



Stochastic Automata for Outdoor Semantic Mapping using Optimised Signal Quantisation

Caponetti, Fabio; Blas, Morten Rufus; Blanke, Mogens

Published in:
Control Engineering Practice

Link to article, DOI:
[10.1016/j.conengprac.2010.11.010](https://doi.org/10.1016/j.conengprac.2010.11.010)

Publication date:
2011

[Link back to DTU Orbit](#)

Citation (APA):
Caponetti, F., Blas, M. R., & Blanke, M. (2011). Stochastic Automata for Outdoor Semantic Mapping using Optimised Signal Quantisation. *Control Engineering Practice*, 19(3), 223-233.
<https://doi.org/10.1016/j.conengprac.2010.11.010>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Stochastic Automata for Outdoor Semantic Mapping using Optimised Signal Quantisation

Fabio Caponetti^{*,a,b,e}, Morten Rufus Blas^{c,a}, Mogens Blanke^{a,d}

^a*Technical University of Denmark, Department of Electrical Engineering, Automation and Control Group, Elektrovej build. 326, DK-2800 Kgs. Lyngby, Denmark*

^b*Università Politecnica delle Marche, Dipartimento di Ingegneria Informatica Gestionale e dell'Automazione, Via Brecce Bianche, 60131 Ancona, Italy*

^c*Claas Agrosystems, Bøgeskovvej 6, 3490 Kvistgaard, Denmark*

^d*CeSOS, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

^e*Integra Software srl, Via Leopardi 1, 60035 Jesi, Italy*

Abstract

Autonomous robots require many types of information to obtain intelligent and safe behaviours. For outdoor operations, semantic mapping is essential and this paper proposes a stochastic automaton to localise the robot within the semantic map. For correct modelling and classification under uncertainty, this paper suggests quantising robotic perceptual features, according to a probabilistic description, and then optimising the quantisation. The proposed method is compared with other state-of-the-art techniques that can assess the confidence of their classification. Data recorded on an autonomous agricultural robot are used for verification and the new method is shown to compare very favourably with existing ones.

Key words: Stochastic automata, Robotics, Classification, Probabilistic models, Quantisation

1. Introduction

Recent developments in outdoor robots and sensor technology have made autonomous field operations a realistic aim. The challenge is to develop and add functionality so that vehicles will behave in a safe and reliable manner under unmanned operation. Safe behaviour is crucial if outdoor robots are to become acceptable to authorities and society. High reliability is also required if robots are to be attractive to farmers and other professional users. The biggest technological challenge in such autonomous outdoor systems is ability to sense the environment, as well as to classify and use perceptual information for control.

Classification and perception in natural environments are difficult. High dimensional data are involved, including those coming from video streams, and with noisy sensor signals that have limitations

in accuracy and range, it is hard to model and generalise objects and contexts. Metric mapping, which is commonly based on position from Global Positioning System (GPS) Satellites combined with motion sensing from Inertial Measurement Units (IMU), provides very useful information, but needs to be supplemented by environment sensing and perception to obtain safe autonomous operation. Semantic mapping is a technique to extract non-metric features from sensors such as cameras and laser scanners. In a changing natural environment autonomous operation is a challenge. GPS availability is irregular or prone to outliers, and feature extraction from advanced sensors frequently suffers from artifacts. Robust methods to recognise objects and scenes are, therefore, a key to assure that a robot shows correct and safe behaviour, even in faulty conditions or unforeseen situations (Bouguerra et al., 2008). Safe autonomous operation therefore requires robust semantic mapping and object recognition, and the technology needed must reach far beyond conventional robotic motion planning, (LaValle, 2006; Mettler et al., 2010). The semantic mapping problem is in essence a classifica-

*Corresponding author. Tel.: +45 4525 3565; fax: +45 4588 1295

Email addresses: fcap@vestas.com (Fabio Caponetti), rufus.blas@claas.com (Morten Rufus Blas), mb@elektro.dtu.dk (Mogens Blanke)

Preprint submitted to Control Engineering Practice

tion problem, and several techniques are available from in-door robotic applications, from computer vision and from the film industry. Detailed references and discussion of main features of classification algorithms are presented in Section 4.2.1.

This paper suggests a novel approach based on stochastic automata (Lunze, 2001) to create and use outdoor semantic maps for safe, autonomous navigation. In this framework, automaton states are shown to conveniently correspond to different characteristic environments, hence giving models an intuitive interpretation. States are interconnected by probabilistic transitions, which represent topological relations. Transitions are associated with activation conditions in the form of perception patterns and vehicle motion history. It is discussed how signal quantisation can enhance robustness against noise and achieve fault-tolerant performances, and how quantisation can be optimised according to the probability distributions of observations. Combining well-known task planning methods (Galindo et al., 2008) with robust semantic mapping, it is shown how safe operation can be achieved. The proposed method for semantic mapping is compared to existing state-of-the-art techniques on data recorded from an autonomous agricultural robot in an experimental orchard.

The paper is organised as follows. The problem and the context are first outlined. Then, perception for semantic mapping is introduced using a stochastic automaton framework, signal space quantisation is discussed and an optimisation is suggested to minimise false alarm probability. Detailed results are then presented using field test data, comparison with other algorithms is discussed and conclusions finalise the paper.

2. Autonomous orchard operations

Semantic mapping is of extreme importance for mobile robots. With a good state estimate of the semantic map, this information can be used to supervise plan execution, redefine controllers, or aid the localisation process. This helps to assure safety and reliability. Semantic mapping is commonly done by matching the metric position of the robot in a known map. This is still viable outdoors but falls short when evaluated in terms of robustness. Missions have to continue, regardless of weather or season, leading to the need for a sensing system robust against vegetation shape, environment changes

and possible map or localisation faults. Fig.1 illustrates the use of an automaton model to represent semantic information of an orchard.

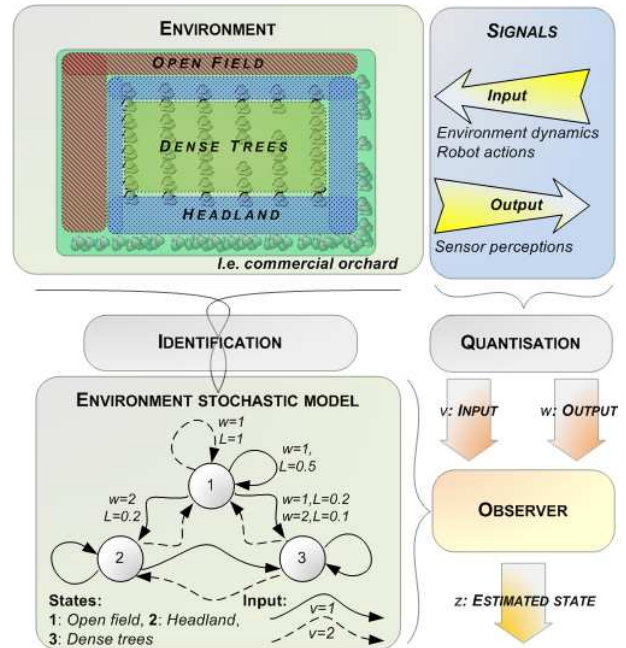


Figure 1: Environment-distinctive areas and their topological relations are modelled by states and transitions of a stochastic automaton. To identify the model probabilities, the real-valued, measured input and perceived output are quantised and are tuned through frequency count probabilities. Robust semantic localisation is achieved in real time by recursive evaluation in an observer.

