Abstract—Humans refer almost to everything by their character- 
arization rather than their detailed descriptions. For example, in indoor environments places are specified as: rooms, corridors, etc. Such categorizations, if learned by a robot, could improve the capabilities in the areas of navigation, localization, or human-robot cooperation. This paper studies the problem of categorizing environments into semantic categories. A new approach based on Support Vector Machine (SVM) is proposed and described for learning to perform classification of environment. The SVM is trained using a supervised training algorithm. This method uses simple features extracted from laser range measures, using methodologies normally used in computer vision. In the present paper the proposed method is used to distinguish between two classes of places from sensor data: rooms and corridors. The real-time experimental architecture designed for classification is presented. Experimental results obtained with real sensor data demonstrate the feasibility and effectiveness of the proposed approach.

I. INTRODUCTION

In mobile robot navigation the problem of learning accurate metric or topological environment maps from sensor data is important for localization, path planning, etc. In many applications (such as indoors, outdoors, industrial, etc), robots can improve their service if they are able to recognize different places and semantically distinguish between them. If a robot has semantic information about places it can be easily commanded to perform its tasks, with instructions like “go to room A”. Only a few works considered the problem of learning semantic information into the map of the world. In this work we address the problem of semantic classification of environment locations using range finder data. This problem has potential applications for example in human-robot communication, better environmental understanding and description, in semantic algorithms, and in conjunction with localization and path-planning. To the best of our knowledge no commercial system is available in this area.

The key idea of this work is to categorize the robot location, in indoor environments, using range finder data. Indoor environments, normally, are decomposed in areas according to their functionality and/or structure. Typical indoor environment can be decomposed in areas such as rooms, corridor, doorways, etc. Each of these areas has a distinct structure. For example rooms are more cluttered than a corridor. These differences in structure motivate the application of learning techniques to develop place classification systems.

1This research was supported in part by Fundação para a Ciência e Tecnologia (FCT), under contract MDT/S04: POSC/EEA-SRI/58279/2004, and by CyberC3 project (European Asia ITC Programme).

The problem of classifying environments has been previously considered. In [1] a topological map was derived from an occupancy grid map. First the grid map was constructed. After that, a topological map was generated by splitting the world into regions, separated through critical points, corresponding to narrow passages such has doorways. Fabrizi and Safiotti in [2] use techniques borrowed from the field of image processing to partition an occupancy grid into ”large open spaces and narrow passages”. In [3], which also uses occupancy grid, an algorithm to automatically segment the space into rooms and corridor regions is proposed. Opposing to [2], the method of [3] has the ability to loose information referencing to older occupancy grid map data, which makes the method better prepared for large environments. In [4] it is proposed a method for extracting the topological maps from occupancy grid maps using the Voronoi Random Fields to classify room environments.

Other works performed classification directly from data gathered from sensors, using learning algorithms. Althus and Christensen [5] applied the Hough transform to sonar readings to extract a pair of parallel lines in order to detect corridors. Doors can also be recognized by detecting the gaps between the lines. Topological maps are constructed from the results of the classifier. Oore et al [6] trained a neural network (NN) with information from odometry and sonar sensors to detect the robot position. In another perspective Posner et al [7] uses an unsupervised algorithm to extract meaningful relations between different scenes.

An interesting work was made by Mozos and Burgard [8]. They approached the problem by classifying the place where the robot was located at each instant using only data coming from a 360 degree planar range system. Classification can be performed by differentiating between room, corridor, door and hallway places. Instead of requiring the extraction of a set of high-level features, the collected raw data is transformed using simple geometrical features. Then, the features are applied to an Adaboost classifier. Adaboost involves the design integration of a collection of weak classifiers, each of which can have a performance only slightly better than random guess. The idea was expanded by employing vision [9] and using Haar features to capture the number of specific objects present in the local environment.

