bob's mug). For category-level recognition, we randomly leave one object instance out from each category for testing, and train models on the remaining $300 - 51 = 249$ objects. We report the accuracy averaged over 10 random train/test splits. For instance-level recognition, we train models on videos captured from 30° and 60° angle, and test them on the videos of 45° angle.

Table 3. The object recognition accuracy on the UWRO dataset. *set size* denotes the number of frames sampled from each video sequence. *cate.* denotes category-level recognition, and *inst.* denotes instance-level recognition.

set	LLC		HMP		DCC		MDA		AHISD		SANP		SAH - learned	
size	cate.	inst.	cate.	inst.	cate.	inst.	cate.	inst.	cate.	inst.	cate.	inst.	cate.	inst.
50	0.632	0.668	0.812	0.738	0.827	0.774	0.852	0.769	0.842	0.749	0.821	0.804	**0.901**	**0.847**
100	0.663	0.607	0.829	0.768	0.844	0.792	0.872	0.807	0.856	0.793	0.839	0.826	**0.913**	**0.882**
all	0.678	0.634	0.849	0.792	0.873	0.852	0.914	0.843	0.883	0.834	0.842	0.841	**0.924**	**0.908**

Table 4. The face recognition accuracy on the ITVF dataset. SAH-equal and SAH-learned denotes using equal weights and learned weights to integrate the RGB and depth data channel, respectively.

LLC	HMP	DCC	MDA	AHISD	SANP	SAH - equal	SAH - learned
0.939	0.949	0.973	0.914	0.969	0.842	0.962	**0.989**

The results are given in Table 3. As can be seen, the proposed algorithm, SAH-learned, outperforms all other algorithms in both category and instance recognition. Examination of the learned weights shows that RGB channel is more effective than depth channel for both category and instance recognition. However, depth channel is relatively more effective in category recognition, while RGB channel is relatively more effective in instance recognition. This is expected, since a particular object instance has fairly consistent texture across views, while objects in the same category can have different texture. On the other hand, shape tends to be stable across many instances of a category.

6.2.3 Results on the ITVF Dataset

ITVF is a dataset of face images acquired with thermal and visible light sensors. There are 31 subjects in this dataset. For each subject, 8 sets of face images are captured corresponding to 3 facial expressions (surprise, anger, and laughing) and 5 lighting conditions (left light on, right light on, both lights on, both lights off, dark room). In each set, thermal and visible face images are captured at 11 poses.

In our experiment, we randomly leave one set out from each subject for testing, and train models on the remaining 31×(8-1) = 217 sets. For each set, the set size is all, *i.e.*, all 11 thermal & visible frames are used. We report the accuracy averaged over 10 random train/test splits. Faces in the thermal images are localized using the bi-modal thresholding algorithm in [32]. Faces in the visible images are localized using the Viola-Jones face detector [33]. We extract HOG features from the rectangular face region to represent face appearance in the thermal and visible images. For algorithms 1∼6, thermal and visible HOG features are concatenated to form a single feature. For SAH-equal, thermal and visible features are integrated by equal weights. For SAH-learned, thermal and visible features are integrated by the MKL framework.

The results are given in Table 4. As can be seen, SAH-learned gives the highest recognition accuracy. The visible channel has an average weight of 0.736 across the 31 subjects, which is much higher than that of the thermal infrared channel (0.264). Nonetheless, the complementary effects of thermal infrared channel are quite important – when only using the visible channel, our recognition algorithm tends to fail at test cases from the *dark room* setting, and the overall recognition accuracy would drop to 0.941.

7. Conclusion

This paper focuses on the problem of recognizing objects in RGB-D egocentric videos. Our recognition method consists of two stages: 1) The target object is first segmented by exploiting the RGB and depth cues; 2) Then a novel kernel function is used to classify a set of features corresponding to the varied object appearance in the video. Using our kernel function, the similarity between two sets of features is measured by the minimum distance between their sparse affine hulls. Our kernel function also allows convenient integration of heterogeneous data modalities. We compare the proposed classification method to single-frame based methods and other set-based methods on three datasets, including two RGB-D object datasets and one thermal/visible face dataset. All experimental results clearly show that the proposed method outperforms state-of-the-art methods.

