

# Semantic Categorization of Outdoor Scenes with Uncertainty Estimates using Multi-Class Gaussian Process Classification

Rohan Paul<sup>‡</sup> Rudolph Triebel<sup>‡</sup> Daniela Rus<sup>†</sup> Paul Newman<sup>‡</sup>

<sup>‡</sup>Mobile Robotics Group, University of Oxford, UK

<sup>†</sup>Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA

{rohanp, rudi, pnewman}@robots.ox.ac.uk and rus@csail.mit.edu

**Abstract**—This paper presents a novel semantic categorization method for 3D point cloud data using supervised, multi-class Gaussian Process (GP) classification. In contrast to other approaches, and particularly Support Vector Machines, which probably are the most used method for this task to date, GPs have the major advantage of providing informative uncertainty estimates about the resulting class labels. As we show in experiments, these uncertainty estimates can either be used to improve the classification by neglecting uncertain class labels or - more importantly - they can serve as an indication of the under-representation of certain classes in the training data. This means that GP classifiers are much better suited in a *life-long* learning framework, where not all classes are represented initially, but instead new training data arrives during the operation of the robot.

## I. INTRODUCTION

To be able to perform complex tasks in its environment and at the same time communicate with a human user on a semantic level, any mobile robotic system needs some kind of semantic information about the environment. In most cases, and also in the context of this work, this semantic information is given in terms of object or class labels attached to sensor data that was acquired by the robot. To obtain such a labeling automatically, a mapping is usually learned from a set of feature vectors extracted from the sensor data into a given set of class labels. This can only be done using *supervised learning* methods, where a human expert manually annotates training examples which are then presented to a classification algorithm. The reason is obvious: class labels are defined by humans and can therefore not be “discovered” with unsupervised or similar learning techniques. In the robotics literature, a large body of work is already available on supervised learning methods for semantic annotation (e.g. object detection, scene analysis). Most of them use learning methods such as AdaBoost [1], Support Vector Machines (SVMs) [2], Probabilistic Graphical Models [3], [4], or other techniques such as Implicit Shape Models (ISM) [5]. Despite the impressive results of some of these systems, they all have one major drawback, which is of particular importance in mobile robotics: They assume the number of different class labels to be known beforehand. This means that the training data has to contain examples of all classes that can potentially be encountered during operation of the robot. All instances of unknown classes are then forced to correspond to one of the known classes, which leads to incorrect classifications.

In this paper, we propose a supervised learning method that has the potential to overcome this drawback.

We achieve this with a classifier that is based on a multi-class Gaussian Process (GP) classification algorithm. As we will show in experiments, our GP classifier can report uncertainty estimates about class labels in cases where the training data contained less classes than encountered in the test set. Thus, from these uncertainties there is implicit evidence that the classifier was trained with too few classes. Furthermore, these uncertainties can be used to select the next sensor observation that should be annotated by the human and added to the training data. This is a key requirement for an *active* learning system that is able to adapt its knowledge as it moves into new environments and thus learns during operation. Such an active and *life-long* learning system is currently the major goal of our line of research, which is the motivation for the need of the multi-class GP classification approach presented in this paper.

## II. RELATED WORK

Several methods for classification and labeling of 3D point cloud data have been presented in the literature. Here we review some related efforts. Anguelov *et al.* [3] proposed a classifier based on an undirected graphical model (UGM), that automatically distinguishes between buildings, trees, shrubs and ground. This was later extended and applied to indoor data by Triebel *et al.* [4]. Posner *et al.* [6] present a multi-level classification framework for semantic annotation of urban maps using vision and laser features. The algorithm combines a probabilistic bag-of-words classifier with a Markov Random Field (MRF) model to incorporate spatial and temporal information. Xiong *et al.* [7] explicitly avoid the use of graphical models and suggest learning contextual relationships between 3D points based on logistic regression.

In contrast, Nüchter and Hertzberg [2] use a Support Vector Machine (SVM) to classify indoor objects in 3D range data. Marton *et al.* [8] introduce global radius-based surface descriptors (GRSD) and then use also an SVM for object classification. Golovinskiy *et al.* [9] segment 3D point clouds and compare several classifiers such as SVMs and random forests to detect objects in urban environments.

Several researchers have applied GPs in robotics mostly for regression applications rather than classification problems. For example, Plagemann *et al.* [10] and Vasudevan

*et al.* [11] use GPs for terrain modeling. Krause *et al.* [12] use GP regression for the problem of optimal placement of sensors and present a near-optimal mutual information based selection criterion. Stachniss *et al.* [13] employ a GP regression to determine a two-dimensional spatial model of gas distributions.

