3-D Laser-Based Multiclass and Multiview Object Detection in Cluttered Indoor Scenes

Xuesong Zhang, Yan Zhuang, *Member, IEEE*, Huosheng Hu, *Senior Member, IEEE*, and Wei Wang, *Senior Member, IEEE*

***Abstract*— This paper investigates the problem of multiclass and multiview 3-D object detection for service robots operating in a cluttered indoor environment. A novel 3-D object detection system using laser point clouds is proposed to deal with cluttered indoor scenes with a fewer and imbalanced training data. Raw 3-D point clouds are first transformed to 2-D bearing angle images to reduce the computational cost, and then jointly trained multiple object detectors are deployed to perform the multiclass and multiview 3-D object detection. The reclassification technique is utilized on each detected low confidence bounding box in the system to reduce false alarms in the detection. The RUS-SMOTEboost algorithm is used to train a group of independent binary classifiers with imbalanced training data. Dense histograms of oriented gradients and local binary pattern features are combined as a feature set for the reclassification task. Based on the dalian university of technology (DUT)-3-D data set taken from various office and household environments, experimental results show the validity and good performance of the proposed method.**

***Index Terms*— Imbalanced learning, laser scanning, multiclass and multiview 3-D object detection, multitask learning, sharing features.**

I. Introduction

INDOOR scene understanding is extremely challenging due to the presence of a large amount of object categories, pose variations, background clutter, and partial occlusions. Any service robot operated in such a complex indoor scene should have the ability to detect and recognize objects accurately. A variety of 3-D object recognition and detection systems with RGB-D cameras and associated machine learning algorithms have been developed for such a task. Lai *et al.* [1] proposed a view-based approach for labeling objects in 3-D scenes reconstructed from RGB-D videos. Sliding window detectors trained from multiple object views were utilized to assign class probabilities to pixels in every RGB-D frame. As introduced in [2], a real-time visual odometry and mapping system was proposed for RGB-D cameras. Anand *et al.* [3] used a graphical model that captured various features and contextual relations to guide semantic labeling and search for RGB-D images.

Recently, 3-D laser scanners have been widely deployed, as they are highly robust against illumination changes and typically have a larger field of view. Wang *et al.* [4] utilized the implicit shape model to describe object categories, and extended the Hough forest framework for object detection in 3-D point clouds. Steder *et al.* [5] addressed the problem of online object detection in 3-D laser range data. Their approach relied on the analysis of range images obtained from raw 3-D laser data and was based on the extraction of point features from the range images.

In order to detect multiview objects in an indoor scene, pose estimation is an important issue to be solved. Some excellent object pose estimation approaches have been developed in recent years. Shotton *et al.* [6] proposed two approaches to perform human pose estimation, which could quickly and accurately predict the positions of body joints from a single depth image without using any temporal information. De Figueiredo *et al.* [7] addressed the problem of object detection and pose estimation using 3-D dense data in a multiple object library scenario.

Traditional 3-D object recognition and detection approaches directly extract 3-D features from point clouds, such as spin image [8], fast point feature histogram [9], and 3-D SURF [10], but the computational burden of these 3-D features is very heavy for time-critical applications. To perform rapid 3-D object detection, 3-D point clouds can be transformed to different 2-D image representations, including depth image [5], [11], [12], bearing angle (BA) image [13], [17], [30], and reflectance image [14], [15]. After the transformation, many existing 2-D key point detectors and descriptors can be used for 3-D object detection.

Bo *et al.* [16] developed a set of kernel features on depth images and showed that for object recognition they were superior to pose-invariant features such as spin images. Xu *et al.* [15] proposed a segmentation method by integrating graph theory and region growing. A reflectance image was created directly from the terrestrial point clouds and segmented by the graph theory-based method. In [17], BA images were used to alleviate the computational burden in the process of segmenting and classifying 3-D point clouds for outdoor scene understanding.

Manuscript received January 25, 2015; revised July 25, 2015 and October 13, 2015; accepted October 22, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61375088 and the Training Program Foundation through the University Talents by Liaoning Province under Grant LJQ2013008.

X. Zhang is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China, and also with the Software Technology Institute, Dalian Jiaotong University, Dalian 116028, China (e-mail: zhangxuesongcn@163.com).

Y. Zhuang is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: zhuang@dlut.edu.cn).

H. Hu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hhu@essex.ac.uk).

W. Wang is with the Research Center of Information and Control, Dalian University of Technology, Dalian 116024, China (e-mail: wangwei@dlut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2496195

Another important problem in object detection is how to make full use of a fewer and imbalanced training data to learn good classifiers. Fei-Fei *et al.* [18] proposed a generative probabilistic model to represent the shape and appearance of a constellation of features and to learn visual object categories from a few training examples. The parameters of the model were learned incrementally in a Bayesian manner. Wang *et al.* [19], [20] evaluated the state-of-the-art online object tracking algorithms and proposed an algorithm that transfers visual prior learned offline for online object tracking. Shao *et al.* [21] surveyed the state-of-the-art transfer learning algorithms in visual categorization applications, such as object recognition, image classification, and human action recognition.

In [22]–[24], multitask learning was used in multiclass object detection to share features between objects from different categories and decrease the amount of training data. The domain adaptation technique was utilized in [25] to transfer useful knowledge from another domain. They utilized some objects from Google's 3-D warehouse to train an object detection system for 3-D point clouds collected by robots navigating through both urban and indoor environments. Many algorithms and models have also been proposed for imbalanced learning. In [26], two models of evolutionary fuzzy ARTMAP neural networks were proposed to deal with the imbalanced data set problems. A critical review of the state-of-the-art technologies and the current assessment metrics used to evaluate learning performance under the imbalanced learning scenario is presented in [27] and [28].

This paper proposes a novel multiclass and multiview 3-D object detection system framework that is based on the 3-D point clouds acquired by a mobile robot equipped with a custom-built 3-D laser scanner. In this framework, a joint boost algorithm is deployed to train multiclass object detectors [23]. Then, the RUS-SMOTEboost algorithm is proposed to train a group of binary classifiers with imbalanced training data. These binary classifiers are used to reclassify the low confidence bounding boxes generated from the multiclass object detection phase. This paper is mainly focused on how to improve the accuracy of multiclass and multiview 3-D object detection in cluttered indoor scenes with the following features.

1) BA images, instead of the raw 3-D point clouds, are used to perform multiclass and multiview 3-D object detection so that a service robot can accomplish scene understanding task at a low computational cost. Multitask learning is deployed in our system to cope with a small number of BA images and obtain the fast training time. Related object detection tasks are learned simultaneously by extracting and utilizing appropriate shared features across tasks. Moreover, multiple object detectors learned simultaneously using shared features tend to have better generalization ability. Fragment features are extracted from BA images, and a joint boost algorithm is utilized to train four binary classifiers for each object category. Since common fragment features are shared between similar appearance objects in different categories, object detectors can be quickly trained with a fewer BA images. In order to find objects, each weak learner votes for possible positions of the object center and consistent hypothesis are searched as local maxima in the voting space. A generalized Hough voting approach can easily deal with partial occlusions, and a fewer training examples are required.
2) To effectively use the limited training data and make up for the weakness of fragment features, histograms of oriented gradients and local binary pattern (HOG-LBP) features are used to reclassify the detected uncertain bounding boxes. A novel imbalanced learning algorithm, called RUS-SMOTEboost, is proposed to train a group of independent binary classifiers for the reclassification task. The detected low confidence bounding boxes are passed to these binary classifiers, and some false positive detection outputs can be eliminated in this phase. Since there are exact correspondences between laser point and pixel in a BA image, the foreground objects can be easily segmented out from the final detected bounding boxes by using depth information and agglomerative hierarchical clustering algorithm.

The rest of this paper is organized as follows. Section II introduces a new 3-D point cloud data set dalian university of technology (DUT)-3-D. In Section III, our multiclass and multiview 3-D object detection system framework is described briefly. The principle and the algorithm of multiclass and multiview 3-D object detection with feature sharing are explained. Moreover, the experimental results on the DUT-3-D data set are presented in this section to prove the validity of multitask learning using BA images. In Section IV, a novel algorithm called RUS-SMOTEboost is proposed to perform the reclassification task with imbalanced training data. The experimental results are given to demonstrate the feasibility and effectiveness of the proposed method. Finally, the conclusion is given in Section V.

II. DUT-3-D Point Cloud Data Set

A new 3-D point cloud data set, namely, DUT-3-D, is collected by a 3-D laser scanner on a mobile SmartROB2 robot traveling in the Yuan building of Dalian University of Technology in China. It is used for 3-D object detection in cluttered indoor scenes. The deployed 3-D laser scanning system is homemade and realized by rotating a 2-D SICK range finder (LMS200, 180° scan and 0.5° resolution) on a rotate platform. The point clouds obtained from this system are shown in polar coordinates (ρ, θ, and φ) in which ρ is the distance between the optical center of the laser range finder and the detected object, θ is the angle of each laser beam in the laser scanning plane, and φ is a rotating angle in coordinates.

