Hierarchical Bag-of-Words Model for Joint Multi-View Object Representation and Classification

Xiang Fu, Sanjay Purushotham, Daru Xu, Jian Li and C.-C. Jay Kuo
University of Southern California, Los Angeles, CA 90089, USA
E-mail: {xiangfu, spurusho, daruxu, jli9}@usc.edu, cckuo@sipi.usc.edu

***Abstract*—Multi-view object classification is a challenging problem in image retrieval. One common approach is to apply the visual bag-of-words (BoW) model to all view representations of each object class and compare them with the representation of the query image one by one so as to determine the closest view of the object class. This approach offers good matching performance, yet it demands a large amount of computation and storage space. To address these issues, we propose a novel hierarchical BoW model that provides a concise representation of each object class with multi-views. When the higher level BoW representation does not match with that of the query instance, further comparison can be saved. We can also incorporate similar views to reduce the storage space. We conduct experiments on a dataset of 3D object classes, and show that the proposed approach achieves higher efficiency in terms of lower computational complexity and storage space while preserving good matching performance.**

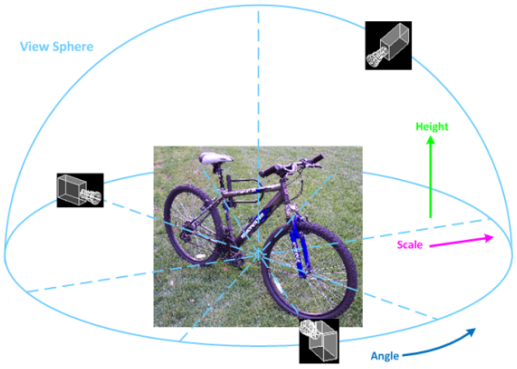

Fig. 1. There are three components to comprise the view sphere: angle, scale, and height.

I. Introduction

Rapid development of video sharing over the Internet creates a large number of videos every day. Take YouTube for example. It was reported in [1] that sixty hours of video had been uploaded to the site every minute in January 2012, and there were four billion videos viewed by users on an average day. It is an essential task to organize or classify tons of Internet videos automatically online, which will be mostly helpful to the search of useful videos in the future, especially in the applications of video surveillance, image/video retrieval, etc.

One classic method to categorize videos for human is based on the subject or the content, which makes content-based video analysis a hot topic. An object is undoubtedly the most significant component to represent the video content. For sports video, athletes, balls and the athletic field are essential objects. For education video, students and classrooms are important. Without object recognition, no system can understand video contents automatically. Therefore, object recognition and classification plays a significant role for intelligent information processing. The classical problem is to determine whether a single image contains some pre-specified objects, such as face or pedestrian recognition.

The traditional tasks of object recognition and classification include two parts. One is to identify a particular object in an image from an unknown viewpoint given a few views of that object for training, which is called "multi-view specific object recognition". Later on, researchers attempt to get the internal relation of object classes from one specific view, like computer and table, which develops to another task called "single-view object classification". In this case, the object class diversity in appearance, shape, or color should be taken into consideration. For example, some chairs have the back while others do not, and some chairs have armrest and others do not. Functionally, a chair is a raised surface used to sit on. The chair back or armrest is not a required component of a chair. These variations increase the difficulty in classification. Over the last decade, many researchers have solved the last two tasks using a concept called intra-class similarity. To further reduce the semantic gap between machine and human, the problem of "multi-view object classification" [2] needs to be well studied. As shown in Fig.1, there are three elements to define a view: angle, scale (distance), and height, which form the view sphere.

Although the viewpoint as well as intra-class variations exist as illustrated in Fig.2, some common features can still be found for one object class by a certain technique. The question is how to get the discriminant common features for each object class and how to represent these features effectively. Several successful recognition retrieval systems have been developed based on powerful local features. As compared with global features that describe the whole image, local features pay more attention to local objects and their patches. Among all local features, SIFT [3] is the most popular one for object classification and image matching in recent years since it effectively captures the stable local points for the object regardless of rotation and scale changes.

