Fast Indoor Scene Classification Using 3D Point Clouds

Lei Shi, Sarath Kodagoda and Ravindra Ranasinghe
Centre for Autonomous Systems
Faculty of Engineering and IT
University of Technology, Sydney
Australia
{ Lei.Shi-1 | Sarath.Kodagoda | Ravindra.Ranasinghe}@uts.edu.au

Abstract

A representation of space that includes both geometric and semantic information enables a robot to perform high-level tasks in complex environments. Identifying and categorizing environments based on onboard sensors are essential in these scenarios. The Kinect™, a 3D low cost sensor is appealing in these scenarios as it can provide rich information. The downside is the presence of large amount of information, which could lead to higher computational complexity. In this paper, we propose a methodology to efficiently classify indoor environments into semantic categories using Kinect™ data. With a fast feature extraction method along with an efficient feature selection algorithm (DEFS) and, support vector machines (SVM) classifier, we could realize a fast scene classification algorithm. Experimental results in an indoor scenario are presented including comparisons with its counterpart of commonly available 2D laser range finder data.

1 Introduction

Scene classification endows a robot with the ability to describe an environment at a more conceptual level, resulting in a common representation of semantic information which can be effectively and efficiently shared between humans and robots [Shi et al., 2010a]. Therefore the robots could be designed with the capabilities to carry out complex tasks in populated environments smoothly interacting with humans.

Monocular camera based scene classification has been in the forefront for many years [Shi and Samarabandu, 2006; Viswanathan et al., 2009]. As discussed by Quattoni et al. [2002], indoor scene classification is a challenging task given the presence of diversity of objects in different environments. Further, it becomes extremely challenging, in places where the images consist of multiple environment types. This could be due to shorter partitions or openings such as doors or glass walls. Given the application of this paper is to classify places that inherently have such environmental features, we have focussed more on range and bearing sensors.

In this respect, 2D laser range finders have been popular sensors used for scene classification. Buschka and Saffiotti [2002] proposed a rectangle fitting algorithm to incrementally extract room-like nodes and segment the space into room and corridor regions. Mozos et al. [2005] extracted a large number of simple features and employed the AdaBoost classifier to label an environment consisting of rooms, corridors, doorways and halls, with the accuracies between 82% and 92% on different datasets; and Sousa et al. [2007] obtained the accuracy of about 80% using a subset of abovementioned features but Support Vector Machines (SVM) as a binary classifier. We have obtained better accuracies (> 96%) on similar data sets as multiclass classification [Shi et al., 2010a; 2010b].

Use of 3D data for object classification in indoor environments is reported in literature [Xiong and Huber, 2010; Rusu et al., 2009] with a reasonable success. On the contrary, outdoor scene classification based on 3D data has been reported in several literatures such as [Gibbins and Swierkowski, 2009; Munoz et al., 2007; Lodha et al., 2007]. Gibbins and Swierkowski [2009] extracted few features from 3D range data and images to distinguish structures like ground, earthworks, trees and buildings. Munoz et al. [2007] classified urban environments using 3D geometric features including statistics of local tangent and normal vector of each observed point. Lodha et al. [2007] sorted 3D Lidar data into four categories: road, grass, buildings, and trees using AdaBoost algorithm and obtained an accuracy of more than 92%. Rather than taking gross environment type classifications, those techniques label part of a scene (local) which is not the focus of our paper.

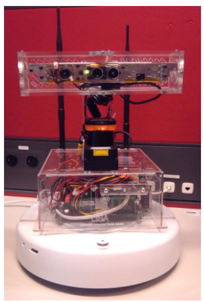


Figure 1: LISA robot (Lightweight Integrated Social Autobot)

In our previous studies [Shi et al., 2010a; 2010b], it was observed that 2D laser range data alone can give rise to classification ambiguities in some environment types. Therefore, as a cost-effective solution, in this work we have focused mainly on an inexpensive 3D range sensor, the Kinect™. Data from an indoor environment is collected using an iRobot® Create based robotics platform, LISA (Lightweight Integrated Social Autobot as shown in Figure 1), equipped with a Hokuyo laser range finder (2D) and a Kinect™ sensor (3D). A feature set that can be easily extracted from 3D range data is proposed and a subset of features which is capable of discriminating between four different spaces is selected by DEFS algorithm. A supervised learning algorithm, Support Vector Machines, is employed to evaluate the classification performance.