A 3-D laser point $P(x, y, z)$ in a Cartesian coordinate system can be calculated by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\sin(\varphi - \varepsilon) & \sin\varphi \\ \cos(\theta)\cos(\varphi - \varepsilon) & \cos\varphi \\ \sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} \rho \\ c \end{bmatrix} \tag{1}$$

where $c \approx 10$ mm and $\varepsilon \approx 4°$ [30].

The DUT-3-D data set contains approximately 400 groups of the 3-D laser data of real indoor scenes, including more than

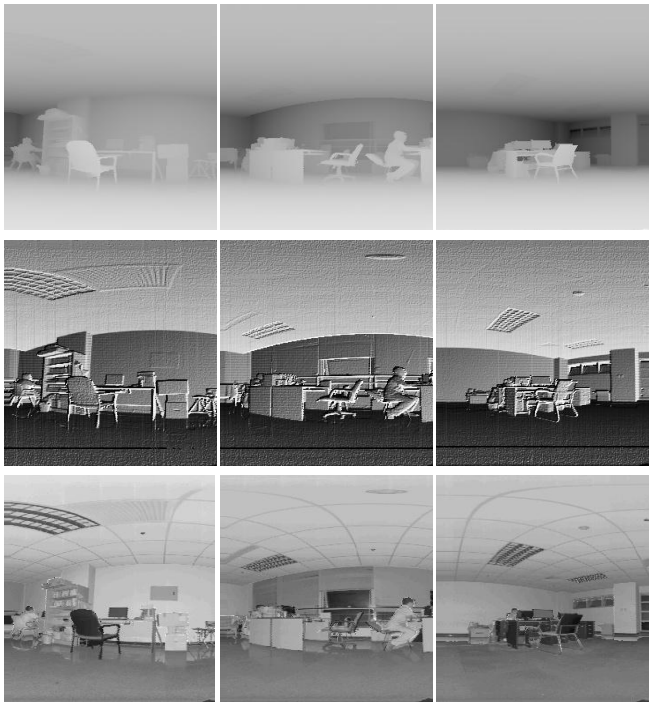


Fig. 1. Examples of filtered depth images, BA images, and reflectance images in the DUT-3-D data set (examples from row 1 to 3 are filtered depth images, BA images, and reflectance images).

600 household and office objects with different pose variations. The objects in the data set can be categorized into four groups: 1) *Chair*; 2) *Monitor*; 3) *Desk*; and 4) *Sofa*. In order to reduce the computational cost in 3-D object detection, the raw point clouds have been transformed into different 2-D image representations, such as depth images, BA images, and reflectance images.

An adaptive median filter is used to fill holes in the raw depth image, which takes the median values in a 5×5 mask centered on the current pixel. This filter is recursively used until all the holes in a depth image are filled. Fig. 1 shows three types of 2-D image representations for three groups of point clouds in cluttered indoor scenes. In addition to category-level 3-D point clouds, the DUT-3-D data set also includes 12-view 3-D point clouds for some typical object classes, which have obvious intraclass variations.

A large part of BA images in the DUT-3-D data set has been manually annotated using LabelMe [29], and these annotated images have been carefully verified for consistency and correctness. Therefore, these BA images can be used to train classifiers and evaluate their performance. All the BA images used in the experiments of this paper can be download from http://scse.dlut.edu.cn/English/Research/Projects/Datasets.htm.

III. Multiclass and Multiview 3-D Object Detection With Feature Sharing

A. Novel 3-D Object Detection System Framework

Fig. 2 shows the proposed 3-D object detection system framework, which includes two Cascade phases: 1) multiclass object detection using fragment features and 2) low confidence bounding boxes reclassification using HOG-LBP features. In order to train a stronger multiclass object detector with a fewer BA images, the multitask learning technique was deployed to share the fragment features of objects from different categories. Jointly trained object detectors vote for the possible positions of object centers in a test BA image. Using fragment features and generalized Hough voting contribute to accommodate partial occlusion and a fewer training images problems, but there still exist some false detection outputs due to some local appearance similar regions in the cluttered indoor scene.

In [30], we extracted the indoor scene framework from 3-D point clouds and utilized the semantic information to rule out some incorrect outputs. However, semantic elimination may be helpless when incorrect detection outputs are around the semantically correct position. Therefore, the RUS-SMOTEboost algorithm is proposed in this paper to train a group of binary classifiers with HOG-LBP features for the reclassification task, which can perform well on imbalanced training data.

B. Searching Best Shared Features Using Joint Boost

Up to now, several multitask learning algorithms have been proposed to solve the joint feature selection problem [23], [31], [32]. In this paper, the joint boost algorithm, first proposed in [23], was used to train multiclass and multiview object detectors.

The joint boost algorithm is a variant version of multiclass gentle boost algorithm. At each round of boosting, it solves the weighted least squares problem

$$J(n) = \sum_{c=1}^{C} \sum_{i=1}^{N} w_i^c \left(y_i^c - f_m^n(x_i, c)\right)^2 \tag{2}$$

where $y_i^c \in \{-1, +1\}$ is the membership label of training example x_i for class c and $f_m^n(x, c)$ is the mth weak classifier for class c and subset $S(n)$. Each example x_i has $|C|$ weights for C classes, and w_i^c are the weights for class c. N is the total number of training examples.

At the mth round of boosting, it will search all the possible $(2^C - 1)$ candidate subsets and fit a shared regression stump for each subset. The shared regression stumps have the form

$$f_m^n(x, c) = \begin{cases} a_s \delta\left(x_i^f > \theta\right) + b_s \delta\left(x_i^f \le \theta\right), & \text{if } c \in S(n) \\ k^c, & \text{if } c \notin S(n) \end{cases} \tag{3}$$

where x_i^f denotes the fth feature of training example x_i, θ is a threshold value, $\delta(\cdot)$ is an indicator function, a_s and b_s are the regression parameters, and k^c is a class-specific constant. The joint boost algorithm first evaluates the corresponding error of each fitted regression stump using (2). Then, it picks the best subset $S(n^*)$ such that $n^* = \arg\min J(n)$. Finally, it updates the class estimates $F(x, c)$ and the weights w_i^c of each training example by

$$F(x, c) = F(x, c) + f_m^{n^*}(x, c) \tag{4}$$

$$w_i^c = w_i^c e^{-y_i^c f_m^{n^*}(x_i, c)}. \tag{5}$$

The joint boost algorithm is implemented by two steps.

1) To repeatedly fit a shared regression stump involving scanning over all features and candidate thresholds.

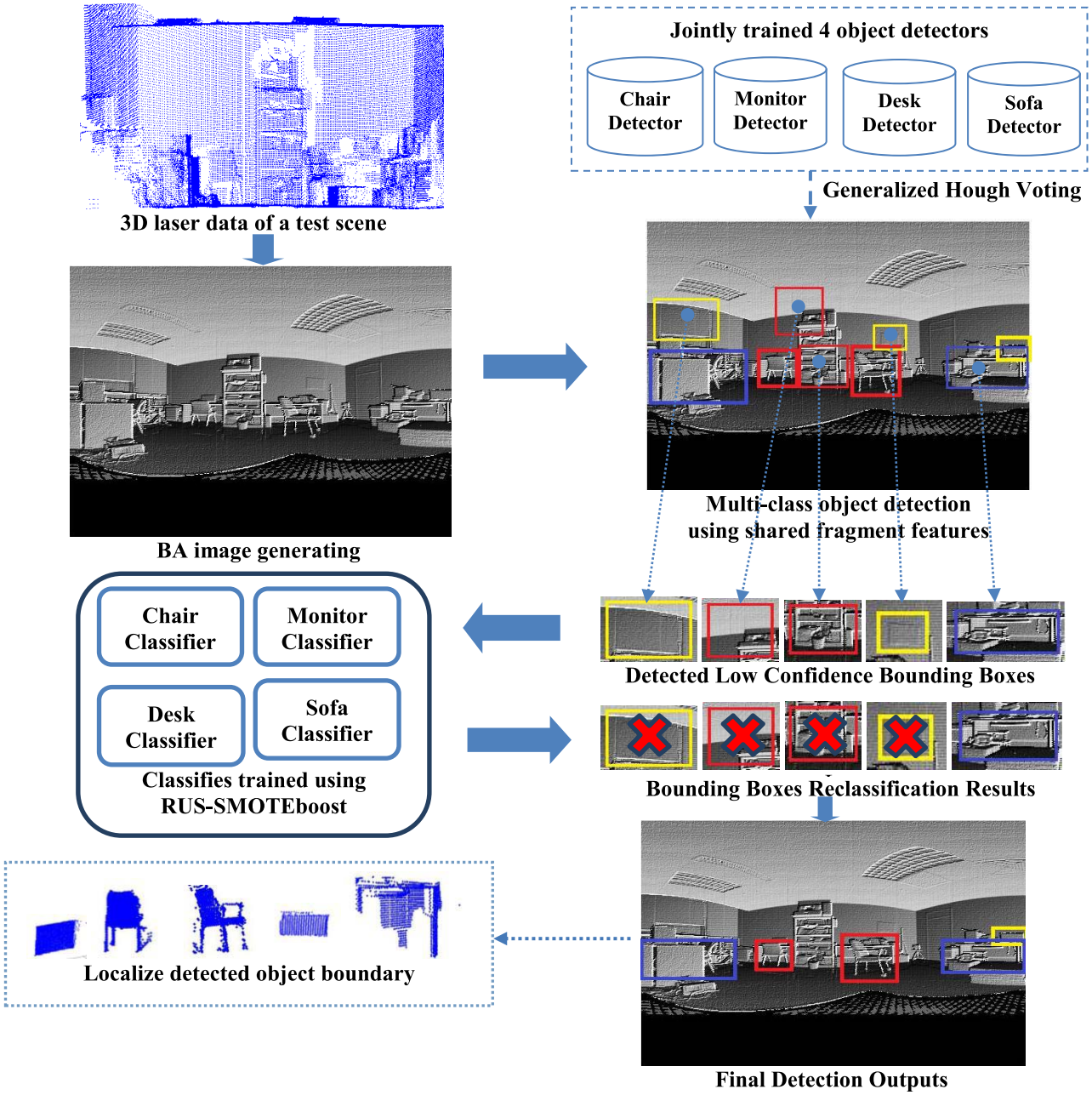


Fig. 2. Proposed 3-D object detection system framework.