In this paper we address the problem of semantic classification of the environment using range finder data in Wheeled Mobile Robots (WMRs). For this purpose, a support vector machine (SVM) approach is used for classification. The input
to the classifier is a set of simple features extracted from a single sensor scan. Thus the method has the advantage of not requiring the design of a set of weak classifiers. The classification method is integrated into a real-time architecture used to control the WMR.

Compared with NN, the SVM method has some advantages. 1) Less tunable variables: The number of tunable variables inside a SVM depends on the chosen kernel, but there exist kernels with good results with just two tunable variables. Opposed to this, in typical NNs (e.g. backpropagation), the number of tunable variables increases with the number of neurons. 2) Replicable solutions: If the same training data is used, it is guaranteed that the SVM always finds the same optimal solution hyperplane to separate classes. This is not guaranteed when using NN.

The overall structure of the place-classification system is illustrated in Figure 1. The approach is supervised, which has the advantage that the resulting semantic labels correspond to user-defined classes. Examples of typical range scans obtained in an office environment are shown in Figure 4. The used test environment is an example of a common research environment, without any modification for testing purpose.

The rest of paper is organized as follows. Section II presents the WMR real-time control architecture. Section III presents the feature extraction methods. Section IV presents the SVM algorithm used for place classification. Section V presents experimental results. Section VI discusses directions of future work. Finally, Section VII presents concluding remarks.

II. REAL-TIME ARCHITECTURE

For the experiments reported in this paper, a differential WMR, named Robchair, was used (Fig. 2). Previous works have used the same robot [10], [11], but recently Robchair has suffered both hardware and software reformulation. This section describes part of the implemented architecture. The real-time architecture is divided in two parts.

1) Robot motion: The WMR movement is controlled directly by the user using a joystick. The mechatronic equipments distributed through the wheelchair, and used on the robot that directly controls the movement are:

- 2 power drives (each connected to a motor);
- 2 encoders;
- 1 joystick.

These devices, and an embedded computer are all connected through a Controller Area Network (CAN) fieldbus [12], as illustrated in Fig 3. This organization guarantees flexibility, e.g. the possibility of incorporating new devices. An embedded micro-controller module [13] was developed to control robot devices, including the ones listed above, and to interface them to the CAN bus.

At the embedded computer board, an Advantech PCM 3680 card, is used to perform the interface to the CAN bus. A real-time driver was developed to this card, guaranteeing a reliable and synchronized access to the data. For guaranteeing a deterministic data flow from and to the device nodes, a periodic message is used to synchronize all the nodes.
2) Environment Classification: Range data for environment classification is collected using one Hokuyo URG-04LX laser range finder [14] connected to the embedded computer through an Universal Serial Bus (USB). Range measurement scans are periodically acquired and post-processed into a collection of features for classification. The methods used for feature extraction are described in section III. The decision of not integrating the laser communication into the CAN was due to the maximal bit rate of the network. The CAN protocol only permits a maximal $1Mbs^{-1}$, while the URG-04LX laser range sensor has a maximal throughput of $9Mbs^{-1}$, which makes it impossible to communicate all the sensor data through CAN. The amount of data sent from the laser to the network could be reduced by enlarging the sampling interval. However, if more nodes are required to be connected through the network that would lead to network overload problems. The acquisition through USB permits, in the future, the integration of more laser devices through USB, and other sensor and actuator nodes through CAN.

III. FEATURES FROM LASER RANGE DATA FOR PLACE CLASSIFICATION

In this work the input raw sensor data is not directly fed into SVM as inputs. Instead, a set of simple features is first extracted from raw data, and then the features are used as inputs. This section describes the features used in our current system. It will be assumed that each sensor observation $z = \{b_1, \ldots, b_M\}$ is composed of a set of range-bearing measures $b_i = (\alpha_i, d_i)$ where $\alpha_i$ and $d_i$ are the bearing and range measures, respectively.

