Lv X, Jiang SQ, Herranz L *et al.* RGB-D hand-held object recognition based on heterogeneous feature fusion. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(2): 340–352 Mar. 2015. DOI 10.1007/s11390-015-1527-0

RGB-D Hand-Held Object Recognition Based on Heterogeneous Feature Fusion

Xiong Lv (吕 雄), Shu-Qiang Jiang* (蒋树强), *Senior Member, IEEE, Member, CCF, ACM*, Luis Herranz, and Shuang Wang (王 双)

Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences, Institute of Computing Technology Chinese Academy of Sciences, Beijing 100190, China

E-mail: xiong.lv@vipl.ict.ac.cn; sqjiang@ict.ac.cn; {luis.herranz, shuang.wang}@vipl.ict.ac.cn

Received December 29, 2014; revised February 22, 2015.

Abstract Object recognition has many applications in human-machine interaction and multimedia retrieval. However, due to large intra-class variability and inter-class similarity, accurate recognition relying only on RGB data is still a big challenge. Recently, with the emergence of inexpensive RGB-D devices, this challenge can be better addressed by leveraging additional depth information. A very special yet important case of object recognition is hand-held object recognition, as manipulating objects with hands is common and intuitive in human-human and human-machine interactions. In this paper, we study this problem and introduce an effective framework to address it. This framework first detects and segments the hand-held object by exploiting skeleton information combined with depth information. In the object recognition stage, this work exploits heterogeneous features extracted from different modalities and fuses them to improve the recognition accuracy. In particular, we incorporate handcrafted and deep learned features and study several multi-step fusion variants. Experimental evaluations validate the effectiveness of the proposed method.

Keywords RGB-D, hand-held object recognition, heterogeneous features fusion

1 Introduction

Objects are basic components in human interaction, and thus object recognition is a fundamental problem in computer vision, multimedia retrieval, and human-machine interaction. Although widely investigated, reliable object recognition relying only on visual information (i.e., RGB images in the paper) is still very challenging, as instances of the same object class may have very different visual appearances, while sometimes objects from different classes may look very similar[1]. In recent years, the emergence of inexpensive RGB-D devices (e.g., Kinect), provides an additional depth channel, which can be exploited to alleviate this problem. However, there are still many challenges in object recognition: 1) how to locate and segment target objects, removing them from the background, 2) how to design representative and discriminative features to describe objects, and 3) how to design robust models that exploit these features to accurately recognize the objects of interest.

Object recognition[2-4] typically involves extracting class-discriminative features and training multi-class classifiers to make predictions over unknown images. Because visual features can be greatly influenced by background, image segmentation can separate objects from background and obtain descriptors for different image regions respectively. In the case of including depth information, these descriptors can be combined with additional spatial information. While image seg-

Regular Paper

Special Section on Object Recognition

This work was supported in part by the National Basic Research 973 Program of China under Grant No. 2012CB316400, the National Natural Science Foundation of China under Grant Nos. 61322212 and 61450110446, the National High Technology Research and Development 863 Program of China under Grant No. 2014AA015202, and the Chinese Academy of Sciences Fellowships for Young International Scientists under Grant No. 2011Y1GB05. This work is also funded by Lenovo Outstanding Young Scientists Program (LOYS).

*Corresponding Author

mentation itself is a difficult and ill-posed problem, the presence of complex backgrounds makes it especially hard to solve. Feature extraction is also critical for good recognition performance. Different handcrafted features like SIFT[5], spin images[6], fast point features histogram[7], and ensembles of shape features (ESFs)[8] have been proposed to represent low-level 2D and 3D information, being robust to transformations such as rotation and scale. However, higher level cues are more difficult to be included in the design of a good feature. More recently, a new trend is to learn deep features directly from some training data instead of designing them manually. In particular, convolutional neural networks (CNNs)[9] can learn higher order properties leading to features that can describe higher level properties of the images. This higher level of abstraction in the feature can help to discriminate better between categories. In order to combine the advantages from both types of features, we use different strategies to fuse handcrafted and deep features in multiple stages.

On the other hand, advanced interaction with devices such as computers or smartphones[10-11], requires the capability to extract and interpret high-level messages from low-level multimedia signals, in order to enable the user to communicate with the device in a more natural and intelligent way. Instead of traditional inputs such as keyboards and mouse devices, we have recently witnessed the emergence of new ways to interact with computing devices. Natural language processing techniques have brought applications such as personalized intelligent assistants, capable of listening, interpreting, and communicating with the user using speech. The body can also be used to interact with the system which can track parts of the body using motion tracking techniques and recognize different body gestures. In particular, hands can be used to communicate with systems in a natural and intuitive way. We could foresee many potential scenarios involving hands to interact with computers or with other human users. For instance, a user holds a damaged toy or a worn-out sweater and shows it to the system for online shopping or product recommendation. In such scenarios, the server must recognize the object being held in hand.

Motivated by the previous discussion, in this work we focus on the special and important case of hand-held object recognition. Hand-held object recognition can exploit prior knowledge, take advantage of RGB-D devices, and use specific segmentation techniques to find hand-held objects, which effectively eliminates noises due to complex backgrounds. In this paper, we introduce the hand-held object recognition (HOR) task and a related hand-held object dataset (HOD) for evaluating these techniques, describe a segmentation method based on skeleton and depth information, and study the fusion of heterogeneous multimodal features to improve the representation ability for this problem. In this paper, we extend our preliminary work on HOR[12-13] with comprehensive experiments including additional features and feature fusion variants, and provide more detailed analysis and discussion of the results.

The HOR framework has two stages. First, the hand-held object is detected using a segmentation algorithm designed for this particular problem, leveraging depth and skeleton information. Then, in the object recognition stage, different features are extracted, including RGB-D handcrafted features and deep convolutional features, which are combined in a two-step fusion. Conventional support vector machines (SVMs) are used for the final classification. Our experiments show that combining multiple complementary features can improve the performance of individual features.

This paper is organized as follows. The next section reviews some related work. Sections 3 and 4 describe the hand-held object segmentation and recognition methods, respectively. Section 5 introduces the hand-held object dataset (HOD) and experimental results. The last section summarizes the results and describes future research.

2 Related Work

2.1 Hand-Held Object Recognition

Previous studies on hand-held object recognition are focused on first-person (egocentric) interfaces[14-15]. Thus here we distinguish between first-person and second-person interfaces. In the former, hand-held images are captured from a first-person point of view, typically with a smartphone. Note that in this case the user previews the image and can interactively control the camera and the object to optimize the coverage and the quality of the image, and minimize hand occlusion. The Small Hand-Held Object Recognition Test (SHORT)[14] focuses on this case. Another dataset including hand-held objects captured from a first-person point of view is Text-IVu[15]. Objects are similar, but the main purpose of this dataset is text recognition in hand-held objects. In both datasets, the objects appear centred and cover most of the image, which makes the detection simpler, even without segmentation.

