Online Generation of Scene Descriptions in Urban Environments [1]

Ingmar Posner, Derik Schroeter, Paul Newman
Robotics Research Group, Dept. Engineering Science,
Oxford University, Parks Road, Oxford, OX1 3PJ, United Kingdom
{hip,ds,pnewman}@robots.ox.ac.uk

Abstract

The ability to extract a rich set of semantic workspace labels from sensor data gathered in complex environments is a fundamental prerequisite to any form of semantic reasoning in mobile robotics. In this paper we present an online system for the augmentation of maps of outdoor urban environments with such higher-order, semantic labels. The system employs a shallow supervised classification hierarchy to classify scene attributes consisting of a mixture of 2D/3D geometric and visual scene information into a range of different workspace classes. The union of classifier responses yields a rich, composite description of the local workspace. We present extensive experimental results using two large urban data sets collected by our research platform.

Key words: semantic robot maps, outdoor mapping, support vector machine

1. Introduction

Significant advances in recent years in the development of localisation and mapping frameworks have inspired an expectation for mobile robots to operate in increasingly complex environments, both autonomously and in concert with human beings. In recent years, appearance-based techniques developed in the computer vision domain have emerged as a valuable complement to standard SLAM solutions [40,32]. As a result, mapping techniques have reached adolescence in the sense that low-level geometric representations can be built for an environment over several hundred meters of track [26]. However, the maps that are produced are typically agglomerations of laser points or an arrangement of geometric primitives (often simply points, lines and planes). Such representations only have a limited discriminative capacity and fail to adequately represent

[1] The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre (DTC) established by the UK Ministry of Defence.

the subtleties of complex environments. In particular, they are of limited use to the operational decisions required by an autonomous agent.

We argue that successful environmental interaction in complex outdoor urban environments requires at least a rudimentary operational awareness of higher-level *semantic* concepts. At the most basic level, such an operational awareness can be obtained by automatically extracting meaningful and pertinent semantic labels from a range of sensor data. Consider, for example, a navigational policy which prefers operation on pavements (appropriate for our ATRV Junior vehicles) to operating on busy roads. Similarly, exploration can be driven by semantic cues in the sense that roads, pavements and paths lead to other, possibly interesting places. The extraction of appropriate labels thus forms the basis of effective semantic reasoning.

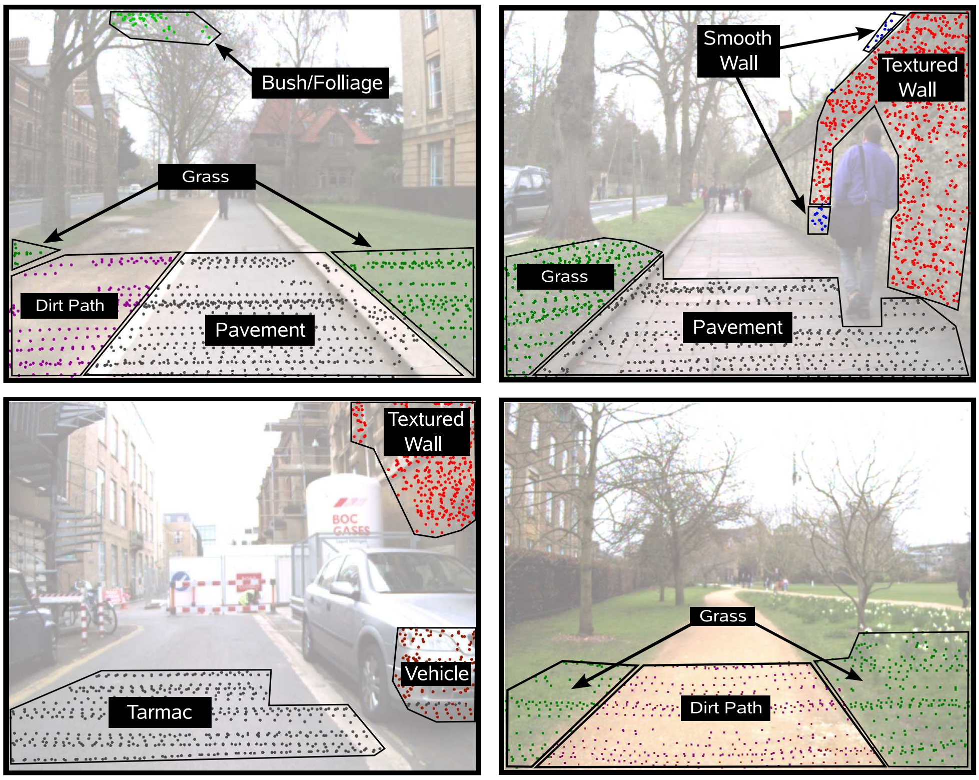

Fig. 1. Semantic labels for typical urban scenes. The images depict the output produced by the presented system, except for the text boxes and arrows that were added manually for illustration purposes. Note that the points in the images refer to 3D points in the robot's workspace. An alternative representation would be a coloured 3D point cloud, where the colour encodes the semantic labels.

This paper presents an appearance-based method for augmenting maps of outdoor urban environments with higher-order, semantic labels using both scene appearance and 2D/3D geometry. A 3D laser scanner is utilised to sense the local workspace geometry and a camera to capture its visual appearance. In combination these two sensors provide a rich source of information with which to characterise different aspects of the local area. In particular, we will focus on describing regions of the ground plane, the surface type of walls, and the presence or otherwise of vehicles and foliage. The geometric and visual properties of a particular scene are passed through a shallow hierarchy of classifiers each trained to respond to a given scene attribute — like pavement, tarmac or bush. The combination of all positive classifications yields a composite description of the scene in question, for example, *'Path and Grass and Foliage'* or *'Road and Brick-Wall and Car'*, see Fig. 1.

We make several extensions to our previous work [34], as presented at the IEEE International Conference on Robotics and Automation (ICRA) in 2007. Our classification framework has been substantially refined and restructured, leading to significant savings in computational cost. This enables us to present here a system which runs online and in near real time. Local smoothing of the classification results is achieved by majority consensus based on automatic image segmentation. Furthermore, we consider the benefits gained by leveraging a much richer set of features in both 3D geometry and visual appearance and we provide a detailed comparison motivating our final choice. In addition, we elaborate on our method for automatic 3D laser/vision cross-calibration. Finally, we present an extensive analysis of system performance in the field.

The next section gives a comprehensive account of related works. Section 3 describes the research platform and data used for the experimental validation of our approach. This is followed by a motivation of our choice of workspace labels in Section 4. A description of the features considered as well as the feature extraction stage of our processing pipeline is given in Section 5. Section 6 describes the classification framework. Finally, the efficacy of the system in a real urban setting is demonstrated in Section 7, along with a detailed discussion.

2. Related Work

The semantic interpretation of sensor data in the context of robotic map building has received much attention in recent years. The first step usually involves the extraction of suitable features from the data. Given, for example, 2D laser range data, common geometric attributes include 2D lines and corners as well as different moments or heuristics drawn from the distribution of distance and angle measurements. Assuming that certain semantic classes can be characterised and distinguished by means of the resulting feature vector, machine learning techniques [4] are commonly used to address the implied classification problem. Martinez et al. [24], for example, classify 2D range scans into classes such as *'Corridor'*, *'Room'*, and *'Door'*, applying *AdaBoost* and *Hidden Markov Models*. The respective labels are then assigned to the global scan positions leading to a semantic annotation of 2D metric maps of indoor environments. In [8] a rectangular shape model is used to detect rooms from 2D range data. Anguelov et al. [1] propose a method that learns the position of doors in a hallway from 2D line segment maps using the expectation-maximisation (EM) algorithm. The latter problem was also considered in [23], where contextual information is used, by means of relational Markov networks, to classify 2D line segments in indoor hallways as being *'Door'* or *'Wall'*. These and other works indicate that augmenting 2D maps of indoor environments with the explicit notion of doors, hallways and rooms has valuable benefits for robot navigation, in particular, path planning and localisation, see also [36]. Mainly it allows to represent common structural properties and to refer to them by means of semantics. Similar ideas have been put forward in the context of topological map building [21,22], where the focus has been primarily on the definition of distinctive places and the navigation between them, often only with limited use of semantics.

Although sufficient for many applications indoors, the information, semantic or otherwise, that can be extracted from 2D laser range data is rather limited. A natural extension is to utilise 3D data. In [10], for example, elevation maps are used to annotate 2D maps with different navigational behaviours. The notion of *'Floor'*, *'Wall'* and *'Ceiling'* is utilised in [27] to support 3D scan matching in indoor environments. Anguelov et al. [2] use a segmentation of 3D data to detect cars and classify terrain using Graph Cuts on *Markov Random Fields* (MRF). The performance of the MRF framework is compared to that obtained using *Support Vector Machines* (SVMs). This work is closely related to our approach in that we also employ SVMs to classify 3D laser data. However, in combining information from two complementary modalities – 2D/3D geometry and visual appearance – our approach gains the capacity of providing more detailed workspace descriptions, such as the surface-type of buildings encountered or the nature of ground traversed.

Over the last decade, a large body of work in computer vision has focused on the semantic interpretation of image content, in particular object detection and recognition as well as scene description. The resulting algorithms, whether they apply probabilistic feature-based approaches [30] or use 3D geometric models [31] have matured to an impressive level. A detailed overview, however, goes beyond the scope of this paper. It should be noted that most of these works do not address the problem of robotic map building and as such are not directly comparable to the work presented here. However, robot mapping has frequently drawn inspiration and benefit from the field of computer vision. In particular, the use of visual appearance has recently attracted increasing attention, see for example [33,9]. Hadsell et al. [15] use visual appearance to classify outdoor terrain regarding its traversability by a mobile robot. In [33] image similarity is utilised to perform an unsupervised partitioning of outdoor workspaces and thereby defining descriptive classes such as *'Park'* and *'Building'*. Several approaches for the detection of doors in office environments using simple geometric models have been proposed, for example, in [20,35]. Visual appearance has also been successfully applied in topological mapping and place recognition [9], although with limited or no notion of semantics.