The rest of this paper is arranged as follows. Section 2 discusses the details about the feature extraction and selection strategies as well as the classifier. Section 3 describes the experimental setup. In Section 4, experimental results are presented; and Section 5 concludes the paper.

2 Classification

2.1 Feature Extraction

In machine learning tasks, features are descriptors of the target concept and the classifier works on features directly. Feature extraction is of great importance because it affects the ability of generalization, overhead and over-fitting of the system [Shi et al., 2010a]. In indoor environments, the appearance of different areas could be drastically affected by the in-class variations and the presence of people and furniture. Therefore, classification based only on a few trivial features such as the gross shape of a scene would be unreliable [Shi et al., 2010a].

There are various features that could be used for scene classification. For 3D range data, Osada et al. [2001] proposed five features as shape signatures. Gumhold et al. [2001] performed surface reconstruction based on line features extracted directly from point clouds. Johnson and Hebert [1999] proposed spin images as features for surface matching. Sadjadi and Hall [1980] derived a set of 3D moment invariants which are invariant under size, orientation, and position change. Gibbins and Swierkowski [2009] reviewed several local 3D features which have been used for 3D target recognition and scene analysis.

Using 2D range data, Mozos et al. [2005] extracted two sets of simple features from the raw range data and the polygonal approximation of the observed area respectively. Specifically, they employed about 150 single-valued features (considering different thresholds) of 22 categories, and used a multi-class AdaBoost classifier to process them. Similarly, Sousa et al. [2007] selected 14 single-valued features from above mentioned feature set and performed a classification task using a binary SVM classifier. In our previous work, we found that using more features normally results in an increased computational complexity and does not guarantee better performance. In two previous publications [Shi et al., 2010a; 2010b], we have shown that optimized combinations of single-valued features will achieve satisfactory classification accuracy.

For scene classification using 3D point clouds, the following 27 features were proposed as computationally simple solutions.

- Five descriptors derived from 3D moment invariants [Sadjadi and Hall, 1980], including three relative invariants $J_{1\mu}$, $J_{2\mu}$ and $J_{3\mu}$ - invariant under rotations, and two absolute invariants $J_{1\mu}^2 / J_{2\mu}$ and $\Delta_{2\mu} / J_{1\mu}^3$ - invariant to transformations.
  $$J_{1\mu} = \mu_{200} + \mu_{020} + \mu_{002}$$
  $$J_{2\mu} = \mu_{020}\mu_{002} - \mu_{011}^2 + \mu_{200}\mu_{002} - \mu_{101}^2 + \mu_{200}\mu_{020} - \mu_{110}^2$$
  $$J_{3\mu} = \mu_{200}\mu_{020}\mu_{002} + 2\mu_{110}\mu_{101}\mu_{011} - \mu_{002}\mu_{110}^2 - \mu_{020}\mu_{101}^2 - \mu_{200}\mu_{011}^2$$
  $$\Delta_{2\mu} = \det\begin{bmatrix} \mu_{200} & \mu_{110} & \mu_{101} \\ \mu_{110} & \mu_{020} & \mu_{011} \\ \mu_{101} & \mu_{011} & \mu_{002} \end{bmatrix}$$
  where μ_{ijk} are normalized central moments
- Number of observed points, which are within the range of 1.2m - 3.5m from the sensor (specified by sensor manufacturers)
- The volume of the convex hull which comprises all the observed points, and the number of points of the point cloud which are comprised in the facets of the convex hull
- The average and the standard deviation of the distance from the sensor to all observed points
- The average and the standard deviation of distance between the centroid of point cloud and all observed points
- The lengths of three semi-axes of a best-fit ellipsoid which comprises all observed points, and the 9 parameters describing the quadric surface of the best-fit ellipsoid
  $$ax^2 + by^2 + cz^2 + 2dxy + 2exz + 2fyz + 2gx + 2hy + 2iz = 1$$
- The Activity, Mobility and Complexity features which were derived by Hjorth [1970] to describe time domain signal

For the purpose of comparison, we use the following features extracted as candidate features from 2D range data. For the mathematical definitions of these features please refer to [Mozos et al., 2005].