Classification using GPs has been addressed by Murphy and Newman [14], who present an approach for planning paths using terrain classification information from overhead imagery. Image regions are first classified using a multi-class GP classifier followed by spatial interpolation of uncertain terrain costs. Furthermore, Kapoor *et al.* [15] use GPs in an active learning framework for object categorization. However, in contrast to our approach, the problem there is not explicitly modelled as a GP classification problem, but rather as a GP regression where the labels are determined by *least-squares* classification. Also, the authors use a one-vs-all strategy based on binary classification rather than an explicit multi-class classifier.

### III. SEGMENTATION AND FEATURE EXTRACTION

Our algorithm operates on 3D point clouds acquired with a rotating laser scanner device. The first step in our tool chain after acquiring a new 3D point cloud, is to produce a triangular mesh by connecting neighboring data points if they are closer than a given threshold. We then compute normal vectors for all triangles and apply a segmentation algorithm based on the work of Felzenszwalb and Huttenlocher [16], where the similarity of two adjacent triangles is defined by the angles of their normal vectors. Each resulting mesh segment consists of a single connected component and is consistent with respect to the orientation of the triangles it contains. Thus, segments are consistently shaped, e.g. all triangles are all mostly co-planar or they are all similarly distributed in orientation. An example result of our segmentation algorithm can be seen in Figure 1.

In the next step, we compute feature vectors for all mesh segments. We use similar features as in earlier work [17], namely *shape factors*, *shape distributions* based on Euclidean distance, on angles between normal vectors and on the elevation of the normal vectors, and finally *spin images*, where the latter are computed per data point and then an average is computed per mesh segment. As a result, we obtain a 113 dimensional feature vector for each mesh segment, where 50 account for the  $5 \times 10$  spin image, 20 for each shape distribution (i.e. the number of histogram bins) and 3 for the shape factors. These feature vectors, together with a set of ground truth class labels are then fed into the training algorithm of the GP multiclass classifier as described next.

### IV. MULTI-CLASS CLASSIFICATION USING GAUSSIAN PROCESSES

Let  $\mathbf{x} = \mathbf{x}_1, \dots, \mathbf{x}_n$  be a given set of  $n$  feature vectors with dimensionality  $d$  and  $\mathbf{y} = y_1, \dots, y_n$  corresponding class labels where  $y_i \in \{1, \dots, k\}$  and  $k$  is the number of classes. To formulate the multi-class classification problem using a Gaussian Process (GP), a *latent* function  $f_j(\mathbf{x})$  is introduced

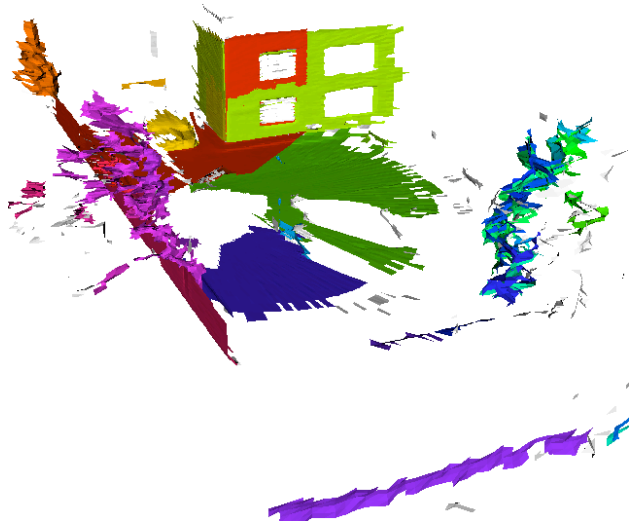


Fig. 1: Mesh segmentation. Every segment has a different color assigned. In this example, the mesh has self-overlapping parts, which is why e.g. the building is split into several segments. Note that “rough” surfaces such as those on the trees are segmented as well as smoother regions such as the ground.

for each class along with the *probit regression* model. The probability of a class label  $y_i$  for a given feature vector  $\mathbf{x}_i$  is defined as:

$$p(y_i = j | \mathbf{x}_i) = \Phi(f_j(\mathbf{x}_i)) \quad i = 1, \dots, n, \quad j = 1, \dots, k, \quad (1)$$

where  $\Phi$  denotes the standard normal cumulative distribution function, i.e.  $\Phi(z) = \int_{-\infty}^z \mathcal{N}(x | 0, 1) dx$ . The latent function  $f$  is represented by a Gaussian Process, determined by a mean function – which in our case is the zero function – and a covariance function  $k(\mathbf{x}_p, \mathbf{x}_q)$ , usually denoted as the *kernel function*. Several different kinds of kernel functions are used in the literature, where the most common ones are the squared exponential, which is also denoted as the Gaussian kernel function. It is defined as

$$k(\mathbf{x}_p, \mathbf{x}_q) = \exp\left(-(\mathbf{x}_p - \mathbf{x}_q)^T D (\mathbf{x}_p - \mathbf{x}_q)\right) \quad p, q = 1, \dots, n, \quad (2)$$

where  $D$  is a  $d \times d$  diagonal matrix. The diagonal entries of  $D$  are known as the *hyper-parameters* of the model.