To stimulate new robust solutions, the data used in this paper were recorded during autonomous operations of a tractor in an experimental orchard owned by Copenhagen University. The area is characterised by geometric features: trees or plants, are set along straight and parallel lines as visible in Fig.2. The distance between each tree in a row depends on the type of tree, while rows are wide enough to let a tractor drive through. A typical work plan involves driving from the docking station to the headland, choosing a row and driving through while performing tasks like cutting grass or spraying. Missions are defined by the user, in terms of trajectories and operations. Each plan is composed of one or more alternatives, useful in unexpected situations, like faults or new obstacles.



Figure 2: A typical tour covers the track shown in the image as GPS route on Google Earth background. The numbers denote the zones in which the area has been divided for semantic mapping. 1. Open field (Road) 2. Headland 3. Dense trees 4. Sparse trees

2.1. Autonomous tractor

The tractor is a standard orchard tractor that has been retrofitted with additional sensors and computing power (Griepentrog et al., 2009). A Sick laser scanner mounted in front of the robot scans a maximum of $8m$ for $180deg$ s with a configurable resolution of 0.5 or $1deg$ at $\sim 70Hz$. The tractor has a second distance sensor, a stereo vision device, which makes available a 3D point cloud of the field of view. It gives more information than the laser although it is quite noisy, and light and weather dependent. The tractor is also equipped with a global positioning device, based on real-time kinematics (RTK) technology and odometry sensors.

2.2. Semantic states

A state is a geographical area, or context, which the tractor must recognise. Referring to Fig.2, a typical orchard can be divided into four interconnected zones, enumerated and described as follows.

1. *Open field* Few obstacles and freely traversable space in the field of view of the tractor, i.e. a road or a low vegetation field, see Fig.4.1.
2. *Headland* Defines the start/end of the agricultural area. It is usually delimited by fences, markers or open space, see Fig.4.2.
3. *Dense trees* The distance between trees is very tight so there re very limited manoeuvring possibilities and the robot may need to push through branches to get by, see Fig.4.3.

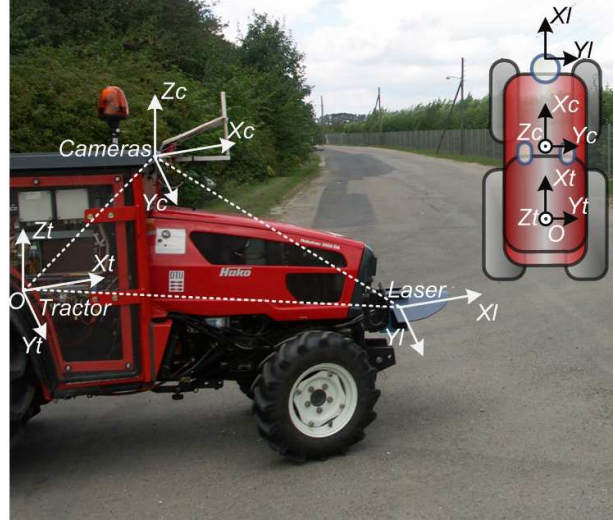


Figure 3: Picture of the operational tractor. Sensor reference systems are overlaid and also shown on a top-down sketch. The position of the tractor is relative to its centre of mass in UTM coordinates.

4. *Sparse trees* Sparse vegetation, robot can easily manoeuvre between the trees while avoiding physical contact, see Fig.4.4.

2.3. Perception

The perception system is designed to provide a compact set of signals that can discriminate between the selected states (Sec.2.2). Due to the limited resolution of the sensors, and to obtain robust features, the system estimates: (a) Amount of visible ground. (b) Space occupied by obstacles. (c) Linear features such as fences or walls. (d) Free space around the tractor. (e) Vegetation permeability. Visible ground plane and space occupied by obstacles are not mutually exclusive because the 3D data makes it possible to observe large amounts of ground while in presence of obstacles. Sparse trees, for example, do not prevent observing the ground plane, even if the crown of the trees form large obstacles that block the line of view. Fig.5 shows how laser and vision streams are fused to get the above features.

2.3.1. Signal y_{gp} : Visible Ground plane

The approach followed for extracting ground plane is similar to Konolige et al. (2009). Given a 3D point cloud a RANSAC technique (Fischler and Bolles, 1981) is used to construct a ground plane hypotheses. This is done by: (a) Choosing three non-collinear points at random from the point cloud; (b)

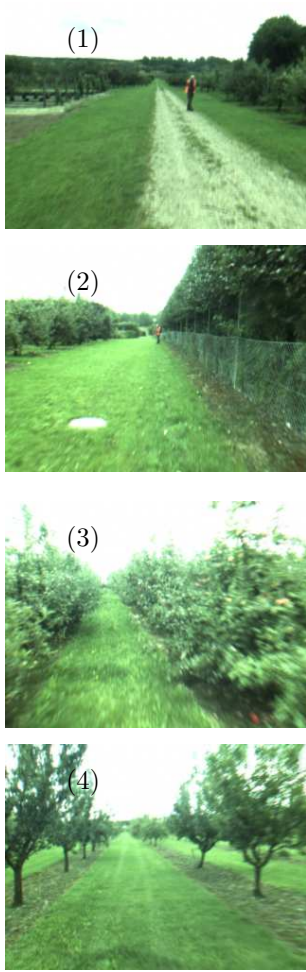


Figure 4: Pictures from the left stereo-camera of the tractor for each semantic state during summer. 1. Open field 2. Headland 3. Dense trees 4. Sparse trees

Estimating a plane from the chosen points; (c) Ranking the plane estimates by the number of inliers. Fig.6.b shows an on-line estimated plane from the 3D point cloud \mathcal{P} in the form of the 3D grid map.

Let the found plane be written in the Hessian normal form defining the unit normal vector $\hat{\mathbf{n}}$ and the distance along the line, p . A 2D cell grid is created on the estimated plane. Each cell c_{xy} , defined by the interval \mathcal{Q}_{xy} , is classified as ground if a cloud point projected on the plane falls within it and its distance to the plane is less than D_{\max} . The amount of ground plane is the sum of these grid cells evaluated from $n_0 - n_1$ along the x-axis to $m_0 - m_1$ along the y-axis. For $\mathbf{x} \in \mathcal{P}$,

