Markov Random Field-based Clustering of Vibration Data

Philippe Komma and Andreas Zell

Abstract— **A safe traversal of a mobile robot in an unknown environment requires the determination of local ground surface properties. As a first step, a broad structure of the underlying environment can be established by clustering terrain sections which exhibit similar features. In this work, we focus on an unsupervised learning approach to segment different terrain types according to the clustering of acquired vibration signals. Therefore, we present a Markov random field-based clustering approach taking the inherent temporal dependencies between consecutive measurements into account. The applied generative model assumes that the class labels of neighboring vibration segments are generated by prior distributions with similar parameters. A temporally constrained expectation maximization algorithm enables the efficient estimation of its parameters considering a predefined set of neighboring vibration segments. Since the size of the neighbor set proves to be data-dependent, we derive a general means of estimating this set size from the observed data. We show that the Markov random field clustering approach generates valid models for a variety of driving speeds even in situations of frequent terrain changes.**

I. Introduction

Employed for a growing number of tasks like rescue missions or agricultural assignments, mobile robots are exposed to a variety of terrain types. To ensure a safe terrain traversal the robot must be aware of ground surface hazards induced by the presence of slippery and bumpy surfaces. These hazards are known as non-geometric hazards [19].

In this paper, we consider an unsupervised learning approach for the problem of terrain clustering, in which we assume the number of clusters k to be known. Inertial sensors, that is, accelerometer sensors, provide the data on which the terrain segmentation is based on. Although the exact terrain types (such as grass, asphalt, etc.) are not inferred an unsupervised clustering of varying ground surfaces still yields beneficial information: when the robot navigates over unknown terrain, meta data such as the degree of bumpiness or slippage of the ground surface can be stored along with the accelerometer data. After data clustering this meta data is an integral part of each cluster. Hence, whenever the robot traverses a certain terrain type, and this terrain type reveals potentially hazardous characteristics according to the clustering, adjustments to its driving style should be made. Note that in some situations some of the meta information may change whereas other meta information remains the same (e.g., wet grass can become dry while a bumpy surface most likely remains bumpy). Here, a simple binary classification strategy which separates the data into a hazardous class and a non-hazardous one is inappropriate since class assignments have to be retrained whenever the degree of hazardousness of some of the data changes. In a multi-class setting, however, we can simply modify the meta information assigned to a certain cluster without the need of retraining the classifier.

Unsupervised terrain clustering is a non-trivial problem. Problems arise due to the potentially large overlap between terrain classes in feature space. In this work, we use temporal dependencies between consecutive measurements to obtain a better clustering, taking ambiguous cases into account. Here, an ambiguous case occurs if two measurements belong to the same class, but are clustered into two different clusters. The applied clustering technique is based on the Markov random field (MRF) model of [4] for incorporating temporal coherences between consecutive measurements. Exploiting temporal dependencies for vibration data clustering is justified by the fact that terrain changes only appear infrequently resulting in a high probability that the terrain type does not change from one measurement to the next. Further justification is provided by previous work which successfully adopted the technique of Bayesian filtering to the problem of supervised classification [9], [10]. In this work, we show how temporal coherences can be integrated in unsupervised clustering tasks as well.

The remainder of this paper is organized as follows: In Sect. II we present an overview of related work. There, we focus on research about the integration of both constraints and temporal dependencies into a clustering framework. Sect. III briefly summarizes the adopted techniques for data clustering: a Gaussian mixture model approach and a Markov random field approach which embeds temporal coherences into the clustering process. In Sect. IV, we provide details of our clustering experiments whose results are presented and discussed in Sect. V. Finally, Sect. VI gives conclusions.

II. Related Work

Terrain identification using vibration data was considered by many authors: Originally proposed in [8], it was successfully applied to planetary robots [2], to autonomous ground vehicles [18], and passenger vehicles [17]. Yet, all these techniques only handle supervised classification and thus cannot be adopted to our unsupervised clustering problem. The sole exception is the work of [7]. There, they established a cost function to find an optimal classifier with respect to its parameter set. The applied cost function also incorporates temporal dependencies of vibration signatures as it constrains the posterior distributions of consecutively performed terrain estimates to be similar. The main drawback of this approach is the time spent on minimizing the proposed cost function.