Our work, in contrast, is designed for second-person interfaces, where the camera is located in the robot or system the user is interacting with, and consequently images are captured from a second-person point of view. The proposed HOD is designed for this scenario. The user does not visualize the image in real time, thus having less control over the captured image. A natural interaction requires certain distance, thus the objects cover only a fraction of the captured image. For this reason, segmentation is critical, and additional data such as depth and skeletons are necessary for better recognition. In contrast to SHORT and Text-IVu that only include RGB images, HOD also includes depth and skeletal data.

As both cases are very different, we consider first-person hand-held object recognition and second-person hand-held object recognition as different tasks, where specific techniques need to be developed and evaluated with the corresponding datasets.

2.2 RGB-D Object Segmentation

Most prior studies on RGB-D image and scene understanding have focused on perceptual and semantic segmentation. Silberman *et al.*[16] used depth information for bottom-up segmentation and then used context features derived from contextual relationships in the scene to perform semantic segmentation. Some work uses features based on kernel descriptors extracted on superpixels and their ancestors from a region hierarchy. Koppula *et al.*[17] studied the problem of indoor scene parsing with RGB-D data in the context of mobile robotics. Gupta *et al.*[3] used geometric contour cues for scene understanding. For some applications, such as gesture recognition, the system can focus on a particular object (i.e., hand) ignoring the rest of the scene. For instance, Chai *et al.*[4] used depth and RGB information to locate the position of the hand, and then track its trajectory. Based on this trajectory, they can model different gestures.

In contrast to general object or scene parsing methods, our approach is designed specifically for the HOR problem. In contrast to gesture recognition, we still focus on the object held on the hand, rather than the hand itself. In HOR, the hand remains mostly static. We exploit skeleton information to infer the location of the hand, and focus on that region to segment the object of interest. By exploiting prior knowledge about the HOR problem, our task-specific detection and segmentation method is more robust than general segmentation methods, having three inherent advantages: it eliminates more effectively the background; recognition is more reliable, as there is only one candidate; and the computational complexity is significantly reduced.

2.3 RGB-D Object Recognition

RGB-D combines specific techniques to exploit RGB and depth data. For the case of single object recognition (i.e., the image covers a close-up of the object of interest), Bo *et al.*[2] introduced hierarchical matching pursuit (HMP) for RGB-D data, which uses sparse coding to learn hierarchical feature representations directly from raw RGB-D data in an unsupervised way. In the case that the image contains multiple objects (e.g., an indoor scene), the different objects must be located and recognized. Typically, object candidates are first detected and then classified into known categories (or discarded as false detections). Kanezaki *et al.*[18] used a sliding window approach to detect all candidate objects. It can find the position of each object, but it is very time-consuming. Gupta *et al.*[3] generalized the deformable parts model (DPM) detector[19] to RGB-D images by computing additional features from the depth image. Alexandre[20] applied convolutional networks (CNNs)[9] to RGB-D data by using four channels (three of color and one of depth). An alternative method[21] encodes each RGB-D pixel as a triplet (horizontal disparity, height above ground, and the angle to the local surface), and then trains a CNN model for detection.

2.4 Feature Fusion

Different features often capture complementary properties of the image. For images, such properties may include local or global patterns related to color, shape, or texture. Thus, combining features may be beneficial to obtain a better representation of the image, and feature fusion has been used in computer vision to improve recognition accuracy. Many recognition tasks can benefit from combining heterogeneous features. For instance, Cimpoi *et al.*[22] combined attribute-based descriptors and the improved fisher vector (IFV) for texture recognition. Xiao *et al.*[23] combined unsupervised CNN features and a number of handcrafted ones for scene recognition.

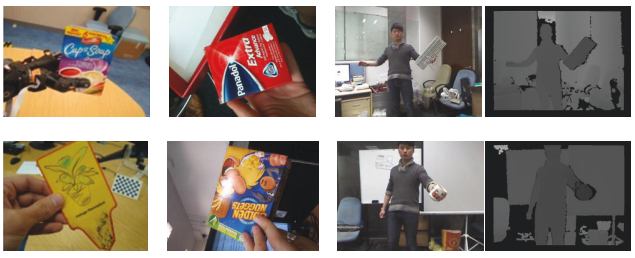
There are many ways for feature fusion[24]. A simple way is simply concatenating feature vectors together into a longer vector. However, this method may not work properly if the information conveyed by different features is not equally represented or measured. Sun

et al.[25] used partial least squares (PLS)[26] regression to fuse pairs of features. The basic idea of PLS is to find a pair of directions such that the covariance between the projections of two feature sets is maximized. Multiple kernel learning (MKL) models feature fusion in kernel space, by designing a multi-feature kernel as a linear combination of feature-specific kernels. Thus, MKL can cope with heterogeneous features via their kernels. This new kernel can be used in kernel-based classifiers such as SVMs. Xiao *et al.*[23] used a weighted sum of kernels, but the weights are learned empirically.

Most computer vision studies on feature fusion focus on RGB images. Our approach mainly focuses on exploiting multiple complementary RGB-D features, extracted from different modalities, and exploring different multi-step fusion architectures.

3 Hand-Held Object Segmentation

Color and depth images are captured by an RGB-D camera, whose API also provides skeletal data. Prior to segmentation, depth map is preprocessed to filter noise and recover part of the missing depth data. Depth is interpolated in pixels where more than half of their eight neighbors have valid depth. As part of the skeletal data, an estimation of the location of the hand is provided. Based on the assumption that hand-held objects are connected to the hand, an initial segmentation is obtained. The hand position will typically fall in the object region (see Fig.1(c)).



(a) (b) (c)

Fig.1. Hand-held recognition datasets. (a) Text-IVu. (b) SHORT. (c) HOD (RGB and depth). Images in Text-IVu and SHORT are captured from a first-person point of view and do not include depth. Images in HOD are captured from a second-person point of view, and include RGB, depth, and skeleton information.

Using the hand position as reference and initial seed, we obtain the object mask using a region-growing algorithm (see Fig.2). This algorithm examines the neighbors of the points in the seed set and includes them in the seed set when they are at a similar depth as the hand.