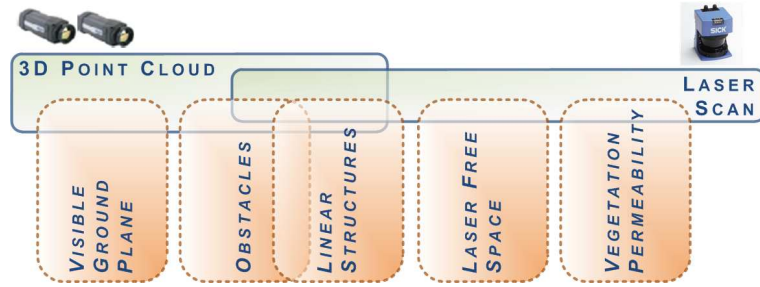


Figure 5: A single laser scan is enough to estimate the amount of free space and to classify the vegetation. By fusing the laser measurements with the synchronized 3D point cloud a 3D grid map of the field of view is built and populated. This is in order to determine the proportion of occupied space and its characteristics. Although the 3D cloud is noisy, it is alone sufficient to reliably estimate the ground plane.

$$c_{x,y} = \left| \left\{ \mathbf{x} \mid \mathbf{x}_x \in \mathcal{Q}_{xy} \wedge \hat{\mathbf{n}} \cdot \mathbf{x} + p < D_{\max} \right\} \right| > 0$$

$$y_{gp} = \sum_{x=n_0}^{n_1} \sum_{y=m_0}^{m_1} c_{x,y}.$$

2.3.2. Signal y_o : Obstacles

The obstacle signal y_o is constructed by projecting each measure from the stereo and laser into a 3D grid map. The value of 1 is given to a 3D grid cell c_{xyz} if a point falls inside the cell and its height above the ground plane is larger than D_{\max} . The number of occupied cells is counted and used as a measure of the obstacles in view. Fig.6 provides a visual illustration of the signal.

$$c_{xyz} = \left| \left\{ \mathbf{x} \mid \mathbf{x} \in \mathcal{Q}_{xyz} \wedge \hat{\mathbf{n}} \cdot \mathbf{x} + p \geq D_{\max} \right\} \right| > 0$$

$$y_o = \sum_{x=n_0}^{n_1} \sum_{y=m_0}^{m_1} \sum_{z=k_0}^{k_1} c_{xyz}.$$

2.3.3. Signal y_{ls} : Linear structures

This signal quantifies the presence of linear structures in the environment such as walls, fences, or hedges. The 3D grid map of obstacles, described in Sec.2.3.2, is collapsed into a 2D grid map on the ground plane by summing the number of occupied cells along the vertical component,

$$g_{xy} = \sum_{z=k_0}^{k_1} c_{x,y,z}.$$

A RANSAC line-fitting algorithm is then used on the 2D grid map to extract the strongest line. A function $l(a,b)$ returns the grid cells that intersect with the line parameterised by a and b , $l(a,b) = \{j \mid j \in g_{x,y}, j_y = aj_x + b\}$. The RANSAC algorithm finally attempts to maximise

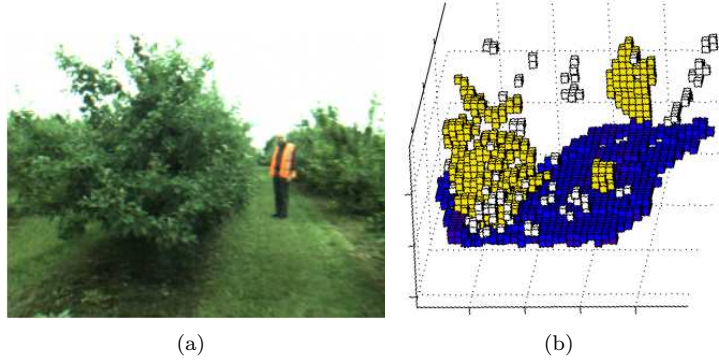


Figure 6: Ground plane and obstacles. (a) Left camera image taken in between dense trees. (b) Corresponding 3D grid map. The estimated ground plane and the obstacles are coloured with a dark and a light colour, respectively. Trees and the human are correctly segmented out and classified in the grid map.

the sum of grid cells that intersect with the line, $y_{ls} = \max_{a,b} \sum_{i=1}^{|w|} l(a,b)$. Fig.7 shows the steps of the process using experimental data.

2.3.4. Signal y_{fs} : Laser free space

The free space observed by the laser is the area spanned by a laser scan. As depicted in Fig.8, each pair of adjacent observations, s_i, s_{i+1} ($i = 1, \dots, n$), defines a triangle from the scanner. Its area can be evaluated by using the Heron's formula,

$$d_i = \|l_i - l_{i+1}\|_2,$$

$$s_i = \frac{l_i + l_{i+1} + d_i}{2},$$

$$A_i = \sqrt{s_i(s_i - l_i)(s_i - l_{i+1})(s_i - d_i)}.$$

By summing the area of all triangles from a laser scan, an estimate of the free space is $y_{fs} = \sum_{i=0}^{n_i-1} A_i$.

2.3.5. Signal y_{vp} : Vegetation permeability

Considering a region of interest, the local spatial laser range distribution can be captured by the principal components of the spatial covariance matrix (Lalonde et al., 2006). By singular value decomposition the covariance matrix is decomposed into principal components ordered by decreasing eigenvalues. By examining the eigenvalues the vegetation can be classified. Eigenvalues of similar magnitude can be used to model/recognise bushes or general foliage. If one eigenvalue dominate significantly, this is the signature of walls or other solid objects. A maximum likelihood strategy for point

wise classification is used to distinguish between the two classes of objects. The details are provided in Caponetti and Blanke (2009).

3. Methods

This section recalls elements of stochastic automata theory and introduces other methods used in the paper.

3.1. Stochastic automaton

An autonomous vehicle in its environment is here modelled as a discrete-event system subject to input that may cause a change of state of the model. Output is state-specific. The states in the model are the locations identified in Sec.2.2. The dynamic behaviour of the model is described by changes in the discrete values of signals, referred to as events in this framework. The system's discrete input, state and output are denoted by v , z and w and their discrete value sets are enumerated as:

$$\text{Input: } v \in \mathcal{N}_v \subset \mathbb{Q}, \mathcal{N}_v = \{1, 2, \dots, M\},$$

$$\text{State: } z \in \mathcal{N}_z \subset \mathbb{Q}, \mathcal{N}_z = \{1, 2, \dots, N\},$$

$$\text{Output: } w \in \mathcal{N}_w \subset \mathbb{Q}, \mathcal{N}_w = \{1, 2, \dots, R\}.$$

Using the notation of Schroder (2003), an initialised stochastic automaton is described by the *five-tuple*: $\mathcal{S} = (\mathcal{N}_z, \mathcal{N}_v, \mathcal{N}_w, L, P(z_k))$. Where $P(z_k)$ represents the set of state probabilities at time k . L is the behavioural function, the law that governs the stochastic process underlying the automaton.

$$L : \mathcal{N}_z \times \mathcal{N}_w \times \mathcal{N}_z \times \mathcal{N}_v \rightarrow [0, 1] \subset \mathbb{R}$$

$$L(z', w, z, v) = P(z_{k+1} = z', w_k = w | z_k = z, v_k = v).$$

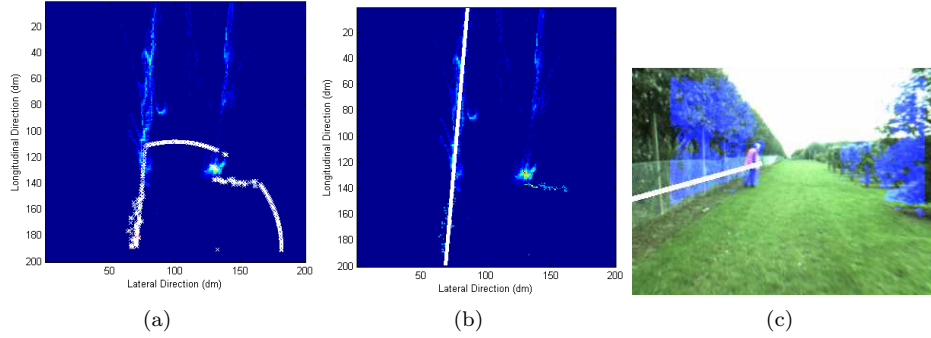


Figure 7: Linear structures. (a) Laser reading overlaid on a top-down view of the 3D grid map. The lighter the colour, the bigger the number of measurements contained in the bin. (b-c) The strongest line estimated corresponds to the fence delimiting the orchard.

