Learning Hierarchical Representation with Sparsity for RGB-D Object Recognition

Kuan-Ting Yu, Shih-Huan Tseng, and Li-Chen Fu, *Fellow, IEEE*

***Abstract*—RGB-D sensor has gained its popularity in the study of object recognition for its low cost as well as its capability to provide synchronized RGB and depth images. Thus, researchers have proposed new methods to extract features from RGB-D data. On the other hand, learning-based feature representation is a promising approach for 2D image classification. By exploiting sparsity in 2D image signals, we can learn image representation instead of using hand-crafted local descriptors like SIFT or HoG. This framework inspired us to learn features from RGB-D data. Our work focuses on two goals. First, we propose a novel Hierarchical Sparse Shape Descriptor (HSSD) to form learning-based representation for 3D shapes. To achieve this, we analyze several 3D feature extraction techniques and propose a unified view of them. Then, we learn hierarchical shape representation with sparse coding, max pooling and local grouping. Second, we investigate whether RGB and depth information should be fused at lower level or higher level. Experimental results show that, first, our HSSD algorithm can learn shape dictionary and provide shape cues in addition to the 2D cues. Using the proposed HSSD algorithm achieves 84% accuracy on a household RGB-D object dataset and outperforms a widely used VFH shape feature by 13%. Second, fusing RGB-D information at lower level does not improve recognition performance.**

I. Introduction

Object recognition is an important capability for robots to serve in the environment. In many tasks, such as object search, and object manipulation, robots must be able to detect and recognize objects. Moreover, robots are required to have very high accuracy or otherwise they would be very difficult to serve correctly.

Thanks to the recent development of RGB-D sensors with low cost and high accuracy. The synchronized color image and depth image it captures allow engineers to extract more diverse information from both channels and compensate each other. The former is known to have rich information and the latter can provide physical size, shape and distance which are very challenging to estimate in normal color images. Despite the rich information we can gain from RGB-D camera, the recognition ability of robots is still imperfect. One of the key elements in recognition procedure is data representation. Researchers have proposed features for particular recognition tasks, *e.g.* SIFT [1] for object categorization, and HoG [2] for human detection. In order to make features automatically adjust to specific task, researchers employ deep learning framework to learn hierarchical representation from data. One key component in the success of representation learning is sparse coding.

Signals captured from nature are actually sparse. Although the number of all possible signal patterns is large, not all patterns appear, or, more generally appear with the same probability. For object recognition task, the classifier only need to know features of low dimension that capture the essence of a signal and leave unimportant information and noise aside. SIFT and HoG can be seen as hand-crafted methods to form a sparse representation. They capture gradient information from low level. However, we expect that a recognition algorithm can automatically extract essential cues for a given dataset. Also, as we build up a hierarchical recognition model, feature extraction of the second layer or above must be learned because it is hard to observe features like gradient in the raw image.

This paper is organized as follows. We first review related work. Then, we introduce the proposed HSSD for learning hierarchical shape representation from data. Second, we describe the configuration that we experiment on fusing RGB-D information. Finally is the experimental result and conclusion.

II. Related work

RGB-D sensor such as Kinect [3] has gained its popularity in object recognition research for its rich information and low price. In [4], Lai *et al.* collected a dataset of 51 household object categories and a total of 300 instances.

In [5], Lai *et al*. proposed sparse distance learning, which is a view based learning mechanism. By applying group lasso regularization they can select representative views as object model. They combined a number of features from 2D and 3D object classification into one descriptor.

For representation learning, in [6], feature hierarchy is constructed directly from data by stacking layers. Each layer can be seen as a function that maps input to output. Typically, each layer consists of filter banks, non-linearity, as well as pooling and subsampling [7]. Yang *et al.* used densely sampled SIFT descriptors, and encoded them with a learned dictionary by applying l_1 regularization. After that, they use a max pooling operation to allow small translation [8]. In [7], the author shows that by learning filters in an unsupervised method and then refining them in a supervised manner can

This work was supported by National Science Council, Taiwan, under Grant NSC 100-2221-E-002-096, and by Microsoft, Taiwan.