Input: depth image $d(\boldsymbol{x})$, initial hand location $\boldsymbol{x}_{\mathrm{H}}$ (from skeletal data); threshold T
Output: object segmentation mask $S(\boldsymbol{x})$

```
// x = (x, y)
// N(x) =8-connected neighbors of x
S(x) ← 0, seen(x) ← false for all x
queue.push(x_H)
d_H ← (1/9) (d(x_H) + Σ_{z∈N(x_H)} d(z))
repeat
    x ← queue.pop()
    if not seen(x) and |d(x) − d_H| ⩽ T then
        S(x) ← 1
        for each z in N(x) do
            if not seen(z) then
                queue.push(z)
            end if
        end for
        seen(x) ← true
    end if
until queue.empty()
return S(x)
```

Fig.2. Hand-held object segmentation algorithm.

The algorithm is simple yet robust. Compared with vision-based segmentation methods, the proposed method locates the target object more accurately based on the *hand-held* assumption. The objects of interest are better recognized using depth information than those using visual appearances, especially when background is cluttered. The segmentation pipeline is shown in Fig.3 and some results are shown in Fig.4. Although we also study more complex variants (e.g., better estimation of hand position depending on skeletal relations, skin detection), in practice we observe that the performance of this simple method is very similar and satisfactory enough.

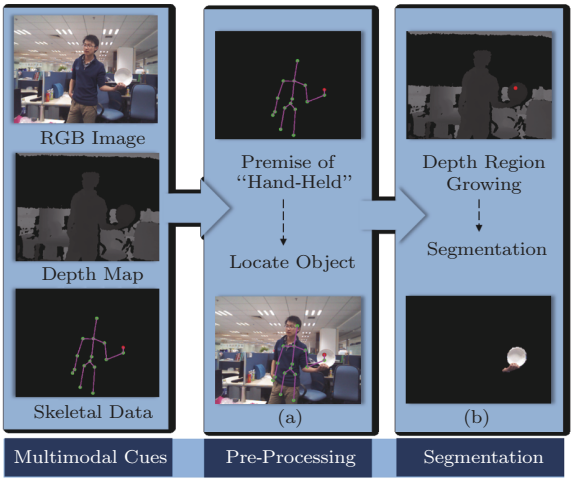


Fig.3. Segmentation pipeline.

Fig.4. Segmentation results.

4 Hand-Held Object Recognition

The recognition stage includes feature extraction, feature fusion and classification. In the following, we give more details about the different features and methods we use to combine them. For classification, we use conventional SVMs. Fig.5 shows the recognition pipeline (one of the variants).

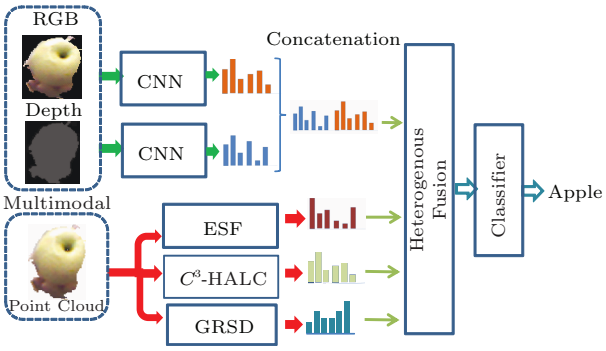


Fig.5. Example of hand-held object recognition pipeline.

4.1 Feature Extraction

We consider four different types of features in this paper (see Fig.5): ESF[27], C^3-HALC (Circular Color Cubic Higher-Order Local Auto-Correlation)[28], GRSD (Global Radius-Based Surface Descriptor)[30] and CNNs[9]. ESF, C^3-HALC, and GRSD are handcrafted features while CNNs features are automatically learned from a set of training data. These features also are extracted from different modalities. While CNNs features are extracted directly from either RGB or depth images, ESF, C^3-HALC, and GRSD are extracted from a point cloud representation, indirectly obtained from the RGB and the depth data. ESF and GRSD are mostly focused on shape properties while C^3-HALC is useful to represent color and texture. CNNs features in general can capture higher-level properties of the image. Thus, the different features cover multiple modalities and represent complementary aspects of the image (see Table 1).

Table 1. Different Features and Their Characteristics

Notation	Modality	Feature	Dimension	Properties
C-CNN	RGB	CNN	4 096	Color, shape, texture, high-level
D-CNN	Depth	CNN	4 096	Shape, high-level
ESF	Point cloud	ESF	640	Shape
C3	Point cloud	C^3-HALC	117	Color, texture
GRSD	Point cloud	GRSD	20	Shape

4.1.1 *Handcrafted Features*

The Ensemble Shape Functions (ESF)[27] descriptor is an ensemble of 10 64-bin histograms of shape functions, describing the characteristic properties of the point cloud. The 10 shape functions include three angle functions, three area functions, and one distance-ratio function. Thanks to the availability of the depth channel, a point cloud can be reconstructed easily. The ESF descriptor is a global shape descriptor based on three distinct shape functions as above. It can be efficiently computed directly from the point cloud without requiring preprocessing steps such as hole filling, smoothing or surface normal calculation. Furthermore, ESF can gracefully handle data errors such as outliers, holes, noise and coarse object boundaries.

C^3-HALC[28] descriptor is a high-dimensional vector that measures the correlation of colors between two neighboring voxels. It combines the multiple RGB values of neighboring voxels in a local $3 \times 3 \times 3$ grid computed in a voxel grid of arbitrary size.

GRSD[30] was proposed to model everyday objects for mobile manipulation applications. Based on a voxel representation, each voxel is classified into one of a set of predefined geometric labels (plane, cylinder, edge, rim, or sphere), based on the radius-based surface descriptor (RSD) of the voxel. Then GRSD is computed from these local RSDs.

4.1.2 *Convolutional Neural Networks*

Recently, CNNs have been used successfully in many computer vision tasks, including (RGB) object recognition[9] and recent RGB-D object recognition[9,20-21]. In order to use CNNs on HOD, we use the binary mask map which is computed during segmentation to get the object mask in both RGB and

depth images. As the input to CNNs is a rectangular image and the segmented object has irregular shape, we pad the empty pixels in the bounding box with zeros (see Fig.6). We separately extract a CNNs feature from both the RGB image and the depth image using the Caffe implementation[29]. The architecture has eight layers. The first five layers are convolutional layers, while the sixth and the seventh layers are fully connected layers and the final layer is a softmax classifier. We discard the classifier and use the output of the seventh layer as a feature (4 096-dimensional).