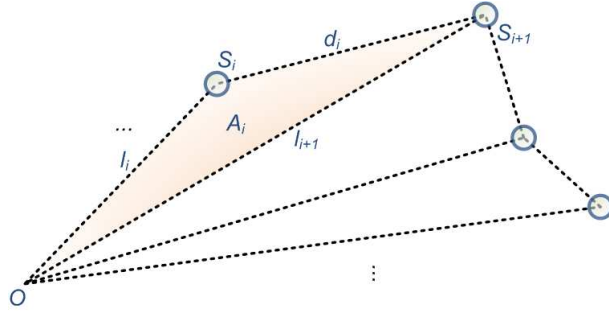


Figure 8: Each pair of consecutive laser readings forms a triangle. By summing the area of all the triangles forming a scan, the *laser free space* is estimated.

3.2. Classification

Semantic mapping using the available perceptual data is equivalent to solving an observation problem for the modelling stochastic automaton. Given an input and output sequence and an initialised automaton \mathcal{S} , the current state is obtained by determining the conditional probability distribution $P(z_k|k) = P(z_k|V(0\dots k), W(0\dots k))$, (Schroder, 2003). The solution of the observation problem is given by the set of all the states z_k to which the automaton may move with non-zero probability, while accepting the input and generating the output sequence specified. The a-posterior state probability distribution can be evaluated on-line by iterative application of a prediction and correction schema discussed in Schroder (2003), and briefly reported

here.

$$P(z_k|k) = \frac{\sum_{z_{k+1}} L(k)P(z_k|k-1)}{\sum_{z_k, z_{k+1}} L(k)P(z_k|k-1)}, \quad k \geq 0, (1)$$

$$P(z_k|k-1) = \frac{\sum_{z_{k-1}} L(k-1)P(z_{k-1}|k-2)}{\sum_{z_k, z_{k-1}} L(k-1)P(z_{k-1}|k-2)} (2)$$

$$P(z_0|-1) ::= P(z_0)$$

The input v_k and the output w_k are arguments in $L(k) = L(z_{k+1}, w_k|z_k, v_k)$. Eq.(1) describes how the prediction obtained from the previous time point, Eq.(2), have to be corrected after the new measurements $v(k)$ and $w(k)$ become available. The recursive formulation of the algorithm is published in Blanke et al. (2006), Chap. 8.

3.3. Quantisation of the signal spaces

Through quantisation, the real-valued signals described in Sec.2.3 are fed to the automaton.

For generality, consider a generic continuous signal function of time t , $x(t)$, $\mathbb{R} \rightarrow \mathbb{R}$. A quantiser splits the signal space \mathbb{R} into a finite number of disjoint sets $\mathcal{Q}_x(\xi)$ where $\xi \in \mathcal{N}_x \subset \mathbb{Q}$. With $\mathcal{Q}_x(\xi)$ denoting the set of values in x associated with the quantised value ξ , the quantiser function reads, in terms of intervals,

$$\mathcal{Q}_x(\xi) = (x_\xi^{low}, x_\xi^{up}], \xi \in \mathcal{N}_x = \{1, \dots, N_x\} \quad (3)$$

where x_ξ^{low} and x_ξ^{up} are the lower and upper bound of the quantisation respectively. The index ξ of the partition $\mathcal{Q}_x(\xi)$ to which the current value of x belongs, represents the qualitative level of the signal. Schroder (2003) designs arbitrarily the quantisers, hence this paper will describe data-driven procedure to define the discretisation levels.

Let $x \subset \mathbb{R}$ be a generic continuous signal sampled in the time with uniform frequency. N samples are drawn independently from the acquired signal to define the subset $\gamma = \{x_i, i = 1 \dots N\}$. Each sample is associated to the corresponding class z to define the training data $(x_i, \theta_i), i = 1, \dots, N$.

Since all x_i are independent and identically-distributed samples of a random variable, the probability density function $\hat{f}(\rho)$ can be estimated by kernel density estimation, Parzen (1962).

$$\hat{f}(\rho) \approx \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{\rho - x_i}{h}\right), \rho \in \mathbb{R}, \min(\gamma) \leq \rho \leq \max(\gamma) \quad (4)$$

where K is a kernel function, N is the number of samples and h is a smoothing parameter. Given a Gaussian kernel in the form of Eq.(5), the value of h can be chosen to maximise the reconstruction performance (Bowman and Azzalini (1997)),

$$K(\varphi) = \frac{1}{2\pi} e^{-\frac{1}{2}\varphi^2}. \quad (5)$$

Given the approximated probability density function the cumulative density function $\hat{F}(x)$ is,

$$\hat{F}(x) = \int_{-inf}^x \hat{f}(\rho) d\rho \approx \sum_{\rho=-inf}^x \hat{f}(\rho). \quad (6)$$

Defining

$$\gamma_z = \{(x_j, \theta_j); \forall j : \theta_j = z\} \quad (7)$$

as the subset of samples related to the state z , it is possible to use the above results to estimate

$\hat{F}(x|\theta = z)$, for each $z \in \mathcal{N}_z$. Fig.9 shows the results of the procedure when applied to a synthetic data-set.