In contrast to other supervised learning methods for classification such as Support Vector Machines (SVMs), Gaussian Processes are non-parametric, which means that there is no explicit computation of model parameters in the training step. However, in GP classification there still needs to be done some training to obtain the hyperparameters of the covariance function and the posterior distribution of the latent function. More specifically, the aim of the classifier is to find the distribution over values of the latent function given the training data and some test input  $\mathbf{x}^*$ , i.e.

$$\begin{aligned} & p(f^* | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}, \mathbf{x}^*) \\ &= \\ & \int p(f^* | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}^*, \mathbf{f}) p(\mathbf{f} | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}) d\mathbf{f}, \end{aligned} \quad (3)$$

where we use the notation  $\mathbf{x}^*, f^*$  to refer to the test input and its function value. This distribution is then used to compute the class probabilities:

$$p(y^* = j | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}, \mathbf{x}^*) = \int p(y^* | f_j^*) p(f_j^* | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y}) df_j^*. \quad (4)$$

The main problem here is that the latent posterior  $p(\mathbf{f} | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{y})$  is not Gaussian and hence Equation (3) can not be computed in closed form. Therefore, approximations need to be done, and the main approaches to do this are the Laplace approximation and Expectation Propagation, as described in [18]. In this paper, we follow the approach of Girolami and Rogers [19], where a variational Bayes formulation is used. During training the hyperparameters are learned by gradient ascent on the estimated marginal likelihood. Once the hyperparameters and the latent posterior are obtained from training data, inference is performed on new test input by applying Equation (4).

The full GP classification procedure scales as  $O(kn^3)$  where  $k$  is the number of classes and  $n$  is the total number of sample points. The scaling is dominated by the cubic dependence on  $n$  due to the matrix inversion required to obtain the posterior mean for the GP variables. The variational bayes multi-class GP formulation [19] is amenable to a sparse approximation by constraining the maximum number of samples  $s$  included in the model. This results in an  $O(kns^2)$  scaling where  $s \ll n$ . The informative points are picked according to the posterior predictive probability of the target value, intuitively picking points from class boundary regions which are most influential in updating the approximation of the target posteriors. For a detailed exposition please refer to [19], [20] and [21].

Compared to a discriminative classifier like SVM, the GP classification framework offers certain benefits making it particularly suitable for our application. GPs possess a probabilistic formulation and express belief over the latent function via *marginalization* as opposed to *minimization* in SVMs and hence provide uncertainty estimates for the distribution over classification labels [22]. This is more principled than a heuristic approach of using the distance from the classification boundary (margin). A high uncertainty in the GP classification output distribution can give evidence for a category not modeled during training and hence can be used to actively seek examples for incremental training. Additionally, the GP kernel parameters and noise models are interpretable and can be learned without cross validation, which is significant if less data is available for a rare category.

## V. EXPERIMENTAL RESULTS

In the following experiments, four statements will be shown. First, the multi-class GP classifier gives very good classification results on our 3D outdoor data. Second, misclassifications can be detected, because the GP classifier provides uncertainty estimates about the resulting labels. This



Fig. 2: Our robotic car, equipped with a 3D laser range finder on the top of the roof.

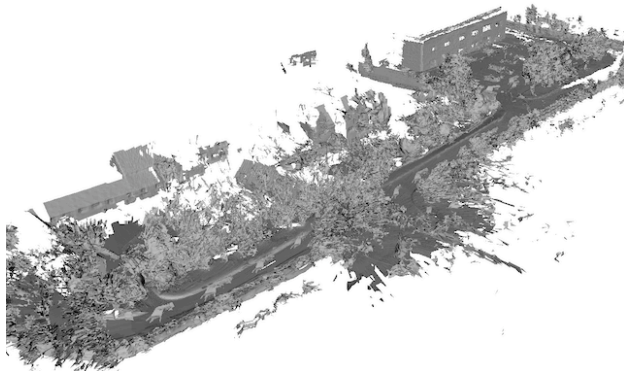


Fig. 3: Mesh representation of our test site. The area mainly consists of buildings, trees, hedges, and roads. The data was acquired with our robotic car “Wildcat”, which is equipped with a 3D laser scanning device and very accurate positioning sensors.

can be used to improve the precision of the classifier even further. Third, in comparison with SVMs, which are probably the most often used method in robotics, GPs perform at least equally well, even if they are chosen to be sparser than the SVM. And finally and most importantly, when trained with too few classes (in our case two instead of six), the estimated class label uncertainties are much higher when using GPs, making them much more useful for detecting unknown classes in the training set.