Finally, there exists a sizeable amount of work that leverages a combination of sensor modalities. Douillard et al. [11] present a probabilistic framework for object recognition using *Conditional Random Fields* that supports the integration of arbitrarily many sensors. They present preliminary results based on image and 2D range data to detect cars. In [25] similar sensor modalities are utilised to classify cars and pedestrians. The classification is carried out separately for image and 2D range data. The results are combined by a Bayesian sum decision rule. Several approaches to the classification of traversability utilise a monocular camera and a fixed 2D laser range finder that faces downwards in front of the vehicle [42,38]. The assumption is that the 3D pose is known or can be determined with sufficient precision. As a consequence, the laser measurements from different poses can be accumulated and form a 3D point cloud, from which features like planarity or goodness of plane-fit can be computed. Together with visual appearance, these features are used to classify whether or not the terrain in front of the vehicle is traversable. These approaches are related to our work in that they draw their features from image as well as 3D laser range data. However, multi-class classification is not considered. Similar work by Happold et al. [16] utilise 3D data from stereo vision along with appearance features using a neural network for terrain classification.

3. Robot System Setup and Urban Data Sets

The work presented in this paper makes use of two extensive data sets, spanning nearly 18 km of track, gathered with our research platform *Marge* (ATRV, Fig. 2). The robot is equipped with a colour camera (Marlin, Allied Vision Technologies), an inertial sensor (XSens), a GPS sensor and odometry from wheel encoders. The camera records images to the left, the right and the front of the robot in a pre-defined pan-cycle triggered by vehicle odometry at 1.5 metre intervals. 3D laser data are acquired using a 2D laser range finder (SICK) that is run with one degree resolution

(180 degree range). It is mounted in a reciprocating cradle driven by a constant velocity motor, see also [17]. Data were gathered in two different locations: *Jericho/Oxford* (13.2 km, 16,000 images) and the *Oxford Science Park* (3.3 km, 8536 images), see also Fig. 2.

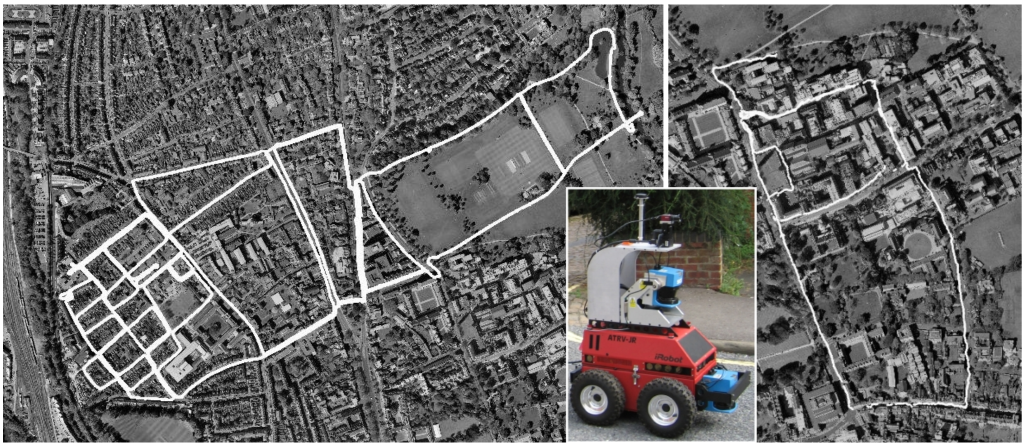

Fig. 2. Aerial map of the *Jericho* data set - 13.2 km, 16000 images (**Left**), and the *Oxford Science Park* data set - 3.3 km, 8536 images (**Right**). Vehicle trajectories are marked in white. **Middle:** *Marge* - our ATRV research platform

4. Workspace Classes in Urban Environments

When navigating in an urban context a higher-order knowledge of the environment is indispensable: self-preservation dictates avoidance of highly dynamic regions such as roads; robust localisation depends on distinguishing features beyond the recognition of ubiquitous general objects such as *'Ground'*, *'Wall'* or *'House'*. This necessity motivates the definition of classes and the closely linked selection of features in this work. Intuitively, in an urban environment places can be distinguished by the type of ground that is present, the colour and texture of surrounding houses (or, more appropriately, of surrounding walls) and the presence or absence of other features such as bushes or trees. The detection of cars (moving or stationary) is also beneficial. These considerations give rise to the classes defined in Tab. 1.

	Class Name	**Description**
Wall	Brick	red or yellow brick
	Nat. Stone	natural stone, sandstone
	Concrete	modern (e.g. concrete, glass)
	Rendered	rendered, plastered, painted
Ground	Pavement	tiled, patched
	Dirt Path	sand, dirt, gravel
	Grass	grass
	Tarmac	common road, pavement
Misc	Bush or Foliage	bushes and parts of trees
	Vehicle	cars or vans

Table 1: Workspace classes.

5. Geometric and Appearance Features

The classes as defined in Tab. 1 suggest that both visual appearance and 2D/3D geometric features are suitable to facilitate reliable classification. For example, it seems straightforward to distinguish between '*Wall*' and '*Ground*' using the 3D plane normal of the 'neighbourhood' of a particular 3D point, but discriminating different '*Ground*' classes using only 3D geometry may be difficult. As described in Sec. 3, our robot platform is equipped with a monocular colour camera and a 3D laser range finder, which supply visual data as well as direct measurements of 3D geometry. Knowing the intrinsic parameters of the camera as well as the relative pose between the two sensors allows for meaningful combination of the information from both. To this end, we developed an accurate cross-calibration method that automatically determines the 3D transformation between the two sensors (Sec. 5.1). As a consequence, each laser measurement, i.e. 3D point, is augmented with appearance information from the image data. The feature extraction takes as input a colour image and a 3D point cloud, and compiles a feature vector incorporating 3D geometric (Sec. 5.2) as well as 2D geometric and visual appearance (Sec. 5.3) features. Sec. 5.4 summarises the overall feature extraction process and the different feature types considered.

5.1. *Camera and 3D Laser Cross-Calibration*

In order to use the information of both a monocular colour camera and a 3D laser range finder in a common frame, the relative position between these sensors must be estimated. Our first approach was to use a planar target as proposed in [28]. However, we found that the accuracy of the target localisation in the 3D point cloud is limited due to (A) the discrete nature of the spatial sampling process as performed by a laser scanner, and (B) the well-known problem of mixed measurements at depth discontinuities. (A) means that 3D measurements - depending on the angular resolution and the distance to the target object - can only be close to the object's boundaries, but never represent its full physical extent. In addition, (B) means that measurements that fall on edges frequently return distance readings that are between the actual object and the background, but lack physical evidence. Together this causes the localisation of the planar target in a 3D point cloud and, in turn, the relative pose estimation between camera and 3D laser scanner to not provide the accuracy we sought for our applications. Therefore, we developed the cross-calibration scheme described here. The primary advantage is that, given the proposed calibration target, the localisation of the target object in the 3D point cloud is performed using robust plane fitting, which is more precise than finding particular corner points or edges directly or assuming that the 3D measurements adequately represent the object's actual size. In fact, if the calibration object is observed (scanned) long enough, the respective planes can be sampled with arbitrary density, and thus, arbitrarily accurate plane estimates can be obtained. Note, since this process is performed while stationary the only source of error is the measurement noise of the laser range finder. Assuming that this noise has zero-mean, it will be compensated for by the plane estimation.

The objective of our approach is to first automatically determine the 3D corner points of both target rectangles, i.e. in the foreground (red) and in the background (white-blue transition), from the image as well as the laser data, see Fig. 3. The resulting 3D corner correspondences between the laser and camera coordinate frame are then used to compute the 3D transformation between the two sensors. From the **Laser Range Data**, 3D corner points are determined using intersections of planes, which are automatically extracted using iterative plane fitting based on MLESAC [39]. Fig. 3 (right) shows the results of this segmentation step. Note that for all but the background plane only circular areas around the plane's centre of gravity are used for the final plane fitting to avoid errors induced by plane segment margins. Using the topology of the calibration target, the 3D corner points are determined by plane-plane and plane-line intersections. The final step applies constrained non-linear optimisation on the 3D point positions to improve the compliance with the calibration target, i.e. the side length of the target rectangles as well as their inner angles.

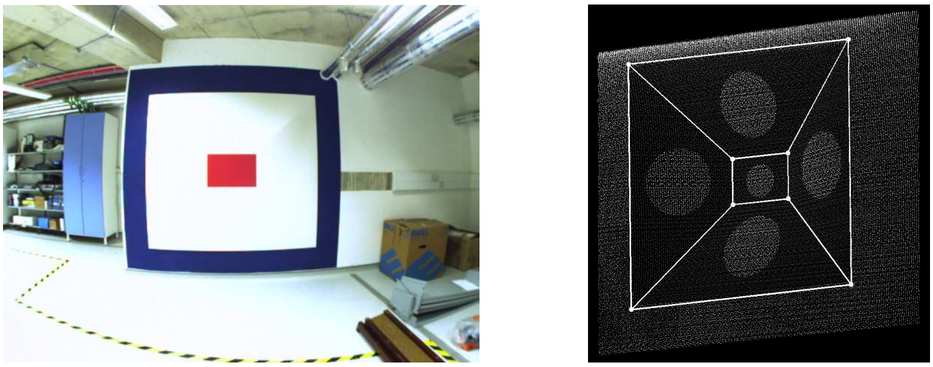

Fig. 3. **Left:** Our 3D calibration target as 'seen' by the camera. **Right:** The target as 'seen' by the 3D laser scanner. Also shown are the 3D planes (grey) and the wire frame model (white) that were automatically segmented (see text below).