Each training example for the SVM algorithm is composed by one observation $z$, and its classification $v_i$. The set of training examples is then given by

$$E = \{(z_i, v_i) : v_i \in \mathcal{Y} = \{\text{Room, Corridor, …}\}\} \quad (1)$$

where $\mathcal{Y}$ is the set of classes. In this paper it is assumed that the classes of the training examples are given in advance. The objective is to learn a classification system that is able to generalize from these training examples and that can later classify unseen places in this environment or other environments.

If the training examples of the set (1) were used directly as inputs to the SVM classifier, then all possible situations should be trained in order to attain a good classification rate. Thus, the raw data was transformed into a group of simple geometrical features from which the classification of places could be extracted. They are called simple because they are single-valued features. In order for the classification system to depend only on the $(x, y)$ position of the robot and be invariant to robot rotation, the features should be invariant to rotation. The features should also be computationally not heavy. The choice was to use a set of geometrical features often used in shape analysis [15], [16], [17], [18], [19].

Define $Z$ as the set of all possible observations; i.e., observations obey $z \in Z$. A feature $f_i$ is a function that takes one observation $z$ as argument and transforms it to a real value $f_i(z)$; i.e., $f_i : Z \rightarrow \mathbb{R}$.

Two methodologies were used to extract features from observations. Thus two different sets of simple features were produced for each observation $z$. The first set $B$ was calculated directly from the sensor data in $E$, i.e., from the raw range measures $z$. The following features were used:

- Average difference between the length of consecutive measures.
- Standard deviation of the differences between the length of consecutive measures.
- Average difference between the length of consecutive measures considering a maximal possible value.
- Standard deviation of the differences between the length of consecutive range-limited measures.
- Average measure length.
- Number of gaps in a scan (we consider a gap when absolute difference between consecutive measures is higher than a given threshold).

The second set $P$ of features is calculated from polygonal approximations $P(z)$ of the area covered by the observations $z$. The vertices of each closed polygon $P(z)$ correspond to the Cartesian coordinates of the end points of each range measure $b_i \in z$ relative to the robot, i.e.: 

$$P(z) = \{v_1, \ldots, v_M\} \quad (2)$$

where $v_i = (x_i, y_i)$ and $x_i = d_i \cos(\alpha_i)$ and $y_i = d_i \sin(\alpha_i)$. An example, of polygonal representations of laser range scans is shown in Fig 4. The used features are the following:

- Area of $P(z)$.
- Perimeter of $P(z)$.
- Area of $P(z)$ divided by perimeter.
- Seven invariants calculated from the central moments of $P(z)$.
- Normalized feature of compactness of $P(z)$.
- Normalized feature of eccentricity of $P(z)$.
- Form factor of $P(z)$.
- Circularity defined as $\frac{\text{perimeter}^2}{\text{area}}$.

Let $l$ be the total number of features. Let all feature functions $f_i(z)$ be grouped into a feature mapping vector function $f : \mathbb{R}^M \rightarrow \mathbb{R}^l$, with $f(z) = [f_1(z) \ldots f_l(z)]^T$. Then, the input data set $S$ to the SVM which is obtained by transforming raw data, becomes:

$$S = \{(f(z_1), v_1), (f(z_2), v_2), \ldots, (f(z_l), v_l)\} \quad (3)$$

where $v_i$ are the desired output classes, which are considered to be defined beforehand.

IV. SVM ALGORITHM

A classification problem consists in categorizing input objects (e.g. documents, images, places, etc) into classes. A pertinent approach is based on learning the input/output classification functionality from examples. The SVM algorithm [20] is a supervised learning approach that can be used to learn models employed for regression and classification. The input to the SVM learning algorithm is a set of training examples