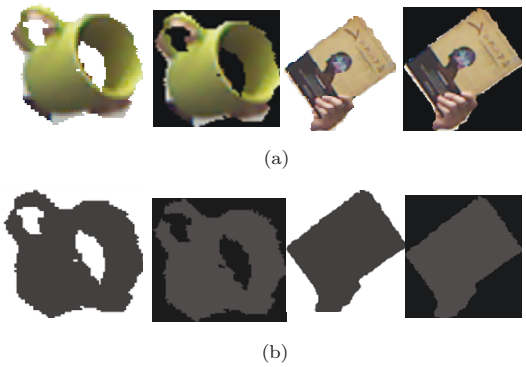


Fig.6. Empty pixels are filled with zeros (black pixels) in both (a) RGB and (b) depth images.

4.2 Feature Fusion

Depending on the task and the features, different fusion methods lead to very different classification performances. There are two main types of fusion schemes, depending on the stage information being combined[31]. Late fusion is often achieved using multi-classifier or multi-expert combination strategies. Early fusion is at the feature level, which plays an important role in the process of data fusion. Fusing at feature level tries to select the most discriminatory information from the original feature sets involved in fusion. In most cases, features extracted from different modalities are highly correlated. Thus, simple normalization or weighting is often not enough to fuse features effectively.

We consider two operations to fuse data: concatenation and multiple kernel learning (MKL), which we sometimes combine in multiple steps. For convenience, we denote these two feature fusion operations as $[A, B]$ and $A+B$, respectively (see Table 2 for examples). The former consists of stacking the feature vectors of the different features in a longer vector, and uses it as a single feature. However, concatenating heterogeneous features from different modalities must be done carefully, since the information conveyed by different features is not equally represented or measured[24]. MKL, in contrast, combines the features as the combination of kernels. Each feature is related with one kernel, and MKL tries to combine the advantage of each of them. It is particularly suitable to deal with heterogeneous data.

Table 2. Notation for the Multi-Step Fusion Variants

	Step 1	Step 2
Example 1	[C-CNN, D-CNN]	[C-CNN, D-CNN]+
	[ESF, C3]	[ESF, C3]
Example 2	[C-CNN, D-CNN]	[C-CNN, D-CNN]+
	ESF	ESF+C3
	C3	
Example 3	[C-CNN, D-CNN, ESF, C3]	

Note: $[A, B]$: concatenation of features A and B; $A + B$: MKL fusion of features A and B.

MKL[24] usually considers two types of kernel functions. Local kernel functions are suitable in learning, but can lead to overfitting. Global kernel, in contrast, can lead to better generalization. In order to combine both advantages, we use a weighted combination of local and global kernel functions. In particular, we use a polynomial kernel and a Gaussian kernel, which are typical global and local kernel function, respectively. The combined kernel is

$$K(x_i, x_j) = \sum_{z=1}^{N} \lambda_z^{(G)} K_z^{(G)}(x_i, x_j) + \sum_{z=1}^{N} \lambda_z^{(P)} K_z^{(P)}(x_i, x_j)$$

$$s.t.\, \lambda_z \geqslant 0, \sum_{z=1}^{N} \lambda_z^{(G)} + \sum_{z=1}^{N} \lambda_z^{(P)} = 1,$$

where N is the total number of features, $K_z^{(G)}$ and $\lambda_z^{(G)}$ are the candidate Gaussian kernel and the corresponding weight for feature z respectively, and $K_z^{(P)}$ and $\lambda_z^{(P)}$ are the corresponding polynomial kernel and weight respectively.

5 Experiments

5.1 Hand-Held Object Dataset

To the best of our knowledge, there is no suitable dataset for this new HOR task. For this reason, we in-

troduce the hand-held object dataset (HOD), designed specifically to evaluate HOR.

HOD contains a total of 12 800 video frames recorded with a Kinect camera, placed at about 1.5 m above the ground. For each frame, we capture one RGB image, one depth map, and the skeletal data for the human (obtained using the Kinect API). The dataset includes 16 common object categories (see Fig.7), each of which has four instances (a total of 64 object instances). For more reliable depth and skeletal data, the target human body should stay within the distance recommended by Kinect (1∼3 meters).

The data was collected in two different scenes (i.e., locations) and by two different users (see Fig.4). Each instance is used four times, corresponding to the four combinations of users and scenes. For each combination of user, location, and instance, we capture 50 frames (640×480 pixels, 30 fps, subsampling ratio 1/30), and thus we collected a total of 200 frames per instance and 800 frames per object category. During the data collection process, the pose and the distance of the hand-held objects are varied, and accordingly the dataset covers multiple views of each object (see Fig.4).

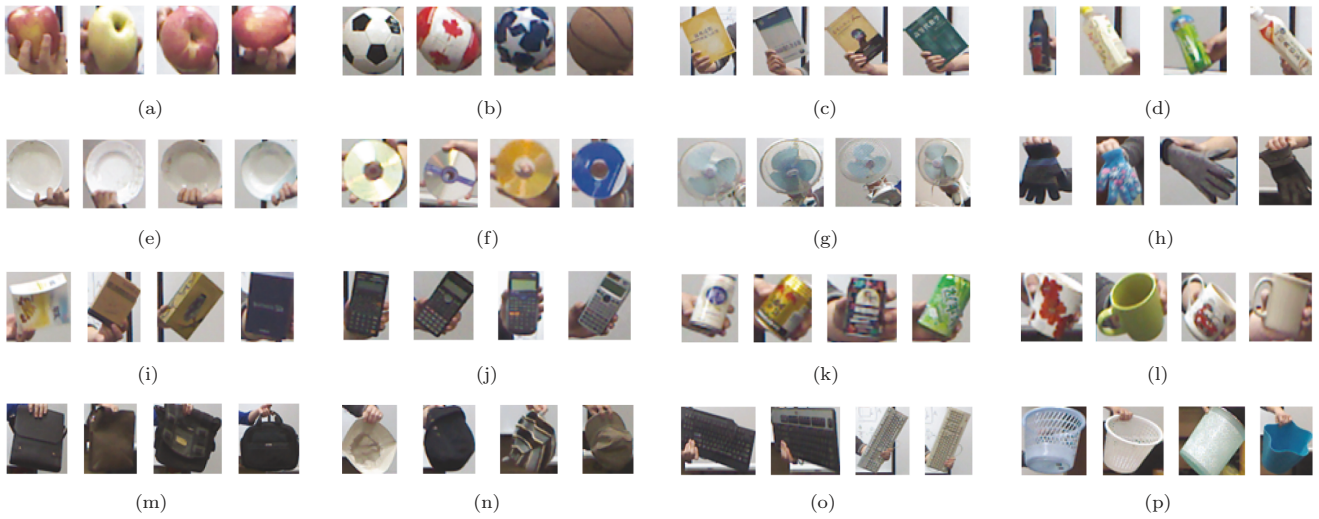


Fig. 7. Hand-held object dataset. We show the four instances of each of the 16 object classes. (a) Apple. (b) Ball. (c) Book. (d) Bottle. (e) Dish. (f) Disk. (g) Fan. (h) Glove. (i) Box. (j) Calculator. (k) Can. (l) Cup. (m) Handbag. (n) Hat. (o) Keyboard. (p) Trashcan.