- The area A of the polygon Z specified by the observed points; the perimeter P of Z; the normalized circularity of Z; the quotient of A / P
- The average and the standard deviation of both the beam length and the normalized beam length
- The average, the normalized average and the standard deviation of the difference between the length of consecutive beams

- The average and the standard deviation of the relation between the length of consecutive beams
- The average, the normalized average, the standard deviation and the normalized standard deviation of the distances between the centroid and the shape boundary
- The major axis M_a and minor axis M_i of the ellipse that approximates Z; the quotient of M_a / M_i
- Kurtosis of the laser range sequence

2.2 Feature Selection

According to a prevalent point of view that a small subset of features is sufficient to approximate the target concept well in most learning tasks [Ng, 2004], finding the dominant features becomes a key issue. In machine learning problems, feature selection algorithms are employed to reduce dimensionality and remove redundant features [Khushaba et al., 2011]. As a novel and effective feature selection algorithm, DEFS is a population based method which modified the differential evolution float-number optimizer to select desired number of features from high dimensional feature set.

The pseudo-code of DEFS algorithm for a single iteration is shown as follows. As a rule of thumb, for a feature set with dozens of features, 150 iterations are sufficient to find a competitive subset containing any desired number of features.

- Input: DNF (desired number of features)
  NF (total number of features)
  NP (number of population)
  Target population (NP x DNF)
- Generate mutant population (NP x DNF) from the target population using differential evolution algorithm
- For 1,…,NP
  a. Generate trial vector through a crossover of a target vector from target population with a mutant vector from mutant population
  b. If there is a feature redundancy in the trial vector, employ the feature distribution factor in a roulette wheel to substitute redundant feature, and then update the feature distribution estimation model
- Select the target vector or the trial vector whichever provides a smaller testing error to be put into the new target population.

As shown in the pseudo code, to deal with the feature redundancy problem introduced by the real number optimizer, a roulette wheel weighting scheme is implemented. The probability of each feature is calculated from its distribution factor considering its frequency of occurrence in good and less competitive subsets (known as positive and negative distributions). For more details about the DEFS algorithm, please refer to literature [Khushaba et al., 2011].

2.3 Classification Algorithm

For the work reported in this paper, we use Support Vector Machines (SVM) for supervised learning. SVM is a prominent learning algorithm originally developed by Vapnik [1999] based on a theoretical foundation rooted in statistical learning theory [Xue et al., 2010]. The basic idea of SVM is to map data into a high dimensional feature space and find an optimal separating hyper-plane with the maximal margin which has been proved to offer the best generalization ability.

Consider a training set of instance-label pairs $D=\{(x_1,y_1),...,(x_m,y_m)\}$, $x_i \in R^n$ and $y_i \in \{1,-1\}$, $i=1,...,m$ where m is the number of samples, instances x_i are firstly mapped into a higher dimension feature space F via a nonlinear mapping $\phi : R^n \rightarrow F$.

In the case that data are linearly separable in F, SVM constructs an optimal separating hyper-plane by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2} w^T w \tag{1}$$

$$s.t.\ y_i\left(w^T x_i + b\right) \geq 1,\ i=1,...,m \tag{2}$$

where w and b denote the weight vector and bias in the optimal hyper-plane $w^T x + b = 0$ respectively.

Otherwise, if data are linearly non-separable in F, the optimization problem is extended to allow a few of noisy data to exist by introducing a non-negative slack variable ξ_i, which accounts for the amount of misclassification. SVM then constructs an optimal hyper-plane $w^T\phi(x)+b=0$ with maximum-margin and bounded error by solving:

$$\min_{w,b} \frac{1}{2} w^T w + C\sum_{i=1}^{m}\xi_i \tag{3}$$

$$s.t.\ y_i\left(w^T\phi(x_i)+b\right) \geq 1-\xi_i,\ \xi_i \geq 0,\ i=1,...,m \tag{4}$$

where the constant C is a penalty parameter used to control the amount of regularization.