To simplify the notation, define

$$\hat{F}(x)_z = \hat{F}(x_j), x_j \in \gamma_z, z \in \mathcal{N}_z. \quad (8)$$

Reverting to classification, the quantiser defined by Eq.(3) can be interpreted as a linear machine, which splits the continuous time signal x into segments. A simple classifier would select the state by looking at the discrete level in which the signal falls. The probability that the robot is in state z while observing a discrete output equal to ξ is,

$$P(z|\mathcal{Q}_x(\xi)) = \frac{P(\mathcal{Q}_x(\xi)|z)P(z)}{\sum_{\zeta \in \mathcal{N}_z} P(\mathcal{Q}_x(\xi)|\zeta)P(\zeta)}. \quad (9)$$

Rewriting Eq.(9) using Eq.(3),

$$\begin{aligned} & P(\theta = z | x_\xi^{low} < x \leq x_\xi^{up}) \\ &= \frac{P(x_\xi^{low} < x \leq x_\xi^{up} | \theta = z) P(\theta = z)}{\sum_{\zeta \in \mathcal{N}_z} P(x_\xi^{low} < x \leq x_\xi^{up} | \theta = \zeta) P(\theta = \zeta)}. \end{aligned}$$

Further, from Eq.(8), the conditional probability is a function of the conditional cumulative density function,

$$P(\mathcal{Q}_x(\xi)|\theta = z) = P(x_\xi^{low} < x \leq x_\xi^{up} | \theta = z) = \hat{F}(x_\xi^{up})_z - \hat{F}(x_\xi^{low})_z.$$

Hence,

$$\begin{aligned} P(z|\mathcal{Q}_x(\xi)) &= \frac{P(\mathcal{Q}_x(\xi)|z)P(z)}{\sum_{\zeta \in \mathcal{N}_z} P(\mathcal{Q}_x(\xi)|\zeta)P(\zeta)} \\ &= \frac{(\hat{F}(x_\xi^{up})_z - \hat{F}(x_\xi^{low})_z) P(z)}{\sum_{\zeta \in \mathcal{N}_z} (\hat{F}(x_\xi^{up})_\zeta - \hat{F}(x_\xi^{low})_\zeta) P(\zeta)}. \end{aligned} \quad (10)$$

Eq.(10) describes the probability that the system is in the semantic state z while the continuous signal x is contained in the quantised level ξ . This information is used to define the quantisation levels by maximising the probability that the robot is in a state z while $x \in \mathcal{Q}_x(\xi)$ and minimising the number of levels ξ , ($|\mathcal{N}_x|$).

$$\min_{|\mathcal{N}_x|} \max_{\mathcal{Q}_x(\xi)} P(z|\mathcal{Q}_x(\xi)), \xi \in \mathcal{N}_x, z \in \mathcal{N}_z. \quad (11)$$

Non-informative intervals, with low discriminative performance are merged with the confining interval.

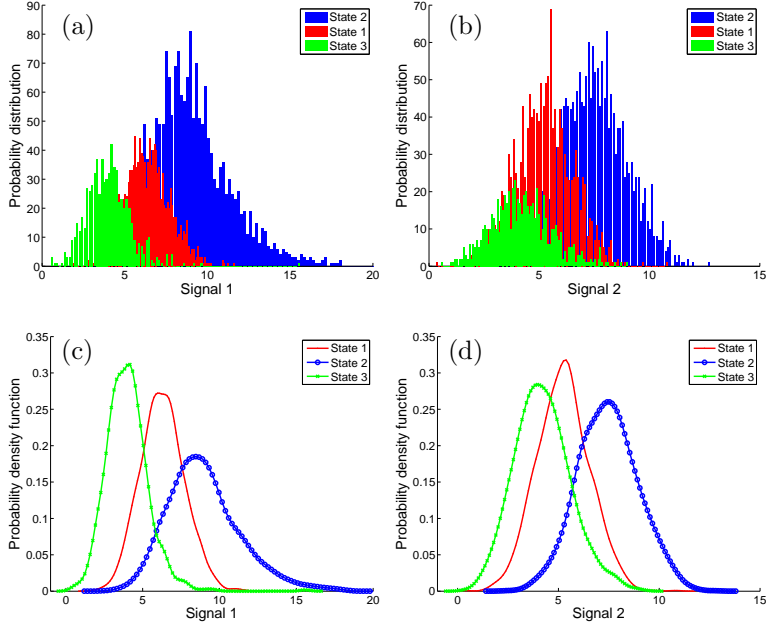


Figure 9: Synthetic data-set composed of two signals generated by three states. (a)(b) Two continuous independent signals are uniformly sampled and grouped in states according to a hand-labelled classification. (c)(d) State-conditioned probability distribution estimated via kernel density estimation

Since the variability space of continuous variables cannot be known exactly, the quantisation levels are defined in the probability space. Probabilities are delimited to the set $[0, 1] \subset \mathbb{R}$ so the whole space is sampled as: $\{p_j\}, p_j \in [0, 1] \subset \mathbb{R}; j = 1, \dots, N_p$. For each p_j a value in the signal space can be found from the state conditional cumulative density function. Summarising, the algorithm to find initial estimates of the quantisation levels is,

$$\forall p_j \text{ find } x_j \text{ such that } \hat{F}(x_j)_z = p_j, z \in \mathcal{N}_z$$

$$\mathcal{X} = \{x_i : \hat{F}(x_i)_z = p_j, \forall z \in \mathcal{N}_z, j = 1 \dots N_p, i = 1 \dots N_x\}$$

Fig.10 shows the procedure an example signal of Fig.9.

Supposing the model composed by N states, \mathcal{X} contains $N_x = N \cdot N_p$ points. By sorting \mathcal{X} , the initial quantisation intervals are defined as:

$$\begin{aligned} \mathcal{Q}_x(1) &= (-\infty, x_1] \\ \mathcal{Q}_x(2) &= (x_1, x_2] \\ &\vdots \\ \mathcal{Q}_x(\xi) &= (x_i, x_{i+1}], \xi \in \mathcal{N}_x \\ &\vdots \\ \mathcal{Q}_x(N_x) &= (x_{N_x}, \infty) \end{aligned} \quad (12)$$

Each quantisation level $\xi \in \mathcal{N}_x$ is associated to the state z for which the condition $\max_z P(\theta_\xi = z | \mathcal{Q}_x(\xi))$ holds. Two quantisers, $\mathcal{Q}_x(\xi)$ and $\mathcal{Q}_x(\xi + 1)$ are therefore merged if:

$$P(\theta_{\xi+1} = \theta_\xi | \mathcal{Q}_x(\xi+1)) \geq P(\theta_{\xi+1} = \zeta | \mathcal{Q}_x(\xi+1)), \zeta \neq \theta_\xi, \forall \zeta \in \mathcal{N}_x, \quad (13)$$

The conditional probability of the resulting merged quantisation interval is re-evaluated before being compared to the succeeding interval $\mathcal{Q}_x(\xi + 2)$. By this procedure the number of levels decreases dynamically while fulfilling the merging condition, as shown by Fig.11 referring to Fig.9 datasets.