From the **Image Data**, 3D corner points are determined by means of projective geometry. After the outlines of the target rectangles have been segmented, the resulting 2D line segments are used to reconstruct the 3D corner points as described in [35]. The sought corner points can be determined up to scale, which is resolved using the known size of the 3D rectangles. As a final step (and similar to the case of laser data), we use constrained non-linear optimisation to allow small deviations of the plane normal and the corner points on the sensor in order to improve the 3D reconstruction. Finally, given the 3D corner correspondences between the laser p^i_L and camera p^i_C coordinate frame, non-linear optimisation is employed to find the parameters for R and t, that minimise the sum of the squared differences between p^i_L and p^i_C, where:

$$p^i_C = R \cdot p^i_L + t, \;\; i = 1...8.$$

The resulting minimum error is in the order of four to twenty millimetres per 3D point. More interestingly, the pixel error of p^i_L and p^i_C back-projected into the image is less than one pixel for all points.

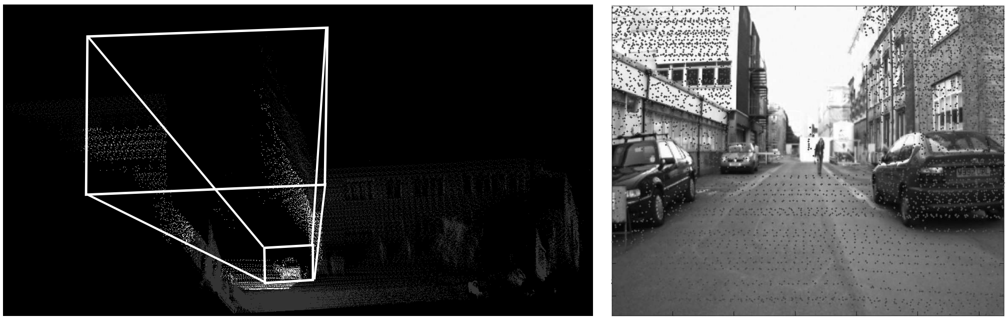

Fig. 4. Camera-laser cross-calibration. **Left:** A typical 3D laser point-cloud. Laser points within the camera frustum are highlighted (white). The frustum outlines have been added for clarity. **Right:** The respective 3D points (from within the frustum) as projected into the corresponding camera image.

5.2. *Extracting Geometric Features from 3D Laser Data*

Given a colour image captured at time t_I, 3D laser points are accumulated over a time window $(t_I - \Delta T,\ t_I)$. The resulting 3D point cloud refers to the same scene that the camera observed at t_I. The *'Wall'* and *'Ground'* classes in Tab. 1 can be approximated geometrically with a planar model. Therefore, the 3D point cloud is first segmented into planar patches following a divide-and-conquer approach outlined in [41]. The given point cloud is discretised into cubic cells and planes are fitted locally using MLESAC [39]. Plane patches obtained in neighbouring cells are merged according to the following constraints, which relate to the relative surface orientation and the distance between plane segments:

$$|\mathbf{n_i} \cdot \mathbf{n_j}| > \arccos(\alpha_{max}) \quad and \quad \tfrac{1}{2}(d_{ij} + d_{ji}) < d_{max}$$

$\mathbf{n_i}$ and $\mathbf{n_j}$ denote the plane normals in cells *i* and *j* and '$\cdot$' denotes the scalar product. d_{ij} and d_{ji} denote the distances from the centre of gravity (cog) of one plane to its orthogonal projection onto the other plane (Fig. 5). α_{max} and d_{max} denote an angle and distance threshold, respectively. Finally, merged plane patches are kept, if they comprise more than N_{min} laser points. A typical result of this segmentation process is shown in Fig. 6.

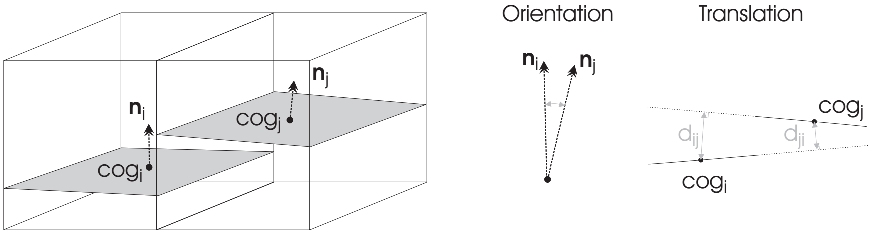

Fig. 5. Plane-merging constraints for two adjacent cubic cells *i* and *j*. **Left:** for orientation. **Right:** for translation. $\mathbf{n}$ - plane normal, *CoG* - centre of gravity

From the segmented plane patches and the respective 3D points we derive the following 3D geometric features that are assigned to each individual 3D point:

- Absolute cosine distance between the normal of the respective plane patch and the normal of the ground plane π_N. The z-axis of the coordinate system (CS) of the laser scanner is aligned with the z-axis of the robot CS, and is pointing upwards. Assuming local approximate planarity, π_N is thus given by the z-axis, i.e. $\pi_N = [0\ 0\ 1]^T$.
- Goodness of plane fit: ratio of smallest/largest SV [2] .
- Patch size: largest × 2nd largest SV, normalised by number of points and subject to a threshold.
- Height of 3D point wrt. ground plane and subject to a threshold.

Note that finally we aim at classifying single 3D points as observed by both the camera and the 3D laser range finder. The fact that certain 3D points stem from the same planar patch and that 3D point classes should be spatially consistent facilitates a post-processing step by means of spatial smoothing using, for example, *Majority Voting* (Sec. 6.4 and 7.3) or *Markov Random Fields*.

5.3. *Extracting Appearance Features from Image Data*

The processing steps as described so far provide 3D laser points which lie on planes fitted to the original laser data, representing the scene depicted in the image and beyond. 3D points that

[2] SV - singular value, comes from the final plane fitting using SVD.

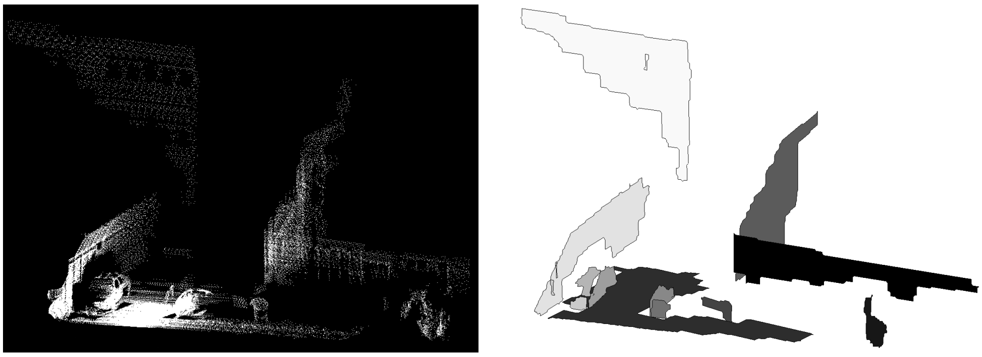

Fig. 6. **Left:** Original 3D point cloud. **Right:** Approximation of the 3D point cloud by planar patches as generated by the segmentation algorithm.

would not project into the image, because they lie outside the camera's viewing frustum, are discarded using frustum culling. The remaining 3D points are projected into the image (Fig. 4) and constitute 2D points of interest (POI). For each of the POIs, appearance features are calculated over a local neighbourhood in the image. These features together with 2D geometric attributes are assigned to the feature vector of the respective 3D point. In this work we consider the following appearance and 2D geometric features:

- Hue and saturation histograms (15 bins each) to characterise colour appearance using a fixed-size neighbourhood of 15×15 pixels.
- Standard deviation of hue and saturation as a simple texture feature using a fixed-size neighbourhood of 15×15 pixels.
- SURF descriptors [3] for the POIs. The scale, and thus the size of the local neighbourhood, is inversely proportional to the distance of the respective 3D points. These descriptors capture primarily texture properties, and are to a certain degree scale, lighting and view point invariant.
- Normalised position of the 2D POIs, as proposed by Hoiem et al [18].

5.4. *Summary*

Fig. 7 shows a flowchart of the processing pipeline that we employ for feature extraction. The 2D/3D geometric and appearance-based features considered in this work are summarised in Tab. 2. This information is used to learn appropriate classifiers that distinguish between the different classes as defined in Tab. 1 (Sec. 4). We address this problem using *Support Vector Machines* in Sec. 6, which describes our classification framework. In Sec. 7.2 we investigate the influence of different feature combinations on the classification performance.

Feature Type	**Dims.**	**Feature Descriptions**
3D Geometry	1	Orientation of surface normal of local planar patch
	1	Quality of plane fit
	1	Size of planar patch
	1	Height of 3D point wrt. the ground plane
2D Geometry	2	Location in image as normalised x and y position
Colour	30	Hue & saturation histograms (15 bins)
Texture	2	Standard deviation of hue & saturation
	64	SURF descriptors

Table 2: Summary of the features considered for classification.

(i) For image I taken at pose x_I and time t_I:
  (a) Obtain 3D laser data (L, t_L) from time window $t_I - \Delta t < t_L < t_I$
(ii) Segment planar patches from 3D point cloud, keep patches that comprise more than N_{min} points. Note that N_{min} is different from the inlier threshold used for MLESAC.
(iii) Filter out 3D points that do not lie within the viewing frustum of the camera (frustum culling).
(iv) For each of the remaining 3D points, see also Tab. 2:
  (a) Assign the 3D geometric features from the respective plane patch.
  (b) Project the 3D point into the image.
  (c) Compute 2D geometric, colour and texture features from local neighbourhood.

Fig. 7. The processing pipeline employed for feature extraction.