P. Komma and A. Zell are with the Chair of Computer Architecture, Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany {philippe.komma, andreas.zell}@uni-tuebingen.de

In contrast, the clustering technique presented in this work is based on an efficient expectation maximization approach which yields a valid clustering within several seconds.
Other approaches regard temporal coherences as (pairwise) constraints which are to be integrated into the Gaussian mixture model-based clustering process [16]: If two measurements are acquired at approximately the same time a hard constraint can be established to enforce a certain Gaussian mixture component to contain both measurements. The problem of this approach is its error-proneness in the presence of false hard constraints. Even a small number of erroneous hard constraints can result in inferior results [14]. This is a significant observation in our case, since these hard constraints have to be estimated and cannot be guaranteed to be valid. Soft constraints as proposed in [12] can handle these situations more adequately, yet, it is still not clear of how to derive a stable measure to estimate the degree of cohesion between two consecutive measurements.
In [6] and [1], Markov random fields (MRF) models were introduced into the domain of image analysis. In the following, they were applied to a number of image analysis tasks such as image segmentation [20] or image interpretation [13]. MRF models offer a convenient means for incorporating context, or dependence among neighboring pixels. Context is important because contiguous pixels are likely to belong to the same region. In this work, we exploit spatial dependencies among neighboring terrain patches instead of pixel arrays: The applied MRF model [4] assumes that the class labels of observations are generated by prior distributions which share similar parameters for neighboring observations. As we will show in Sect. V, the integration of local environment properties can significantly improve the clustering results if the number of considered neighbors is chosen adequately. As a further contribution we derive a general technique which estimates this neighbor set size from the given data.

III. Applied Techniques

A. Gaussian Mixture Model-based Clustering

The Gaussian mixture model (GMM) is a semi-parametric technique for modeling an unconditional probability density function $p(x)$, given a set of unlabeled, d-dimensional data points $u_{0:n} = \{u_0, u_1, \ldots, u_n\}$. The probabilistic model is expressed as a linear combination of k basis functions $p(u_i) = \sum_{j=1}^{k} \pi_j p(u_i|j)$, where k denotes the number of components of the model, π_j is the mixing coefficient of component j, and $p(u_i|j)$ is the component likelihood. The latter defines the probability of a data point u_i to belong to a certain mixture component j. For Gaussian mixture models the basis functions are given by Gaussian distribution functions with parameters $\{\mu_j, \Sigma_j\}$:

$$p(u_i|j) = \frac{1}{(2\pi)^{d/2}\,|\Sigma_j|^{1/2}} \cdot e^{-\frac{1}{2}(u_i-\mu_j)^T \Sigma_j^{-1}(u_i-\mu_j)}. \quad (1)$$

Here, μ_j denotes a d-dimensional mean vector and Σ_j is a positive definite $d \times d$ matrix. The corresponding generative model is shown in Fig. 1(a). Here, two neighboring vibration segments i and j are shown. The model assumes a common prior distribution π which independently generates all vibration segment labels u_i.

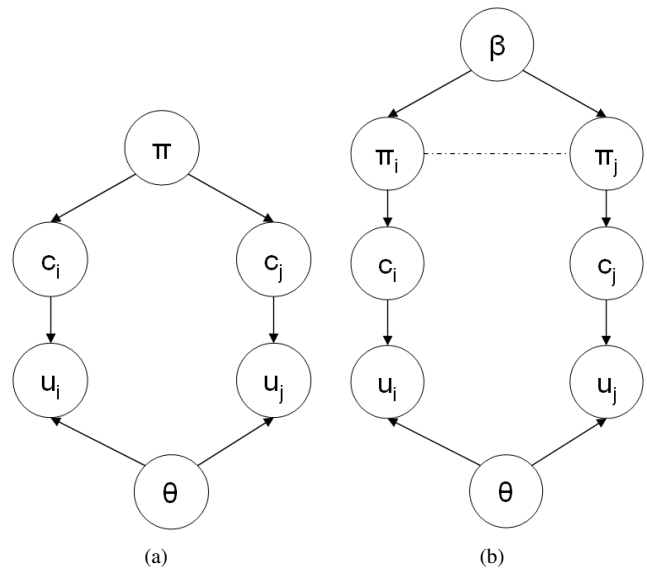

Fig. 1. The employed probabilistic graphical models: (a) Standard Gaussian mixture model. (b) MRF on vibration segment priors.