3.4. Model identification

The estimation of the behavioural relation for each transition is done by applying the identification algorithm documented by Blanke et al. (2006). From a hand-classified data-set the transition probabilities are approximated by frequency count. For large sample sizes not all state transition for all the possible input/output couples are found, yielding to $L(z', w|z, f, v) = 0$ even if the transition $z \rightarrow z'$ is feasible. To overcome these limitations, the same proportion of training points were selected for each state while a bias in the state transition matrix was added for transitions that might not be

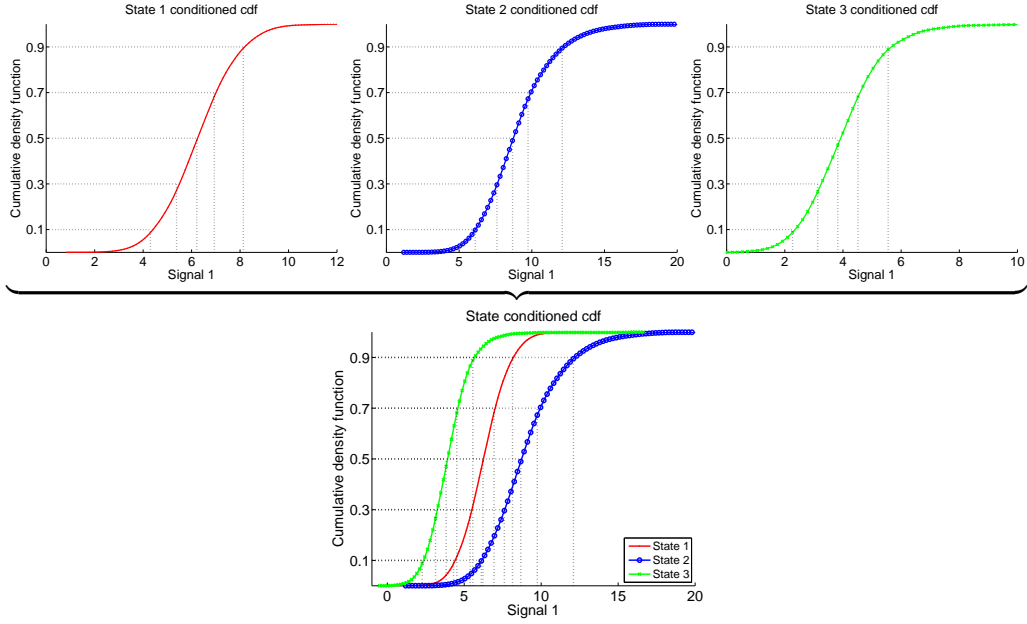


Figure 10: The probability space is uniformly sampled between 0.1 and 0.9 with steps of 0.1. For each level, the corresponding signal value is mapped on the state-conditional cumulative distribution function.

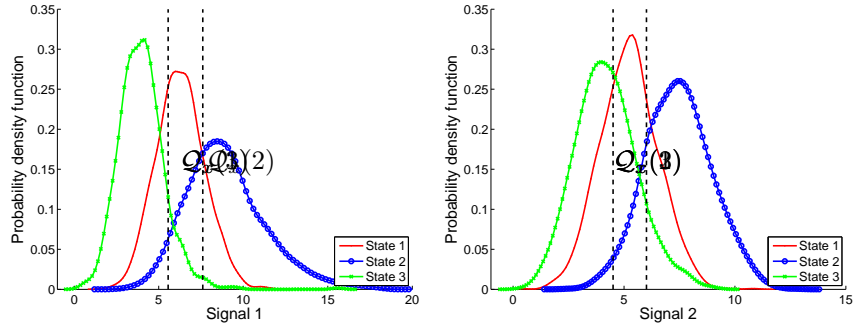


Figure 11: Estimated discretisation levels resulting by the application of the proposed procedure on the synthetic data. The initial quantisation levels estimated, shown in Fig.10, are reduced by application of Eq.(13).

represented in the sample sequence but are physically feasible.

4. Semantic mapping results

This section describes how the above methods are used for semantic mapping and field test results are presented.

4.1. Model design

According to Sec.2.2 the modelling stochastic automaton is composed of $N = 4$ states,

$$\mathcal{N}_z = \{\text{Open field, Headland, Dense trees, Sparse trees}\} = \{1, 2, 3, 4\}$$

The signals used by the robot for semantic mapping are divided in two sets with respect to the automaton: input $\mathbf{u} \in \mathbb{R}^m$ and output $\mathbf{y} \in \mathbb{R}^r$.

4.1.1. Discrete-valued input

For semantic mapping, robot motion is the only input considered $\mathbf{u} = [u_m]$, hence $\mathcal{N}_v = \{\text{moving, stand still}\} = \{1, 2\}$. This is done to disallow state transitions when the tractor is stopped.

4.1.2. Discrete-valued output

Collecting the signals described in Sec.2.3, the observed output consist of a vector $\mathbf{y} = [y_{gp}, y_{fs}, y_{ls}, y_o, y_{vp}]$ of continuous signals. The procedure introduced in section 3.3 is here used to design quantisers for the perceptual data. The γ set of Eq.(7) is populated by randomly picking samples from a training set. The state-conditioned probability distributions are obtained by kernel density estimation and shown in Fig.12. Perceptual aliasing is recognisable in the probability space as an overlap of the distribution curves. This problem is handled by creating quantisation levels for each region of interest and by combining all the signals in the automaton and using the state transition model.

4.2. Classification results

The validation data-set was recorded during a run which covered the track shown in Fig.2. The path first passed through an apple orchard (dense trees), then followed the back fence (headland) to a pear orchard (sparse trees), the path returned along the line, crossed over behind some nut-trees and took the back route home to the garage. The run was made in summer time to catch one extremum of the scenario. Full grown foliage, bushes and tree branches hanging increased the variability of each zone. To stress more the robustness of the methods, people were moving or standing in the tractor field of view during the data recording.

4.2.1. State-of-art algorithms

Classification problems appear frequently in different areas of science and technology and several state-of-art methods are available. *Support Vector Machines* (SVM), *Adaboost*, *Gaussian mixture emitting Hidden Markov Models* (GHMM) (Wolf and Sukhatme, 2008) and *Finite State Machines* (FSM) provide different classification methods for use in semantic mapping. The SVM technique is based on statistical learning theory and is used for classification and regression problems, Vapnik (2006). Adaboost is a technique introduced by Freund and Schapire (1997) that linearly combines

simple, weak classifiers on the basis of classification performance obtained on a training set. Such classifiers have been used previously Mozos et al. (2005) to classify indoor places. Later, several approaches were proposed in literature to improve performance by taking advantage of object recognition (Nuchter et al., 2005) and probabilistic environment models (Mozos et al., 2007). A GHMM consists of a discrete-time and discrete-space Markov process that contains some hidden parameters and emits observable outputs (Rabiner, 1989). For semantic mapping, a GHMM could be built for each possible state by using labelled observations to train Gaussian mixtures that characterise emission. Finite state machines were used to classify film scenes for information retrieval in Zhai et al. (2004) where structural information, together with low and mid-level features, were used to classify the scenes.

4.2.2. Comparison of algorithms

To benchmark the solutions, the Matlab implementation of the automata was compared to open source libraries for SVM, Adaboost and GHMM. The automaton was configured so that all the state transitions were allowed in order to make the comparison fair with the other methods. Four standard SVM kernels were tested: linear, polynomial (of degree 3), radial basis function (RBF), and sigmoid. Kernel parameters were fine-tuned by an iterative procedure based on the training data. The package LIBSVM by Chang and Lin (2001) was used for learning and classification. Adaboost was set to use a maximum number of weak classifiers of 10. The implementation was based on Matlab and configured with tree stumps as weak classifiers. The Bayes Net Toolbox for Matlab was used to produce the results presented regarding the GHMM.