6. Classification Framework

In [34] we employ a bank of *Support Vector Machines* (SVMs) for classification. This choice was predominantly motivated by the wide-ranging successes achieved by SVM classifiers. The classification framework adopted here extends our previous work by introducing a hierarchical combination of two distinct discriminative approaches. At the top of the hierarchy a Bayes classifier is employed to distinguish between ground and non-ground classes. For each of these categories a combination of SVMs yields the final class decisions. In addition, the class posterior from the raw SVM output is estimated such that the final classification amounts to a maximum *a posteriori* decision amongst the individual classes [29]. An illustration of the classification framework is given in Fig. 8. The hierarchical approach provides a speed-up of factor two compared to the system presented in [34] and thus constitutes a significant gain in terms of online workspace classification. The remainder of this section describes the individual components of this framework.

6.1. *Bayes Decision Rule for Ground/Non-Ground Separation*

The first step in the classification hierarchy separates ground from non-ground classes. Intuitively, the height (wrt. ground) of the datum as well as the orientation of the plane patch the datum is associated with will be the most conducive to this purpose. For reasons of computational efficiency we propose a simple thresholding scheme on these features. Similar approaches operating on different features have been proposed, for example, in [37,43]. In the work presented here, thresholds are determined such that the resulting probability of misclassification is minimised. This is achieved by employing the Bayes decision rule [12]. Suppose a feature vector $\mathbf{x} \in \Re^2$ derives from two classes C_1 and C_2. A given threshold divides the feature space into two adjacent and non-overlapping volumes, V_1 and V_2. The probability of error is given by

$$p(error) = \int_{V_2} p(\mathbf{x}, C_1) dx + \int_{V_1} p(\mathbf{x}, C_2) dx, \tag{1}$$

where $p(\mathbf{x}, C_i)$ represents the joint probability of feature $\mathbf{x}$ and class C_i. The first and second terms represent the cumulative density of $p(\mathbf{x}, C_1)$ over the volume V_2 and the cumulative density of $p(\mathbf{x}, C_2)$ over the volume V_1. Intuitively, the probability of error is minimised when $\mathbf{x}$ is assigned to that class for which $p(\mathbf{x}, C)$ is at a maximum. In this case the threshold is estimated

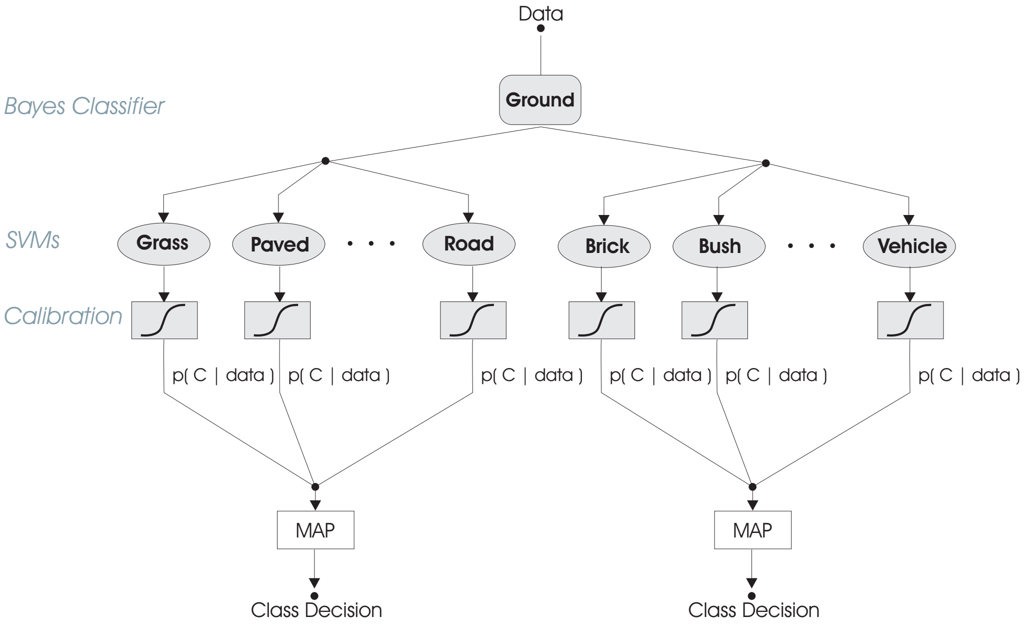

Fig. 8. The classification hierarchy employed in this work.

from the available training data. $p(error)$ is therefore estimated directly for a putative set of threshold values such that

$$p(error) \approx \frac{FP + FN}{N}, \tag{2}$$

where, FP, FN and N denote the number of false positives, false negatives and the total number of data in the training set, respectively. This is directly analogous to Equation 1. Thus, the value which minimises Equation 2 is chosen for further classification.

6.2. *Support Vector Machine Classification*

Support Vector Machines (SVMs) are based on a linear discriminant framework which aims to maximise the margin between two classes. They are a popular choice since the model parameters are found by solving a convex optimisation problem. This is a desirable property since it implies that the final classifier is guaranteed to be the best feasible discriminant given the training data. A detailed discussion of SVM training and classification lies outside the remit of this paper[3]. However, pertinent to the remainder of this section is a brief overview of the mechanism by which future predictions are made.

Consider a set of N training data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathbf{x} \in \Re^d$ denotes a datum in *d*-dimensional feature space. Associated with $\mathcal{X}$ comes a set of labels $\mathcal{Y} = \{y_1, \ldots, y_N\}$ where each $y_i \in \{-1, 1\}$. Once training has been completed, predictions on future observations are made based on the signed distance of the observed feature vector from the optimal hyperplane [7], such that:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{3}$$

[3] For more details on SVM-classification the interested reader is referred to, for example, [4] or [7].

where α_i refers to a Lagrange multiplier associated with datum i, b denotes a bias parameter and $K(\mathbf{x}_i, \mathbf{x}_j)$ denotes the kernel function. Both α_i and b are obtained by training. Note that, in practice, α_i will only be non-zero for a subset of the training data. Members of this subset are referred to as the *support vectors* of the classifier. The kernel function amounts to a scalar product between two data, which have been transformed from d-dimensional feature space into some higher dimensional space. The nature of this mapping between spaces is inherent in the choice of kernel and need not be specified explicitly (the *kernel trick*).

One disadvantage of SVMs lies in the necessary choice of the kernel and the computational burden usually associated with determining the corresponding parameters. In this work we employ a Gaussian kernel [7], which is a common choice and has been found to perform well in a variety of applications. The kernel parameter as well as a trade-off parameter specifying a tolerance for misclassifications during training are commonly determined by grid-search over the parameter space.

SVMs are inherently binary classifiers. However, several schemes exist by which to extend the SVM framework to multi-class problems. In this work, multi-class classification is performed by training a chain of binary classifiers – one for each class – as one-versus-all [7].

6.3. *Probabilistic Calibration*

The use of an inherently binary classification framework such as SVMs in a one-versus-all configuration comes with a caveat: the possibility of individual classifiers assigning an input to multiple classes simultaneously is addressed using a winner-takes-all heuristic where the winner is the classification resulting in the greatest margin (i.e. the largest distance from the separating hyperplane). Even though satisfactory results are obtained in practice, there is no guarantee that the real valued quantities representing the margins for different classifiers will have appropriate scales. This problem can be addressed by a process referred to as probabilistic calibration: the distance of a data point from the separating hyperplane is mapped onto a posterior probability $p(C|f(x))$ where $f(.)$ represents the classification function resulting in an (uncalibrated) distance from the separating hyperplane for each data point x (cf. Equation 3). In this work we adopt a method of probabilistic calibration introduced by Platt [29]. In this approach a parametric model is fitted directly to the posterior probability $p(C|f(x))$. Inspired by empirical data on the class-conditional densities between the margins – Platt observes that they take an exponential form – the parametric model takes the form of a sigmoid as obtained when applying Bayes' rule to two exponentials.

$$p(C|f(x)) = \frac{1}{1 + exp(Af(x) + B)} \tag{4}$$

The parameters A and B are found by minimising the negative log likelihood of the training data. In this work we employ the same model-trust minimisation algorithm used by Platt. The datum is finally assigned to the class with the maximum posterior probability.

6.4. *Voted SVM Classification*

The multi-class SVM approach outlined so far does not take into account information about the spatial cohesion of structures and objects in the real world. Voted SVM classification incorporates this information by assigning a given neighbourhood of data a class label determined by majority consensus of individual, *independent* classifications. In particular, given the probabilistic calibration of the classifiers, a *weighted* majority vote can be performed where the estimated class label $\hat{C}$ is given by

$$\hat{C} = \max_{i} \sum_{r \in N} p(C_i|f(x_r)) \tag{5}$$

where N refers to the set of points in the neighbourhood and $p(C_i|f(x_r))$ is the probability of class i given the uncalibrated SVM output $f(x)$ for datum x (cf. Equation 3). The nature of the data available allows a natural determination of a neighbourhood set N: rather than fixing a distance threshold, neighbourhoods are formed over regions of contiguous appearance within the appropriate image. This patch-segmentation is performed automatically for the results shown in the next section using an off-the-shelf image segmentation method [14], see Fig. 9 for segmentation examples.

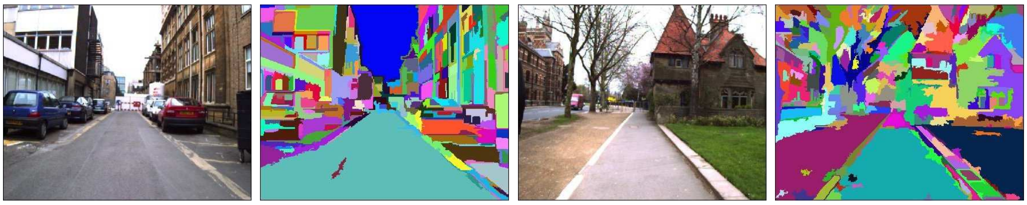

Fig. 9. Examples for the image segmentation used for *Majority Voting* in our online classification system.