Kuan-Ting Yu, and Shih-Huan Tseng are with the Department of Computer Science and Information Engineering, National Taiwan University (e-mail: r99922070@ ntu.edu.tw, shihhuan.tseng@gmail.com).

Li-Chen Fu is with the Department of Computer Science and Information Engineering and Department of Electrical Engineering, National Taiwan University (Phone: +886-2-23622209, e-mail: lichen@ntu.edu.tw).

further improve the performance. Also, using multiple layers can provide better performance, but what is the best number of hierarchy is still unclear. Our work is inspired by [9], in which Yang *et al*. build a hierarchy with each stage composed of sparse coding, max pooling, and local grouping. Their Hierarchical model with Sparsity, Saliency and Locality (HSSL) directly learns features from pixel level and groups lower level features to form complex features for upper level. Using the learned hierarchical representation, they achieved state-of-the-art performance on several well-known image classification datasets, *e.g.* Caltech 101, Caltech256, and Oxford Flowers. The work mentioned above focus on RGB images. To the best of our knowledge, only the work [10] attempts to extend the representation learning model into learning of RGB-D features. However, they consider depth images as simply 2D images and apply the model directly to learn RGB-D features. In this paper we propose a novel learning-based feature to learn hierarchical physical shape representation from depth images.

Capturing features in 3D space has the main challenge of aligning the 3D coordinate of each feature. We need to fix two axes and the third axis can be determined by cross product of the two. Johnson *et al*. proposed spin image [11], which uses the normal vector from a point. For the second axis, the orientation is not determined, instead, it creates a histogram of the presence of points at specific position to achieve rotational invariance along the second axis. Fast point feature histogram [12] extends the same idea to make histogram of not only the relative position but the normal differences between the source point and the vicinity points. We propose a generalization of the two methods by using rotational convolution along the normal axis. By doing so, this work borrows the idea of feature learning to learn shape features by exploiting sparsity on possible shapes that occurs in nature. The dictionary learned via sparse coding can adapt to specific tasks. Given a set of observations, sparse coding can analyze what major shape components occur and suppress insignificant components using l_1 regularization.

The contributions of this paper are as follows. First, we propose a novel Hierarchical Sparse Shape Descriptor (HSSD) feature to learn structural representation for 3D shape by analyzing several 3D feature extraction techniques and proposing a unified view of them. From the viewpoint of hierarchical representation learning, we incorporate 3D physical shape information. In the perspective of 3D local feature extraction, we provide a structural way to build a global descriptor from local feature. Second, we investigate whether RGB and depth information should be fused at lower level or higher level in the representation learning hierarchy.

III. Hierarchical Sparse Representation Learning

In this section, we first describe Hierarchical Sparse Shape Descriptor. We provide a generalization of two shape feature extraction techniques, *e.g.* Spin image and Fast Point Feature Histograms (FPFH) as the foundation for shape representation learning. Next, we explain how we investigate the best fusion configuration for RGB-D images.

A. Hierarchical Sparse Shape Descriptor

The hierarchical representation learning contains several levels that map input from output. Each layer consists of three component functions: sparse coding, spatial pooling, and local grouping. The system overview is shown in Fig. 1 and we discuss each part in detail as follows.

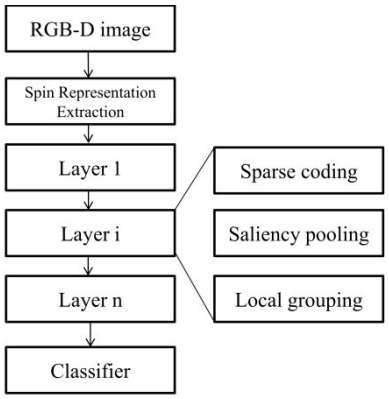
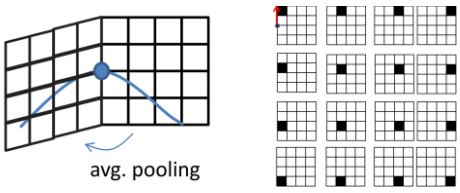


Fig. 1. HSSD System architecture

Spin Representation Extraction

Traditional representation learning only focuses on 2D images without taking into account the physical shape information. One challenge of describing shape information is to achieve rotational invariance in feature. Here, we use filter bank (dictionary) and pooling to describe the spin image and integrate it into the learning framework.