The Gaussian mixture parameters $\vec{\theta}_j = \{\mu_j, \Sigma_j, \pi_j\}$, $j \in [1;k]$ can be efficiently trained using the expectation maximization (EM) algorithm. The EM algorithm is an iterative technique guaranteeing a monotone decrease of the negative log-likelihood of the data set during optimization. The log-likelihood is defined as:

$$\mathcal{L}_1^{(t)} = \sum_{i=0}^{n} \log p(u_i|\vec{\theta}^{(t)}), \quad (2)$$

which is the sum of the probability of each data point given the current model parameters $\vec{\theta}^{(t)}$. Provided with an initial estimate of the mixture model parameters $\vec{\theta}^{(0)}$, the EM algorithm iteratively reestimates these parameters until convergence of the data log-likelihood:

1) E-step:

$$p(j|u_i) = \frac{p(u_i|j)\pi_j}{\sum_{c=1}^{k} p(u_i|c)\pi_c} \quad (3)$$

2) M-step:

$$\widehat{\mu}_j = \frac{1}{(n+1)\pi_j} \sum_{i=0}^{n} p(j|u_i)u_i \quad (4)$$

$$\widehat{\Sigma}_j = \frac{1}{(n+1)\pi_j} \sum_{i=0}^{n} p(j|u_i)u_i u_i^T - \widehat{\mu}_j \widehat{\mu}_j^T \quad (5)$$

$$\widehat{\pi}_j = \frac{1}{n+1} \sum_{i=0}^{n} p(j|u_i) \quad (6)$$

While the E-step determines the probability of the jth mixture component given the data and model parameters $\vec{\theta}^{(t)}$, the M-step performs a reestimation of the model parameters. The algorithm terminates if $1 - \frac{\mathcal{L}_1^{(t-1)}}{\mathcal{L}_1^{(t)}} < \epsilon$. In our experiments, we used $\epsilon = 0.001$.

A trained GMM can then be employed for classification and clustering tasks by assuming a one-to-one correspondence between mixture components and classes and mixture components and clusters, respectively. That is, we assign a certain class c_i or cluster i to the mixture component i. Finally, the class (or cluster) belonging to a given data point u_l is chosen by selecting the mixture component j that maximizes the posterior probability $p(j|u_l)$. Since in the unsupervised learning case each cluster represents a certain terrain class to discriminate, both "cluster" and "class" are used as equivalent terms in the remainder of this paper.

B. MRF-based Clustering

1) The Markov Random Field Model: The model presented in the previous section does not exploit temporal a priori-dependencies between vibration segments since the prior π_k is independent from the vibration segment index i. In our context, this assumption does not (necessarily) hold since, during robot traversal, it is very likely that the terrain does not change from one measurement to the next. To incorporate temporal dependencies, we applied the generative model of [4] (Fig. 1(b)). There, each label is generated by an individual prior distribution π_i. Further, it is assumed that these priors are similar among neighboring vibration segments.

We enforce the latter by including a penalty term $p(\pi|\beta)$ to the log-likelihood function of (2). This term penalizes neighboring pixels with different priors:

$$\mathcal{L}_2 = \mathcal{L}_1 - p(\pi|\beta).$$

To model the joint density over vibration segment priors, the following Besag approximation is used:

$$p(\pi|\beta) \approx \prod_i p(\pi_i|\pi_{N_i}, \beta),$$

where N_i is the set of neighboring vibration segments of vibration segment i, β is a non-negative scalar, and π_{N_i} denotes the mixture distribution over the priors of neighboring vibration segments of vibration segment i:

$$\pi_{N_i} = \sum_{l \in N_i, l \neq i} \lambda_{il} \pi_l. \tag{7}$$

Here, the mixture weights $\lambda_{il}, l \in N_i, l \neq i$ are constrained to be non-negative and sum to one over all l. They determine the influence of each prior to the mixture relative to the offset between vibration segments i and l.

The conditional density $p(\pi_i|\pi_{N_i}, \beta)$ is then approximated by the following log-model (ignoring constants):

$$\log p(\pi_i|\pi_{N_i}, \beta) = -\beta \left[D(\pi_i||\pi_{N_i}) + H(\pi_i) \right]. \tag{8}$$

$D(\pi_i||\pi_{N_i})$ denotes the KL divergence between π_i and π_{N_i} and is defined as $D(\pi_i||\pi_{N_i}) = \sum_{j=1}^{k} \pi_{ij} \log \pi_{ij} - \sum_{j=1}^{k} \pi_{ij} \log \pi_{N_{ij}}$. The KL divergence is a measure of similarity between the prior of a vibration segment i and the one of its neighbors. The KL divergence is always positive and becomes zero if $\pi_i = \pi_{N_i}$. Hence, by minimizing the KL divergence, we constrain the neighbors to have similar class labels. $H(\pi_i)$ is the entropy of the distribution π_i. It is a non-negative measure which is the larger the more similar is π_i to a uniform distribution. The minimization of the entropy $H(\pi_i)$ is necessary since although regions of the same terrain should exhibit similar priors, we do not expect these priors to be distributed uniformly.