5.2 Experimental Setup

For each frame in the dataset, we first segment the hand-held object. Then we obtain the point cloud and extract ESF (sampling point set to 20 000), C^3-HALC (voxel size 0.01 m) and GRSD (search radius for the surface normal set to 0.02 m) features. From RGB and depth images, we also extract CNNs features. Classifiers are implemented using libSVM ($\lambda = 0.000\,5$). For MKL, we use Gaussian kernels with different sizes (0.5, 1, 2, 5, 7, 10, 12, 17, and 20), polynomial kernels of different degrees (1, 2, and 3) and $C = 1\,000$.

In order to reduce the possible dependency of the results on the particular person and scene involved, we use different combinations of scene and person for training and testing. For example, if we use the data collected in scene 1 by user 1 as the training set, then we use the data collected in scene 2 by user 2 as the test set. Thus we have four combinations and we report the average results.

5.3 Seen and Unseen Instances

In contrast to other object recognition datasets, HOD has two different levels: category and instance. The variability between frames collected from the same instance is lower than the variability within the same category but across different instances (see Fig.8). For this reason, it is easier to recognize a frame from an instance that has been *seen* in the training set. However, even if the same object is seen, but no frame from a particular instance is included in the training set (i.e., the instance is *unseen*), it is more difficult to recognize that particular instance than a seen instance.

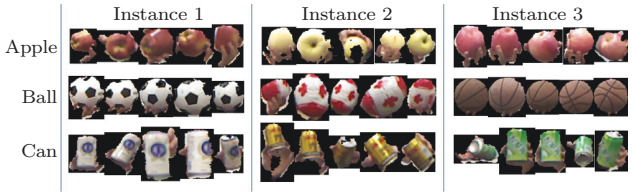


Fig.8. Different poses and segmentation results for different objects and instances.

For this reason, we consider two different evaluation settings, depending on how the training and test sets are organized:

• *Seen*: the training and the test sets both contain frames from all the instances;

• *Unseen*: the training set contains frames from the instances 1, 2, and 3; the instance 4 is used in the test set.

5.4 Segmented vs Unsegmented

We first evaluate the influence of the segmentation in the recognition performance. Table 3 shows the results for both unsegmented and segmented regions. These results show that the segmentation stage is crucial for good recognition performance. We also observe the CNNs features over RGB images that obtain the best performance over the handcrafted features.

Table 3. Accuracy (%) with and Without Segmentation (Seen Setting)

Feature	ESF	C3	GRSD	C-CNN	D-CNN
Unsegmented	7.00	11.20	13.00	12.81	12.88
Segmented	54.66	64.45	32.28	80.86	50.20

5.5 Deep Features

We compare the performance of different CNNs variants for both the seen and the unseen settings in Table 4. Depth information achieves a remarkable accuracy, but far from the much better performance using RGB data, which is much richer information. Nevertheless, the drop in the accuracy for the unseen setting, which is more challenging, is more significant for RGB data. We evaluate the impact of the two fully connected layers over RGB data by comparing classifiers trained with the output of the layer 5 and the layer 7 respectively. This deeper architecture improves the recognition accuracy around 2% for the seen setting, but more interestingly, around 4% for the unseen setting.

Table 4. Test Average Accuracy (%) for Different RGB CNNs and Depth CNN Combinations with 7 Layers

Setting	C-CNN	C-CNN (Layer 5)	D-CNN
Seen	80.86	78.50	50.20
Unseen	55.56	54.67	40.50

In contrast to handcrafted features, CNNs leverage real visual data to learn filters that are capable of detecting intrinsic properties of natural images. In order to illustrate how the CNNs used in the experiments can find class-specific patterns, we show the response of several filters of the last convolutional layer (i.e., layer 5) in Fig.9. Note that we only show 10 representative filters out of 256 filters in layer 5. But considering all the responses in layer 5, we observe that the number of filters with significant response is much higher for RGB images than for depth images. As depth images mainly contain contour information while RGB images also include richer texture and color information, many filters that can detect texture and color patterns do not show any response to depth data. Thus, the classifier can exploit these richer filter responses to achieve better performance with RGB data. Note however that depth information (see Fig.9(b)) leads to less noisy and more invariant responses.

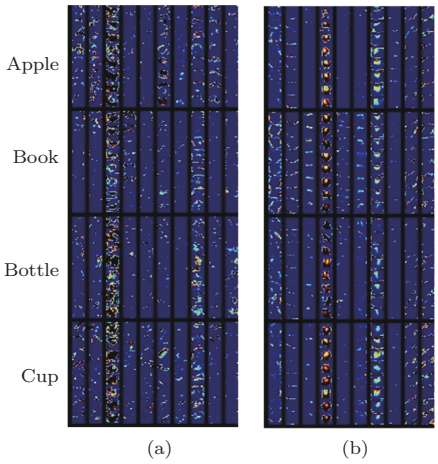


Fig.9. Filter responses at layer 5 of the CNN. (a) RGB. (b) Depth. Each row corresponds to one instance (showing 4 instances per category and 2 images per instance) and each column to a filter.

5.6 Feature Fusion

We evaluate different fusion methods by combining concatenation and MKL over different combinations of features. The results are shown in Table 5, where we

use the notation described in Table 2, and separate the methods in different groups for easier comparison. Although C-CNN has already had a performance significantly better than the other features, combining it with them can improve the performance, especially in difficult cases where including other modalities, such as depth and point cloud, can be helpful.

One first observation is that, as expected, evaluation in the unseen setting is more difficult than in the seen setting, and consequently the accuracy drops in the unseen setting when compared with the seen setting. However, different methods have different drops in accuracy, which shows that some of them may generalize better to unseen instances while others may perform better in recognizing the same instance under different capturing conditions. This can be seen as some type of overfitting to the particular seen instances but still robust to variations in pose, illumination, rotations, and tolerant to problems during segmentation.