### A. Data sets and training

We acquired data with our autonomous car *Wildcat* (see Figure 2), equipped with a 3D scanning device consisting of three SICK LMS-151 laser scanners that are mounted vertically on a rotating turn table. The rotation frequency was set to 0.1Hz. For our experiments, we drove the car slowly ( $\approx 15\text{km/h}$ ) around our research site at Begbroke science park in Oxfordshire. A mesh representation of the acquired data is shown in Figure 3. The data we obtained is comparably dense: each point cloud consists of 100,000 to 150,000 points.

PCA was used for dimensionality reduction retaining 10 principal components from the original 113 dimensional feature vector. Figure 4 (left) plots the eigen magnitudes obtained. Note that very few principal directions capture most of the data variance. To perform the multi-class GP classification we used the Variational Bayes Sparse Gaussian Process approach by Girolami and Rogers [19]. During training the (in-sample) marginal likelihood was monitored

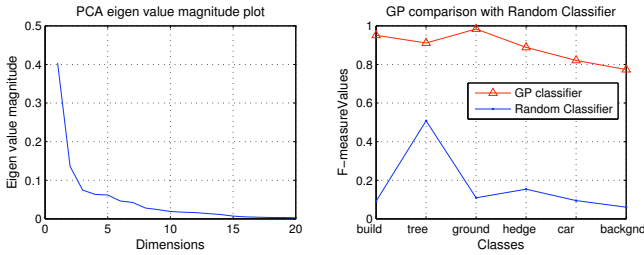


Fig. 4: Left: Plot of eigen value magnitudes after PCA on 113 dimensional feature vectors. Note that very few principal directions capture most of the data variance. Right:  $F_{0.5}$ -measure comparison of GP classifier with a naive classifier making random decisions based on relative sample frequency. The random classifier performs much worse than the GP classifier.

for convergence within 1% increase tolerance. The process converged for all runs within 45 conjugate gradient iterations.

For evaluating the classification performance of the system, a subset consisting of 1497 segments from 53 lidar point clouds was hand-labeled into six categories frequently encountered in outdoor urban scenes: *building*, *tree*, *ground*, *hedge*, *car* and *background*. The data set was randomly split into test and training with test fractions varying as 0.3, 0.5 and 0.8. Note that our test data was unbalanced, there were more segment instances for some classes like *trees* and *ground* compared to *building* and *cars* due to their shape complexity (reflecting in the number of segments) and natural occurrence frequency in the environment. As suggested in [6], for a more realistic evaluation of the classifier in real settings the data set was not equalized. Thus, we report classifier performance per-class instead of average due to unbalanced class sizes.

### B. Quantitative results

The GP classifier gives a distribution over labels for test data. By taking the maximum-likelihood class assignment, the per-class precision and recall values were estimated and listed in Table I for the run with test fraction 0.5. Precision and recall can be combined into a  $F_\beta$ -measure as given in Equation 5. Here, parameter  $\beta$  refers to the relative importance assigned to recall performance over precision. As suggested in literature [6], we use  $\beta = 0.5$  assigning greater importance to precision accuracy over recall.

$$F_\beta = \frac{(1 + \beta^2)(precision \times recall)}{(\beta^2 precision + recall)} \quad (5)$$

The classifier attains high  $F_{0.5}$ -measure performance for *ground* (0.98) and *building* (0.95) and lower accuracy for classes *hedge* (0.89), *car* (0.82) and *background* (0.77).

Figure 4 (right) compares the GP classifier performance ( $F_{0.5}$ -measure) with a naive classifier making random decisions based on class frequencies in training data. The accuracy of the random classifier is much worse than the GP classifier.