$$S = \{(x_1, v_1), (x_2, v_2), \ldots, (x_l, v_l)\} \quad (4)$$

where $x_i (i = 1, \ldots, l)$ are inputs and $v_i$ are the desired outputs of the SVM. In this paper $x_i = [x_1, \ldots, x_N]^T$ are vectors of sensor data gathered from the environment,
and \( v_i \) are the corresponding classes of places. \( N \) is the dimensionality of the input vectors \( x_i (i = 1, \ldots, l) \). Suppose that there are only two classes, and \( v_i \in \{ -1, 1 \} \). Initially assume that the classes are linearly separable. Under this assumption the next step is to determine a linear function that separates the classes. This problem falls into the category of linear learning machines [21] and to solve it, it is required to determine a hyper-plane that separates both classes. The definition of the hyper-plane is given by

\[
\sum_{i=1}^{N} w_i x_i + b = 0,
\]

where \( w = [w_1, \ldots, w_N]^T \) is the weight vector and \( b \) is defined as the bias.

The SVM algorithm tries to learn the separating hyperplane which better separates the data classes, and its learning operation is described as follows. Consider the definition of

\[
\gamma_i = v_i \langle \langle w, x_i \rangle + b \rangle
\]

as the (functional) margin of example \( (x_i, v_i) \) with respect to a hyperplane \( (w, b) \) in the same feature space (\( \gamma_i > 0 \) implies correct classification of the example). The (functional) margin distribution of hyperplane \( (w, b) \) with respect to training set \( S \) is the distribution of margins of the examples in \( S \). In the sequel we refer the minimum value of the margin distribution as the (functional) margin distribution of hyperplane \( (w, b) \) with respect to training set \( S \).

Now consider that a set of example points are available, and we want to determine the hyperplane which probably gets better classification of unseen data. For this purpose it is natural to define and use the hyperplane which has the maximal functional margin (see Fig. 5). This hyperplane is defined as maximal margin hyperplane, and constitutes the solution to the linearly separable classification problem.

Given a linearly separable training sample set \( (4) \), the hyperplane \( (w, b) \) that solves the following optimization problem

\[
\begin{align*}
\text{minimize}_{w,b} & \quad \langle w, w \rangle \\
\text{subject to} & \quad v_i (\langle w, x_i \rangle + b) \geq 1, \\
& \quad i = 1, \ldots, l
\end{align*}
\]

is the maximal margin hyperplane. The notation \( \langle a, b \rangle \) represents the inner product of vectors \( a \) and \( b \). By applying Kuhn-Tucker generalization of the Lagrangian multipliers [21], the maximal margin hyperplane \( (w^*, b^*) \) can be obtained as follows:

\[
w^* = \sum_{i=1}^{l} v_i^* \alpha_i x_i, \\
b^* = \max_{v_i = -1} \left( \langle w^*, x_i \rangle \right) + \min_{v_i = 1} \left( \langle w^*, x_i \rangle \right)
\]

where \( \alpha^* = [\alpha_1^*, \ldots, \alpha_l^*] \) is the Lagrangian vector which solves the following quadratic optimization problem,

\[
\begin{align*}
\text{maximize}_{\alpha} & \quad W (\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} v_i v_j \alpha_i \alpha_j \langle x_i, x_j \rangle, \\
\text{subject to} & \quad \sum_{i=1}^{l} v_i \alpha_i = 0, \\
& \quad \alpha_i \geq 0, i = 1, \ldots, l
\end{align*}
\]

which can be solved using the Sequential Minimal Optimization (SMO) algorithm [22].

There are cases where the training data set cannot be classified by a linear discriminant function. In such cases it could in principle be possible, to find a non-linear discriminant
function to solve the classification problem. In many cases it would be difficult to find such a non-linear separating function. The SVM uses a different approach. It performs a non-linear mapping of the original data set into a higher dimensional feature space. Figure 6 illustrates such a mapping. The idea is to design a non-linear SVM where input vectors \( x \in \mathbb{R}^n \) are transformed by a mapping function \( \Phi \) into vectors \( u = \Phi(x) \) in a higher dimensional feature space \( \mathcal{F} \). The mapping function is characterized by \( \Phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^l \). Then the problem is solved in the higher dimensional feature space using the linear classification approach described above.