Table 5. Results for Different Feature Fusion Variants

		Feature Fusion Variant	Accuracy (%)	
			Seen	Unseen
Multiple Kernel Learning	More Complex	[C-CNN, D-CNN, C3, ESF, GRSD]+ [C-CNN, D-CNN]+C3+ESF+GRSD	86.71	73.19
		[C-CNN, D-CNN, C3, ESF]+ [C-CNN, D-CNN]+C3+ESF	85.10	**75.31**
		[C3, ESF, GRSD]+C3+ESF+GRSD	82.19	67.63
		[C3, ESF]+C3+ESF	80.70	60.30
	[C-CNN, D-CNN]	[C-CNN, D-CNN]+ESF+C3+GRSD	86.53	72.56
		[C-CNN, D-CNN]+C3+ESF	86.60	72.06
		[C-CNN, D-CNN]+ESF	81.92	71.56
		[C-CNN, D-CNN]+C-CNN+D-CNN	78.71	61.56
	Simple	C-CNN+D-CNN	49.65	35.75
		C-CNN+D-CNN+C3+ESF+GRSD	**88.59**	73.31
		C-CNN+D-CNN+C3+ESF	88.31	72.75
		C3+ESF+GRSD	81.50	66.89
Concatenation		[C-CNN, D-CNN, C3, ESF, GRSD]	83.33	69.50
		[C-CNN, D-CNN, C3, ESF]	82.85	69.63
		[C-CNN, D-CNN, C3]	79.22	59.56
		[C-CNN, D-CNN, ESF]	82.67	68.86
		[C-CNN, D-CNN, GRDS]	78.79	61.19
		[C-CNN, D-CNN]	82.54	61.75
		[C3, ESF, GRSD]	70.05	58.56
		[C3, ESF]	68.85	51.81
Independent		C-CNN	80.86	55.56
		D-CNN	50.20	40.50
		C3	64.45	22.38
		ESF	54.66	50.25
		GRSD	32.28	27.31

Note: $[A, B]$: concatenation of features A and B; $A + B$: MKL fusion of features A and B.

Focusing first on handcrafted features, we observe that their individual performance varies significantly. In particular, C^3-HALC has the best performance for the seen setting and the worst one for the unseen setting, with a drop of more than 40% in accuracy. This suggests that this feature is more suitable to reidentify already seen instances, but has very poor generalization capability. On the contrary, ESF has a remarkable performance in both the seen and the unseen setting, with a very small difference of only 4.4%. This suggests much better generalization capability. Combining the three features using concatenation or MKL improves the recognition performance over the best single feature in both settings. MKL leads to better accuracy.

On the other hand, deep features lead to improved performance, in particular using RGB data. For the seen setting, C-CNN improves the best handcrafted feature by more than 16%, while for the unseen setting, the performance is less than 3% better than ESF. Combining both deep features leads to better performance in the case of concatenation, while, unexpectedly, leading to a much worse performance in the case of MKL.

However, the best performances are achieved when we combine both deep and handcrafted features. In particular, concatenating the five features leads to seen and unseen performances of 83.33% and 69.5%, outperformed by using MKL over the five features (88.59% and 73.31%, respectively).

We also explore multi-step fusion architectures. First, we consider the concatenation of both deep features [C-CNN, D-CNN] as a new feature that is combined with other features using MKL. Combining it with handcrafted features leads to better performance, but combining it again with single deep features leads to worse performance. The second architecture concatenates a number of features to an extended new feature and combines it again with the same individual features using MKL. This architecture leads to the best performance in both settings using only handcrafted features. A variation with deep features leads to the best performance over the unseen setting of 75.31%, but still cannot outperform the single-step MKL combination in the seen setting.

5.7 Accuracy per Category

Previous subsections report the accuracy averaged over all the categories. In this subsection, we analyze the accuracy per category. Fig.10 compares some of the results.

Comparing single features, the deep feature C-CNN outperforms handcrafted ESF in all the categories for the seen setting, but for the unseen setting, the best one highly depends on the particular category. In particular, ESF seems to be much better than C-CNN in difficult categories such as handbag, glove, and apple. Interestingly, the gain of combining features is much higher in the unseen setting than in the seen one (e.g., around 19% and 23% for concatenation and MKL, respectively, compared with around 3% and 8%). This suggests that, although in both cases, feature fusion improves the accuracy, combining heterogeneous features that perform differently in different categories is more

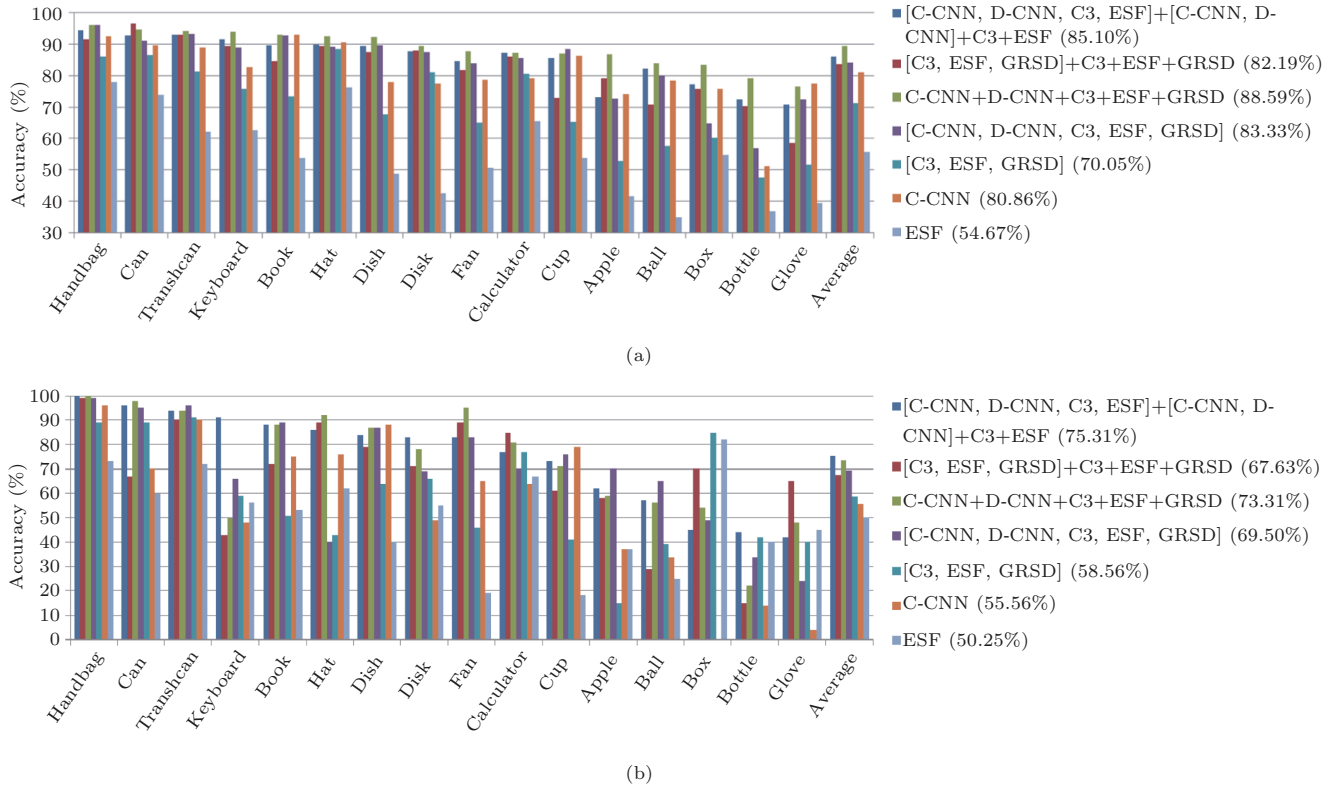


Fig.10. Test accuracy per category. (a) Seen. (b) Unseen. Classes are ordered by decreasing accuracy according to the best method.

beneficial. And in particular, MKL achieves better results than concatenation.