Figure 5 visualizes the confusion matrix where values are normalized along rows. Hence, diagonal values represent per-class recall, indicating the extent to which the ground truth assignments are retrieved. Note that categories *car*, *background* and *hedge* are confused in recall with the *tree*

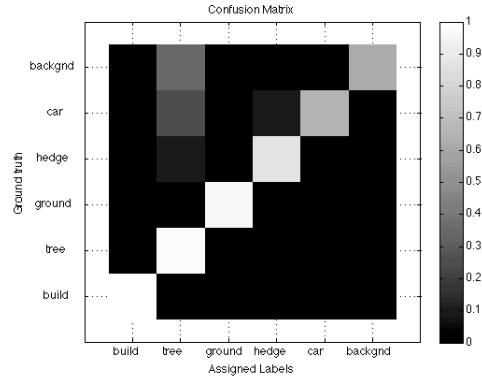


Fig. 5: Confusion matrix (normalized) resulting from the GP classifier. Recall values appear along the diagonal. Results with test fraction: 0.5.

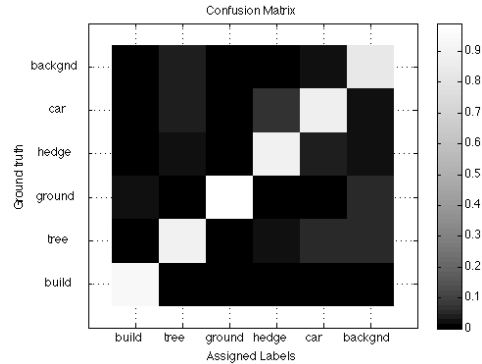


Fig. 6: Confusion matrix (normalized) resulting from the GP classifier. Precision values appear along the diagonal. Results with test fraction: 0.5.

class. Figure 6 presents the confusion matrix with values normalized vertically. Each column represents the accuracy of the classifier labeling and the diagonal values represent precision. Overall, the classifier shows good precision performance. Some confusion is observed between *hedge* and *car* categories.

Next, we calculated the entropy values for each label distribution for a segment to quantify the uncertainty in the classification of that segment. This was normalized to 0 to 1 range by dividing by  $\log(k)$  the maximum entropy of a uniform distribution over  $k$  class labels. An incorrect maximum likelihood assignment frequently results when the label distribution entropy is high. By thresholding the normalized entropy in increments from 0 to 1 and considering assignments only where the classifier is certain above the threshold, the class-specific precision and recall values are calculated. Figure 7 plots the precision-recall curves for two runs with test fractions 0.5 and 0.8. For the case with test

TABLE I: Precision, Recall and  $F_{0.5}$ -measure performance for GP classification for all six categories in the data set. Test fraction: 0.5

Name	train:test	Precision	Recall	$F_{0.5}$ -measure
<b>Building</b>	70 : 62	0.94	1.00	0.95
<b>Tree</b>	362 : 357	0.90	0.98	0.91
<b>Ground</b>	114 : 115	0.99	0.96	0.98
<b>Hedge</b>	91 : 100	0.90	0.86	0.89
<b>Car</b>	74 : 75	0.88	0.64	0.82
<b>Background</b>	37 : 45	0.82	0.62	0.77

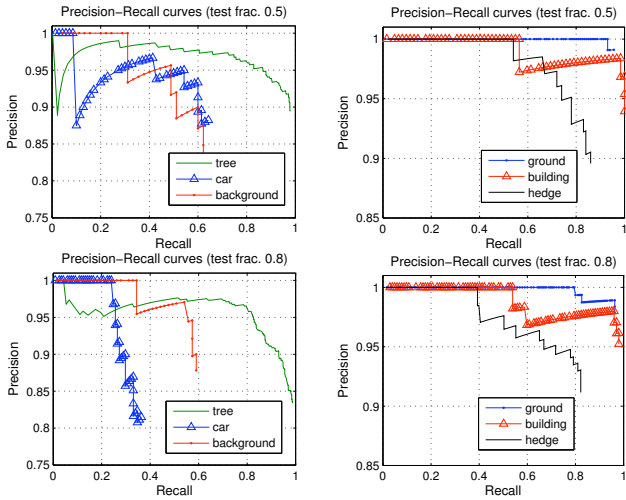


Fig. 7: Precision-recall curves (per-class) obtained by thresholding on the normalized entropy of the label distribution for classes. Left: Classes *tree*, *car* and *background*. Right: Classes *building*, *ground*, and *hedge*. Top: Plots with test fraction: 0.5. Below: Plots with test fraction: 0.8. Note the scale on y-axis.

fraction 0.5, Figure 7 (top), classes *ground*, *building* and *hedge* attain 100% precision at a maximum recall of 93%, 56% and 54% respectively. By accepting a slightly lower precision of 90%, nearly 100% of the ground truth can be retrieved for *building* and *ground* classes and 86% for class *hedge*. The curves for classes *car* and *background* are lower. At 90% precision both classes have a lower recall of 60%.