2) To exhaustively search all the possible $|(2^C - 1)|$ candidate subsets to fit a shared regression stump and evaluate the corresponding error.

For any given dimension of a feature vector, the feature values in the training data constitute the candidate thresholds.

In order to fit a regression stump, it must scan over all the candidate thresholds and pick the best one. To reduce the computational cost of scanning over all the candidate thresholds, we uniformly sampled the thresholds between minimum and maximum candidate thresholds. Furthermore, we propagated most of the computation from leaf nodes to its parent nodes bottom–up as the work in [23].

Generating the candidate subset $S(n)$ is a typical combination problem, which can be viewed as picking k $(k = 1, \ldots, C)$ labels from a label set $L = \{1, 2, \ldots, C\}$ and the order does not matter. The labels in the nth picking form the candidate subset $S(n)$. Fig. 3 shows all the possible ways for the joint boost algorithm to share features on 15 possible

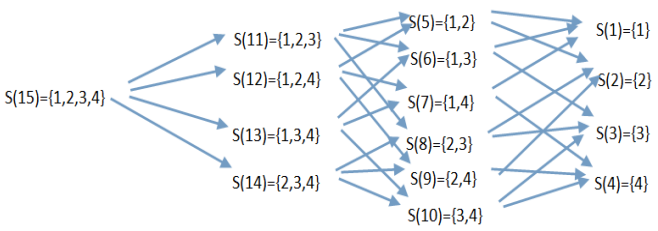
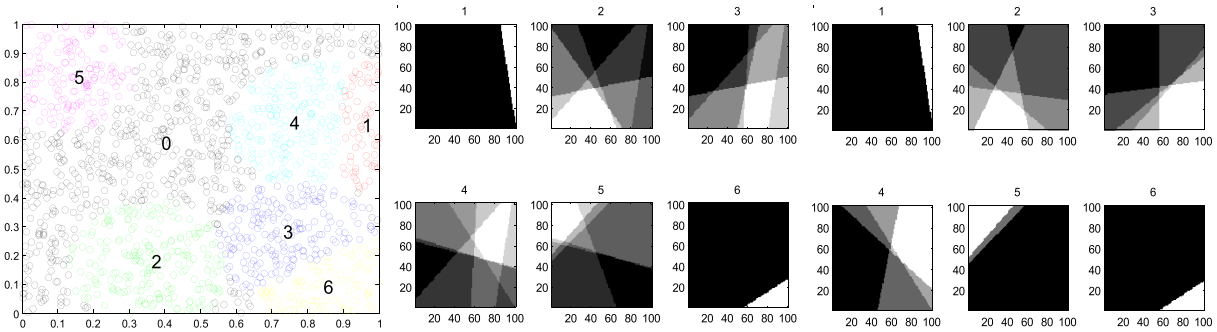


Fig. 3. All possible ways to share features of joint boost algorithm.



(a)

Number of Classes	Exhaustive Search	Best First Search	Naive Best First Search	Pair Best First Search	No Feature Sharing
12	133	156	160	139	176
11	122	137	147	114	215
10	100	94	133	115	144
9	82	59	95	48	115
8	40	58	76	59	87
7	56	61	60	74	77
6	28	64	57	53	64
5	26	25	24	26	25
Average Rounds	**73.375**	**81.75**	**92.75**	**78.5**	**112.875**

(b)

Fig. 4. (a) Top left: visualization of the artificial 2-D data points. Top middle: feature sharing manner. Top right: no feature sharing manner. (b) Boosting rounds to achieve the same performance (area under receiver operating characteristic (ROC) equal to 0.95) when using exhaustive search, best first search, naive best first search, pair best first search, and no feature sharing at each round.

candidate subsets. In this paper, three types of heuristic greedy search methods are considered.

1) *Best First Search:* It has theoretic complexity $O(C^2)$, which first deals with isolated class nodes $S(1)$, $S(2)$, $S(3)$, and $S(4)$, and then selects one with the best error reduction. Suppose it has selected $S(2)$. It will then select the second class, which has the best error reduction jointly with class 2 from the parent nodes of $S(2)$. The best first search continues to add the next best class until it has added all the classes.
2) *Naive Best First Search:* It is supplemented by us, and has theoretic complexity $O(C)$. This method first deals with isolated class nodes $S(1)$, $S(2)$, $S(3)$, and $S(4)$, and then sorts the isolated class nodes by their cost in an ascending order. Suppose that the costs of these nodes are in an order of $J(4) < J(2) < J(1) < J(3)$. The method will greedily choose nodes $S(9)$, $S(12)$, and $S(15)$ following the cost ascending order and compute the corresponding cost of each new node.
3) *Pair Best First Search:* It is a simplified version of the best first search method, and forces the joint boost algorithm to search the pairs of classes. This method works similar to the best first search, but does not take nodes $S(11)$, $S(12)$, $S(13)$, $S(14)$, and $S(15)$ into account, since there are more than two classes in these nodes.

The term feature sharing manner is defined as the joint boost algorithm using a specified search method to traverse some candidate subsets at each round of boosting and pick the best candidate subset to fit a regression stump, and all the classes in the current best candidate subset share the regression stump (feature). In contrast, no feature sharing manner means that the joint boost algorithm only traverses the one element candidate subset at each round of boosting to fit a regression stump. Considering Fig. 3, feature sharing manner needs to traverse the rightmost four leaf nodes and some internal nodes in the graph at each round of boosting, while no feature sharing manner only needs to traverse the rightmost four leaf nodes.

Fig. 4(a) shows that the classifiers trained with feature sharing manner are superior to no feature sharing manner on artificial data. We separately boosted 15 rounds to select 15 features (features are lines from 60 angles) using the joint boost algorithm with feature sharing and no feature sharing manners. As shown in Fig. 4(a), the data points of classes 4 and 5 are classified better when we use feature sharing manner.

Fig. 4(b) compares different methods of searching for the best shared stump, which can be used in a joint boost algorithm. Considering two dimensions and seven classes cases (six classes of points plus background class), 4000 data points and 6 center points are randomly generated. The class label c is assigned to a point when its distance to the cth center point is less than 0.05. We compared the number of boosting rounds to achieve a fixed level of performance (area under ROC is 0.95) for different search methods.

As shown in Fig. 4(b), using feature sharing manner always requires a fewer average boosting rounds to achieve a given area under the ROC curve (AUC) than no feature sharing manner. The results of this experiment show that, at least on artificial data, the best first search is the most stable approximate searching method. The pair best first search method requires the least average boosting rounds, and the

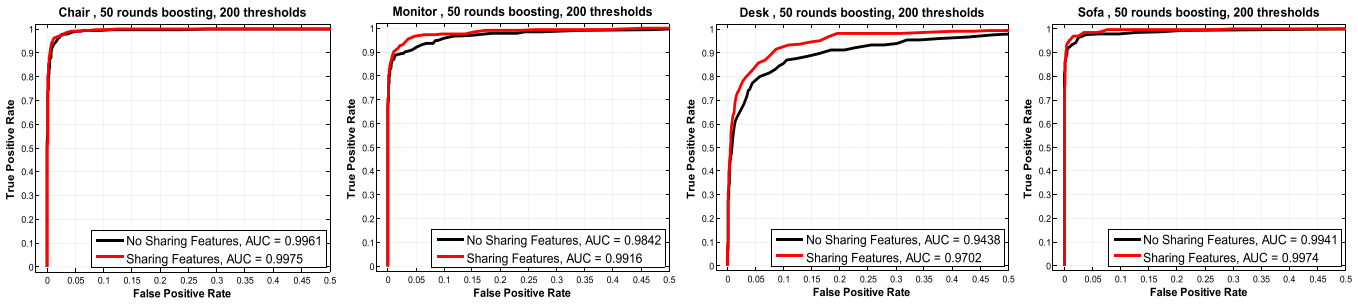


Fig. 5. ROC curves for four object categories. Red curves: classifiers trained with feature sharing manner. Black curves: classifiers trained with no feature sharing manner. ROC curves are calculated with 50 rounds boosting and uniformly sampling 200 candidate thresholds to fit a regression stump. The plot range of the x-axis is adjusted to $[-0.02, 0.5]$ to facilitate observation.

naive best first search method works better than the other two approximate searching methods when the number of classes is less than 7.