The kernel function is defined as:

$$K\left(x_i,x_j\right)=\phi(x_i)^T\phi(x_j) \tag{5}$$

If the kernel function exists, then it is not required to find out the specific definition of $\phi(x_j)$.

Given that the SVM is inherently a binary classifier, there exist various multi-class solutions. In this paper, we utilize the multi-class implementation of C-Support Vector Classification scheme included in LibSVM package [Chang and Lin, 2001].

3 Environment and Dataset

In the following experiments, datasets were collected by the LISA robot (Figure 1) operating in an indoor environment (Lab / Office Area, Level 6, Building 2 of the University of Technology, Sydney), consisting of office rooms, cubicles for student workstations, corridors and a common area for seminars and gatherings. The layout of the environment is shown in Figure 2 where black areas are occupied spaces and the other coloured areas indicate the free space representing the above four categories. The environment contains glass walls and inherits people during the experiments (see Figure 3). The cubicles and the offices contain similar furniture but are arranged differently based on the needs of the occupants.

The LISA robot is equipped with a Hokuyo UTM-30LX laser range finder and a Kinect™ sensor device, as shown in Figure 1. The Kinect™ provides 3D point clouds of the environment and has 57° horizontal field of view, 43° vertical field of view. The reliable range of depth is approximately 1.2m ~ 3.5m. In the experiment discussed in this paper, RGB images from Kinect™ sensor are only used for visualization purposes. The Hokuyo laser range finder has a span of 30m with 270° horizontal field of view.

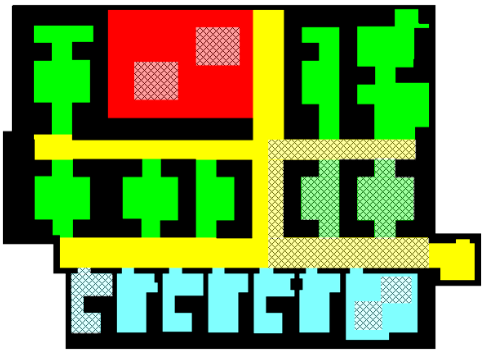


Figure 2: Blueprint of the indoor environment. Green, yellow, blue and red areas represent cubicle area, corridor area, office rooms and common area respectively. The training and testing samples were collected by robot moving randomly in the cross patterned areas, and the validation samples are gathered from coloured (except black) solid areas.

In this experiment, we have utilized a combination of six snapshots of the Kinect™ to provide 360° field of view (3D) and called it a 360° sample for the rest of the paper. Six snapshots were aligned through coordinate transformation according to the odometry information. In addition, experiments on other combinations / coverage are also conducted to provide comparable results.

Typically a 360° 3D sample comprises ~1.5 million unorganized points lying in a space with a 43° vertical field of view. By contrast, a 360° 2D sample contains ~ 6.5 thousand (it has 6 overlapping laser scans). Examples of 3D / 2D sample pairs from each class are shown in Figure 3, in which colours superimposed on the range readings are presented for ease of visualization.

The whole dataset has been divided into training set, testing set and validation set according to different robot positions. For learning, two hundred 3D / 2D sample pairs were collected in the patterned areas shown in Figure 2 for training and testing. The supervised learning and the feature selection algorithms work on both training and testing dataset to tune model parameters and select optimized feature combinations. For validation, another six hundred sample pairs were collected from the remainder of the environment, which were excluded in the learning process. The classification results on validation dataset reflect the performance of the model.