We assume that object instances belonging to the same object class should have more similar local features than others,

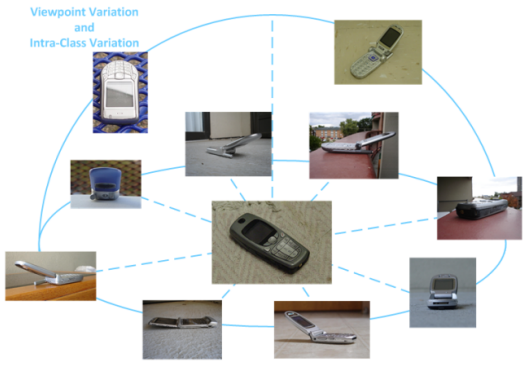

Fig. 2. Illustration of viewpoint and intra-class variations for the cell-phone class.

which is statistically true to some extent. This assumption results in the success of the visual bag-of-words (BoW) model. The BoW model in [4] is a simplifying tool originally used in text analysis and information retrieval by ignoring the position of each word. It is a dictionary-based modeling approach, where each document is treated as a bag that contains some words from the dictionary. The visual BoW model offers an effective representation of discriminant local features. In spite of its popularity, the classic BoW model suffers from several restrictions. It is far more difficult to define visual words than textual words, which are ordered combination of 26 letters. There are numerous patch patterns to represent various kinds of objects, which make it difficult to determine the visual vocabulary size. Another problem arises from the ignorance of the spatial correlation in the word frequency histogram, which results in unexpected false matches. A number of approaches [5,6,7,8] have been proposed to solve this problem by imposing various spatial constraints.

It is difficult to build a model to represent each object class for all views. Take a car for example. We can see two wheels in the side of a car, but cannot see them in the front or the back. For twenty dollars, we cannot see President Andrew Jackson and the White House at the same time. In other words, it is impossible to represent all features from different views in one model since they do not appear concurrently. On the other hand, it is expensive to make too many models for one object class even if differences always exist from different views. According to the structural characteristics of each object class, we can design a sequence of coarse-to-fine models to represent one object class to facilitate the hierarchical matching. Furthermore, for objects with holohedral symmetry like ball, appearance from all views looks similar. For objects with bilateral symmetry like bicycle, their left and right views are almost the same. Based on the view similarity, we can develop some methods to reduce the number of views for each object class model.

In addition, some object classes are so distinctive from others, which makes the complete BoW model representation wasteful in terms of storage and computation. For object classes with complex structures, we need more words to represent them. In contrast, for object classes with basic structures, we need fewer words and views. These modifications improve the effectiveness of the object class representation.

The main contributions of this paper include the following.

(1) We propose a hierarchical BoW model to effectively represent each object class using different levels. This model produces a compact yet powerful representation for each object class with a flexible number of levels depending on its shape and geometric appearance complexity. It largely reduces the model redundancy of the traditional visual BoW model.

(2) We develop a local-comparison method to build up the model to decrease the computational cost from $O(k^n)$ to $O(kn)$, where $k \geq 2$ is a constant.

(3) We describe a rule to reduce the view redundancy of each object class according to its view similarity. It efficiently represents an object class with fewer views.

II. Review of Previous Work

Many researchers attempted to solve the "multi-view object classification" problem in the last decade. Generally speaking, there are three main approaches as detailed below.

The first one is to use as many view samples as possible to model an object class for the whole view sphere. This method can yield precise models for each object class as long as the number of views is sufficient, but it is limited to a few discrete viewpoints in the training process practically. The systems cannot deal with new viewpoints that have not been trained before. Thomas [9] built a bucket model to support the general conditions of viewpoints distributed over the view sphere, and established links between adjacent views of the same object class with this concept.

The second one is to establish a 3D representation using connections from a 2D training set. This approach has little confidence on images with high angle variation, not to speak of unseen object even in the same class. Among them, Su [10] proposed a part-based probabilistic model to learn affine constraints between object parts. Savarese [11] formed the 3D object class model by connecting the canonical parts in the 3D space.

The third one is feature resorting on existing 3D models. It suffers from rendering artifacts and causes large differences between real images and synthetic models. In [12], patches were collected from multiple 2D training images and labeled on an existing 3D CAD model. They utilized the model views (multi-view specific object) as bridges to connect the supplemental views (multi-view same class object) to the model and constructed a codebook by combining all the mapped features with 3D locations. Liebelt [13] established a link between 3D geometry and local 2D image appearance using synthetic CAD models.