Fig.11 shows the confusion matrices for the best performing method in each setting. For the seen case, the accuracy is very high and errors are relatively randomly distributed. These errors sometimes may result from unsatisfactory segmentations. More interesting is the unseen case, in which we observe certain misclassification patterns, which suggest certain overfitting to some properties and problems to be generalized in the particular dataset. For example, the test instance of can is often misclassified as bottle or cup, and the box one as a calculator or a can.

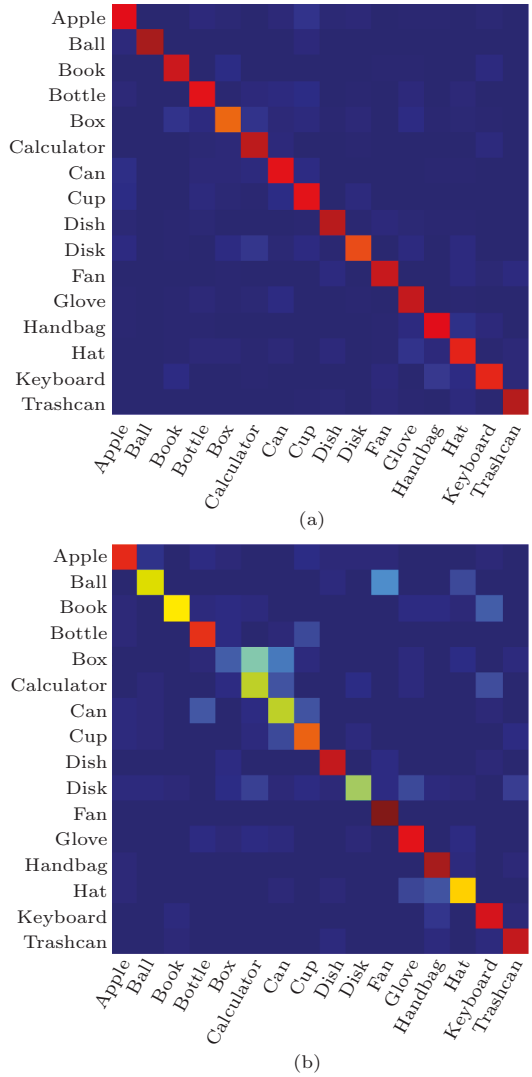


Fig.11. Confusion matrices of the best performing methods. (a) Seen (C-CNN+D-CNN+C3+ESF+GRSD). (b) Unseen ([C-CNN, D-CNN, C3, ESF]+[C-CNN, D-CNN]+C3+ESF).

6 Conclusions

Interacting with people often involves object being manipulated, and these objects are often the main topic of such interaction or at least provide important contextual information. In this paper, we studied the problem of hand-held object segmentation and recognition which is fundamental to enable this type of capability in human-computer interaction.

We described a framework to detect and recognize objects held in hand. We exploited depth and skeletal information to detect and segment the target object, even in complex and cluttered backgrounds, without complex and less reliable scene parsing methods. In order to accurately recognize hand-held objects, we exploited deep features implemented using CNNs. Deep features extracted from RGB images are the best performing features, but including other modalities via multi-feature fusion also leads to improved performance. Multi-step fusion of multiple features can improve the performance over simpler one in some cases. However, we found that MKL fusion of the five features is a simple yet good strategy which provides excellent performance in both cases, which makes it very suitable in practice.

As this particular type of interaction involves recognizing known and unknown instances of known objects, it is important to distinguish between the object category and the particular object instance level. For that reason, we evaluated separately seen and unseen instances. Both cases have different complexity, and we found that some methods may accurately reidentify an instance already seen, but are not so capable to recognize the category when the instance is different. Other features capturing more abstract properties, such as CNNs, are capable of recognizing the category even when the particular instance was unknown to the system.

References

[1] Li L, Jiang S, Huang Q. Learning hierarchical semantic description via mixed-norm regularization for image understanding. *IEEE Transactions on Multimedia*, 2012, 14(5):1401–1413.

[2] Bo L, Ren X, Fox D. Unsupervised feature learning for RGB-D based object recognition. In *Springer Tracts in Advanced Robotics 88*, Desai J P, Dudek G, Khatib O, Kumar V (eds.), Springer, pp.387–402.

[3] Gupta S, Arbeláez P, Girshick R, Malik J. Indoor scene understanding with RGB-D images: Bottom-

up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision*, 2014. http://link.springer.com/article/10.1007/s11263-014-0777-6#, Feb. 2015

[4] Chai X, Li G, Lin Y, Xu Z, Tang Y, Chen X, Zhou M. Sign language recognition and translation with Kinect. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, April 2013.

[5] Lowe D G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, 60(2):91–110.

[6] Johnson A E, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, 21(5):433–449.

[7] Morisset B, Rusu R B, Sundaresan A, Hauser K, Agrawal M, Latombe J C, Beetz M. Leaving flatland: Toward real-time 3D navigation. In *Proc. IEEE International Conference on Robotics and Automation*, May 2009, pp.3786–3793.

[8] Hinterstoisser S, Holzer S, Cagniart C *et al.* Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proc. IEEE International Conference on Computer Vision* (*ICCV*), Nov. 2011, pp.858–865.

[9] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems*, Dec. 2012.

[10] Zhang Z, Zhou C, Xin B,Wang Y, Gao W. An interactive system of stereoscopic video conversion. In *Proc. the 20th ACM International Conference on Multimedia*, Oct. 29–Nov. 2, 2012, pp.149–158.

[11] Izadi S, Kim D, Hilliges O *et al.* KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. the 24th Annual ACM Symposium on User Interface Software and Technology*, Nov. 2011, pp.559–568.