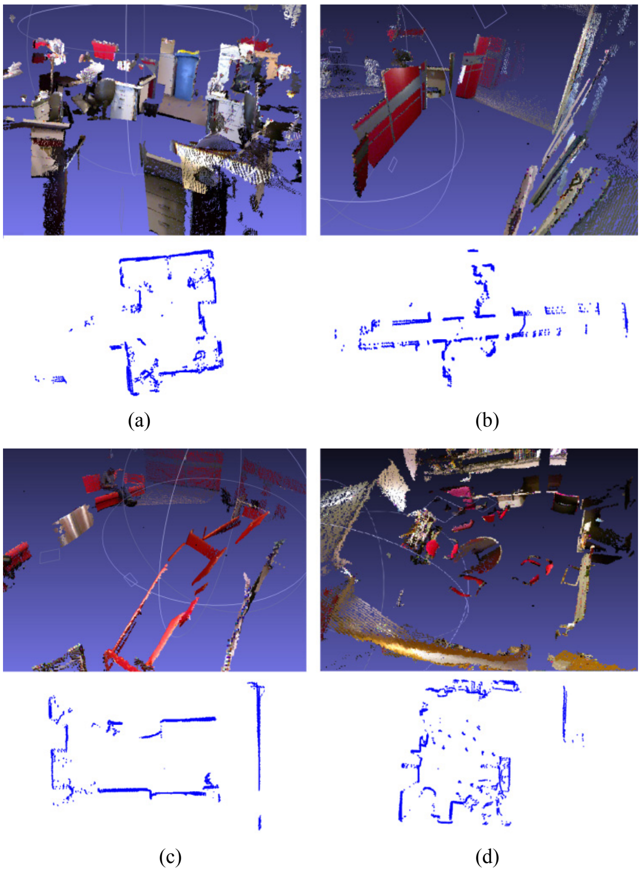


Figure 3. Examples of 3D / 2D sample pairs from (a) cubicle, (b) corridor, (c) common area, (d) room. Colours in the 3D view are for visualization purposes only, and there are people appearing in (a) and (c). (Note: figures are not drawn to scale)

4 Results

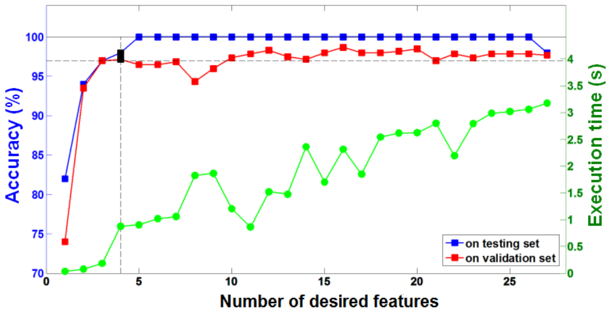
4.1 DEFS Feature Selection

In order to find a subset of features that best interact together to solve the scene classification problem, DEFS feature selection algorithm was applied on 3D and corresponding 2D samples. Therefore, a sweep in the whole feature space among all possible desired dimensions is performed, and the testing accuracies of the best-n features are recorded, where n ranges from one to the dimension of the feature space.

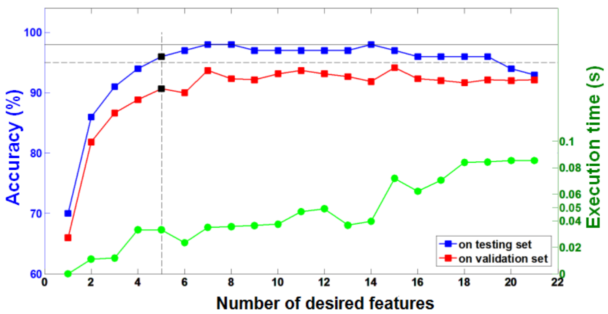
The feature selection process operated on training and testing sets and the optimization criteria is to minimize testing error. Validation set did not involve in the selection, therefore the accuracy on validation set with corresponding selected subset applied reflects the real-world scenario.

The results of feature selection on 3D and 2D samples are shown in Figure 4 (a) and Figure 4 (b) respectively. In these figures, feature selection has proved essential to machine learning problems by highlighting the outstanding performance of certain feature combinations. Furthermore, using an optimized subset of features avoided the cost of constructing insignificant or even misleading features.

There is no particular rule to accept certain n-dimensional optimized subset, because the choice is a compromise between the performance and the time overhead. However, for the sake of efficiency, it would be a reasonable choice to focus on the end of the steep rising stage of the performance curve. Therefore, without considering the accuracy on validation set which is actually unavailable to the feature selection algorithm, the optimized subset which first gives a testing accuracy of within 95% of maximum testing accuracy, is adopted in the following experiment.