The next test was focused on whether the classifiers trained with feature sharing manner outperform the ones without feature sharing manner on the DUT-3-D data set. We selected four object categories (chair, monitor, desk, and sofa) from the DUT-3-D data set and jointly trained four classifiers using the joint boost algorithm with the feature sharing manner and no feature sharing manner separately. For each object category, we picked 70 BA images and extracted 2400-D fragment features to generate the training and validation data set. We extracted one positive example from each object center and 30 hard negative examples in the background. Therefore, there are 70 positive examples and 2100 negative examples for each object category. A stratified holdout method (repeated ten times, 30% for training and 70% for testing) was utilized to sample the training examples.

At each round of boosting, the false positive rate and the true positive rate were computed using the test data. ROC curves were obtained by computing the mean values of the truth-positive rate and the false-positive rate, which were computed at each round of boosting. The AUC is used to evaluate the performance of a classifier. We adopted the exhaustive search method to find the best shared stump in this experiment.

As shown in Fig. 5, the AUC of the feature sharing manner is larger than the ones without the feature sharing manner for all the four object categories. This experiment indicates that the feature sharing manner can be used to jointly train stronger classifiers when using BA image fragment features. The AUC of the desk category is the smallest one among the four object categories, i.e., the most difficult one to be classified in DUT 3-D. This is because of the heavy occlusion and scale variance of desks in DUT 3-D. Note that the details of extracting fragment features from a BA image will be discussed in Section III-C.

In real world applications, we may have the limited numbers of BA images to train classifiers. In order to uncover the influence of the amount of training data on the classifier's performance, we used the stratified holdout sampling (repeated ten times) at different percentages (5%, 10%, and 15%) and iterated 50 rounds to train classifiers. We used the same data set in the previous experiment. As shown in Table I,

TABLE I

AUC VALUES OF CLASSIFIERS WHICH ARE JOINTLY TRAINED FOR FOUR OBJECT CATEGORIES (BOOST 50 ROUNDS)

	Chair		**Monitor**	
	No feature sharing	Feature sharing	No feature sharing	Feature sharing
5%	0.9469	0.9465	0.7427	0.6916
10%	0.9859	0.9856	0.8768	**0.8819**
15%	0.9834	**0.9882**	0.9733	**0.9853**
	Desk		**Sofa**	
	No feature sharing	Feature sharing	No feature sharing	Feature sharing
5%	0.7654	0.7844	0.8750	**0.8942**
10%	0.7963	**0.8256**	0.9565	**0.9850**
15%	0.8514	**0.9227**	0.9876	**0.9956**

TABLE II

AUC VALUES OF CLASSIFIERS WHICH ARE JOINTLY TRAINED FOR FOUR OBJECT CATEGORIES (BOOST 100 ROUNDS)

	Chair		**Monitor**	
	No feature sharing	Feature sharing	No feature sharing	Feature sharing
5%	0.9934	**0.9944**	0.8342	0.8054
15%	0.9950	**0.9959**	0.9796	**0.9901**
	Desk		**Sofa**	
	No feature sharing	Feature sharing	No feature sharing	Feature sharing
5%	0.6785	**0.7154**	0.9467	**0.9492**
15%	0.9150	**0.9488**	0.9928	**0.9963**

the classifier's performance seriously degraded for all the four object categories when the percentage of training examples is 5%. Moreover, the AUC of no feature sharing manner (0.7427) is even larger than feature sharing manner (0.6916) for monitor category. The reason may be that the monitor category has too many similar negative and positive examples, which cannot be distinguished well by the commonly shared generic features. We increased the boosting rounds to 100 and found that the AUC values of all the four object categories were all increased, as shown in Table II. In particular, the AUC of 15% case (boosting 100 rounds) was close to the AUC of 30% case (Fig. 5, boosting 50 rounds).

From this experiment, it became clear that the limited numbers of shared generic features were not better than

the limited numbers of class-specific features for classifying hard negative examples. The number of BA images used for training should not be lower than some specified amount and boosting by sufficient rounds may be helpful. When there are insufficient training data, boosting too many rounds will result in overfitting the training data and decrease the classifier's generalization ability. As we know, the HOG-LBP features are better than the fragment features to represent the global visual appearance of an object. This is the intuition of using HOG-LBP features to compensate for the weakness of BA image fragment features and reclassify the detected uncertainty bounding boxes.

C. Multiclass Object Detection in BA Images

Motivated by the work in [23] and [36], we built BA image fragment features by extracting a random set of 2400 fragments from a subset of training BA images that includes four object categories (each object is normalized in scale to fit in a bounding box of 28×56 pixels). Note that 28 is the maximum size of object height and 56 is the maximum size of object width.

Suppose that the actual width and the height of an object in a BA image are w and h, a BA image will be scaled at min $(28/w, 56/h)$. The fragments have size ranging from 7×7 to 21×21 pixels. When a fragment f_j was extracted, we also recorded the location w_j with respect to the object center where it was taken (within the 28×56 windows).

Once the fragment dictionary was built, the normalized cross correlation was made between each fragment f_j and the training images. Both the BA image I and the fragment f_j were filtered using a spatial filter s_j before applying the normalized cross correlation with f_j to produce more robust features. We used the exponent $e = 3$ to perform element-wise exponentiation of the normalized cross correlation result. It approximated a local maximum operation and was good for template matching. As a result, the fragment feature $v_j(I, x, y)$ can be computed by

$$v_j(I, x, y) = \left[|I * s_j| \otimes f_j\right]^e * w_j. \tag{6}$$

Each BA image could produce a large number of training examples by using (6) to compute features. We obtained one positive example at the object center and a large number of hard negative examples in the background. The dimension of a fragment feature was equal to the number of fragments used to compute features. Since each object was normalized within a 28×56 pixels window for training, they were only detected at a normalized scale of 28×56 pixels. In this paper, only single scale and view-invariant multiclass 3-D object detection was considered. Each testing BA image was cropped and scaled before running object detectors. Multiscale object detection can be realized by running the single scale object detector on a scale space. The detailed description of using the generalized Hough voting approach to detect objects in a BA image can be seen in [30].

In the training phase, 2400 fragments were randomly chosen from the fragment dictionary to compute fragment features. The BA images in the DUT-3-D data set were divided into training images and testing images. Table III presents the exact numbers of training images, testing images, and total images. We jointly trained four object detectors for four object categories using feature sharing manner and no feature sharing manner separately. We boosted 150 rounds and scanned 400 candidate thresholds to fit a regression stump. Although 2400 fragments can produce a 2400-D feature vector, 150 rounds of boosting will only select 150 features from them. A detected bounding box was considered correct if it overlapped more than 50% with a groundtruth bounding box. Otherwise, the bounding box was considered as a false positive detection.

TABLE III
NUMBER OF TRAINING IMAGES, TESTING IMAGES, AND TOTAL IMAGES FOR MULTICLASS OBJECT DETECTION EXPERIMENT

	Chair	Monitor	Desk	Sofa
Training images	20	20	20	20
Testing images	120	63	110	60
Total images	140	83	130	80

Moreover, if we detect an object that we thought was too small (less than 10×10 pixels), we do not penalize its performance for this. Nonmaximum suppression was used to greedily select high-scoring detections and skip detections that were significantly covered by a previously selected detection. Precision–recall curves were used to evaluate the detector's performances, as shown in Fig. 6. It is clear that the feature sharing manner outperformed the no feature sharing manner for all the four category-level object detections in BA images. The desk detector worked very poorly in our experiment, since a large number of desks in the DUT-3-D data set had significant scale changes and were severely occluded by the other objects.

Fig. 7 shows some correct detection results for chairs, monitors, desk, and sofa. The detected bounding boxes of each object detector can be mapped to the original BA image and produce the complete multiclass object detection results. Fig. 8 presents some false detection outputs for chairs, monitors, desk, and sofa. There are two types of common mistakes when running multiclass object detectors: 1) false negatives due to intraclass variance of objects and 2) false positives in cluttered background regions.

D. Multiview Object Detection in BA Images

If each object pose is viewed as an object category, the multiview object detection becomes a typical multiclass object detection problem. The joint boost algorithm can be used in multiview object detection to improve the detection accuracy and reduce the computing cost. Since sharing 3-D features is computationally expensive, we shared BA image fragment features between objects with different poses.