Li [14] produced two kinds of models for objects in the Semantic Robot Vision Challenge in 2007. For objects with little intra-class variation such as branded items and books, the model could be simplified by learning. A template matching could be applied to separate them from others. For general objects with high intra-class variation such as chair, bag, or

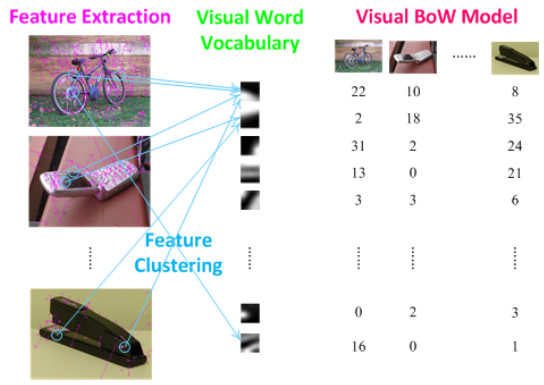

Fig. 3. Illustration of visual BoW establishment, where all features from the training set are first extracted to construct the visual word vocabulary using feature clustering, and then, the visual BoW model is built for each testing instance according to the occurrence of visual words.

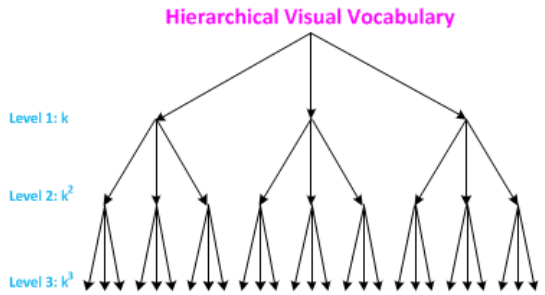

Fig. 4. Illustration of hierarchical visual vocabulary establishment for an L-level vocabulary. First, all feature descriptors of the training set in the foreground region are clustered into k groups at the first level. The centroid of each clustered subgroup represents the corresponding word at that level. For each subgroup, all children feature descriptors are clustered into k groups at the second level and there are totally k^2 words at level 2. Afterwards, the same process is recursively applied until the maximum level L is reached.

computer, we need more training to create a more general model. To reach a wider range of applications, we focus on generic object classification in this work.

Here, instead of building complicated 3D appearance model or connections on spatial restriction, we modify the fundamental local feature representation in an efficient way. In particular, we consider the characteristics of an object class and determine the object model cost according to its overall structural complexity. Furthermore, we integrate the shared features among different views depending on the object appearance similarity, which will reduce the view redundancy on the model representation.

III. Hierarchical BoW Model

The local features such as the SIFT or STIP descriptors in a frame describe the regions of interest of an object. The visual word vocabulary can be established using the k-means clustering. The centroid of each cluster yields the descriptor of the region of interest. All centroids comprise the visual vocabulary.

To describe each frame using the BoW, we count the occurrence of visual words to form the frequency histogram. For each local feature, we try to find the most similar word in the vocabulary. The whole process is shown in Fig. 3.

As compared with the textual vocabulary, it is more difficult to determine the number of feature words for the visual vocabulary. We pre-define the number for training empirically. In the literature, 1000, 2000, or 10000 feature words have been trained for various visual vocabularies previously. Here, the problem lies in the determination of the proper number of feature words suitable for a specific database and the underlying application.

To solve this problem, we adopt a hierarchical visual vocabulary [15] that has a scalable size of visual words. There are totally L levels in this model. The number of words at the next level is $k\,(k \geq 2)$ times of that at the current level. The parameter k is called the branching factor. Initially, there are k words at level 1 and there are k^l words at level l.

In words, all training feature descriptors in the foreground region are clustered into k groups at the first level. The descriptor centroid of each cluster group represents the corresponding word at that level. Afterwards, the same operation is recursively applied to subgroups to generate centroids (or visual words) for each group. The model is determined level by level, where the maximum level is denoted by L. The development progress is explained in Fig. 4, where $k = 3$ and $L = 3$.

After obtaining all words at all L levels using a hierarchical k-means algorithm, the hierarchical BoW models are established for each view of the object class. For training, we use a similarity comparison between each key point feature in the segmented foreground region and each word of that level in the hierarchical visual vocabulary. It takes time to find the closest word at each level for each key point feature, especially when the level number is high. It is however a one-time effort to train the model. For different instances in the same view for that object class, we average their feature contributions to each level of vocabularies.