7. Experimental Results and Evaluation in Urban Environments

Previous sections have introduced the classification framework and a selection of features. In the following we present results obtained when applying the proposed approach to real data as gathered by a mobile robot. Throughout this section the *Jericho* dataset is used for training purposes. The *Oxford Science Park* data are used as an independent test set. We proceed by deriving the Bayes optimal threshold required for the top level of our classification hierarchy (see Section 6.1). This is followed in Section 7.2 by a description of results obtained with different combinations of features introduced in Section 5. Finally, using a set of selected features, we present more detailed results of system performance on an independent test set.

7.1. *Determining the Bayes Optimal Decision Threshold*

The Bayes optimal threshold required for ground/non-ground separation was determined using approximately 201,000 unbalanced data from the *Jericho* dataset. As indicated in Section 6.1, each datum here consists of a two-element vector associated with a 3D point and containing the relative height above ground as well as the orientation of the associated plane. The left and middle panel of Fig. 10 show the histograms corresponding to the distributions $p(\mathbf{x}, C_1)$ and $p(\mathbf{x}, C_2)$. The right panel illustrates the corresponding $p(error)$ estimated according to Equation 2 using a grid search with a resolution of 100 steps per dimension. The estimate of the Bayes optimal decision threshold results in a mis-classification rate of approximately 2.9%.

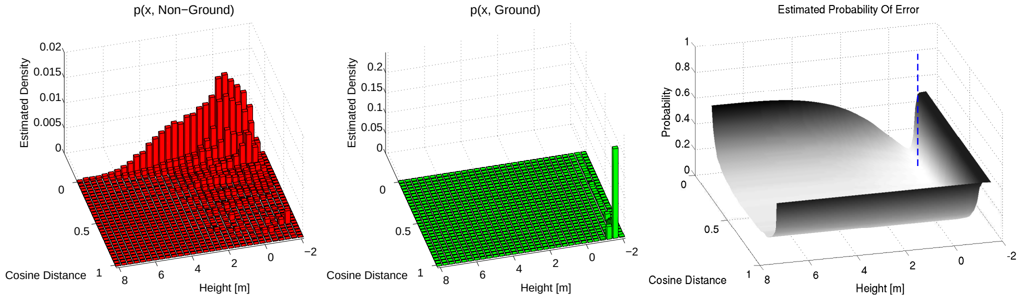

Fig. 10. **Left, Middle:** The joint densities as estimated from training data. Note the difference in scale. **Right:** The estimate of $p(error)$. The dashed vertical line indicates the threshold which minimises the misclassification rate. The cosine-distance of a patch is derived from the normal of the plane of which a point is a member and measured wrt. the normal of the ground plane.

7.2. *Feature-Set Selection*

Section 5 provided a selection of features amongst which to choose. The purpose of this section is to provide an intuition of how system performance varies with the choice of feature combinations. Rather than provide a feature-by-feature analysis, we aim to show that a collection of simple colour-based features and a single geometric feature provides a reasonable speed/performance trade-off compared to more elaborate feature sets. For this purpose we have defined four distinct combinations of features as detailed in Tab. 3, which will provide the input for SVM training and classification.

Name of Feature Set	**Dimension**	**Description**
Minimal	33	Orientation of the surface normal of the local plane patch, normalised $x-$ and y positions within the image, hue- and saturation histograms.
ICRA07	35	All features of the *Minimal* set, standard deviations of hue/saturation histograms. This is identical to the feature set used in [34] from where it takes its name.
Extended Geometry	36	All features of the *Minimal* set, goodness of plane fit, plane patch size, height of 3D point wrt. ground plane.
Ext. Geom. and Texture	100	All the features of the *Extended Geometry* set, SURF descriptors for additional texture information.

Table 3: Feature sets considered in our comparison.

SVM training was conducted using the *Jericho* data set [4] . The appropriate kernel width and the regularisation parameter (i.e. the tolerance for misclassifications) were determined using a grid-search over a section of the parameter space with five-fold cross-validation. The grid-search was conducted with 8,000 training data per class. The data were balanced so that training was conducted at an equal ratio of positive to negative examples. The parameter-set resulting in the highest mean classification accuracy was chosen for each class and the corresponding classifier was re-trained using the entire training set of 8,000 data points. Probabilistic calibration for each class was performed as per Section 6.3 using a hold-out set of 2000 data, again with an equal ratio of positive to negative data. Our comparison of feature sets is based on the ability of the resulting classifiers to separate the relevant class from all other classes. To this end we consider

[4] SVM training and classification were performed using SVMLight [19].

the receiver operating characteristics (ROC) for the classifiers obtained after training using the various feature sets for every class considered in our system. Five-fold cross-validation gives rise to five ROC curves from five independent validation sets for every model considered. Therefore, for every feature set considered, a mean ROC curve was obtained for every class by threshold averaging [13]. These mean ROC curves are shown in Fig. 11. For clarity we avoid the inclusion of error-bars in Fig. 11. Instead, an indication of the variability of classifier response due to the use of different sample sets during cross-validation is provided in the form of the mean and the standard error of the *area under the ROC curve* (AUC). The AUC can be interpreted as equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [13] and provides a convenient single-figure measure of classifier performance commonly used in the machine learning community [5]. The mean and standard error of the AUC have been calculated for every feature set and class as outlined in [5] and are provided in Table 5.

It should be noted that reasonable performance is achieved across all classes even for the worst feature set. However, the richest feature set (*Extended Geometry And Texture*) always performs as well as or better than the others — an intuitive result since the additional information should, by design, aid class separation. Performance gains with respect to the most basic feature set (*Minimal*) are particularly noticeable for the *'Tarmac'* and *'Modern/Glass Wall'* classes, where the added texture information appears to contribute significantly towards the difference. More marginal improvements are achieved for *'Nat. Stone Wall'* and *'Plastered Wall'*. No noticeable improvement is obtained for *'Grass'*, *'Paved'*, *'Dirt Track'*, *'Brick Wall'* or *'Vehicle'*. A further point to note is the consistently equal performance of the classifiers based on the *Minimal* and the *ICRA07* feature sets. This implies that the standard deviations contained in the *ICRA07* feature set do not contribute to class separation and are thus superfluous — this is another intuitive result since the information is already contained in the histograms themselves and is therefore redundant.

Classifier	**Accuracy [%]**	**Precision [%]**	**Recall [%]**
Grass	97.6	97.8	97.4
Paved	81.0	78.8	84.8
Dirt s	82.1	81.7	82.7
Tarmac	88.7	85.5	93.2
Brick Wall	75.7	72.3	83.3
Nat. Stone Wall	84.5	84.4	84.7
Modern/Glass Wall	80.0	74.9	90.4
Rendered Wall	89.6	84.8	96.5
Bushes/Foliage	91.8	92.6	90.9
Vehicles	83.4	82.9	84.2

Table 4: Classifier performance on a balanced hold-out set taken from the *Jericho* data set (2000 data per class). The classifiers are based on the *Minimal* feature set.

In summary, the inclusion of richer geometric and texture-based information only significantly improves the classification result in two cases. For all but these two classes the *Minimal* feature set, based on colour and basic geometry only, remains a competitive alternative. However, there exists a significant difference in computational cost in both the SVM classification and computation of features. The complexity of SVM classification is $O(M \cdot N)$, where M is the number of support vectors (SVs) and N is the dimension of the feature vectors [5] . In our experiments we

[5] In general N is the number of operations necessary to evaluate the distance to one support vector, which in our case is the dimension of the feature vector.

found the number of SVs for the classifiers to be of the same order across the different feature sets. Thus, the dominating factor on SVM classification run-time is the dimension of the feature vector. That means that with less than half the number of features for the *Minimal* feature set as compared to the largest (Table 3), the respective SVM classification is more than twice as fast. This implies a considerable speed-up, given that the SVM classification using the *Minimal* feature set takes about 1.8 seconds on average, as stated in Tab. 4. In addition, extracting more complex textural features like SURF descriptors is computationally expensive due to the relatively large number of points-of-interest (POIs) considered in the presented system. Generally, our system produces of the order of 1500 POIs per image. In comparison, using an image based POI detector usually only 100-400 POIs are found, often less. Consequently, the computation of SURF descriptors in our system would increase the overall processing time, as given in Tab. 4, by about 20 percent. Therefore, with a view towards real-time performance, a decision was made to trade a limited gain in classification accuracy for a notable gain in computational speed by adopting the *Minimal* feature set for this system. Performance figures for the final (i.e. retrained using all available training data) classifiers as applied to a balanced hold-out set are given in Sec. 7.6.