To achieve rotational invariance, 3D descriptor must be able to align the coordinate of each feature when the local point cloud rotates. Spin image utilizes local normal direction to align the first axis. Afterwards, only one degree of freedom to rotate is along the normal. Spin image achieves the invariance by making a histogram of filter response along the normal. Figure 2 illustrates the analogy.



(a) (b)

Fig. 2. An analogy between spin image extraction process and filtering-pooling framework: suppose we use 4×4 spin image. We determine the normal for the filter to work on. The filter-bank is composed of 16 patterns for grids of 4x4 as shown in (b). Each filter contains only one black area that will respond to the presence of points. Average pooling is done in a spinning manner to compute the histogram. The red arrow in (b) indicates the normal direction of the spin image filters.

FPFH (Fast Point Feature Histograms) shares the same idea. In addition to gathering statistics of relative position of vicinity points, it takes the difference of normal direction into account. In FPFH, a Darboux frame $<\mathbf{u},\mathbf{v},\mathbf{w}>$ is defined using a source point $\mathbf{p}_s$ and a nearby target point $\mathbf{p}_t$.

$$\begin{cases} \mathbf{u} = \mathbf{n}_s \\ \mathbf{v} = \mathbf{u} \times \dfrac{(\mathbf{p}_t - \mathbf{p}_s)}{\|\mathbf{p}_t - \mathbf{p}_s\|_2} \\ \mathbf{w} = \mathbf{u} \times \mathbf{v} \end{cases} \quad (1)$$

This coordinate corresponds to the image axis in spin image as depicted in Fig. 3 (e). The relationship between two points is described as a quadruplet $<\phi, d, \alpha, \theta>$.

$$\begin{aligned} \phi &= \mathbf{u} \cdot \frac{(\mathbf{p}_t - \mathbf{p}_s)}{d} \\ d &= \|\mathbf{p}_t - \mathbf{p}_s\|_2 \\ \alpha &= \mathbf{v} \cdot \mathbf{n}_t \\ \theta &= \arctan(\mathbf{w} \cdot \mathbf{n}_t, \mathbf{u} \cdot \mathbf{n}_t) \end{aligned} \quad (2)$$

In the following, we show how to construct filter-bank for FPFH. Orientation ϕ, and distance d describe the position of target points relative to that of the source point. Figure 3 (a) (b) show the corresponding filter. Here, we assume the values of ϕ, d, α and θ are quantized into three bins. On the other hand, α and θ represent the difference of normal directions of the target points relative to the source point. The description of the normal difference is similar to azimuth and latitude. We can think of the normal difference on a unit sphere. Figure 3 (c) shows three spheres respectively segmented into 3 parts, in which the black field is the valid normal difference for each bin of α. We can form filters for each bin by filling the three filters with corresponding normal receptive field. Figure 3 (d) also shows three segmented spheres describing the term θ, which measures the normal orientation in $<\mathbf{u}, \mathbf{w}>$ frame. The filters for each bin can be achieved as for α. The key difference of $<\alpha, \theta>$ from $<\phi, d>$ is that the former works on the normal of point cloud instead of the occupation of point cloud. After filtering along the normal, the responses are also pooled by average operator.

From the above observation, we can find out the main difference of spin image and FPFH is the pattern of the corresponding filter bank. Therefore, we would like to combine the techniques of sparse coding to automatically find out patterns that describe natural shapes most effectively. This is in contrast to manually defining shapes like plane, cylinder, and edge [13].