In order to fairly represent the model for each view, we borrow a technique from text analysis to adjust the weight of each word. First, each histogram representation for each view is normalized to the unit sum. Let $\mathbf{v} = \{v_1, v_2, \cdots, v_{k^l}\}$ be the histogram representation. Its normalized value can be written as

$$norm\,\mathbf{v} = \frac{\mathbf{v}}{v_1 + v_1 + \cdots + v_{k^l}} \tag{1}$$

Then, we adopt the term frequency-inverse document frequency (tf*idf) measure, which is defined as

$$tf * idf\,(w,d) = tf\,(w,d) * idf\,(w)\,, \tag{2}$$

where

$$tf\,(w,d) = \frac{|\{\text{word } w \text{ in particular document } d\}|}{\sum_w \sum_d |\{\text{word } w \text{ in particular document } d\}|}, \tag{3}$$

$$idf\,(w) = \log \frac{|\text{documents}|}{|\{\text{document that contains word } w\}|} \tag{4}$$

to adjust the significance of each word. The tf*idf measure has been widely used in information retrieval and text mining. It reduces the weight of common words that appear frequently in most documents. In contrast, it gives a larger weight to less frequent words in the current document. As a result, it is able to identify the current document from others. In brief, tf*idf helps highlight significant words and downplay unimportant words. We establish the hierarchical BoW model for each view of each object class based on the above description.

To speed up the classification of the query instance, we propose a local-comparison method in the hierarchical BoW model as shown in Fig. 5 (b). For each feature point, if it belongs to word w at level l, the potential word at level $l+1$ can only be from k choices, from word $(w-1)\cdot k+1$ to word $w \cdot k$. In the training process, we still apply the traditional-comparison method, as shown in Fig. 5 (a), to build an accurate model. Thus we compare each feature point with all visual words to find the closest one at each level. It takes a total number of $k + k^2 + \cdots + k^L = k\left(1-k^L\right)/\left(1-k\right)$ in comparison. That is also the storage cost for an L-level hierarchical BoW model with branching factor k. In contrast, for the local-comparison method, we only need to make $k \cdot L$ comparisons for each query instance. When $k = 2$ and $L = 8$, the computation reduced from 510 to 16 by using the local-comparison method, with a saving of 96.9%. The performance of these two models will be further compared in Section V.

For the vector comparison between a trained model and the test query instance, the similarity measure of two models is defined using the cosine similarity as given below:

$$\text{Similarity} = \frac{\langle \mathbf{v_1}, \mathbf{v_2} \rangle}{|\mathbf{v_1}| \cdot |\mathbf{v_2}|} \tag{5}$$

where $\mathbf{v_1}$ and $\mathbf{v}_2$ are normalized model vectors adjusted by the tf*idf measure. If the similarity value is close to 1, we say that $\mathbf{v_1}$ and $\mathbf{v}_2$ belong to the same class.

IV. View Reduction

It may be possible to build a simplified object class model with fewer views by exploiting its inherent symmetry. For each object, the number of views depends on the complexity of the object appearance. For a holohedric object such as a ball or cube box, two or three views are sufficient. For an object class with bilateral symmetry, at least half of the views can be saved. For a complicated object without symmetry, we need more views.

View reduction can be used to remove view redundancy in an object class. The cosine similarity measure takes the value from 0 to 1. We partition the values into three intervals with two thresholds, i.e., T_{down} and T_{up}. When the similarity value is from 0 to T_{down} (not similar), we keep both views in the model. When the similarity value is from T_{down} to T_{up} (marginally similar), we merge these two views. Finally, when the similarity value is from T_{up} to 1 (quite similar), we discard one of the two views.

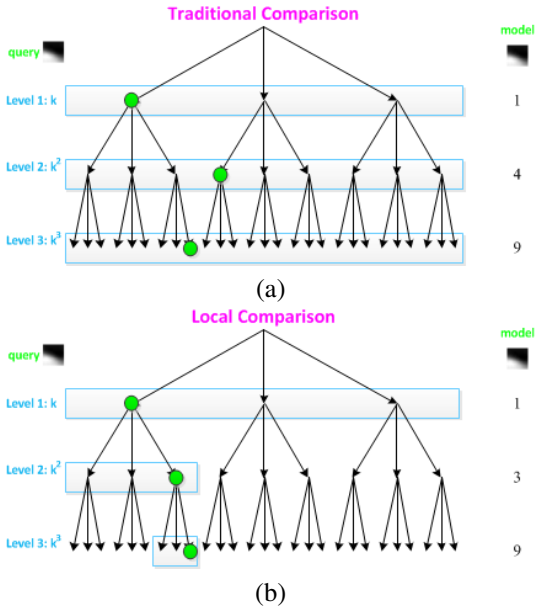

Fig. 5. Comparison of the traditional-comparison and the local-comparison methods in a hierarchical BoW model with L levels. The exemplary hierarchical BoW model has $k = 3$ and $L = 3$. The blue rectangles indicate the search ranges for each level and the green circles denote the closest word at that level for the query instance. (a) Each query has to be compared with all words to find the closest word at each level in the traditional-comparison method, and (b) each query has only to be compared with local words in the group to find the closest word at each level in the local-comparison method.