Instead of optimizing (8) directly, the following approximation of the conditional density is employed:

$$\log p(\pi_i|\pi_{N_i}, \beta, s_i) \approx -\beta \left[D(s_i||\pi) + D(s_i||\pi_{N_i}) + H(s_i) \right], \tag{9}$$

where $\{s_i\}$ is an auxiliary set of distributions. Replacing (8) by (9) turns the constrained optimization problem into an efficient one which can be solved using the EM algorithm described below. We note that s_i (as well as the other auxiliary distribution q_i introduced in the next paragraph) is not user-specified, but arises directly from the optimization process.

In addition to the penalty term of (9), a data-dependent penalty term $\mathcal{P}_s$ is introduced, which incorporates useful domain knowledge. Therefore, we consider the posterior distributions and constrain them to be similar among neighboring vibration segments and to be as informative as possible:

$$\mathcal{P}_d = -0.5 \left[D(q_i||p_i) + D(q_i||p_{N_i}) + H(q_i) \right].$$

Here, q_i is an arbitrary class distribution for a vibration segment i and p_i is the posterior of a vibration segment given the model parameters θ and priors π_i.

The complete penalized log-likelihood of the observed data as a function of the model parameters and the introduced auxiliary distributions $\{s_i\}$ and $\{q_i\}$ then becomes (ignoring constants):

$$\begin{aligned} \mathcal{L}_2(\theta, \pi, s, q) = \sum_i \Big[& \log \sum_j p(u_i|j, \theta) \pi_{ij} \\ & -\beta \left[D(s_i||\pi_i) + D(s_i||\pi_{N_i}) + H(s_i) \right] \\ & - 0.5 \left[D(q_i||p_i) + D(q_i||p_{N_i}) + H(q_i) \right] \Big]. \end{aligned} \tag{10}$$

2) Estimating the Model Parameters: The parameter set $\{\theta, s, q\}$ is estimated using an EM algorithm which maximizes the energy of $\mathcal{L}_2$ by coordinate ascent: in the E-step, we fix θ and π and maximize $\mathcal{L}_2$ over s and q. In the M-step, we fix s and q and we maximize $\mathcal{L}_2$ over θ and π. Pseudo code for the EM algorithm is provided below (see [4] for a complete derivation of the respective formulas). We note the similarity between the EM formulation for training the MRF-based generative model and the one for Gaussian mixtures (cf. Sect. III-A). The main difference is that in the temporally coherent approach the label posteriors are "smoothed" over vibration segments between each E- and M-step by a one dimensional filter.

3) Selection of the Free Parameters: The determination of the mixture distributions over the priors (π_{N_i}), the posteriors (p_{N_i}), and the auxiliary set q_i (q_{N_i}) require the definition of both the mixing weights λ_{ij} and the neighborhood size. Note that the evaluation of (7) is equivalent to a convolution operation $\pi_{\cdot j} \odot \lambda$, for each mixture component j. In this

Algorithm 1 The temporally constrained EM algorithm

1: Initialize the parameter vector θ (including the priors $\{\pi_i\}$) using the k-means algorithm (see Sect. IV-C).

2: **E-step**: Determine posterior probabilities p_i using the current estimates of θ and $\{\pi_i\}$:

$$p_{ij} \equiv p(j|u_i) = \frac{p(u_i|j)\pi_{ij}}{\sum_{c=1}^{k} p(u_i|c)\pi_{ic}}$$

3: Determine $\{s_i\}$:

$$s_i \propto \pi_i \pi_{N_i},\ \pi_{N_i} = \sum_{l \in N_i, l \neq i} \lambda_{il} \pi_l.$$

4: Normalize each s_i such that $\sum_j s_{ij} = 1$.

5: Determine $\{q_i\}$:

$$q_i \propto p_i p_{N_i},\ p_{N_i} = \sum_{l \in N_i, l \neq i} \lambda_{il} p_l.$$

6: Normalize each q_i such that $\sum_j q_{ij} = 1$.