In this paper, we only consider the relative position of the point, but it can be easily extended to the statistics of normal difference. We compute spin image $\mathbf{P}^{spin} \in \mathbb{R}^{w_s \times w_s}$ with physical radius of r_s cm at each sampled point. We chose w_s = 16 and r_s = 5 if not stated otherwise. We form the shape signal as $\mathbf{x} = \mathbf{P}^{spin}(:)$ [1] for sparse coding in the next stage.

Sparse coding

To find a compact shape representation, we learn a set of bases that reconstruct the original signals using weighted sum. The corresponding weight coefficient is the coding result $\mathbf{s}$. The bases can be represented as a set of d-dimensional vectors, a.k.a dictionary, $\mathbf{B} = [b_1, b_2, ..., b_k], \in \mathbb{R}^{d \times k}$. Given a dictionary, we compute the sparse coding of an input signal $\mathbf{x} \in \mathbb{R}$ by solving the following minimization problem,

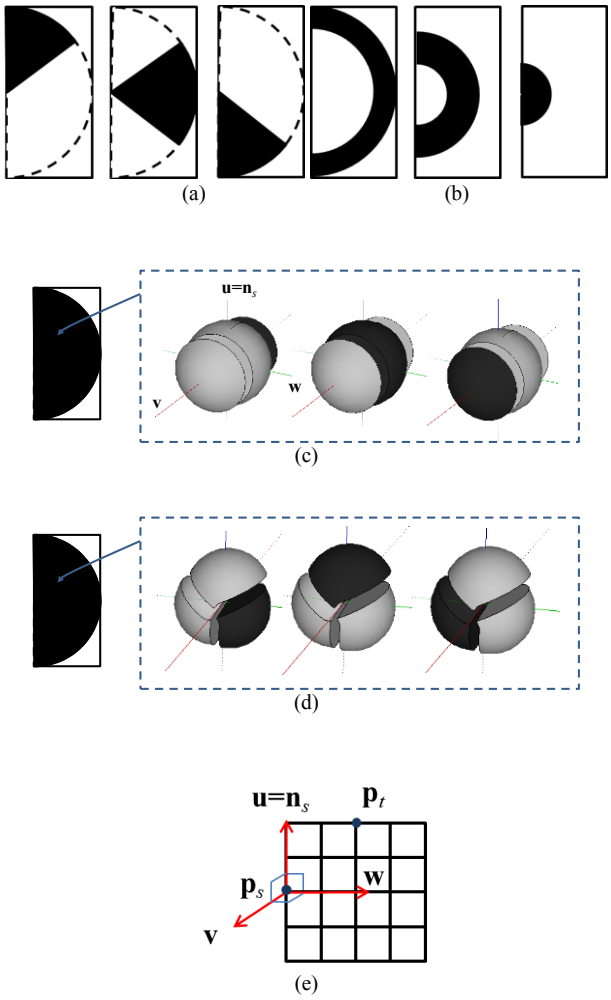


Fig. 3. The equivalent filter for ϕ, d, α, and θ depicted in (a), (b), (c), and (d) respectively. (a) and (b) work on the occupation of point cloud. (c) and (d) are normal filters which convolute with the point cloud using dot products. (e) depicts the correspondence of spin image frame and Darboux $<\mathbf{u}, \mathbf{v}, \mathbf{w}>$ frame in FPFH.

$$\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{s}\|^2 + \gamma\|\mathbf{s}\|_0, \quad (3)$$

where $\|\cdot\|_0$ denotes the l_0 -norm, and γ a regularization parameter. The first term is to minimize reconstruction error and the second is to minimize the number of nonzero coefficient used to reconstruct the observed signal $\mathbf{x}$. However, solving this formulation is an NP-hard problem, so, in the sparse coding literature, researchers use l_1 regularization to approximate (1), as shown in the following:

$$\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{s}\|^2 + \gamma\|\mathbf{s}\|_1, \quad (4)$$

In coding phase, we expect the result to be stable, *i.e.* minor changes have small effect on $\mathbf{s}$. To improve the stability, we introduce an additional l_2-norm regularization to form an elastic net problem [14].