(a)



(b)

Figure 4. Performance and time overhead of selected feature subsets with different desired numbers of features on (a) 360° 3D samples and (b) 360° 2D samples. The horizontal solid / dotted line indicates the best performance and 95% of the best performance on testing set respectively, and the vertical dotted line marked the accepted subset dimension.
(Note: the validation set was not involved in the feature selection)

As a result, a subset of the four features has been selected for the 3D classification scenario:

- The volume of the convex hull which comprises all observed points
- The standard deviation of the distances from the sensor to all observed points
- The standard deviation of the distances between the centroid of the point cloud and all observed points
- The parameter h, which is one of the 9 parameters describing the quadric surface of the best-fit ellipsoid.
  $ax^2 + by^2 + cz^2 + 2dxy + 2exz + 2fyz + 2gx + 2hy + 2iz = 1$

Another subset of the five features has been selected for the 2D classification scenario:

- The perimeter P of the polygon Z specified by all observed points
- The standard deviation and the normalized average of the difference between the length of consecutive beams
- The standard deviation and the average of the distances between the centroid and the shape boundary

For a computer with Intel Xeon W3580, 3.33GHz CPU, it takes typically 0.88s to process a 360° 3D sample containing ~1.5 million unorganized points to extract the four 3D features. To extract the five 2D features from a sample with ~6.5 thousand points typically takes 0.03s.

4.2 Results from Supervised Learning

In this section, a learning model is generated from training set and thereafter has been applied on validation sets. Specifically, in the learning stage, 360° 3D samples (or corresponding 360° 2D samples) from the training set with preselected feature subset and known labels (cubicle, corridor, common area and room) were processed by SVM classifier to generate a model. To demonstrate the effectiveness of learning, the model was then applied on the validation set, resulting in a series of predictions. The degree of closeness between the ground truth and the predictions reflected the correctness of the model.

In this experiment, classification of 360° 3D samples gave an accuracy of 97.17% and 360° 2D samples provided 90.67%. To be specific, the classification performances in Figure 5 and Figure 6 illustrate class-specified accuracies in the 3D and 2D scenarios respectively.

Figure 6 demonstrates that in 2D scenario the classifier suffers from higher classification errors on cubicle (blue) and room (brown) categories. By contrast the 3D information is able to provide adequate discrimination because Figure 5 shows that the class-specified accuracies on 3D dataset are remarkably high and uniform.

The error sources of misclassification come from the presence of people and furniture in the environment, ambiguity of classes and in-class variations. Moreover, both 3D and 2D range sensors "see through" glass walls in the environment, introducing another error source in both cases. This error source may be minimized by further analyzing the depth images, which was not carried out in the current implementation.

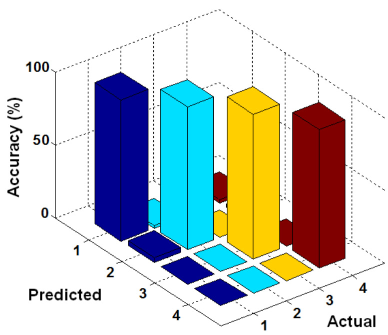


Figure 5. Classification accuracies of using SVM classifier on 360° 3D dataset. Blue, cyan, yellow and brown bars represent accuracies (96.71% / 98.06% / 99.29% / 94.74%) on cubicle, corridor, common area and room categories respectively. The standard deviation of class-specific accuracies is 1.95%.

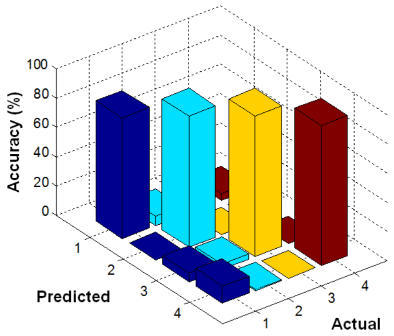


Figure 6. Classification accuracies of using SVM classifier on 360° 2D dataset. blue, cyan, yellow and brown bars represent accuracies (82.24% / 89.68% / 95.74% / 95.39%) on cubicle, corridor, common area and room categories respectively. The standard deviation of class-specific accuracies is 6.32%.