V. Experiments

The experiments in this section are conducted based on the 3D Object Categories of the Caltech dataset [11], which consists of ten common object categories, including bike, car, cell phone, head, iron, monitor, mouse, shoe, stapler, and toaster as shown in Fig. 6. For each object category, there are ten object instances with 72 viewpoints, 8 angles, 3 heights, and 3 scales (distances). The total number of images is around 7000. There are only 48 viewpoints for the car object. All images are of the bmp format with size around 400*300. All objects are located in the center of each image.

Since local features in the background will lower the classification performance, an interactive image segmentation technique is used as a pre-processing step to extract the foreground object in both training and testing whenever it is needed. Then, we can focus on local features inside the foreground object only. Because most objects are quite different from their background in the Caltech dataset, this is not a demanding task.

In a hierarchical BoW model, the larger its branching factor, the more powerful its modeling capability. However, this is achieved at an expense of higher complexity. To simplify the model, we take the branching factor $k = 2$ in the experiment. Similarly, a larger number of levels, better classification performance can be achieved but with higher complexity. We choose $L = 12$ as the highest level number.

Fig. 6. The Caltech Dataset of 3D Object Categories.

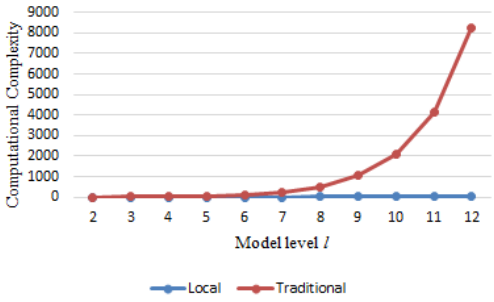

Fig. 7. Computational complexity comparison between the traditional-comparison method and the local-comparison method that have the same number of visual words, where the horizontal axis is the level number l of the hierarchical model and the vertical axis is the number of comparisons.

A. Computational complexity

We compare the computational complexity of visual word matching using the traditional-comparison method and the local-comparison method that have the same number of visual words in Fig. 7. It is clear that the local-comparison method has a much lower complexity. This is especially obvious when the hierarchical BoW model has a higher level number.

Furthermore, we use cosine similarity to see the model difference between these two methods for the same feature points. As shown in Fig. 8, the average similarity between these two methods decreases as the level, l, of the hierarchical BoW increases. When $L = 12$, the average cosine similarity is still above 0.7. This indicates that the local-comparison method still has a good approximation capability with low complexity even with a high level.

B. Object classification without view reduction

For the training data (10 classes, 10 instances, 72 or 48 views), the classification confusion matrix without view reduction is shown in Table I (12-level model). The average accuracy is 94.82%.

For a hierarchical BoW model with a level number lower than 8, the average accuracy is lower than 60% (e.g., 58.92% for the 7-level model). However, the level number depends on the object type. To achieve comparable classification performance, some need more levels while others need less. Thus,

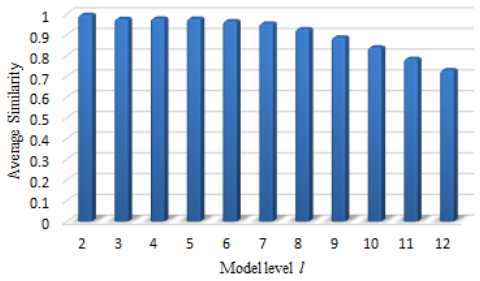

Fig. 8. Similarity comparison between the traditional-comparison method and the local-comparison method as a function of the level number of the hierarchical BoW model.

we may adopt different level numbers for different models according to their appearance.

In Table II, we list the average classification accuracy for each object class using a variable level number, where the level number ranges from 8 to 12.