[1] (:) is an operator to reshape a matrix into a long vector.

$$\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x}-\mathbf{B}\mathbf{s}\|^2 + \gamma\|\mathbf{s}\|_1 + \frac{\lambda}{2}\|\mathbf{s}\|_2^2, \quad (5)$$

This problem can be reformulated into a quadratic form and solved using coordinate decent algorithms.

On the other hand, finding the most suitable dictionary to represent a set of data can be useful. One idea is to solve them simultaneously to achieve the least reconstruction error and the sparsest representation for a set of data randomly sampled from lower layer as:

$$\arg\min_{\mathbf{B},\mathbf{s}_i} \sum_{i=1}^{n} \frac{1}{2}\|\mathbf{x}_i-\mathbf{B}\mathbf{s}_i\|^2 + \gamma\|\mathbf{s}_i\|_1 + \frac{\lambda}{2}\|\mathbf{s}_i\|_2^2, \quad (6)$$

Note that the objective function is not convex if we optimize both $\mathbf{B}$ and $\mathbf{s}_i$ at the same time. Therefore, we iteratively update dictionary $\mathbf{B}$ with fixed $\mathbf{s}_i$, and update $\mathbf{s}_i$ with fixed $\mathbf{B}$. The shape dictionary learned from RGB-D dataset is shown in Fig 4.

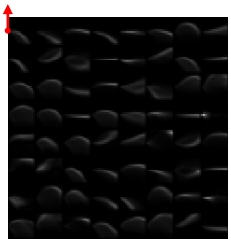


Fig. 4. the shape dictionary learned from RGB-D dataset by calculating spin images, containing 64 code word of size 16×16. The red arrow indicates the normal direction of the first spin image.

Spatial pooling

In this component, we use functions to combine the sparse codes in a working area into one descriptor. Functions typically used are max and average operations:

$$\text{Average-pooling: } \mathbf{z} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{s}_i, \quad (7)$$

$$\text{Max-pooling: } \mathbf{z} = \max_{i=1..M}\{|\mathbf{s}_i|\}, \quad (8)$$

where M is the number of $\mathbf{s}_i$ in the working windows. By doing a statistic of the features, we allow the features to have translational invariance in the working area. In the literature, the work [15] has empirically shown that max-pooling is more robust to noise. Boureau *et al.* gave an explanation in [16] about why max-pooling helps improve the performance. In this paper, we borrow the idea from [9] to use saliency pooling, which uses biological saliency map [17] to raise the weighting of the sparse code that describes the foreground object, *i.e.*,

$$\text{Saliency-pooling: } \mathbf{z} = \max_{i=1..M}\{w_i|\mathbf{s}_i|\} \quad (9)$$

Note that we can pool features in different scales. For example, for spatial pyramid matching, we can divide the working area into 1×1, 2×2 and 4×4 sub-spaces. Then, we apply pooling operation in each and concatenate them into a total of 31 $\mathbf{z}'s$ to form a final descriptor. By doing so, spatial relationship can be retained.

Local grouping

After forming locally translation-invariant descriptors, we group the nearby features to construct a higher-level descriptor that can represent a more complex structure. We can see this as a way to describe co-occurrence and spatial relationship of the local parts of a larger structure. Figure 5 shows how the grouping operation is performed.

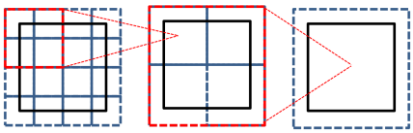


Fig. 5. Local working area (depicted in dashed line) grows larger from left to right. The object described grows from small line segments and corners to larger contour. After combining the contours will form a higher level shape - in this case, a square.

B. Fusion of Multi-channel 2D image

It is important to note that $\mathbf{x}$ may contain multiple channels, e.g. RGB-D. We will show the configuration we have investigated in this section. Given a *d*-channel image patch $\mathbf{P} \in \mathbb{R}^{w\times w\times d}$, we form the observation signal $\mathbf{x} = \mathbf{P}(:)$. We chose $w = 8$ if not stated otherwise. Given $\mathbf{x}$, the Hierarchical Sparse Saliency Locality (HSSL) [18] descriptor is computed with 2D patches. We tried the following configuration to find out what is the best way to extract feature from multi-channel image data. Combining multiple channels into one patch can be regarded as patch level fusion. Another approach is to first compute descriptors from each channel, combine them into one, and use linear SVM to fuse them, which belongs to feature level fusion.