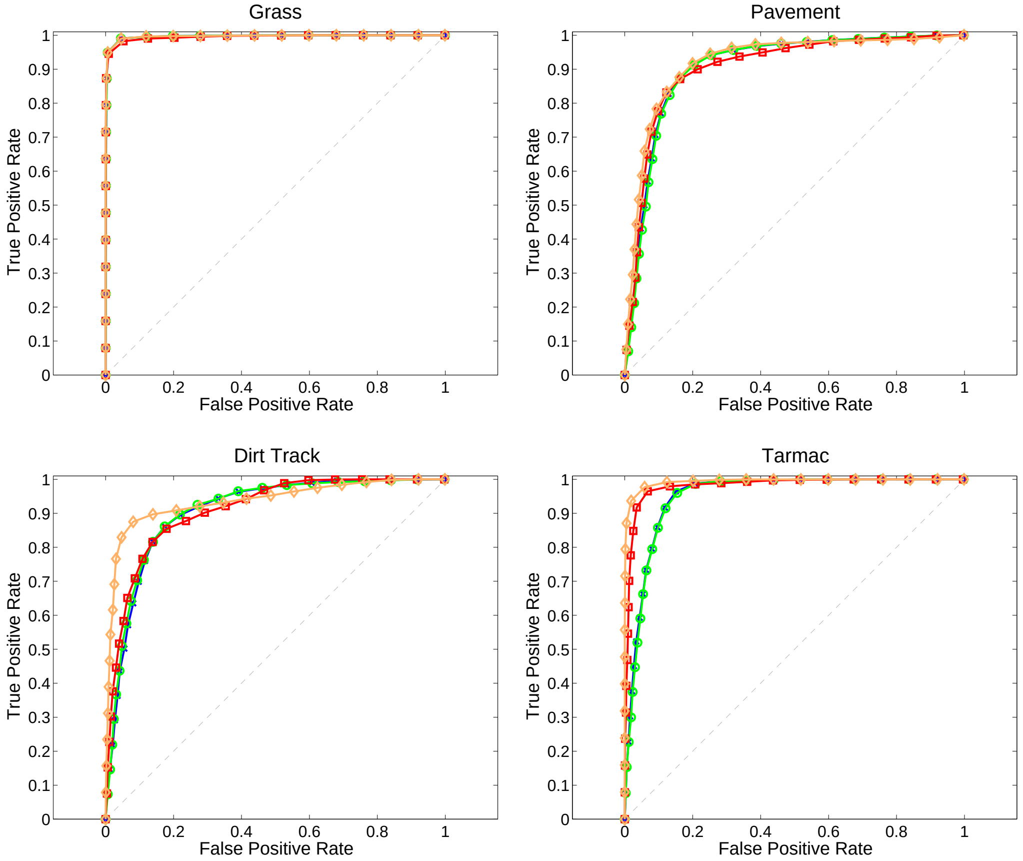

Fig. 11. SVM-ROC curves per class for the feature sets considered: *Minimal* (blue, crosses), *ICRA07* (green, circles), *Extended Geometry* (red, squares), *Extended Geometry and Texture* (orange, diamonds). Each curve represents a combination of results from five independent validation sets as obtained by threshold sampling [13].

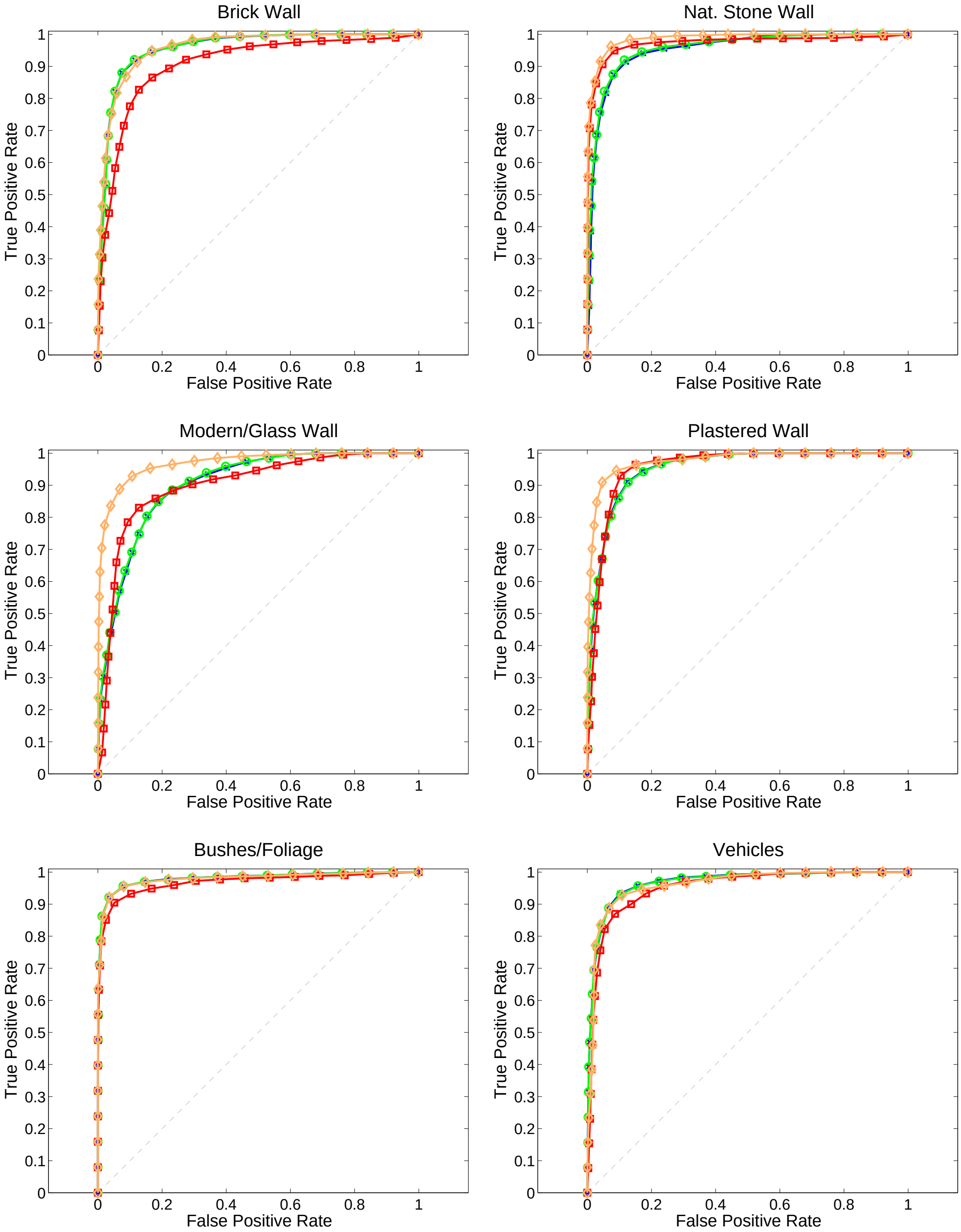

Fig. 11 continued: SVM-ROC curves per class for the feature sets considered.

Feature Set Class	Minimal AUC		ICRA07 AUC		Ext. Geometry AUC		Ext. Geom. and Texture AUC	
	Mean	Std. Error	Mean	Std. Error	Mean	Std. Error	Mean	Std. Error
Grass	0.9968	± 0.0006	0.9968	± 0.0006	0.9957	± 0.0013	0.9969	± 0.0016
Paved	0.9102	± 0.0094	0.9124	± 0.0095	0.9127	± 0.0225	0.9281	± 0.0208
Dirt	0.9077	± 0.0118	0.9050	± 0.0066	0.9093	± 0.0332	0.9360	± 0.0447
Tarmac	0.9513	± 0.0088	0.9531	± 0.0084	0.9825	± 0.0063	0.9946	± 0.0014
Brick Wall	0.9597	± 0.0044	0.9601	± 0.0044	0.9032	± 0.0268	0.9539	± 0.0230
Nat. Stone Wall	0.9588	± 0.0086	0.9549	± 0.0056	0.9730	± 0.0138	0.9879	± 0.0052
Mod./Glass Wall	0.9074	± 0.0155	0.9059	± 0.0160	0.9064	± 0.0472	0.9647	± 0.0210
Rendered Wall	0.9544	± 0.0093	0.9543	± 0.0095	0.9559	± 0.0173	0.9770	± 0.0070
Bush/Foliage	0.9821	± 0.0072	0.9826	± 0.0068	0.9717	± 0.0147	0.9837	± 0.0086
Vehicle	0.9680	± 0.0062	0.9685	± 0.0062	0.9512	± 0.0209	0.9573	± 0.0234

Table 5: Mean and standard error of the area under the curve (AUC) as derived from five-fold cross-validation. The corresponding mean ROC curves are shown in Fig. 11. See text for details.

Class Details			Point-Wise		Voted	
Name	# Patches	# Points	Precision [%]	Recall [%]	Precision [%]	Recall [%]
Gr	99	5393	94.3	91.3	95.3	95.4
Pa	466	11342	21.6	61.9	22.2	69.0
Di	147	7988	37.1	83.4	41.5	84.6
Ta	907	65914	89.8	47.5	92.0	46.5
Br	480	18802	31.0	21.5	32.0	21.2
Na	1760	50739	66.7	56.7	68.6	64.5
Co	437	13037	17.6	17.3	20.6	15.7
Re	469	16844	28.2	42.2	31.1	44.2
Bu	181	8364	66.0	61.4	71.0	66.2
Ve	169	4499	32.5	75.0	35.4	84.6

Legend for class shortcuts: **Gr**ass, **Pa**ved, **Di**rt Path, **Ta**rmac, **Br**ick Wall, **Na**tural Stone Wall, **Co**ncrete Wall, **Re**ndered Wall, **Bu**sh/Foliage, **Ve**hicle

Table 6: Classification results for the *Oxford Science Park* data: Original Classes.

7.3. *Discussion of the Point-Based Classification Performance*

So far in this section the Bayes optimal threshold for ground/non-ground classification has been obtained, an appropriate feature set has been selected and classifiers have been trained together with their respective probabilistic calibrations. Thus, all required components of our classification framework (cf. Fig. 8) are in place. The generalisation performance of the entire system has been tested using labelled data from the *Oxford Science Park* data set (ca. 203,000 data). It should be noted that our test data are unbalanced, in the sense that there are many more instances of some classes than others, reflecting their relative frequency in the world. We deliberately chose not to balance the data because such an evaluation more accurately reflects system performance as obtained online. However, as a consequence, performance figures such as overall or per-class *accuracy* are not informative since they mostly represent classifier performance on the largest class. Instead, we quote the per-class *precision* and *recall*. Detailed numerical results of system performance, for both point-wise classifications and voted classification, using classes as outlined in Section 4 can be found in Tab. 6. We also present a complementary set of results in the form of confusion matrices obtained for voted-SVM classification in Fig. 12. These matrices are normalised, on one hand, such that the values along the diagonals represent per-class *precision* and, on the other hand, such that the values along the diagonals represent per-class *recall*. Thus, the former provides information (along the rows) on how reliable the given labels are compared to ground truth — i.e. how much trust can we put in the obtained labels — whereas the latter provides information (along the columns) of how well ground-truth data are retrieved.