As shown in Table II, the performance improvement between two adjacent levels varies from around 1% to over 10%. To save the storage cost while maintaining good classification performance, we adopt the 8-level model for bicycle, the 10-level model for car and head, the 11-level model for cell, iron, and shoe, and the 12-level model for monitor, mouse, stapler, and toaster. Although the classification performance degrades from 94.82% to 92.72%, the saving of the storage cost is more than 39.38%.

C. View reduction

It is possible to exploit the symmetry of an object class to reduce the viewpoint number furthermore. To give an example, there are 72 views for bicycle and 48 views for car in the dataset. All views are ordered in different scales, heights, and angles. We show the similarity matrix between different views, scales, heights and angles in Fig. 9. The value in the matrix ranges from 0 to 1, which is calculated using the cosine similarity. Clearly, all diagonal elements are ones. We use a gray-level map to represent the cosine similarity from 0 (black) to 1 (white). The similarity matrix of the bicycle object is shown in Fig. 9 (a), where view angles 1 and 5 correspond to the front and the back of the bicycle respectively, and other

TABLE I
THE CLASSIFICATION CONFUSION MATRIX USING A HIERARCHICAL BOW MODEL WITH 12 LEVELS. (UNIT: %)

	Bi.	Car	Ce.	He.	Ir.	Mn.	Mu.	Sh.	St.	To.
Bi.	100	0	0	0	0	0	0	0	0	0
Car	0	98.5	0.2	0	0	0	0.2	0.2	0.4	0.4
Ce.	0	0	97.2	0	0.1	1.0	0.2	0	1.4	0
He.	0	0	0.7	97.9	0	0	0.8	0	0.6	0
Ir.	0	0	0.2	0	96.6	1.2	1.2	0	0.5	0.3
Mn.	0	0.6	1.3	0.2	1.5	82.7	1.9	2.7	7.2	1.9
Mu.	0.3	0.1	0.8	0.1	2.7	1.0	91.3	0.6	2.4	0.7
Sh.	0	0.1	0.4	0.1	0	0.4	0.1	97.4	1.3	0
St.	0	0.1	1.1	0.1	0.6	1.3	2.1	0.3	94.1	0.3
To.	0	0	0.4	0.3	1.4	0.1	1.9	0	1.0	94.0

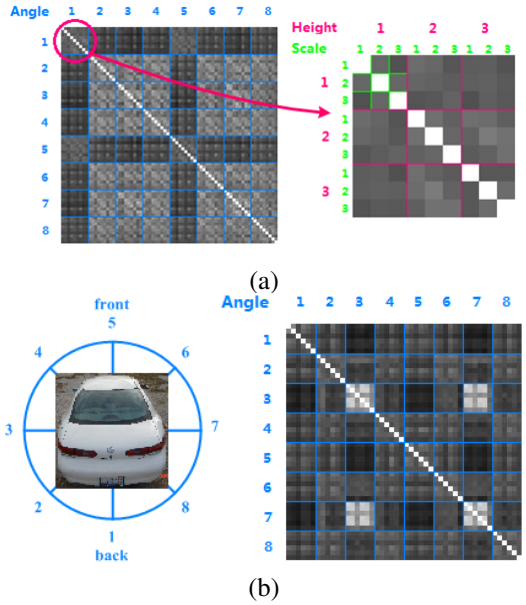

Fig. 9. The similarity matrix of different viewpoints for (a) the bicycle class and (b) the car class.

angles are side views. Due to the 180-degree symmetry, view angles 1 and 5 are similar to each other. View angles 2, 3, 4, 6, 7, 8 are similar to each other. The similarity matrix for the car object is shown in Fig. 9 (b), where view angles 3 and 7 offer the side (the left and the right) view of the car. These symmetric properties allow view reduction.

D. Object classification with view reduction

Based on view reduction discussed in the above subsection, we can reduce the size of the hierarchical BoW model. The classification accuracy and its corresponding saving in view numbers are shown in Table III.

As mentioned in Section IV, we partition the cosine similarity measure into three intervals with two thresholds, i.e., T_{down} and T_{up}. From Table III, when T_{down} is 0.5 and T_{up} is 0.7, more than half of the views are saved while the classification performance reduces around 10% only. Therefore, view reduction is an effective method to reduce the model complexity.