RGB-D HSSL: combining the 4 channel
RGB HSSL: combining the 3 channel
Depth HSSL: one channel
Intensity HSSL: one channel computed from RGB image

Figure 6 shows a dictionary learned from RGB-D dataset by combining the RGB-D (4 channels) together.

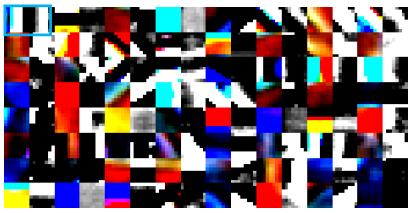


Fig. 6. Each codeword is 8×8 containing RGB-D information and appears in one block with two patches; the left patch is RGB patch and the right one is depth image patch. For example, the top left blue block shows one codeword that contains two patches of RGB and depth.

IV. EXPERIMENTS

A. Dataset

The dataset we use is the first 10 object categories from the large scale RGB-D dataset proposed in [4]. The objects picked are shown in Fig. 7. Average accuracies and standard deviation for each experiment were retrieved across 10 trials. The objects are put on a turn table and captured using a depth sensor and a higher resolution RGB camera.

We subsample the dataset by taking every fifth frame, resulting in 6,258 RGB-D images. The point cloud captured for each view is downsampled to approximately 3000 points for faster evaluation. The testing theme is category level recognition. We follow the testing procedure described in [4]: randomly leave an object out from each category for testing and train the classifiers on all views of the remaining objects.

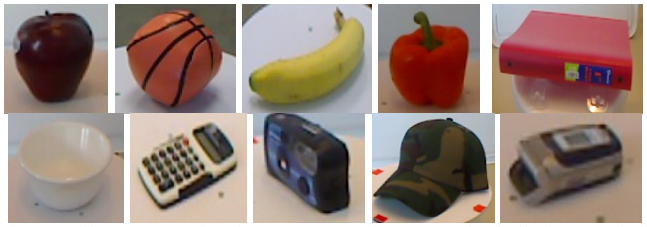


Fig. 7. objects categories from RGB-D dataset: apple, ball, banana, bell pepper, binder, bowl, calculator, camera, cap, and cell-phone.

B. Pre-processing

Before feeding raw images into the first layer, we whiten them as suggested in [9] with Caltech 101 dataset. First, we resize the image to a fixed size of 151 pixels while maintaining the original ratio. If the image has multiple layers, the resizing is conducted independently on each channel. Second, the standard deviation of the whole image and the $9\times9\times d$ local patch is calculated (d=4 for RGB-D image, and d=16×16=256 for spin-image map). We choose the greater one as the normalizer. Then, every pixel is subtracted by the mean of the $9\times9\times d$ window and divided by the normalizer. For image patch having multimodal information such as RGB-D with d=4, the normalization is done independently on RGB with d_1=3 and depth with d_2=1. We found out in this way, the performance is much better than that with normalization in a combined fashion. Third, we zero-pad the image to have $143\times143\times d$ pixels.

C. Configuration of Learning Hierarchy

Here, the configuration is made similar to [9] for Caltech 101 dataset but with some adaptation to spin image map.

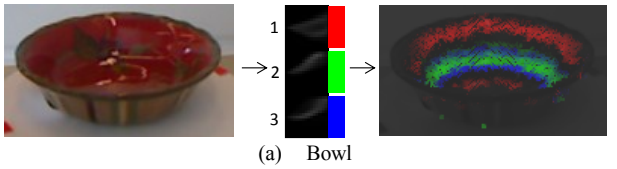
First layer: We randomly sample 200,000 $8\times8\times d$ patches to learn a dictionary of 64 codewords. Given an image, we step over it with step size one, and compute sparse coding for each local patch. As a result, we get 136×136 64-dimensional descriptor map. The sparse code is max-pooled within each 4×4 non-overlapping window, which results in 34×34 64-dimensional descriptors. Descriptors are grouped for each pixel within 4×4 local window making a 31×31 1024-dimensional descriptor map. To reduce the dimensionality, we use PCA to project it down to 96 dimensions.