For $|V|$-view object detection, $|V|$ binary classifiers $F(x, v_i|c)$ are jointly trained for a given object category c and view v_i $(i = 1, 2, \ldots, |V|)$, where $|V|$ depends on the pose variation. In the detection phase, a view-invariant object detector is realized by running $|V|$ binary classifiers at each image location. If multiple single-view detectors detected an object at the same position (bounding boxes were overlapped

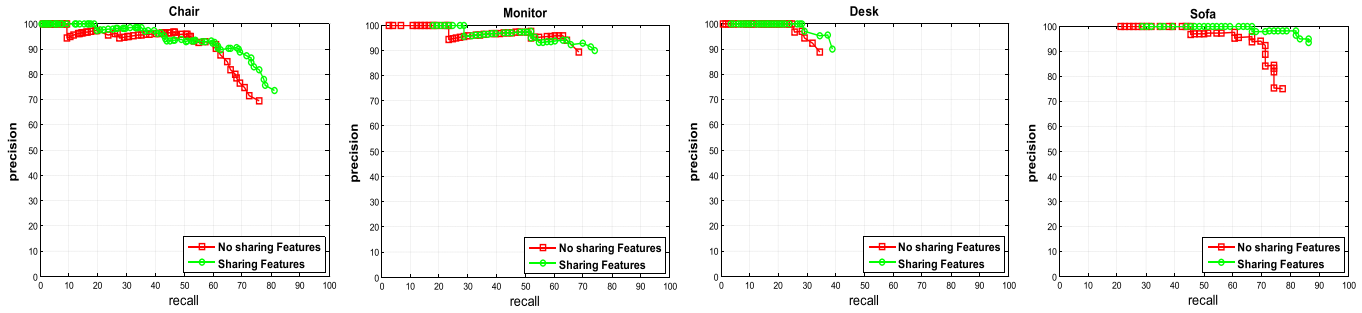


Fig. 6. Precision–recall curves for chair, monitor, desk, and sofa detection. Red curves: no feature sharing manner. Green curves: feature sharing manner. We boost 150 rounds to jointly train all the four classifiers with feature sharing manner and no feature sharing manner separately.

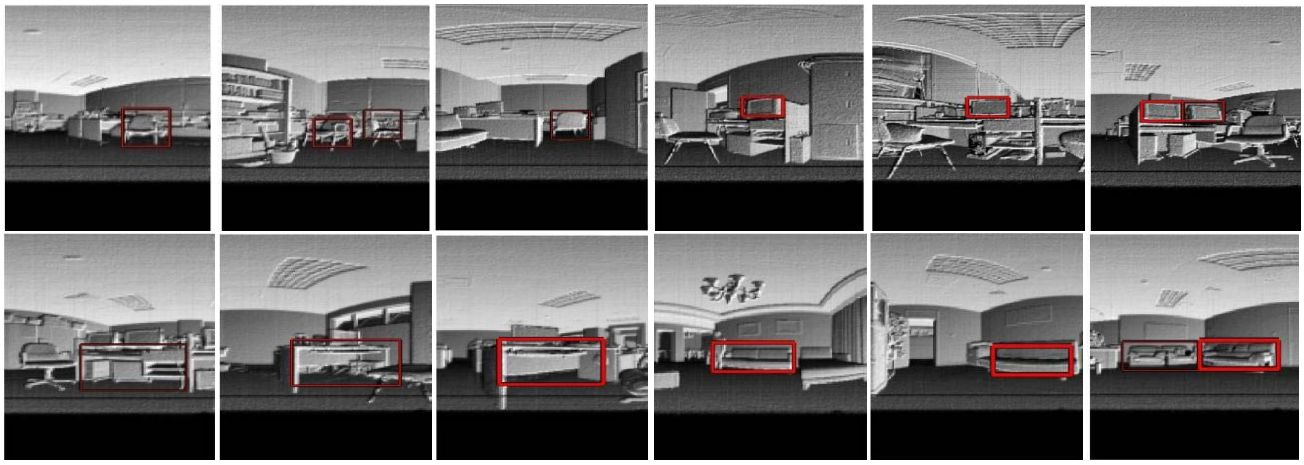


Fig. 7. Examples of correct detection outputs for chair, monitor, desk, and sofa on the DUT-3-D data set. We show object detection outputs in red bounding boxes by running four object detectors separately. These four object detectors are jointly trained using the feature sharing manner.

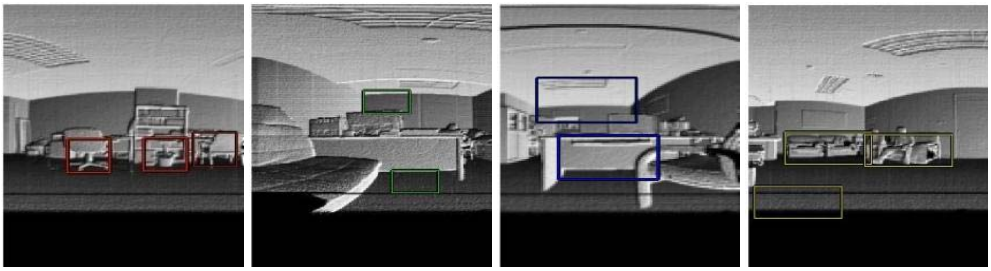


Fig. 8. Examples of false detection outputs for chair, monitor, desk, and sofa on the DUT-3-D data set. Four object detectors are jointly trained using the feature sharing manner. We run a different object detector for each different object categories. The bounding boxes with different colors (red, green, blue, and yellow) stand for different object categories (chair, monitor, desk, and sofa).

more than 50%) in a test image, the view label became $v^* = \arg\max_{vi}\{F(x, v_i|c)\}$.

In order to explore how to learn good multiview chair detectors with limited BA images using the joint boost algorithm, we collected 120 groups of raw 3-D laser point clouds for chair category with 12 original pose variations (≈30°), as shown in Fig. 9.

Four different view-invariant chair detectors were trained using different pose variations: 1) 12 pose variations (≈30°); 2) 6 pose variations (≈60°); 3) 4 pose variations (≈90°); and 4) 2 pose variations (≈180°). Since we only have ten BA images for original pose variation (≈30°), we randomly picked up two BA images to build a dictionary of fragments, five images to compute features, and the remaining three images to evaluate the performance of classifiers. In other words, we have 84 BA images to train and 36 BA images to test. Due to the fixed number of BA images to train, the finer the pose variation, the fewer positive examples belong to each view category.

Suppose that we have 12 positive examples for 12-view chair objects and the original class label set is $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$. We can generate three new class label sets $(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)$, $(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4)$, and $(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)$ for six-view, four-view, and two-view cases. In order to use different pose variations to train classifiers,

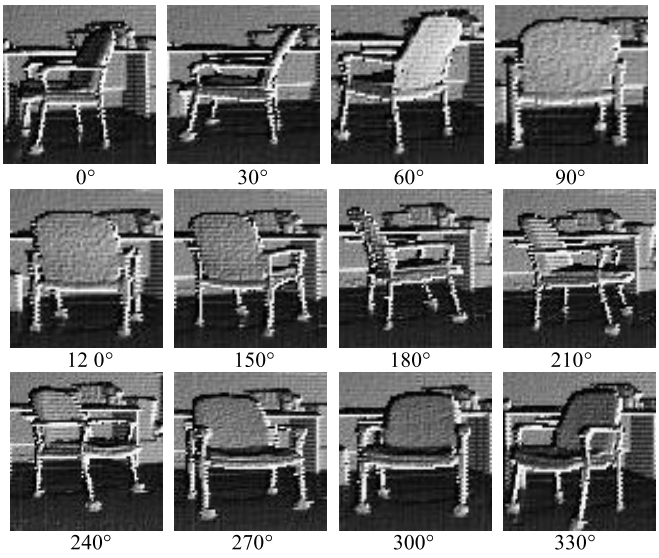


Fig. 9. Examples of original 12 view chairs in BA image with approximately 30° pose variations.

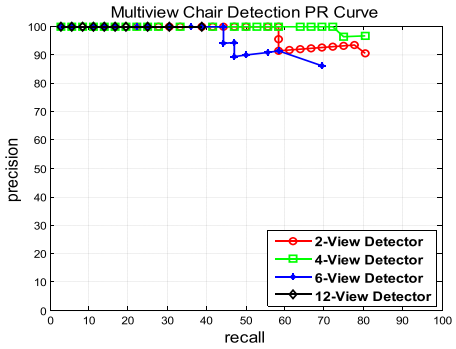


Fig. 10. Precision–recall curves for multiview chair detection. View-invariant chair detectors are jointly trained on the DUT-3-D data set using four types of pose variations (≈30°, 60°, 90°, and 180°, respectively).

we combined the original class labels from 12 pose variations and generated the other three class label sets. We utilized all the four class label sets and the same feature vectors to train 12-view, 6-view, 4-view, and 2-view chair detectors. As shown in Fig. 10, the view-invariant chair detector trained with four pose variations (four pose categories and background category) has the best performance and the view-invariant chair detector trained with 12 pose variations (12 pose categories and background category) has the worst performance.

This indicates that using feature sharing manner does not guarantee to train a stable view-invariant classifier when we have too few examples in each view class. We can infer from this experiment that, if we sampled too fine on a fixed number of training data, sharing fragment features between objects with similar poses may not increase the diversity of the weak learner and not be able to improve the generalization ability of the final strong classifiers. Note that the size of each BA image was cropped and scaled to 128×128 pixels before training and testing in this experiment.