In the optimization problem (5) the input vectors appear in terms of the form \( \langle x_i, x_j \rangle \). In the context of this problem, it is not necessary to explicitly know the actual mapping function \( \Phi(x_i) \); \( \Phi \) can be equivalently represented or performed [23] by kernel functions \( K(x_i, x_j) \) of the following form:

\[
K(x_i, x_j) = \langle u_i, u_j \rangle = u_i^T u_j = \Phi^T(x_i) \Phi(x_j),
\]

Thus, kernel functions can be seen as performing the mapping function. We have only to chose admissible kernel functions and investigate its results. There are several kernel functions proposed in the literature (e.g. [23]). Table I defines the kernel functions used in this work.

**TABLE I**

<table>
<thead>
<tr>
<th>Kernel Functions</th>
<th>Type of classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K(x_i, x_j) = \tanh(x_i^T x_j + c) )</td>
<td>Multilayer perceptron</td>
</tr>
<tr>
<td>( K(x_i, x_j) =</td>
<td>x_i^T x_j + 1</td>
</tr>
<tr>
<td>( K(x_i, x_j) = e^{-\frac{1}{2} (x_i^T x_j)^2} )</td>
<td>Gaussian RBF</td>
</tr>
</tbody>
</table>

If, even after performing the non-linear mapping \( \Phi(x_i) \), no linear separation hyper-plane exists in feature space to separate the training samples into classes, i.e. the transformed data are not linearly separable, then the quadratic optimization problem (5) cannot be solved. To design a classifier in such cases, the constraint problem of (5) is reformulated. Slack variables are introduced to allow relaxation on the margins of the constraints. With this approach the problem can be solved at the cost of obtaining some non-zero misclassification rate on the training data. The constraints in (5) then become

subject to \( \sum_{i=1}^{l} v_i \alpha_i \geq 1 - \xi_i \),

\( \xi_i \geq 0, \alpha_i \geq 0, i = 1, \ldots, l. \)

Then, an element on the objective function can be used to penalize the violation of the constraints of (5) [21].

### V. EXPERIMENTAL RESULTS

The methods described in this paper have been implemented and tested with real sensor data. No simulated data was used for training or testing. The experiments were carried out with one Hokuyo URG-04LX laser range finder [14] carried by Robchair. The Hokuyo has a scanning range of 240 degrees. Each scan is composed of 632 range-bearing readings which are radially equally spaced by 0.36 degrees. Thus each observation is \( z = \{b_1, \ldots, b_{632}\} \). Then each raw sensor range measure is transformed into a set of 14 simple features described in Section III. Thus, the feature mapping function is \( f : \mathbb{R}^{632} \rightarrow \mathbb{R}^{14} \). The objective of the experiments is to demonstrate that the simple features presented in Section III, used in conjunction with the SVM classifier presented in Section IV, can be used to classify places. Additionally, it will be analyzed whether the resulting classifier can be used to classify places using testing data different from the training data.

The sensor data sets were collected in the office-like building of ISR-Coimbra (illustrated in Figure 7). The design of the method enables it to operate independently of specific environments characteristics (both in training and classification phases), i.e. the method is prepared to operate in various environments and to be robust to environment changes. In this context, five data sets were collected in several corridors and rooms. The first data set was used for training the classifier, and the other data sets were used to test the classification system with data representing new observation situations not present in the training data. The training data set was composed of 527 sensor observations \( z_i \) and the corresponding place classifications \( v_i \). This is a relatively small training set. One of the four testing data sets corresponds to a corridor. The other three testing data sets were obtained in different rooms, and are named “Room 1”\(^1\), “Room 2”\(^2\), and “Room 3”\(^3\).