Second layer: In this layer, n_{s2} codewords are learned. We chose n_{s2}=2048 for 2D image HSSL, and n_{s2}=128 for HSSD. For a given output from layer one, sparse coding will produce 31×31 n_{s2}-dimensional descriptor map. Finally, the max-pooling is operated within 1×1, 2×2, and 4×4 subspaces of the whole image. The descriptors after max-pooling are concatenated into one single long descriptor.

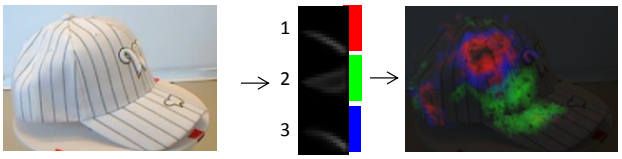
D. Hierarchical Sparse Shape Descriptor

Figure 8 shows two examples that illustrate first layer sparse coding. Spin images at every point are encoded by its major shape component. For the instance of bowl, due to different curvature from its bottom to top, they are encoded by shape words that best describe it.

In Table I, we show the accuracy of category level



(a) Bowl



(b) Cap

Fig. 8. Two examples of first layer sparse coding, *Left*: the object image, *Middle*: the three spin image bases out of 64 that has larger response on the object, *Right*: the response of the three shape words. Red: the response of basis 1, Green: the response of basis 2, Blue: the response of basis 3.

recognition. Our proposed HSSD has comparable performance with directly applying HSSL on depth images, which encodes 2D contour. More importantly, when we combine Depth HSSL with HSSD, the performance increases, which shows HSSD can compensate depth image with physical shape information. We combine different cues by combining the feature vectors and using linear SVM [19] as the classifier. We compare our HSSD with VFH, which encodes viewpoint and geometry cues using FPFH of object point clouds. Although spin image does not include statistics of normal differences as in FPFH, by learning sparse representation and hierarchical structure, our HSSD outperforms VFH by 13%. The confusion matrix for HSSD is shown in Table II (a).

TABLE I. RECOGNITION ACCURACIES ON THE RGB-D10 OBJECT DATASET (IN PERCENTAGE).

Feature	Accuracy
Intensity HSSL	90.7±4.8
RGB HSSL	71.4±11.4
HSSD	84.8±4.8
Depth HSSL	85.7±4.0
HSSD+Depth HSSL	91.3±5.4
VFH [20]	71.5±2.6
RGB-D HSSL	80.8±6.3
Intensity+Depth HSSL	95.5±3.4
RGB+Depth HSSL	89.6±3.8
HSSD+Intensity+Depth HSSL	96.9±2.9

E. Comparison on Fusion method

The question we want to investigate is whether learning multi-channel dictionary help improve performance. From the comparison between intensity and RGB HSSL, we observe that although RGB has three channels of information, it performs poorer than using intensity only. The variance is also very large indicating unstableness of learning dictionaory of RGB channels. Another example is comparing RGB-D with RGB+Depth; learning dictionary by combining the 4 channels of RGB-D is inferior to combining RGB and Depth at feature level. One reason may be that the dimension

of RGB-D patches is too high, and the dictionary may overfit the sampled patch. Therefore, we attempt to reduce the dimension by PCA. However, this only gives limited improvement on learning RGB-D dictionary. The result is shown in Fig. 9.