When the GP is trained with a higher test fraction of 0.8 a general decline is observed in the precision-recall performance as shown in Figure 7 (bottom). The decrease is small for classes like *building*, *ground* and *hedge* and is more significant for the class *car* for which recall decreases from 60% to 27% at 90% precision level. In both experiments, class *tree* displays a gradually declining curve which may be attributed to significant variation in the entropy values from a large number and varied segments obtained as category samples. In general, using a lower normalized entropy threshold was found to improve precision at the cost of lowering recall since uncertain true positive class labels are also suppressed. This allows the user to obtain an application specific value of the entropy threshold.

### C. Qualitative results

Figure 8 shows an example of the classification result for one triangle mesh from our data set. The left image shows the ground truth labeling obtained from manual annotation. The center image depicts our classification result using the multi-class GP classifier. One can clearly see that there are only minor classification errors. The most obvious ones are in the front on the hedge surrounding the car park. Here, the classifier generated the label *car*. However, the labels in that area are not very certain, which can be seen from the right image in the figure. Here, the normalized entropy is visualized with color values between green (no entropy) and red (entropy equal to 1). We can see that the class label distributions of the segments in the front have a much higher entropy than others such as those on the ground. This

means that the classification result can be improved even further, if required, by neglecting all label assignments where the entropy of the label distribution is too high. Of course, such a conservative classifier will have a weaker recall performance (as shown in the previous section), but in some applications the reduction of false-positive classifications is more important.

Figure 9 shows the classification result of 9 consecutive meshes in one common image. We note that there are slight labeling errors even in the ground truth (left image). This is caused by imperfections in the segmentation process, which lead to under-segmentation. For example, some few segments contain sensor readings from the building and the ground. As it is impossible to determine a single *true* label for these segments, we decided to assign the ground truth label based on a majority voting during annotation. From the figure we can see that the qualitative result corroborates the outcome of the quantitative evaluation: in general, the classification is very good, only the under-represented classes such as *car* and *hedge* are classified slightly worse.

### D. Classifier Comparison

We compared the generative GP classifier with a discriminative SVM classifier using the LIBSVM implementation of Chang and Lin [23]. In all cases, we employed the squared exponential kernel to facilitate comparison. Table II compares the  $F_{0.5}$ -measure performance of the two classifiers with test fractions: 0.3, 0.5 and 0.8. The total number of support vectors (indicating model sparsity) obtained during SVM training were noted for each run. The GP classifier sparsity parameter  $S$  was set to a value close to but smaller than the number of support vectors used by SVM. The  $F_{0.5}$ -measure performance for the GP and SVM classifiers was very similar, even with a sparser representation used for the GP. The experimental result accords with similar findings by Naish-Guzman *et al.* in [24].

Next, we compared the uncertainty estimates of the probabilistic classification output of the two classifiers to new object classes not used in training. We trained the GP and SVM classifiers only on segments from two classes (randomly picked): *building* and *ground*. Data from the remaining *un-modeled* classes was presented to both classifiers for inference, resulting in a classification distribution over binary labels. The normalized entropy values measuring uncertainty in the classification decision were computed for each label distribution.

Figure 10 presents the normalized entropy histograms for the inference set. The SVM classifier commits a large majority of the *un-modeled* points to one of the *modeled* classes with high certainty, resulting in a peaked distribution over one of the two labels. As a result, for a majority of the data points, the label distribution has lower normalized entropy. In contrast, the GP classifier assigns higher normalized entropy for a majority of the test points. The same pattern was found consistent for other choices of training and testing classes. The classifier uncertainty for the test points from new classes is expressed as a more uncertain (uniform) distribution over