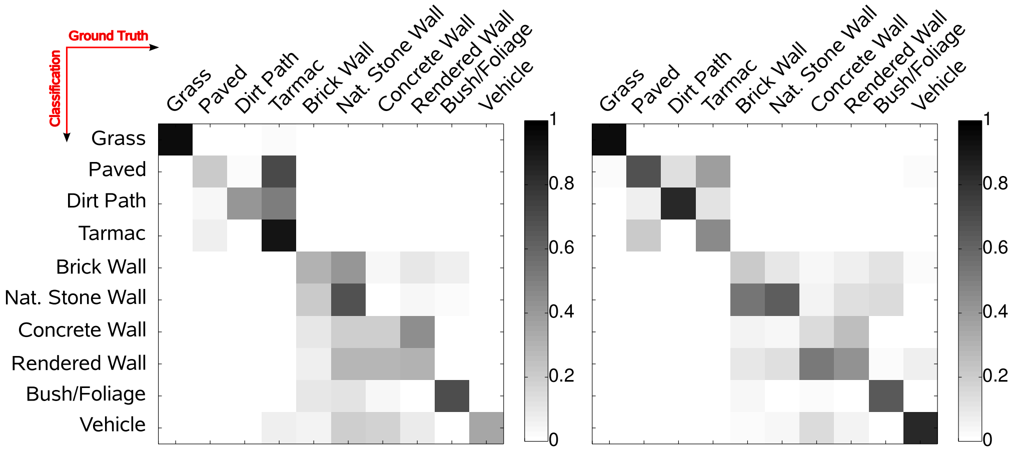

Fig. 12. Confusion matrices for *Oxford Science Park* data obtained using voted-SVM classification. **Left:** Rows are normalised such that the diagonal represents *precision*. **Right:** Columns are normalised such that the diagonal represents *recall*.

Tab. 6 indicates good precision/recall performance in the point-wise classification of *'Tarmac'*, *'Nat. Stone Wall'* and *'Bush'*. Results for *'Grass'* are particularly encouraging. This is attributed to the significant difference in colour between grass and other ground-classes. In comparison, performance of most wall classes other than *'Nat. Stone Wall'* is poor in both precision and recall. *'Brick Wall'*, *'Concrete Wall'*, *'Rendered Wall'* as well as *'Paved'* and *'Dirt Track'* suffer from relatively low precision, implying a high false positive rate. While considering the point-wise classification results it should also be noted that, as expected, the individual performance in precision and recall is consistently worse compared to that obtained on the balanced hold-out data (cf. Tab. 4). This is primarily due to the skew in the number of data for each class present in the test set. It stands to reason that classifiers trained using unbalanced data might perform better in an unbalanced system. We leave this to future work.

7.4. *Incorporating SVM Majority Voting for Patch Classification*

Substantial improvements in performance can be obtained when applying voted-SVM classification where local neighbourhoods are determined automatically as described in Section 6.4. Tab. 6 reveals overall substantial increases in both *precision* and *recall* with only three classes suffering marginally in recall. In this case the image segmentation parameters were determined empirically by inspection of segmentation performance on the original training data. However, significantly larger improvements in performance have been observed when using manually segmented data. It therefore stands to reason that further improvements may be obtained when the optimal image segmentation parameters are determined on independent test data.

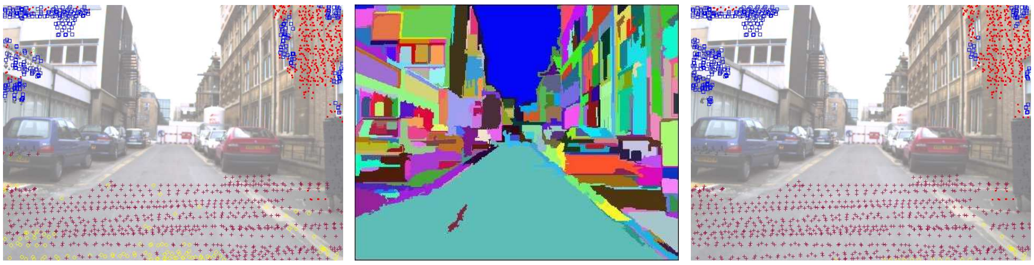

Fig. 13. Example for the majority vote based on patches determined by image segmentation.

Inspection of Fig. 12 reveals both confusion matrices to be broadly diagonally dominant. Good separation is achieved between ground and non-ground classes with a misclassification rate of 1.5%. This is comparable to the figure obtained for the training set in Section 7.1. The strong performance in precision for the *'Grass'*, *'Tarmac'*, *'Nat. Stone Wall'* and *'Bush'* classes is mirrored in the left panel of Fig. 12. In comparison, performance of most wall classes is poor. Considerable confusion exists amongst the wall classes where *'Brick Wall'* and *'Nat. Stone Wall'* are commonly confused as well as *'Concrete Wall'* and *'Rendered Wall'*. Further, *'Tarmac'* is commonly mistaken for both *'Paved'* and *'Dirt Track'*. This again is attributed to a similarity in the colour profiles between these respective classes. In contrast, strong recall performance is obtained for *'Grass'*, *'Paved'*, *'Dirt Path'*, *'Bush/Foliage'* and *'Vehicle'* (cf. left panel of Fig. 12). However, the considerable confusion amongst several of the wall classes is also evident here.

Class Details			**Point-Wise**		**Voted**	
Name	# Patches	# Points	Precision [%]	Recall [%]	Precision [%]	Recall [%]
Gr	99	5393	94.7	92.5	96.6	98.1
Ta	1373	77256	97.5	82.7	97.7	89.0
Di	147	7988	34.5	85.2	46.4	84.8
Te	2240	69541	81.4	71.1	82.7	73.5
Sm	906	29881	53.4	59.3	56.9	64.4
Bu	181	8364	56.8	58.9	60.6	62.8
Ve	169	4499	35.1	76.8	43.7	80.1
Legend for class shortcuts: **Gr**ass, **Ta**rmac/Paved, **Di**rt Path, **Te**xtured Wall, **Sm**ooth Wall, **Bu**sh/Foliage, **Ve**hicle						

Table 7: Classification results for the *Oxford Science Park* data: Meta Classes.

7.5. *Combination of Classes*

Thus, although broadly correct classifications are obtained, the results presented so far indicate that the system can not discriminate adequately between several of our chosen workspace classes. In particular, confusion exists between the class pairs *'Brick Wall'* and *'Nat. Stone Wall'*, *'Concrete Wall'* and *'Rendered Wall'* as well as *'Tarmac'* and *'Paved'*. This is attributed to the similarity in colour profile within these classes, but not across. However, the consistency of the classification results can be improved by combining conceptually related classes for which the current combination of descriptive features does not allow for robust classification. In particular, *'Tarmac'* and *'Paved'* are combined into the class *'Tarmac/Paved'*, *'Brick Wall'* and *'Nat.*

Stone Wall' are combined into the class *'Textured Wall'* and, finally, *'Concrete Wall'* and *'Rendered Wall'* are combined into the class *'Smooth Wall'*. In analogy to our analysis of results with the original workspace classes, Tab. 7 and Fig. 14 show detailed results of this revised system. Typical classification results are shown in Fig. 1.

Comparatively high precision and recall values can be observed for the combined classes. Voted-SVM classification once again improves performance significantly. This is emphasised by much stronger diagonal dominance of the corresponding confusion matrices compared to Fig. 12. In particular the recall-matrix indicates that most ground-truth data over all classes are now retrieved correctly. However, the precision matrix indicates some considerable confusion remains. A significant proportion of the *'Dirt Path'* detections actually originate from the *'Tarmac/Paved'* class. Likewise, a significant proportion of the *'Bush/Foliage'* and *'Vehicle'* detections actually originate from *'Tarmac/Paved'* and/or *'Textured Wall'*. The reason for this can be found in Tab. 7, which indicates that data from each of the two classes *'Tarmac/Paved'* and *'Textured Wall'* outnumbers data from the *'Dirt Path'*, *'Bush/Foliage'* and *'Vehicle'* classes by an order of magnitude. Therefore, a small percentage error in the classification of data from the large classes results in a relatively large drop in the precision of the small classes. In this particular case, 779 out of 77256 ground-truth *'Tarmac/Paved'* data were classified as *'Vehicle'*. Thus, 9.5 % of all vehicle detections (8235 in total) actually originated from the *'Tarmac/Paved'* class (cf. left panel of Fig. 14) whereas that figure only amounts to 1% of all ground-truth *'Tarmac/Paved'* data having been misclassified as *'Vehicle'* (cf. left panel of Fig. 14). It follows, of course, that a small percentage reduction in misclassifications for a large class may have a significant impact on the classification precision of smaller classes. This consideration has currently not been included in the choice of features detailed in Section 7.2, where the two largest classes were amongst the main beneficiaries when more elaborate feature sets were considered.