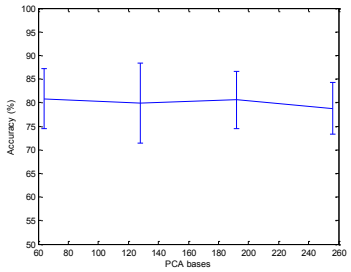


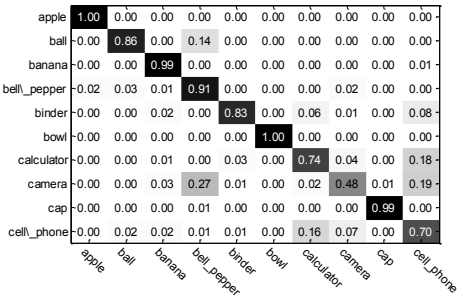
Fig. 9. Dictionary size and PCA dimension reduction in training RGB-D first layer dictionary versus recognition accuracy (in percentage).

Surprisingly, the best performance is achieved by computing Spin, depth, and intensity HSSL descriptors seperately and combine them at feature level using linear SVM. The confusion matrix is shown in Table II (b).

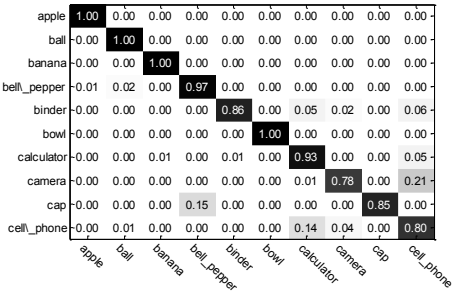
V. CONCLUSIONS

In this paper, we propose a novel Hierarchical Sparse Shape Descriptor (HSSD), which learns shape primitives in multiple hierarchies from the dataset. We achieve this by transforming spin image and FPFH into filter-pooling framework to generalize them for learning shape representation. We apply the similar learning framework to learn representation from 2D RGB-D patches. First, Experiment shows that learned shape descriptors, HSSD, provide informative cues in addition to 2D contour from depth image for improving accuracy. Second, learning dictionary from multiple channels is plausible but usually the performance is inferior to fusing them in feature level. The performance is worse than simply applying HSSL on intensity or depth only images.

TABLE II. THE CONFUSION MATRIX (A) USING HSSD (B) COMBINE HSSD, DEPTH HSSL, INTENSITY HSSL DESCRIPTORS IN FEATURE LEVEL.



(a)



(b)

VI. REFERENCES

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.* , 2005, pp. 886-893 vol. 1.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence, ,* vol. 29, pp. 1052-1067, 2007.

[4] K. Lai, B. Liefeng, R. Xiaofeng, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*, 2011, pp. 1817-1824.

[5] K. Lai, B. Liefeng, R. Xiaofeng, and D. Fox, "Sparse distance learning for object recognition combining RGB and depth information," in *IEEE International Conference on Robotics and Automation (ICRA), 2011* 2011, pp. 4007-4013.

[6] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation,* vol. 18, pp. 1527-1554, 2006/07/01 2006.

[7] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE 12th International Conference on Computer Vision, 2009*, 2009, pp. 2146-2153.

[8] Y. Jianchao, Y. Kai, G. Yihong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.* , 2009, pp. 1794-1801.

[9] J. Yang and M. H. Yang, "Learning Hierarchical Image Representation with Sparsity, Saliency and Locality," *BMVC 2011.*

[10] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "On the Applicability of Unsupervised Feature Learning for Object Recognition in RGB-D Data."

[11] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 21, pp. 433-449, 1999.

[12] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation, 2009.* , 2009, pp. 3212-3217.

[13] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *10th International Conference on Control, Automation, Robotics and Vision, 2008.*, 2008, pp. 643-650.

[14] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 67, pp. 301-320, 2005.

[15] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *IEEE 12th Int. Conf. on Computer Vision*, 2009, pp. 2146-2153.

[16] Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, 2010, pp. 2559-2566.

[17] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, pp. 1254-1259, 1998.

[18] J. Yang and M. H. Yang, "Learning Hierarchical Image Representation with Sparsity, Saliency and Locality," *BMVC,* 2011.

[19] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Mach. Learn. Res.,* vol. 9, pp. 1871-1874, 2008.

[20] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2155-2162.