4.3 Effect of Coverage

As a single snapshot of the Kinect™ sensor provides a limited field of view (57° horizontal by 43° vertical). A sample used in the above experiment comprised of six snapshots of different directions to cover a 360° of view. However, as a closer examination of the issue, a further investigation on sensor coverage and accuracy was conducted.

In this experiment, different sensor combinations were considered and processed for classification.

Figure 7 shows the accuracies of different coverage schemes. It is rather an expected and reasonable observation that by registering adjacent snapshots gradually, classification accuracies on 3D samples increased in a near-linear manner. The accuracies are 62.17%, 71.33%, 79.50%, 86.17%, 93.83% and 97.17% respectively for combinations of one to six snapshots, with a correlation coefficient 0.99 between the number of registered snapshots and the accuracies. However, by arranging the sensor readings more symmetrically around the robot, the classification performance overwhelmed corresponding unsymmetrical configurations (81.33% vs. 71.33% for a combination of two snapshots, and 88.00% vs. 79.50% for a combination of three snapshots).

When comparing with samples from the 2D sensor, a single snapshot of 2D sensor covering 270° field of view provided more useful information and thus outperformed its 3D counterpart which covered about 60° (horizontal) field of view, with an accuracy of 77.67% vs. 62.17%. However, a combination of two symmetrical snapshots of the 3D sensor covering about 120° field of view showed an accuracy of 81.33%, which is better than a single snapshot of the 2D sensor.

Therefore, in practical implementations, a proper 3D data acquisition scheme which registers symmetrical snapshots provided by either a motorized sensor or multiple physically mounted sensors is recommended.

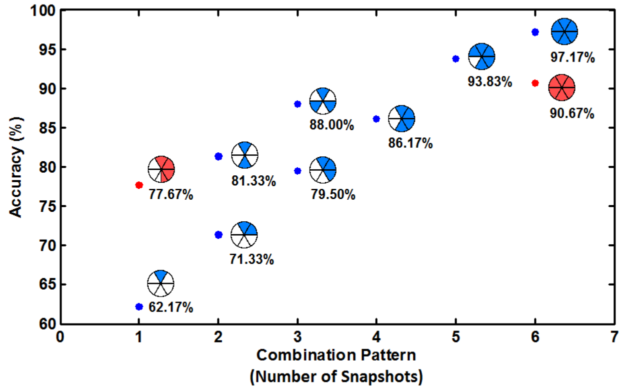


Figure 7. Classification accuracies on different coverage schemes. The blue and red points indicate 3D and 2D scenarios respectively and the filled pattern of the circle reflects the sensor coverage.
(Note: as a single snapshot from 2D sensor provides the coverage of 270°, it does not necessarily need 6 snapshots to cover 360°. Therefore an overlapping exists and is accepted in this experiment.)

5 Conclusion

In this paper, we have proposed a feature set for unorganized 3D point clouds and then presented a supervised learning approach to discriminate the environment around a robot into four categories. DEFS algorithm has been employed to select optimized subset of features and SVM has been applied on the classification tasks. In a similar manner, corresponding observations from 2D laser range finder has been processed for comparison purposes. The comparison results between 3D and 2D datasets demonstrated the superiority of the 3D dataset.

Experiments on the dataset collected by a robot operating in an indoor environment demonstrated that: 1) the feature extraction algorithm spends 0.88s on average to obtain four single-valued features from a typical 360° 3D sample comprising about 1.5 million unorganized points; 2) the classification stage achieves 97.17% gross accuracies using SVM classifier, with a standard deviation of 1.95% on class-specified accuracies; 3) with different data acquisition schemes, a range of solutions with 62.17% to 97.17% accuracies were available to cope with various requirements and constraints.

References

[Shi et al., 2010a] Lei Shi, Sarath Kodagoda and Gamini Dissanayake, Laser range data based semantic labeling of places, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5941 - 5946, 2010.

[Buschka and Saffiotti, 2002] Par Buschka and Alessandro Saffiotti, A virtual sensor for room detection, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 637 - 642, 2002.