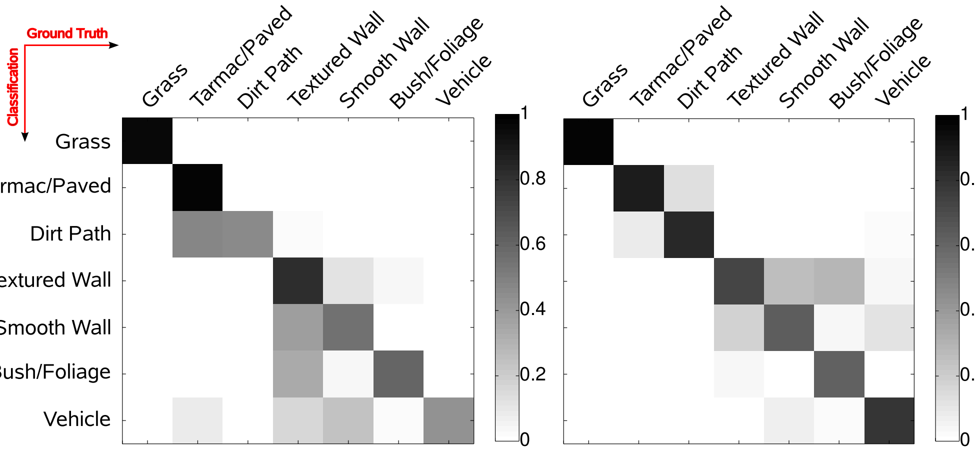

Fig. 14. Confusion matrices for *Oxford Science Park* data obtained using voted-SVM classification and merged classes. **Left:** Rows normalised such that the diagonal represents *precision*. **Right:** Columns normalised such that the diagonal represents *recall*.

7.6. *Overall System Runtime Performance*

The scene classification engine as presented here has been implemented to run online, interfacing to the Mission Oriented Operating Suite (MOOS)[6] installed on our mobile robot *Marge*. Detailed estimates of timing for every stage of the processing pipeline are provided in Tab. 8. The mean total processing time amounts to 4.8 s per frame. The maximum speed of the vehicle is restricted by the need to gather high-quality 3D laser data to ca. 0.5 m/s. An image is recorded every 1.5 m, leading to a real-time processing constraint of 3 s per frame. Although the system currently runs (just) behind time, further computational savings are expected from optimising both the plane segmentation stage as well as the classification stage. In particular, the latter could be achieved by reducing the complexity of the SVMs used via methods such as outlined in [6], where a reduction in complexity by a factor of ten is achieved with no loss in generalisation performance.

Process	Mean [s]	Max [s]
Plane Segmentation	2.00	2.80
Feature Extraction	0.09	0.15
Image Segmentation	0.96	1.13
Classification	1.78	6.70
Overall	4.83	10.78

Table 8: Per-Frame Timing Information. Estimates were obtained on a vanilla 2.0 GHz Pentium laptop as used in the field.

8. Conclusions and Future Work

In this paper we give a detailed account of an appearance-based scene-labelling engine intended for the augmentation of common SLAM maps of outdoor urban environments. The system runs online and close to real-time as per our requirements. Our approach is based on a hierarchy of binary classifiers labelling individual laser data according to their origin. Laser points are characterised by both 3D geometric data and visual cues obtained from monocular vision. Spatial smoothing is performed automatically by considering locally consistent (in appearance) scene patches via image segmentation. We motivate our current choice of features by trading off speed against accuracy amongst several sets of proposed feature combinations. The generalisation performance of the resulting classification scheme is sufficient to consistently separate different types of terrain and walls, including bushes and foliage. The system also has a capacity to recognise common objects such as vehicles.

A natural extention of the current system is the enforcement of scene-wide spatial as well as temporal consistency of the obtained labels. This can be achieved via, for example, a *Markov Random Field* using any of a multitude of available inference methods. Our system is particularly amenable to such an approach since the intuitive labelling by majority vote of local scene patches – rather than the raw laser data – enable the construction of relatively sparse graphs, thereby reducing the computational cost of graph-construction and inference. In addition to incorporating prior knowledge by means of co-appearance of semantic labels, we expect the enforcement of temporal consistency, e.g. by prediction and tracking, to further improve our systems classification performance. However, we have a clear vision of how the semantic workspace descriptions generated by our system will contribute to mobile robot autonomy and human-machine

[6] http://www.robots.ox.ac.uk/∼pnewman/TheMOOS/

interaction. Part of our on-going endeavour are (1) exploration strategies and planning based on semantic knowledge, (2) the enhancement of our appearance-based (natural visual landmarks) navigation system, where landmark grouping according to semantic labels is expected to reduce ambiguities, (3) and the development of human-machine interfaces that, for example, generate semantic path descriptions and allow to address or characterise particular places in the environment by means of semantic attributes.

9. Acknowledgements

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence. The authors gratefully acknowledge the support of the members of the Robotics Research Group who have tirelessly labelled our laser data.

References

[1] D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2004.

[2] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Y. Ng. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data. In *CVPR (2)*, pages 169–176. IEEE Computer Society, 2005.

[3] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *Proc. of the 9th European Conference on Computer Vision (ECCV)*, 2006.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[5] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.

[6] C. J. C. Burges. Simplified Support Vector Decision Rules. In *Int. Conf. on Machine Learning*, pages 71–77, 1996.

[7] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[8] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[9] M. Cummins and P. M. Newman. Probabilistic Appearance Based Navigation and Loop Closing. In *ICRA*, pages 2042–2048, 2007.

[10] C. Dornhege and A. Kleiner. Behavior maps for online planning of obstacle negotiation and climbing on rough terrain. In *In Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2007.

[11] B. Douillard, D. Fox, and F. Ramos. A Spatio-Temporal Probabilistic Model for Multi-Sensor Object Recognition. In *in Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2007.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[13] T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.

[15] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, J. Han, U. Muller, and Y. LeCun. Online Learning for Offroad Robots: Spatial Label Propagation to Learn Long-Range Traversability. In *Proc. of Robotics: Science and Systems*, 2007.

[16] M. Happold, M. Ollis, and N. Johnson. Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning. In *Proc. of Robotics: Science and Systems*, 2006.

[17] A. Harrison and P. Newman. High Quality 3D Laser Ranging Under General Vehicle Motion. In *In Proc. of IEEE Int. Conference on Robotics and Automation (ICRA)*, 2007.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Putting Objects in Perspective. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.

[19] T. Joachims. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, pages 169–184, 1999.

[20] J. Kosecka, L. Zhou, P. Barber, and Z. Duric. Qualitative Image-Based Localization in Indoors Environments. In *in Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[21] B. Kuipers and P. Beeson. Bootstrap Learning for Place Recognition. In *Proc. of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, 2002.

[22] B. Kuipers and Y.-T. Byun. A Robot Exploration and Mapping Strategy based on a Semantic Hierachy of Spatial Representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.

[23] B. Limketkai, L. Liao, and D. Fox. Relational Object Maps for Mobile Robots. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI*, pages 1471–1476. Professional Book Center, 2005.

[24] O. Ḿartínez-Mozos, C. Stachniss, and W. Burgard. Supervised Learning of Places from Range Data using Adaboost. In *Proc. of the Int. Conference on Robotics and Automation (ICRA)*, pages 1742–1747, 2005.

[25] G. Monteiro, C. Premebida, P. Peixoto, and U. Nunes. Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision. In *in Workshop on "Safe Navigation in Open and Dynamic Environments - Autonomous Systems versus Driving Assistance Systems" at the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2006.

[26] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using Visual Appearance and Laser Ranging. In *Proc. of IEEE Int. Conference on Robotics and Automation*, 2006.

[27] A. Nuechter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D Mapping with Semantic Knowledge. In *RoboCup International Symposium*, 2004.

[28] K. Pervoelz, A. Nuechter, H. Surmann, and J. Hertzberg. Automatic Reconstruction of Colored 3D Models. In *In Proc. of Robotik, VDI-Berichte 1841*, pages 215 – 222, 2004.

[29] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In A. Smola, P. Bartlett, B. Schoelkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[30] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors. *Toward Category-Level Object Recognition*. Lecture Notes in Computer Science , Vol. 4170, 2007.

[31] A. R. Pope. Model-Based Object Recognition - A Survey of Recent Research. Technical Report TR-94-04, The University of British Columbia, 1994.

[32] J. M. Porta and B. J. A. Kroese. Appearance-based Concurrent Map Building and Localization. *Robotics and Autonomous Systems*, 54(2):159–164, 2005.

[33] I. Posner, D. Schröter, and P. Newman. Using Scene Similarity for Place Labelling. In *Proc. of the Int. Symposium on Experimental Robotics (ISER)*, 2006.

[34] I. Posner, D. Schröter, and P. Newman. Describing Composite Urban Workspaces. In *Proc. of the Int. Conference on Robotics and Automation (ICRA)*, 2007.

[35] D. Schröter and M. Beetz. Acquiring Modells of Rectangular Objects for Robot Maps. In *Proc. of IEEE Int. Conference on Robotics and Automation (ICRA)*, 2004.

[36] D. Schröter, M. Beetz, and J.-S. Gutmann. RG Mapping: Learning Compact and Structured 2D Line Maps of Indoor Environments. In *Proc. of 11th IEEE ROMAN Conf., Berlin/Germany*, 2002.

[37] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies. Fast and Relaiable Obstacle Detection and Segmentation for Cross-country Navigation. In *Intelligent Vehicle Symposium, France*, 2002.

[38] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the DARPA Grand Challenge. *Journal of Field Robotics*, 2006.

[39] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[40] I. Ulrich and I. Nourbakhsh. Appearance-Based Place Recognition for Topological Localization. In *Proc. of IEEE Int. Conference on Robotics and Automation*, 2000.

[41] J. Weingarten, G. Gruener, and R. Siegwart. A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction. In *Proc. of the 11th Int. Conference on Advanced Robotics (ICAR)*, 2003.

[42] C. Wellington, A. Courville, and A. Stentz. Interacting Markov Random Fields for Simultaneous Terrain Modeling and Obstacle Detection. In *Proc. of Robotics: Science and Systems*, 2005.

[43] C. Wulf, O. andBrenneke and B. Wagner. Colored 2D Maps for Robot Navigation with 3D Sensor Data. In *In Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2004.