[12] Liu S, Wang S, Wu L, Jiang S. Multiple feature fusion based hand-held object recognition with RGB-D data. In *Proc. International Conference on Internet Multimedia Computing and Service*, July 2014, p.303.

[13] Lv X, Wang S, Li X, Jiang S. Combining heterogenous features for 3D handheld object recognition. In *Proc. SPIE Optoelectronic Imaging and Multimedia Technology III*, Oct. 2014.

[14] Rivera-Rubio J, Idrees S, Alexiou I, Hadjilucas L, Bharath A. Small hand-held object recognition test (short). In *Proc. the 2014 IEEE Winter Conference on Applications of Computer Vision* (*WACV*), March 2014, pp.524–531.

[15] Beck C, Broun A, Mirmehdi M, Pipe A, Melhuish C. Text line aggregation. In *Proc. International Conference on Pattern Recognition Applications and Methods* (*ICPRAM*), Mar. 2014, pp.393–401.

[16] Silberman N, Hoiem D, Kohli P, Fergus R. Indoor segmentation and support inference from RGBD images. In *Proc. the 12th ECCV*, Part 5, Oct. 2012, pp.746-760

[17] Koppula H S, Anand A, Joachims T, Saxena A. Semantic labeling of 3D point clouds for indoor scenes. In *Proc. the 25th Neural Information Processing Systems*, Dec. 2011.

[18] Kanezaki A, Suzuki T, Harada T, Kuniyoshi Y. Fast object detection for robots in a cluttered indoor environment using integral 3D feature table. In *Proc. the 2011 IEEE International Conference on Robotics and Automation* (*ICRA*), May 2011, pp.4026–4033.

[19] Felzenszwalb P F, Girshick R B, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, 32(9):1627–1645.

[20] Alexandre L A. 3D object recognition using convolutional neural networks with transfer learning between input channels. In *Proc. the 13th International Conference on Intelligent Autonomous Systems*, July 2014.

[21] Gupta S, Girshick R, Arbeláez P, Malik J. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. the 13th ECCV*, Part 7, Sept. 2014, pp.345–360.

[22] Cimpoi M, Maji S, Kokkinos I, Mohamed S, Vedaldi A. Describing textures in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp.3606-3613

[23] Xiao J, Ehinger K, Hays J, Torralba A, Oliva A. SUN database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 2014. http://link.springer.com/article/10.1007/s11263-014-0748-y, Feb. 2015.

[24] Fu Y, Cao L, Guo G, Huang T S. Multiple feature fusion by subspace learning. In *Proc. the 2008 International Conference on Content-Based Image and Video Retrieval*, July 2008, pp.127–134.

[25] Sun Q S, Jin Z, Heng P A, Xia D S. A novel feature fusion method based on partial least squares regression. In *Proc. the 3rd International Conference on Advances in Pattern Recognition*, Part 1, Aug. 2005, pp.268–277.

[26] Barker M, Rayens W. Partial least squares for discrimination. *Journal of Chemometrics*, 2003, 17(3):166–173.

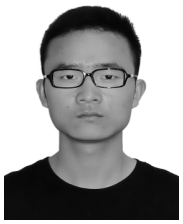
[27] Wohlkinger W, Vincze M. Ensemble of shape functions for 3D object classification. In *Proc. the 2011 IEEE International Conference on Robotics and Biomimetics* (*ROBIO*), Dec. 2011, pp.2987–2992.

[28] Kanezaki A, Marton Z C, Pangercic D, Harada T, Kuniyoshi Y, Beetz M. Voxelized shape and color histograms for RGBD. In *Proc. IROS Workshop on Active Semantic Perception and Object Search in the Real World*, Sept. 2011.

[29] Jia Y, Shelhamer Evan, Donahue J *et al.* Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. http://arxiv.org/abs/1408.5093, Feb. 2015.

[30] Marton Z C, Pangercic D, Rusu R B, Holzbach A, Beetz M. Hierarchical object geometric categorization and appearance classification for mobile manipulation. In *Proc. the 10th IEEE-RAS International Conference on Humanoid Robots*, Dec. 2010, pp.365-370

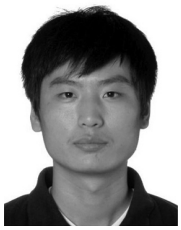
[31] Snoek C G, Worring M, Smeulders A W. Early versus late fusion in semantic video analysis. In *Proc. the 13th Annual ACM International Conference on Multimedia*, Nov. 2005, pp.399–402.



Xiong Lv received his B.S. degree in computer science and engineering from Beihang University, Beijing, in 2013. He is currently a graduate student of the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His research interests include image understanding, human-system interaction with image, 2D and 3D object recognition.

Shu-Qiang Jiang got his Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2006. Currently, he is a professor with ICT, CAS, and is also with the Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences, Beijing. His research interests include multimedia processing and semantic understanding, pattern recognition, and computer vision. He has authored or coauthored more than 100 papers on the related research topics. Dr. Jiang was supported by the New-Star program of Science and Technology of Beijing Metropolis in 2008. He won the Lu Jiaxi Young Talent Award from Chinese Academy of Sciences in 2012, and the CCF Award of Science and Technology in 2012. He is a senior member of IEEE and a member of CCF and ACM. Prof. Jiang is the executive committee member of ACM SIGMM China chapter. He has been serving as a guest editor of the special issues for PR and MTA. He is the program chair of ICIMCS2010, the special session chair of PCM2008, ICIMCS2012, the area chair of PCIVT2011, the publicity chair of PCM2011, and the proceedings chair of MMSP2011. He has also served as a TPC member for more than 20 well-known conferences, including ACM Multimedia, CVPR, ICCV, ICME, ICIP, and PCM.

Luis Herranz received his Telecommunication Engineer degree from the Universidad Politécnica de Madrid, Spain in 2003, and his Ph.D. degree in computer science and telecommunication from the Universidad Autónoma de Madrid, Spain, in 2010. From 2003 to 2010, he was with the Escuela Politécnica Superior of the Universidad Autónoma de Madrid as a researcher and teaching assistant. From 2010 to 2011, he was with Mitsubishi Electric R&D Centre Europe, United Kingdom. He is currently a postdoctoral research fellow with the Institute of Computing Technology of the Chinese Academy of Sciences, Beijing. His research interests include image understanding, video abstraction, and multimedia indexing and retrieval.



Shuang Wang received his B.S. degree in software engineering from Dalian University of Technology, China in 2011 and M.S. degree in technology of computer application from the Institute of Computing Technology of the Chinese Academy of Sciences, Beijing, in 2014. His research interests include image understanding, image retrieval, 2D and 3D object recognition.