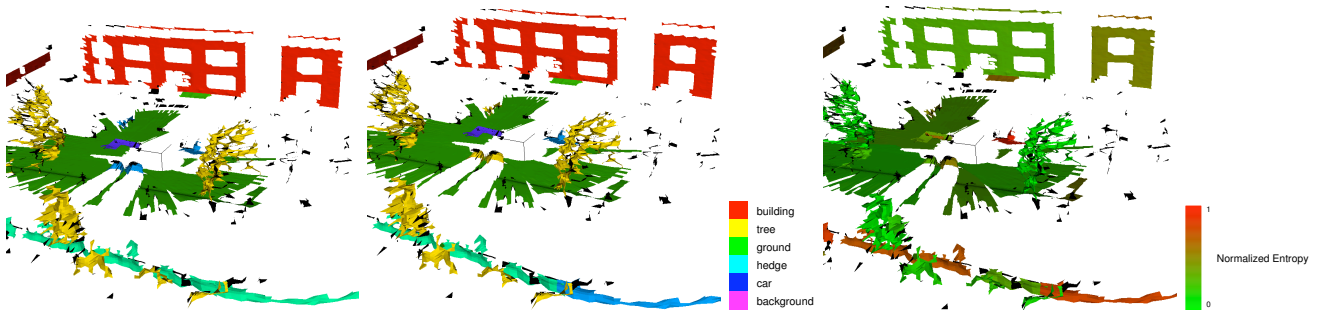


Fig. 8: Classification result and normalized entropy for one example mesh. **Left:** Ground truth labeling. In this scene, no background objects were present. **Center:** Classification result using multi-class GP classification. Note the classification error of the *hedge* in the front, which is classified as *car*. **Right:** Normalized entropy of the class label distributions for each mesh segment. For most segments the classifier is very confident. For some, such as the (wrong classified) *hedge* in the front, the normalized entropy is high and thus the classification confidence low.

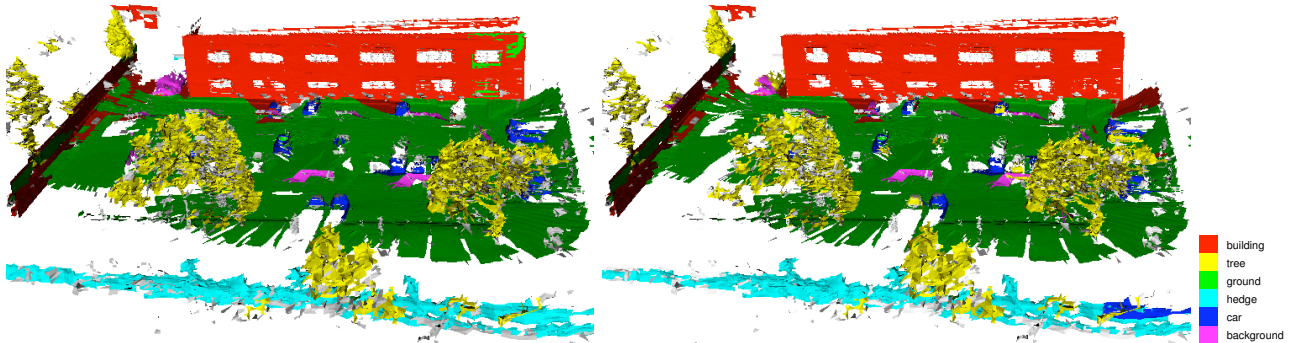


Fig. 9: Classification result after 9 point clouds (time steps). **Left:** Ground truth. Note that even in the ground truth some areas are not labeled correctly, e.g. on the ground close to the *building*. This is due to the fact that the mesh segmentation is not perfect and a correct manual labeling of segments that actually correspond to more than one class is not possible. In our evaluation we abstract from such segmentation errors. **Right:** Classification result. Only minor errors are visible. Note again the *hedge* in the front, but also on some *cars*.

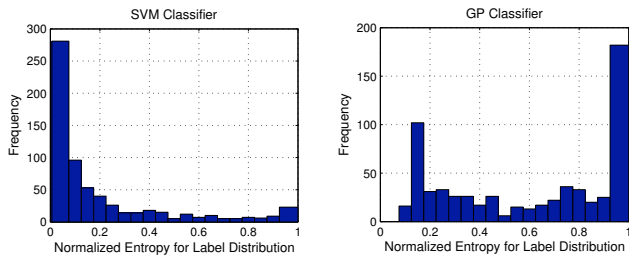


Fig. 10: Histogram of normalized entropy values of the label distribution for SVM and GP classifiers. Both classifiers were trained explicitly on two classes. The data from the remaining classes was presented for inference. **Left:** SVM classifier assigns a majority of the points to a particular class with high certainty. **Right:** As a contrast, GP classifier assigns greater classification uncertainty to a majority of the points, providing evidence for a potential new class. Note the scale on y-axis.

labels, indicating the presence of one or more potentially *un-modeled* classes.