In different training sessions, the SVM classifier was tested with each of the kernels listed in Table I. After obtaining the SVM model, the classifier was tested, with different data sets. Tables II, III, and IV present the results obtained with the three different kernels. The best results were obtained with the

\(^1\)equivalent to 0.3 on map
\(^2\)equivalent to 0.16 on map
\(^3\)equivalent to 0.5 on map
Gaussian RBF kernel, where the hit ratio was always above 80% except for Room 3. These are promising results obtained with the relatively small data set used to train the SVM-based classifier. In comparison with [8], the present approach has the advantage of not requiring the design of a set of weak classifiers.

### TABLE II

**Classification results for different indoor areas and using a multilayer perceptron with c = 30 as the kernel (Table I)**

<table>
<thead>
<tr>
<th>Area</th>
<th>Total samples</th>
<th>Correctly classified</th>
<th>Correct rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corridor</td>
<td>48</td>
<td>36</td>
<td>74%</td>
</tr>
<tr>
<td>Room 1</td>
<td>112</td>
<td>84</td>
<td>75%</td>
</tr>
<tr>
<td>Room 2</td>
<td>27</td>
<td>21</td>
<td>77%</td>
</tr>
<tr>
<td>Room 3</td>
<td>57</td>
<td>43</td>
<td>75%</td>
</tr>
</tbody>
</table>

### TABLE III

**Classification results for different indoor areas and using a polynomial of degree d = 5 as the kernel (Table I)**

<table>
<thead>
<tr>
<th>Area</th>
<th>Total samples</th>
<th>Correctly classified</th>
<th>Correct rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corridor</td>
<td>48</td>
<td>38</td>
<td>81%</td>
</tr>
<tr>
<td>Room 1</td>
<td>112</td>
<td>88</td>
<td>79%</td>
</tr>
<tr>
<td>Room 2</td>
<td>27</td>
<td>21</td>
<td>80%</td>
</tr>
<tr>
<td>Room 3</td>
<td>57</td>
<td>43</td>
<td>76%</td>
</tr>
</tbody>
</table>

### TABLE IV

**Classification results for different indoor areas and using a Gaussian RBF as the kernel (Table I)**

<table>
<thead>
<tr>
<th>Area</th>
<th>Total samples</th>
<th>Correctly classified</th>
<th>Correct rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corridor</td>
<td>48</td>
<td>41</td>
<td>80%</td>
</tr>
<tr>
<td>Room 1</td>
<td>112</td>
<td>92</td>
<td>82%</td>
</tr>
<tr>
<td>Room 2</td>
<td>27</td>
<td>22</td>
<td>84%</td>
</tr>
<tr>
<td>Room 3</td>
<td>57</td>
<td>43</td>
<td>76%</td>
</tr>
</tbody>
</table>

**VI. Future Work**

At the moment, the method presented in this paper distinguishes between two classes of places: rooms and corridors. We plan to extend the number of classes, adding for example the classification of doors. To achieve this goal, a decision-tree of SVMs will be used. Another interesting research direction is the application of more than one laser scanner at different levels from the floor. Parallel and non-parallel organization of the scanners will be investigated. It is plausible that with this approach the hit rate can be improved, and a greater variety of object situations in the environment can be distinguished (e.g. with differences arising only at certain levels from the floor). The place recognition hit rate can possibly be improved if sequences of classifications are integrated, especially if this analysis is also confronted with a map of the environment (possibly including the current robot belief about its location in the environment). This issue will be investigated. The inclusion of more simple features into the SVM-based classification system with the goal of improving the hit rate will be investigated. Another future research direction is the design of topological map learning algorithms integrating the semantic information provided by this place classification learning method.

**VII. Conclusion**

This paper has presented a novel approach to classify places in a robot environment into different classes like rooms, corridors, etc. The proposed technique is based on training an SVM using a supervised learning approach. A set of simple geometric features, extracted from a single laser range scans, are fed into the SVM and used for classification. The architecture of the classifier does not require the design of additional weak classifiers, and the training of the SVM does not put design difficulties. Experiments carried out with real sensor data demonstrate the feasibility and effectiveness of our technique.

**References**