A K -fold cross validation procedure was performed to evaluate the variability of the results. The data collected was split by random sampling in K disjoint sets. $K - 1$ sets were merged and used as training while the remaining data were used as validation. In this case, the data collected was composed of 2281 synchronised laser and vision observations. Due to the amount of data K was chosen equal to 2. In this way the generalisation capabilities together with the robustness were stressed by using half of the available data-set for training and half for testing. The performances were evaluated by collecting the classification results from five independent runs of the 2-fold validation as shown by Fig.13. In addition to the classification rates the

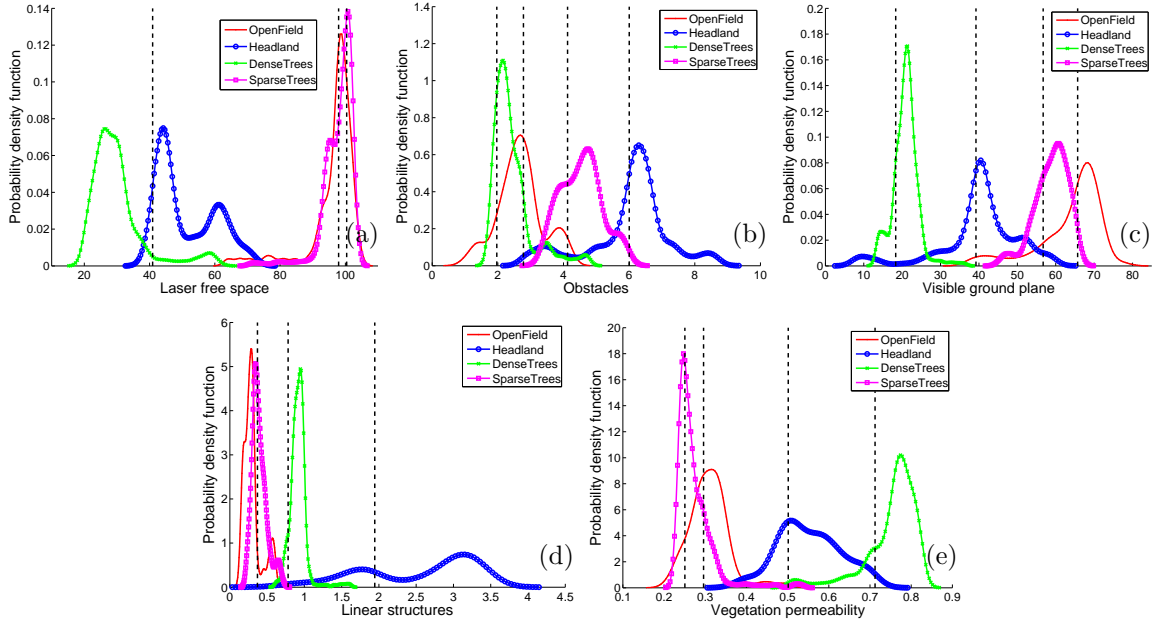


Figure 12: Refined quantisation levels overlaid on state conditional probability density functions. Levels are represented as vertical dotted lines and conditional PDFs as lines with different styles. (a) Laser free space (b) Obstacles (c) Visible ground space (d) Linear structures (e) Vegetation permeability

confusion matrices of each method have been evaluated and shown in Fig. 14. Each row of the matrices represents the instances in a predicted class, while each column represents the instances in an actual class.

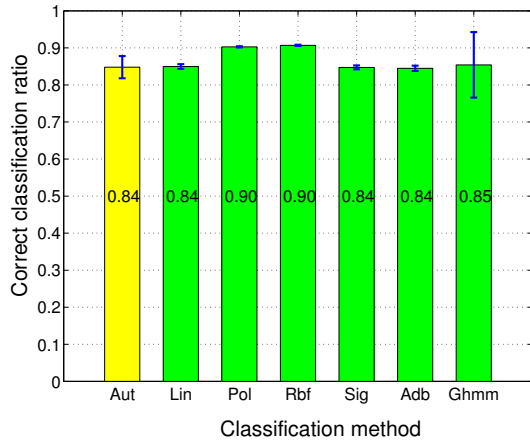


Figure 13: Correct classification rate for all the methods. The performances were evaluated by collecting the results of five runs with a 2-fold cross validation process (both training and validation were done using two disjoint sets of 1141 samples of five features). The standard deviation of the results has been represented as an error line for each bar.

Polynomial and RBF kernel based SVMs show the best classification rates. The GHMM shows the worst overall performance, which is due to singularity problems that in turn are mapped to the classification rate variance. All the methods have problems distinguishing between state 1 and 4, apart from the GHMM. Sparse orchards, are characterised by spaced trees, letting either stereovision and laser perceive only few obstacles. Missing trees in the sparse orchard are then labelled as open areas. For this reason, only the GHMM has the required degree of freedom to discriminate between the two states but suffers from training problems, which in turn worsens the overall performance. The automata shows similar performance to the other state-of-art methods, even though it employs quantised signals. This shows that the information loss in quantising the signals is minimal. The design of the raw signals has been made in order to have signals as independent as possible. This is further demonstrated by the fact that the automata can achieve the same performances as the other methods. The advantage of working with quantised signals is that it simplifies the classification problem by reducing the amount of data that goes in.

In Fig.15 the timings for both training and classification are shown for each method. SVMs clearly

require the longest training time and show the worst scalability. This is due to the fact that the signals are not bounded, and the implementation utilised here has problems handling this. Regarding the classification times, GHMMs are the slowest. The SVMs again seem to scale the worst. One exception is the linear SVM, which is faster than the automata but the confusion matrix shows the worst performance in terms of discrimination. Adaboost performs remarkably well for both training and classification timings. However, the output of Adaboost does not give information about the confidence of the estimate. This makes the method unsuitable for supervision and diagnosis tasks where low confidence estimates should not trigger false alarms. The automata trains faster than SVM and classifies faster than the GHMMs. It shows good scalability properties and outputs a confidence estimate unlike Adaboost. Automata-based classifiers can be implemented efficiently compared to the other methods, which makes them suitable for robotic hardware.

5. Conclusion

This paper presented a novel approach for outdoor semantic mapping by means of a stochastic automaton. A main advantage of the automata compared to other classification methods was a straightforward inclusion of how the dynamical system evolves over time. In the case study of the orchard presented here, the tractor motion and the spatial connection of environments formed an intuitive basis for a model. Spurious observations were effectively dealt with in the updating method for state belief but had a slight penalty in the form of lower adherence to the ground truth during transitions. This behaviour could be fine-tuned according to the needs of a particular use of the algorithm.

A case study of an autonomous vehicle in an orchard showed the properties of the automata-based diagnosis with optimised signal quantisation. A comparison with state-of-the-art classifiers was made on the orchard data. Results showed that the automata-based method trains faster than SVM and classifies faster than the GHMMs. The automata approach shows good scalability properties and outputs a confidence estimate, an essential feature to avoid false alarms from low-confidence hypothesis results. Automata-based classifiers were shown to be efficient compared to the other methods, and they were therefore found attractive for

implementation in robotic environments with hard real-time constraints.

The method was used to optimise quantisation is general and could well be applied to general classification problems.

6. Acknowledgements

The support from the Danish Ministry of Food Agriculture and Fisheries, under contract 3412-06-01729 is gratefully acknowledged. Our colleagues Dr. J.C. Andersen from the Technical University of Denmark, Dr. H-W. Griepentrog and Mr. J. Resting-Jeppesen, both from Copenhagen University, Department of Life Sciences, are gratefully acknowledged for allowing access to equipment and data. Hako Werke is acknowledged for providing the tractor used for the orchard experiments.