## VI. CONCLUSIONS AND FUTURE WORK

The mid-term goal in our current research is an actively learning mobile robotic system that acquires semantic knowledge by supervision, but during system operation, i.e. in a *life-long* learning framework. However, this knowledge needs to be added *incrementally* and *selectively*, because no human would be willing to annotate all new sensor observations from the robot. Unfortunately, none of the currently used supervised learning algorithms can provide sufficient means

to select the next observation that needs human annotation. In this paper, we show that when using multi-class GP classification this selection actually *can* be done based on the uncertainty estimates of the class labels that the GP classifier inherently provides. We also show that there is no loss in performance when using a GP classifier, even if a higher level of sparsification is chosen. These results demonstrate the power of the GP classifier for this purpose, and thus provide an important step towards *life-long* learning robot systems.

## VII. ACKNOWLEDGEMENTS

Authors wish to thank Dr. Ingmar Posner for insightful discussions and Dr. Benjamin Davis for maintaining the robotic platform used for this work. Paul Newman was supported by an EPSRC Leadership Fellowship, EPSRC Grant EP/I005021/1. Daniela Rus was supported for this work in parts by the MAST Project under ARL Grant W911NF-08-2-0004 and ONR MURI Grants N00014-09-1-1051 and N00014-09-1-1031.

## REFERENCES

- [1] Ó. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, “Supervised semantic labeling of places using information extracted from sensor data,” *Journal on Robotics and Autonomous Systems (RAS)*, vol. 55, no. 5, pp. 391–402, 2007.
- [2] A. Nüchter and J. Hertzberg, “Towards semantic maps for mobile robots,” *Journal of Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

TABLE II:  $F_{0.5}$ -measure classification performance comparison for GP and SVM with varying test data fractions.

Experiment	Test Fraction 0.3		Test Fraction 0.5		Test Fraction 0.8	
	GP #s : 450	SVM #sv : 547	GP #s : 300	SVM #sv : 411	GP #s : 100	SVM #sv : 179
<b>Building</b>	0.99	0.99	0.95	0.95	0.96	0.96
<b>Tree</b>	0.89	0.91	0.92	0.92	0.86	0.88
<b>Ground</b>	0.97	0.97	0.98	0.97	0.98	0.98
<b>Hedge</b>	0.83	0.82	0.87	0.87	0.88	0.91
<b>Car</b>	0.66	0.67	0.84	0.82	0.63	0.68
<b>Background</b>	0.62	0.73	0.75	0.82	0.79	0.77

- [3] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of markov random fields for segmentation of 3d scan data," in *IEEE Conf. on Comp. Vis. and Pat. Recog. (CVPR)*, 2005, pp. 169–176.
- [4] R. Triebel, R. Schmidt, O. M. Mozos, and W. Burgard, "Instance-based AMN classification for improved object recognition in 2d and 3d laser range data," in *Proc. of the Intern. Joint Conf. on Artificial Intell.*, 2007.
- [5] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3d range data," in *special track on Physically Grounded AI of AAAI*, 2010.
- [6] I. Posner, M. Cummins, and P. Newman, "A generative framework for fast urban labeling using spatial and temporal context," *Autonomous Robots*, vol. 26, no. 2, pp. 153–170, 2009.
- [7] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-d scene analysis via sequenced predictions over points and regions," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.
- [8] Z. C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D-3D Categorization and Classification for Multimodal Perception Systems," *The International Journal of Robotics Research*, 2011, accepted for publication.
- [9] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3d point clouds in urban environments," in *International Conference on Computer Vision (ICCV)*, 2009.
- [10] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "Learning predictive terrain models for legged robot locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.
- [11] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.
- [12] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *The Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.
- [13] C. Stachniss, C. Plagemann, and A. J. Lilienthal, "Gas distribution modeling using sparse gaussian process mixtures," *Autonomous Robots*, vol. 26, no. 2-3, pp. 187–202, April 2009.
- [14] L. Murphy and P. Newman, "Planning most-likely paths from overhead imagery," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010.
- [15] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [16] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [17] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and unsupervised part-based discovery of repetitive objects," *Proceedings of Robotics: Science and Systems*, Jan 2010.
- [18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006, no. ISBN 0-262-18253-X.
- [19] M. Girolami and S. Rogers, "Variational bayesian multinomial probit regression with gaussian process priors," *Neural Computation*, vol. 18, no. 8, pp. 1790–1817, 2006.
- [20] M. Seeger and M. Jordan, "Sparse gaussian process classification with multiple classes," Citeseer, Tech. Rep., 2004.
- [21] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse gaussian process methods: The informative vector machine," *Advances in neural information processing systems*, vol. 15, pp. 609–616, 2002.
- [22] C. Williams and C. Rasmussen, "Gaussian processes for machine learning," 2006.
- [23] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [24] A. Naish-Guzman and S. Holden, "The generalized fitc approximation," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1057–1064, 2008.