TABLE II
THE AVERAGE CLASSIFICATION PERFORMANCE OF EACH OBJECT CLASS WITH A VARIABLE LEVEL NUMBER. (UNIT: %)

Level:	12	11	10	9	8
Bicy.	100	100	99.72	99.44	99.58
Car	98.54	97.92	93.96	87.08	77.92
Cell.	97.22	92.08	84.72	75.14	65.00
Head	97.92	96.39	93.19	89.03	82.22
Iron	96.60	93.98	87.19	81.48	76.70
Moni.	82.70	75.53	68.14	60.13	51.05
Mous.	91.30	84.43	74.75	67.04	60.59
Shoe	97.41	92.24	86.35	73.85	66.24
Stap.	94.10	87.64	78.51	67.98	59.97
Toas.	94.03	86.94	73.19	54.86	44.03

VI. CONCLUSIONS

A hierarchical BoW model was proposed to improve the efficiency of object class representation and facilitate object classification. We also described ways to reduce model redundancy with a flexible level number and a reduced number of views. The objective is to maintain a good balance between classification accuracy and storage/computational cost. Experimental results were given to demonstrate the superior performance of the proposed hierarchical BoW model.

REFERENCES

[1] "YouTube reports 4 billion video views – a day," in *the Business and Culture of Our Digital Lives from the L.A. Times,* Jan. 23, 2012.

[2] H. Chiu, "Models for Multi-View Object Class Detection," PhD Thesis in *Massachusetts Institute of Technology,* June 2009.

[3] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *IJCV,* vol. 60, pp. 91-110, Nov. 2004.

[4] D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," in *ECML,* Apr. 21-24, 1998.

[5] H. Jegou, M. Douze, and C. Schmid, "Improving Bag-of-Features for Large Scale Image Search," in *IJCV,* vol. 87, pp. 191-212, May 2010.

[6] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling Features for Large Scale Partial-Duplicate Web Image Search," in *CVPR,* June 20-26, 2009.

[7] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building Contextual Visual Vocabulary for Large-Scale Image Applications," in *ACM Multimedia,* Oct. 25-29, 2010.

[8] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li, "Descriptive Visual Words and Visual Phrases for Image Applications," in *ACM Multimedia,* Oct. 19-24, 2009.

[9] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool, "Towards Multi-View Object Class Detection," in *CVPR,* June 2006.

[10] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, "Learning a Dense Multi-View Representation for Detection, Viewpoint Classification and Synthesis of Object Categories," in *ICCV,* Sep. 29-Oct. 2, 2009.

[11] S. Savarese and L. Fei-Fei, "3D Generic Object Categorization, Localization and Pose Estimation," in *ICCV,* Oct. 14-20, 2007.

[12] P. Yan, S. M. Khan, and M. Shah, "3D Model Based Object Class Detection in An Arbitrary View," in *ICCV,* Oct. 14-20, 2007.

[13] J. Liebelt and C. Schmid, "Multi-View Object Class Detection with a 3D Geometric Model," in *CVPR,* June 13-18, 2010.

[14] L. Li, J. C. Niebles, and L. Fei-Fei, "OPTIMOL: a framework for Online Picture collecTion via Incremental MOdel Learning," in *the Association for the Advancement of Artificial Intelligence (AAAI) 2007 Robot Competition and Exhibition,* July 22-26, 2007.

[15] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in *CVPR,* June 17-22, 2006.

TABLE III
THE VIEW REDUCTION CLASSIFICATION PERFORMANCE WITH DIFFERENT PARAMETER SETTING.

Parameter Setting	Saving in View Numbers	Classification Performance
$T_{down} = 0.4, T_{up} = 0.7$	61.06%	73.63%
$T_{down} = 0.5, T_{up} = 0.7$	50.57%	83.72%
$T_{down} = 0.6, T_{up} = 0.7$	40.37%	89.29%
$T_{down} = 0.4, T_{up} = 0.8$	57.47%	72.62%
$T_{down} = 0.5, T_{up} = 0.8$	26.44%	90.03%
$T_{down} = 0.6, T_{up} = 0.8$	13.22%	92.04%
$T_{down} = 0.7, T_{up} = 0.8$	3.88%	92.50%
$T_{down} = 0.4, T_{up} = 0.9$	57.33%	75.70%
$T_{down} = 0.5, T_{up} = 0.9$	46.55%	84.43%
$T_{down} = 0.6, T_{up} = 0.9$	37.07%	88.92%
$T_{down} = 0.7, T_{up} = 0.9$	28.88%	91.36%