Fig. 11 shows some typical detection results from BA images in the DUT-3-D data set. The classifiers were

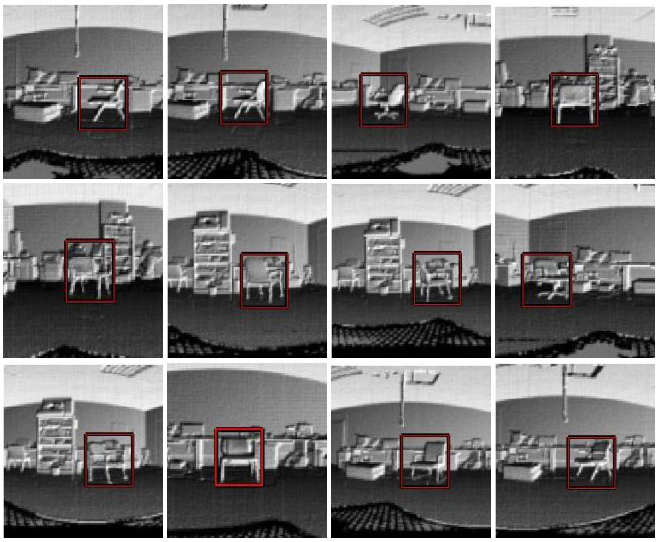


Fig. 11. Typical chair detection results of running view-invariant chair detector trained with four pose variations (≈90°).

trained with four pose variations (≈90°) of chair objects and 200 rounds boosting. The best first search method was used to fit the best shared regression stump.

E. Comparison With Depth Image-Based 3-D Object Detection

As far as we know, many 3-D object detection methods are based on depth image [5], [11] and RGB-D image [1], [16]. But RGB-D sensor (e.g., Microsoft's Kinect) is susceptible to different lighting conditions. When the target objects are under weak light or dark environment, color information is often too noisy or unavailable. Therefore, only the depth information is unchanged under all lighting conditions.

In this experiment, the joint boost algorithm was used to train four object detectors using BA images and depth images separately, which were generated from the same 3-D laser data. We boosted 150 rounds and scanned 400 candidate thresholds to fit a regression stump at each boosting round. Generalized Hough voting was used to detect objects on a test image. Since desk detection had a very poor performance and sofa detection had achieved very high accuracy when using BA images (see Fig. 6), only chair and monitor detection were considered in this experiment. The model evaluations were performed on the DUT-3-D data set. Some raw 3-D laser data in the DUT-3-D data set were transformed to depth images and BA images separately. The generated BA images and the depth images of each object category were split into training and testing image sets, as shown in Table IV. The detection performance was reported as a precision–recall curve on the test images.

Fig. 12 shows some typical detection outputs when using depth images and BA images. The detection outputs suggest that using depth images for object detection tends to produce more false positive outputs. As shown in Fig. 13, detection results using depth images have lower precision than the BA images under the precision–recall curves. The main

TABLE IV
NUMBER OF TRAINING IMAGES, TESTING IMAGES, AND TOTAL IMAGES FOR COMPARATIVE EXPERIMENT

	Chair	Monitor	Desk	Sofa
Training images	20	20	20	20
Testing images	120	40	72	40
Total images	140	60	92	60

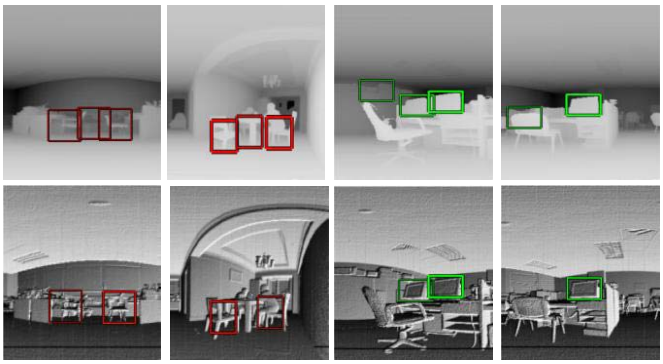


Fig. 12. Some typical detection outputs for chair and monitor. The first row is four groups of detection outputs when using depth images, and the second row is four groups of detection outputs when using BA images. Red bounding boxes: chair detection outputs. Green bounding boxes: monitor detection outputs.

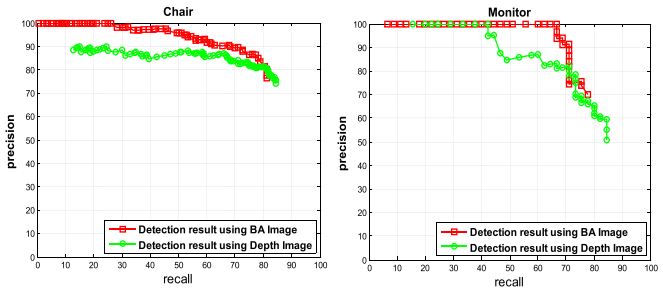


Fig. 13. Precision–recall curves for chair and monitor detection when using BA images and depth images.

reason may be that the BA image contains more detailed information than the depth image, such as texture and edge information.

The preprocess module used for generating BA images and depth images has been implemented in C++. The average time used for generating a BA image is ∼70 ms on a 2.33-GHz 2-core Intel CPU, which is slightly longer than the average time 50 ms used for generating a depth image. Overall, the experimental results indicate that using the BA image is helpful to improve the accuracy of 3-D object detection and the additional time cost of generating BA image is acceptable in practice.

IV. LOW CONFIDENCE BOUNDING BOX RECLASSIFICATION

A. RUS-SMOTEboost Algorithm

As shown in Fig. 8, there are some false positive and false negative output bounding boxes after sharing features in the multiclass object detection. The detection threshold is set to a lower value in the multiclass object detection phase to decrease the false negative detection outputs. For the reclassification phase, we only focus on eliminating the false positive bounding boxes.

Standing on the reclassification views, the multiclass object detection phase can be viewed as a generalized Hough voting process. After the voting, some uncertainty bounding boxes are passed to a group of cascaded classifiers to make the final decision. The generalized Hough voting-then-reclassification process follows the voting-then-reclassification strategy, which does not like a sliding window method. The cascaded classifiers used for the reclassification task only need to classify relatively fewer numbers of uncertainty bounding boxes. Since each test BA image is cropped and scaled in the multiclass object detection phase, the coordinates of each detected uncertainty bounding box should be transformed to its original coordinates before performing reclassification.

The most difficult problem for the reclassification phase was to train a strong binary classifier for each object category with a fewer BA images. For each BA image, we extracted M positive examples from M objects and N ($N \gg M$) negative examples randomly from background. In other words, training examples were predominately composed of a large number of negative examples. Therefore, a default strategy of guessing the majority class always gave a very high prediction accuracy $N/(N + M)\%$. Two effective ways to address the class imbalance problem were to assign distinct costs to training examples, and to resample the original data set. In considering the limited positive training examples in this paper, the synthetic minority oversampling technique (SMOTE) [33] was used to enlarge the number of positive examples. Synthetic positive examples were generated in the following manner.

1) To compute the difference of each positive example and its k nearest neighbor.
2) To multiply the difference by a random number between 0 and 1, and add it to the positive training data set.

Based on the SMOTE and the undersampling technique, we proposed a novel algorithm called RUS-SMOTEboost, which including both SMOTE the minority class and random undersampling (RUS) the majority class by a user-specified percentage. At each round of boosting, a weak classifier is learned on the data set perturbed by RUS the majority class and SMOTE the minority class. As shown in Algorithm 1, at each round of boosting, the negative training examples are randomly undersampled at $M\%$ (the percentages of negative and positive examples are $M\%$ and $1 - M\%$ after sampling), and the positive training examples are SMOTEed at $N\%$. The RUS percentage $M\%$ is specified by the user, and the SMOTE percentage $N\%$ is determined by the number of positive and negative examples after RUS in the training data.