A. Solving Eq. 3

Let us denote $\boldsymbol{A} = [\boldsymbol{X}_k^i, -\boldsymbol{X}_k^j] \in \mathbb{R}^{d\times(p+q)}$, $\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{\beta}_k^i \\ \boldsymbol{\beta}_k^j\end{bmatrix} \in \mathbb{R}^{p+q}$, $\boldsymbol{e} = \begin{bmatrix}1 \\ 1\end{bmatrix} \in \mathbb{R}^2$, $\boldsymbol{C} = \begin{bmatrix}\boldsymbol{c}_1 \\ \boldsymbol{c}_2\end{bmatrix} \in \mathbb{R}^{2\times(p+q)}$, $\boldsymbol{c}_1 = [\overbrace{1,1,\cdots,1}^{p},\overbrace{0,0,\cdots,0}^{q}] \in \mathbb{R}^{1\times(p+q)}$, $\boldsymbol{c}_2 = [\overbrace{0,0,\cdots,0}^{q},\overbrace{1,1,\cdots,1}^{p}] \in \mathbb{R}^{1\times(p+q)}$, Eq. 3 can thus

be rewritten as

$$\hat{\boldsymbol{\beta}} \leftarrow \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{A}\boldsymbol{\beta}\|_2^2 + \lambda|\boldsymbol{\beta}|_1 \tag{6a}$$

$$\text{s.t. } \boldsymbol{C}\boldsymbol{\beta} = \boldsymbol{e} \tag{6b}$$

Since Eq. 6 is a convex problem, the joint convexity of Eq. 3 w.r.t. $\boldsymbol{\beta}_k^i$ and $\boldsymbol{\beta}_k^j$ is proved, and the global solution can be solved by iterative optimization procedures such as ADMM [5].

References

[1] B. Abidi. Iris thermal/visible face database. In *DOE University Research Program in Robotics under Grant DOE-DE-FG02-86NE37968*, 2007.

[2] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, 2005.

[3] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *TPAMI*, 2003.

[4] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *NIPS*. 2011.

[5] 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.*, 2011.

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

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

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

[10] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011.

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

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010.

[13] M. Gönen and E. Alpayd. Multiple kernel learning algorithms. *JMLR*, 2011.

[14] Y. Hu, A. Mian, and R. Owen. Sparse Approximated Nearest Points for Image Set Classification. In *CVPR*, 2011.

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

[16] M. Jones and J. Rehg. Statistical color models with application to skin detection. *IJCV*, 2002.

[17] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *TPAMI*, 2007.

[18] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011.

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

[20] S. Liu, S. Wang, L. Wu, and S. Jiang. Multiple feature fusion based hand-held object recognition with rgb-d data. In *ICIMCS*, 2014.

[21] X. Lv, S.-Q. Jiang, L. Herranz, and S. Wang. Rgb-d hand-held object recognition based on heterogeneous feature fusion. *JCST*, 2015.

[22] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

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

[24] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situ using in vitro training data. In *CVPR*, 2007.

[25] A. Mian, Y. Hu, R. Hartley, and R. Owens. Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning. *TIE*, 2013.

[26] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, 2009.

[27] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *Pervasive Computing*, 2004.

[28] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010.

[29] X. Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *CVPRW*, 2009.

[30] G. Shakhnarovich, J. W. Fisher III, and T. Darrell. Face recognition from long-term observation. In *ECCV*, 2002.

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

[32] L. Trujillo, G. Olague, R. Hammoud, and B. Hernandez. Automatic feature localization in thermal images for facial expression recognition. In *CVPRW*, 2005.

[33] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[34] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[35] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR*, 2009.

[36] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

[37] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 2009.

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

[39] M. Yang, P. Zhu, L. Van Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. In *FGR*, 2013.