References

- Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., 2006. Diagnosis and fault tolerant control, 2nd Edition. Springer.
- Bouguerra, A., Karlsson, L., Saffiotti, A., 2008. Monitoring the execution of robot plans using semantic knowledge. *Robotics and autonomous systems* 56, 942–954.
- Bowman, A. W., Azzalini, A., 1997. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press.
- Caponetti, F., Blanke, M., 2009. Combining stochastic automata and classification techniques for supervision and safe orchard navigation. In: 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.
- Chang, C.-C., Lin, C.-J., 2001. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Fischler, M. A., Bolles, R. C., June 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing* 24 (6).
- Freund, Y., Schapire, R. E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* (55).
- Galindo, C., Fernández-Madrugal, J.-A., González, J., Saffiotti, A., 2008. Robot task planning using semantic maps. *Robotics and autonomous systems* 56 (11), 955–966.
- Griepentrog, H. W., Andersen, N. A., Andersen, J., Blanke, M., Heinemann, O., Madsen, T., Pedersen, S., Ravn, O., Wulfsohn, D., July 2009. Safe and reliable - further development of a field robot. In: Proc. 7th European Conference on Precision Agriculture (ECPA). Academic Publishers, Wageningen.
- Konolige, K., Agrawal, M., Blas, M. R., Bolles, R. C., Gerkey, B., Solà, J., Sundaresan, A., 2009. Mapping, navigation, and learning for off-road traversal. *J. of Field Robotics* 26 (1), 88–113.
- Lalonde, J., Vandapel, N., Huber, D., Hebert, M., October 2006. Natural terrain classification using three dimensional lidar data for ground robot mobility. *Journal of field robotics* 23 (10), 839–861.

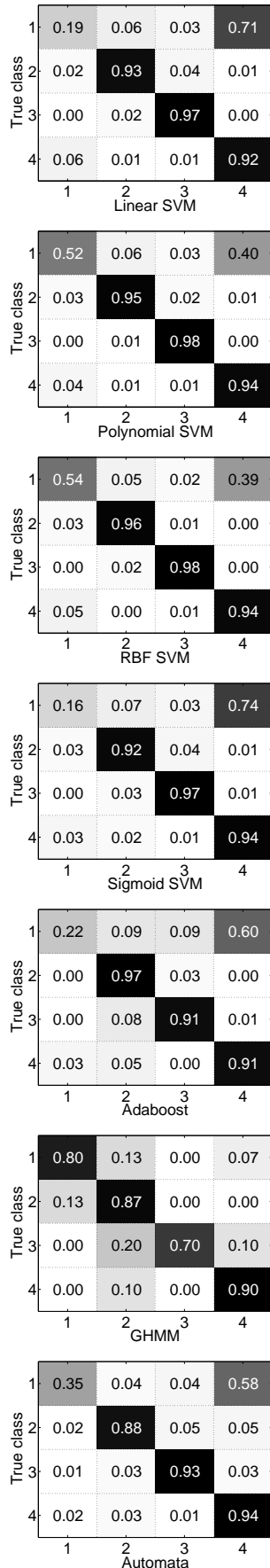


Figure 14: Mean confusion matrices collected evaluating the performances from five trials of a 2-fold cross validation. Rows of the matrices represent the instances of predicted class and columns the instances of an actual class. The darker the square, the higher the classification rate.

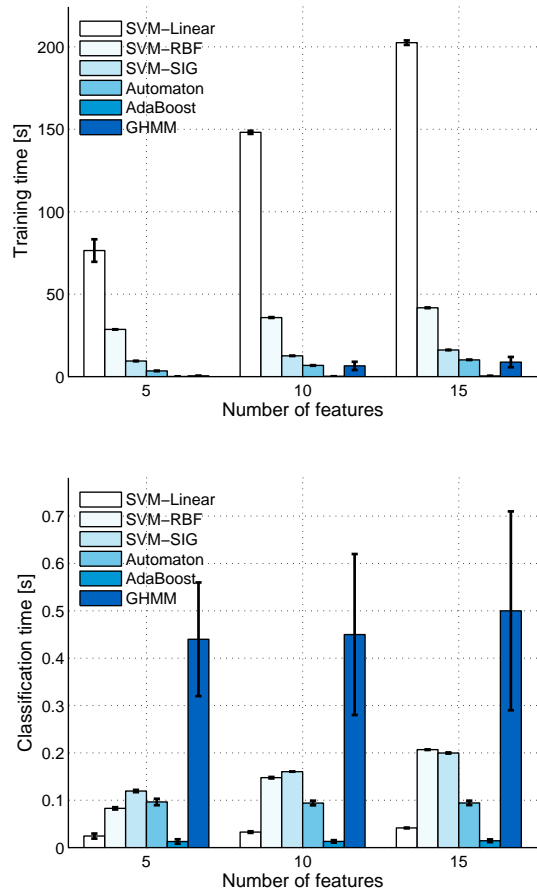


Figure 15: Comparison of training and classification time for evaluated methods. The test dataset is composed of 1141 samples with five features per sample. For 10 and 15 features the existing five features have been replicated two and three times to see how algorithms scale.

LaValle, S. M., 2006. Planning Algorithms. Cambridge University Press, Cambridge, U.K., available at <http://planning.cs.uiuc.edu/>.

Lunze, J., June 2001. Diagnosis of quantised systems. Fault Detection, Supervision and Safety for Technical Processes 2000 1 (1), 29–40.

Mettler, B., Dadkhah, N., Kong, Z., July 2010. Agile autonomous guidance using spatial value functions. Control Engineering Practice 18 (7), 773–788.

Mozos, O., Stachniss, C., Burgard, W., April 2005. Supervised learning of places from range data using adaboost. Proceedings of the 2005 IEEE International Conference on Robotics and Automation.

Mozos, O. M., Jensfelt, P., Zender, H., Kruijff, G. J., Burgard, W., April 2007. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In: Workshop "Semantic information in robotics" at the IEEE International

- Conference on Robotics and Automation.
- Nuchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., Surmann, H., 2005. 3d mapping with semantic knowledge. RoboCup International Symposium, 335–346.
- Parzen, E., 1962. On estimation of a probability density function and mode. *Ann. Math. Stat.* 33, 1065–1076.
- Rabiner, L., February 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* 77 (2), 257–286.
- Schroder, J., 2003. Modeling, state observation and diagnosis of quantised systems. No. 282 in *Lecture notes in control and information science*. Springer.
- Vapnik, V., 2006. *Estimation of Dependences Based on Empirical Data*. Information Science and Statistics. Springer.
- Wolf, D. F., Sukhatme, G. S., April 2008. Semantic mapping using mobile robots. *IEEE Transactions on Robotics* 24 (2), 245–258.
- Zhai, Y., Rasheed, Z., Shah, M., 2004. A framework for semantic classification of scenes using finite state machines. *Lecture notes in computer science* 3115, 279–288.