Three experiments were designed to evaluate whether RUS-SMOTEboost algorithm could outperform RUSboost [34] and SMOTEboost [35] algorithms. For each training BA image, we extracted one positive example from each object window (normalized in 32×32 pixels) and 20 negative examples from the background window (normalized in 32×32 pixels). We concatenated the dense HOG feature (the cell size is 8×8 pixels) and the LBP

Algorithm 1 RUS-SMOTEboost

Inputs: $D=\{(x^{(i)}, y^{(i)})\}$, $i=1, 2, \cdots N$, $x \in R$, $y \in \{-1, +1\}$
Base Learning Algorithm = BLearner(D)
Booting rounds = T
Majority class Random Under-Sampling percentage = $M\%$
Output: $F(x) = \text{sign}(\sum_{m=1}^{T} \alpha_m f_m(x))$
Initialize $w_1 = 1/N$
for $m = 1$ to T do
  Generate temporary negative examples set R by RUS negative examples in D using w_m and $M\%$;
  N = (number of examples in R - number of positive examples in D);
  Generate N synthetic positive examples by SMOTE;
  NEWD′ = $R \cup$ {All the positive examples after SMOTE};
  Train Base classifier $f_m(x)$ = BLearner(NEWD′);
  Compute error of f_m on D: $\varepsilon_m = \sum_{i=1}^{N} w_m . * (f_m(x^{(i)}) \neq y^{(i)})$;
  Set $\beta_m = \varepsilon_m/(1-\varepsilon_m)$ and $\alpha_m = 0.5 * \ln(1/\beta_m)$;
  $w_{m+1} = (w_m . * \beta_m^{I(y=f_m(x))})/Z_m$, where Z_m is a normalization factor.
end for

TABLE V
NUMBER OF POSITIVE EXAMPLES AND NEGATIVE EXAMPLES

	Chair	Monitor	Desk	Sofa
Positive examples	132	135	94	104
Negative examples	2640	2700	1880	2080
Total examples	2772	2835	1974	2184

feature to represent a 32 × 32 pixels subwindow. Table V presents the number of positive and negative examples in the training data, which were used in our experiments.

In Experiment 1, we repeatedly used the stratified holdout method (30% for training and 70% for testing, repeat ten times) to evaluate the classifier's performance. We calculated the mean value of the false positive rate and the true positive rate, which was computed from each fold of testing and utilized the mean values to draw ROC curves. RUSboost, SMOTEboost, and RUS-SMOTEboost algorithms were separately used to train three binary classifiers for the same object category on the same training data. In the training phase, we boosted 30 rounds and then evaluated the performance of three classifiers on the test data. We set the RUS percentage to be 0.75, 0.8, and 0.9 for RUSboost and RUS-SMOTEboost algorithms, which means that the proportion of negative examples and positive examples after RUS is 3:1, 4:1, and 9:1.

On one hand, RUS-SMOTEboost required less training time than SMOTEboost, because it used a fewer training examples. On the other hand, RUS-SMOTEboost was more accurate than RUSboost, because more positive examples were used for training classifier. As shown in Table VI, the average performance of the RUS-SMOTEboost algorithm is better than the RUSboost and SMOTEboost algorithms.

In Experiment 2, RUS-SMOTEboost algorithm boosted different rounds to train different classifiers for the chair category.

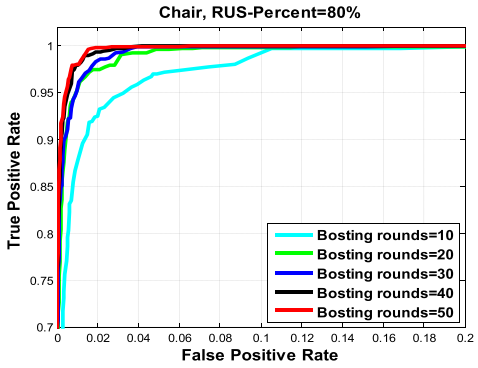


Fig. 14. ROC curves of five classifiers which are separately trained with 10, 20, 30, 40, and 50 rounds of boosting for chair category. The plot range of the x-axis is adjusted to [0, 0.2] to facilitate observation. The AUC values of the learned five classifiers are 0.9936, 0.9972, 0.9983, 0.9990, and 0.9992.

The classifiers were trained using the data set given in Table V. We fixed the RUS percentage to 80% and changed the boosting rounds to 10, 20, 30, 40, and 50. The ROC curves are given in Fig. 14. The experimental results indicate that more boosting rounds can produce a stronger classifier for the RUS-SMOTEboost algorithm.

In Experiment 3, we used logistic regression, support vector machine (SVM) (Sequential minimal optimization algorithm), and decision tree as the base learner in the RUS-SMOTEboost algorithm and trained three classifiers for each object category. The RUS percentage was fixed to 75% and boosted 30 rounds. The repeated stratified holdout method was adopted to sample the training data (30% for training and 70% for testing, repeat five times). ROC curves were calculated to evaluate the classifier's performance, as shown in Fig. 15. The experimental results indicate that using logistic regression as the base learner in the RUS-SMOTEboost algorithm has the best performance when using HOG-LBP features.

B. Sharing Features Versus Sharing Features + Reclassification

This section aims to verify if the reclassification technique is helpful to reduce the false positives outputs produced in the multiclass object detection phase. We used the same classifiers trained with the joint boost algorithm in feature sharing manner in Section III-C, and discriminatively trained four binary classifiers for each object category with the RUS-SMOTEboost algorithm for low confidence bounding box reclassification task. HOG-LBP features were extracted from BA images to build positive and negative examples. Note that the training BA images used in the reclassification phase were the same as the training BA images used in the multiclass object detection phase. Each HOG-LBP feature was extracted from a normalized window. Since objects in different categories may have different aspect ratios, the size of a normalized window should be chosen at the average aspect ratio of each object category. For example, we used a 32×32 pixels window to extract HOG-LBP features for chair category and a 28×56 pixels window to extract HOG-LBP features for sofa.

The RUS-SMOTEboost algorithm was run at 300 rounds of boosting and 80% RUS percentage, and logistic regression was

TABLE VI

AUC VALUES OF THREE CLASSIFIERS TRAINED WITH THE RUSboost, SMOTEboost, AND RUS-SMOTEboost ALGORITHMS FOR FOUR OBJECT CATEGORIES, RESPECTIVELY

	Chair			Monitor		
	RUS	**SMOTE**	**RUS-SMOTE**	**RUS**	**SMOTE**	**RUS-SMOTE**
75%	0.9941	0.9980	0.9978	0.9816	0.9900	**0.9941**
80%	0.9935	0.9981	**0.9988**	0.9843	0.9911	**0.9938**
90%	0.9883	0.9984	**0.9987**	0.9792	0.9911	**0.9915**
	Desk			**Sofa**		
	RUS	**SMOTE**	**RUS-SMOTE**	**RUS**	**SMOTE**	**RUS-SMOTE**
75%	0.9655	0.9812	0.9808	0.9963	0.9982	**0.9982**
80%	0.9695	0.9780	**0.9851**	0.9973	0.9976	**0.9982**
90%	0.9723	0.9816	**0.9840**	0.9981	0.9983	**0.9985**

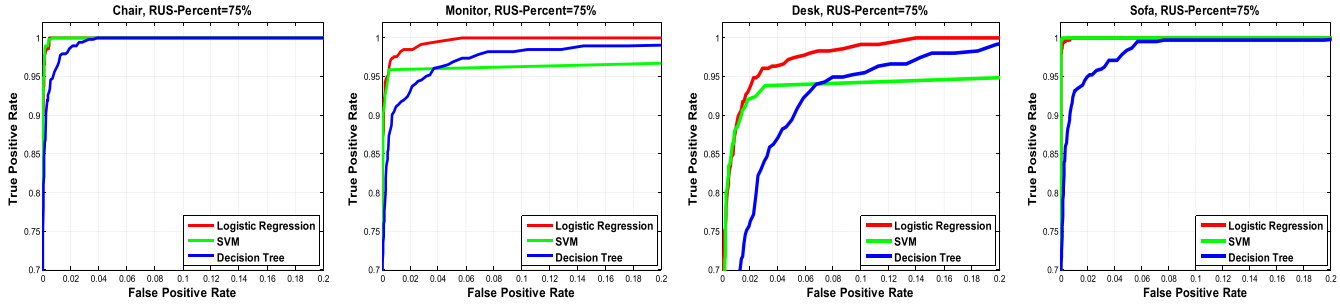


Fig. 15. ROC curves of three classifiers which are trained with the RUS-SMOTEboost algorithm using three different base learners (logistic regression, SVM, and decision tree) for each object category. The plot ranges of the x-axis and the y-axis are adjusted to [0, 0.2] and [0.7, 1] to facilitate observation.

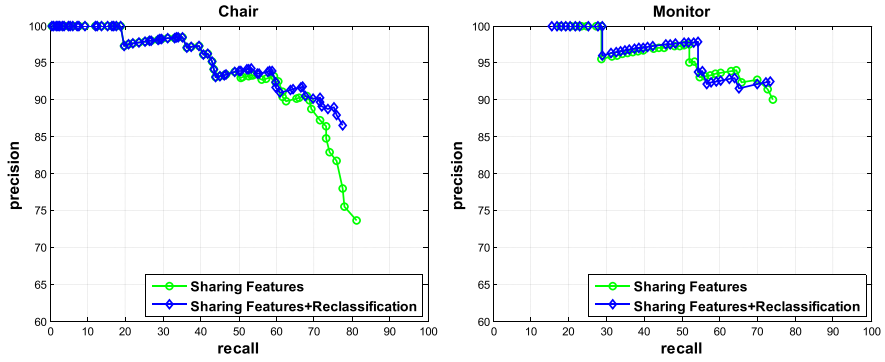


Fig. 16. Precision–recall curves for chair and monitor detection using sharing features and reclassification. The plot range of the y-axis is adjusted to [60, 100] to facilitate observation.

chosen as the base learner in it. Since the desk detector had very poor detection results (see Fig. 6) in the sharing features multiclass object detection phase, we did not consider the desk reclassification problem. For the sofa category detection, it has achieved a very high detection accuracy in the sharing features multiclass object detection phase. Therefore, precision–recall curves were calculated for chair and monitor detection in this experiment, as shown in Fig. 16. It is clear that low confidence bounding box reclassification could effectively improve the object detection accuracy.