[Mozos et al., 2005] Oscar M. Mozos, Cyrill Stachniss, and Wolfram Burgard, Supervised Learning of Places from Range Data using AdaBoost, *Proc. IEEE International Conference on Robotics and Automation*. pp. 1730 - 1735, 2005.

[Sousa et al., 2007] Pedro Sousa, Rui Araujo, and Urbano Nunes, Real-Time Labeling of Places using Support Vector Machines, *IEEE International Symposium on Industrial Electronics*, pp. 2022 - 2027, 2007.

[Shi et al., 2010b] Lei Shi, Sarath Kodagoda and Gamini Dissanayake, Multi-class classification for semantic labeling of places. *Proc. 11th International Conference on Control, Automation, Robotics & Vision*, pp. 2307 – 2312, 2010.

[Shi and Samarabandu, 2006] Wenxia Shi and Jagath Samarabandu, Investigating the Performance of Corridor and Door Detection Algorithms in Different Environments, *International Conference on Information and Automation*, pp. 206 - 211, 2006.

[Viswanathan et al., 2009] Pooja Viswanathan, David Meger, Tristram Southey, James J. Little and Alan Mackworth, Automated Spatial-Semantic Modeling with Applications to Place Labeling and Informed Search, *Canadian Conference on Computer and Robot Vision*, pp. 284 - 291, 2009.

[Quattoni and Torralba, 2002] Ariadna Quattoni and Antonio Torralba, Recognizing indoor scenes, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 413 - 420, 2009.

[Xiong and Huber, 2010] Xuehan Xiong and Daniel Huber, Using Context to Create Semantic 3d Models of Indoor Environments, *British Machine Vision Conference*, 2010.

[Rusu et al., 2009] Radu B. Rusu, Zoltan C. Marton, Nico Blodow, Andreas Holzbach and Michael Beetz, Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3601 - 3608, 2009.

[Gibbins and Swierkowski, 2009] Danny Gibbins and Leszek Swierkowski, A comparison of terrain classification using local feature measurements of 3-dimensional colour point-cloud data, *24th International Conference Image and Vision Computing*, pp. 293 - 298, 2009.

[Munoz et al., 2007] Daniel Munoz, Nicolas Vandapel and Martial Hebert, automatic 3d point cloud classification of urban environments, *Security*, 2007.

[Lodha et al., 2007] Suresh K. Lodha, Darren M. Fitzpatrick and David P. Helmbold, Aerial Lidar Data Classification using AdaBoost, *6th International Conference on 3-D Digital Imaging and Modeling*, pp. 435 - 442, 2007.

[Ng, 2004] Andrew Y. Ng, Feature selection, L1 vs. L2 regularization, and rotational invariance, *International Conference on Machine Learning*, 2004.

[Osada et al., 2001] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, Matching 3D models with shape distributions, *International Conference on Shape Modeling and Applications*, pp. 154 - 166, 2001.

[Gumhold et al., 2001] Stefan Gumhold, Xinlong Wang and Rob MacLeod, Feature Extraction from Point Clouds, *Proc. 10. Int. Meshing Roundtable*, 2001.

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

[Sadjadi and Hall, 1980] Firooz A. Sadjadi and Ernest L. Hall, Three-Dimensional Moment Invariants, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 127 - 136, 1980.

[Vapnik, 1999] Vladimir N. Vapnik, *The nature of statistical learning theory*, New York: Springer, 1999.

[Xue et al., 2010] Hui Xue, Qiang Yang, and Songcan Chen, SVM: Support Vector Machines, in *The top ten algorithms in data mining*, Xindong Wu and Vipin Kumar, Ed, Boca Raton: CRC Press, 2010, pp. 37 - 47.

[Chang and Lin, 2001] Chih-Chung. Chang and Chin-Jen Lin, LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[Khushaba et al., 2011] Rami N. Khushaba, Ahmed Al-Ani and Adel Al-Jumaily, Feature subset selection using differential evolution and a statistical repair mechanism, *Expert Systems with Applications*, vol. 38, pp. 11515-11526, 2011.

[Hjorth, 1970] Bo Hjorth, EEG analysis based on time domain properties, *Electroencephalography and Clinical Neurophysiology*, vol. 29, pp. 306 - 310, 1970.