V. CONCLUSION

This paper was focused on how to accomplish 3-D laser-based multiclass and multiview object detection in cluttered indoor scenes with a fewer groups of laser scanning data. We propose a novel framework that utilizes a voting-then-reclassification approach to improve the overall detection accuracy. Multiclass and multiview object detectors were jointly trained using a feature sharing manner to effectively reduce the influence of limited training data. The experimental results indicate that using feature sharing technique is capable of providing superior performance in the multiclass and multiview object detection. In order to train classifiers with imbalanced training data for the task of low confidence bounding boxes reclassification, a novel algorithm, RUS-SMOTEboost, was proposed to train a group of classifiers with HOG-LBP features. Experimental results are given to show the validity of the proposed approach. It should be noticed that the proposed framework is a general one and can be directly applied to many other 2-D object detection tasks with a fewer training data.

In the future work, we plan to perform rapid multiscale and multiclass 3-D object detection using BA images. Furthermore, some other 3-D features, which can be extracted from raw laser point clouds, will be deployed to improve the classifier's accuracy for object detection in much more complicated indoor scenarios.

REFERENCES

[1] K. Lai, L. Bo, X. Ren, and D. Fox, "Detection-based object labeling in 3D scenes," in *Proc. 29th IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 1330–1337.

[2] I. Dryanovski, R. G. Valenti, and J. Xiao, "Fast visual odometry and mapping from RGB-D data," in *Proc. 30th IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 2305–2310.

[3] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *Int. J. Robot. Res.*, vol. 32, no. 1, pp. 19–34, 2013.

[4] H. Wang *et al.*, "Object detection in terrestrial laser scanning point clouds based on Hough forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1807–1811, Oct. 2014.

[5] B. Steder, G. Grisetti, M. Van Loock, and W. Burgard, "Robust on-line model-based object detection from range images," in *Proc. 22nd IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 4739–4744.

[6] J. Shotton *et al.*, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, Dec. 2013.

[7] R. P. de Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Multi-object detection and pose estimation in 3D point clouds: A fast grid-based Bayesian filter," in *Proc. 30th IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 4250–4255.

[8] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[9] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. 26th IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 3212–3217.

[10] C. Redondo-Cabrera, R. J. López-Sastre, J. Acevedo-Rodríguez, and S. Maldonado-Bascón, "SURFing the point clouds: Selective 3D spatial pyramids for category-level object recognition," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3458–3465.

[11] L.-C. Chen, X.-L. Nguyen, and S.-T. Lin, "Automatic object detection employing viewing angle histogram for range images," in *Proc. 11th IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Kachsiung, Taiwan, Jul. 2012, pp. 196–201.

[12] S.-Y. Kim, S.-B. Lee, and Y.-S. Ho, "Three-dimensional natural video system based on layered representation of depth maps," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 1035–1042, Aug. 2006.

[13] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes," in *Proc. 20th IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, Oct./Nov. 2007, pp. 4164–4169.

[14] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robot. Auto. Syst.*, vol. 56, no. 11, pp. 915–926, 2008.

[15] W.-X. Xu, Z.-Z. Kang, and T. Jiang, "Segmentation approach for terrestrial point clouds based on the integration of graph theory and region growing," in *Proc. Joint Urban Remote Sens. Event*, Shanghai, China, May 2009, pp. 1–8.

[16] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. 24th IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 821–826.

[17] Y. Zhuang, G. He, H. Hu, and Z. Wu, "A novel outdoor scene-understanding framework for unmanned ground vehicles with 3D laser scanners," *Trans. Inst. Meas. Control*, vol. 37, no. 4, pp. 435–445, 2015.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.

[19] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang, "Transferring visual prior for online object tracking," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3296–3305, Jul. 2012.

[20] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "An experimental comparison of online object-tracking algorithms," *Proc. SPIE*, vol. 8138, p. 81381A, Sep. 2011.

[21] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[22] S. J. Hwang, F. Sha, and K. Grauman, "Sharing features between objects and their attributes," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, USA, Jun. 2011, pp. 1761–1768.

[23] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, May 2007.

[24] C. Zhang and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," in *Proc. 14th IEEE Winter Conf. Appl. Comput. Vis.*, Steamboat Springs, CO, USA, Mar. 2014, pp. 1036–1041.

[25] K. Lai and D. Fox, "Object recognition in 3D point clouds using Web data and domain adaptation," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1019–1037, 2010.

[26] S. C. Tan, J. Watada, Z. Ibrahim, and M. Khalid, "Evolutionary fuzzy ARTMAP neural networks for classification of semiconductor defects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 933–950, May 2015.

[27] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[28] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[29] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008.

[30] Y. Zhuang, X. Lin, H. Hu, and G. Guo, "Using scale coordination and semantic information for robust 3-D object recognition by a service robot," *IEEE Sensors J.*, vol. 15, no. 1, pp. 37–47, Jan. 2015.

[31] B. Krishnapuram, A. J. Hartemink, L. Carin, and M. A. T. Figueiredo, "A Bayesian approach to joint feature selection and classifier design," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1105–1111, Sep. 2004.

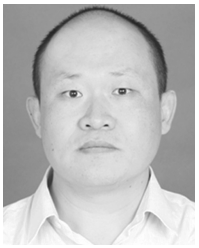
[32] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[34] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[35] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. 7th Eur. Conf. Principles Pract. Knowl. Discovery Databases*, Cavtat-Dubrovnik, Croatia, Sep. 2003, pp. 107–119.

[36] M. Vidal-Naquet and S. Ullman, "Object recognition with informative features and linear classification," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, pp. 281–288.



Xuesong Zhang received the bachelor's and master's degrees in computer application and technology from Liaoning Shihua University, Fushun, China, in 2003 and 2006, respectively. He is currently pursuing the Ph.D. degree with the School of Control Science and Engineering, Dalian University of Technology, Dalian, China.

He joined the Software Technology Institute, Dalian Jiaotong University, Dalian, in 2006, as a Teaching Assistant and became a Lecturer in 2008. His current research interests include robot vision, indoor scene understanding, object recognition and detection, and machine learning.

Yan Zhuang (M'11) received the bachelor's and master's degrees from Northeastern University, Shenyang, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2004, all in control theory and engineering.

He joined the Dalian University of Technology in 2005, as a Lecturer and became an Associate Professor in 2007. He is currently a Professor with the School of Control Science and Engineering, Dalian University of Technology. His current research interests include mobile robot 3-D mapping, outdoor scene understanding, 3-D laser-based object recognition, 3-D scene recognition, and reconstruction.

Huosheng Hu (M'94–SM'01) received the M.Sc. degree in industrial automation from the Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the Robotics Research Group. He has authored around 450 papers in journals, books, and conferences in these areas. His current research interests include behaviour-based robotics, human–robot interaction, service robots, embedded systems, data fusion, learning algorithms, mechatronics, and pervasive computing.

Prof. Hu is a Founding Member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots, a fellow of the Institution of Engineering and Technology and the Institute of Measurement and Control, and a Senior Member of the Association for Computing Machinery. He received a number of best paper awards. He has been a Program Chair or a member of Advisory/Organizing Committee of many international conferences, such as the IEEE International Conference on Robotics and Automation, the International Conference on Intelligent Robots and Systems, the International Conference on Mechatronics and Automation, the International Conference on Robotics and Biomimetics, the International Conference on Information and Automation, the International Conference on Automation and Logistics, and International Association of Science and Technology for Development, Robotics and Applications, Control and Applications, Computational Intelligence conferences. He currently serves as the Editor-in-Chief of the *International Journal of Automation and Computing* and *Online Robotics Journal*, and the Executive Editor of the *International Journal of Mechatronics and Automation*.

Wei Wang (SM'01) received the bachelor's, master's, and Ph.D. degrees from Northeastern University, Shenyang, China, in 1982, 1986, and 1988, respectively, all in industrial automation.

He was a Post-Doctoral Fellow with the Division of Engineering Cybernetics, Norwegian Science and Technology University, Trondheim, Norway, from 1990 to 1992, and a Research Fellow with the Department of Engineering Science, University of Oxford, Oxford, U.K., from 1998 to 1999. He is currently a Professor with the School of Control Science and Engineering, Dalian University of Technology, Dalian, China. He has authored over 200 papers in international and domestic journals. His current research interests include adaptive control, predictive control, robotics, computer integrated manufacturing systems, and computer control of industrial process.

Prof. Wang has been a member of the IFAC Technical Committee of Mining, Mineral and Metal Processing since 1999 and a Steering Commission Member of the Asian Control Association since 2011. He received the National Distinguished Young Fund from the National Natural Science Foundation of China in 1998. He was a Chair of the IFAC Technical Committee on Cost Oriented Automation from 2005 to 2008.