7: **M-step**: Update the parameter vector θ:

$$\begin{aligned} q_{N_i} &= \sum_{l \in N_i, l \neq i} \lambda_{il} q_l \\ \mu_j &= \frac{\sum_i (q_{ij} + q_{N_i,j}) y_i}{\sum_i (q_{ij} + q_{N_i,j})} \\ \Sigma_j &= \frac{\sum_i (q_{ij} + q_{N_i,j}) y_i {y_i}^T}{\sum_i (q_{ij} + q_{N_i,j})} - \mu_j {\mu_j}^T \end{aligned}$$

8: Update $\{\pi_i\}$:

$$\pi_i = \frac{1}{(1+2\beta)} \left[\frac{1}{2}(q_i + q_{N_i}) + \beta(s_i + s_{N_i}) \right].$$

9: Evaluate $\mathcal{L}_2$ using (10)

10: **if** convergence of $\mathcal{L}_2$ **then**

11: terminate.

12: **else**

13: goto step 2.

14: **end if**

context, λ is a linear one-dimensional filter with certain properties: First, the center coefficient has to be zero and second, all coefficients have to sum to one. In our experiments, we applied modified versions (i.e., the center coefficient was set to zero) of a box, tent, quadratic, cubic, and a Gaussian filter. Although we expected relevant differences in the results when adopting the varying filters these differences did not prove to be statistically significant. Hence, we only present the results of the Gaussian filter in the result section which tend to yield slightly better results in comparison with the other filters. Further, we chose the standard deviation of the Gaussian filter to be equal to half of the neighbor set size.

The neighbor set size should be selected data-dependent: Choosing a large neighborhood is only appropriate when the robot navigates over homogeneous terrain for a longer period of time. In this case, the influence of erroneously chosen priors is reduced due to the inclusion of neighboring priors.

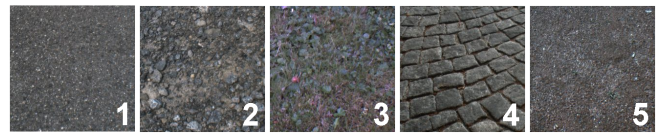

Fig. 2. The terrain types we used in our experiments: 1: asphalt, 2: gravel, 3: grass, 4: paving, 5: clay.

In situations of high-frequency terrain changes, however, a large neighbor set is inadequate since the neighboring vibration segments most likely belong to a another terrain class and thus should provide priors with different distributions.

Although the exact frequency of terrain changes is not known, we propose the following technique to estimate the neighbor set size from the acquired sensor data: first, we perform a k-means clustering of the vibration signals. This clustering provides a broad assignment of each vibration segment u_i to one of the terrain classes. We then determine the homogeneous neighbor size $\mathcal{S}$ for each vibration segment u_i which is defined as:

$$\begin{aligned} \mathcal{S}(u_i) &= \arg\max_{l}\ \mathcal{T}(u_i, \{u_j\}),\ j \in [i-l; i+l] \backslash \{i\}, \\ \mathcal{T}(u_i, v) &= \begin{cases} 0, & \exists v_j \in v,\ v_j = \text{outlier w.r.t. } \mathcal{N}_{\mu_c, \Sigma_c} \\ |v|, & \text{else} \end{cases} \end{aligned}$$

In other words, the homogeneous neighbor set size $\mathcal{S}$ of a vibration segment u_i is the maximum number of contiguous vibration segments which are located symmetrically around observation u_i and belong to the same terrain class c as vibration segment u_i with a certain probability. Since the terrain class c of u_i is represented by a multivariate Gaussian distribution with mean μ_c and covariance matrix Σ_c we can perform the latter verification using the Mahalanobis distance-based outlier test of [15], pp. 224: This test is based on the fact that the Mahalanobis distances between the instances of class c and the cluster center μ_c are approximately χ^2_d-distributed with d degrees of freedom [11]. Here, d denotes the dimensionality of a vibration segment u_i. The test defines a given vibration segment v_j as outlier with respect to class c if the squared Mahalanobis distance between v_j and the cluster center μ_c is larger than $\chi^2_{d,0.95}$, where $\chi^2_{d,0.95}$ is the 95th percentile of the χ^2_d distribution.

Finally, we define the homogeneous neighbor set size of the whole data set $\overline{\mathcal{S}}$ to be the mean neighbor set size averaged over all vibration segments u_i, $i \in [0; n]$, mapped to the next multiple of 2. For the last free parameter β, we follow the suggestion of [4] and set β to 0.5. Experimental results proved this choice to be valid.

IV. Experimental Setup

A. Data Acquisition and Feature Extraction

The objective of our approach is to establish a clustering of different terrain types after the robot has covered a certain distance. The estimation is based on acceleration data collected by an Xsens MTi altitude and heading reference system which was mounted on an aluminum plate on top of

the robot. The inertial sensor measured vibration signals in left-right direction at a frequency of 100 Hz. The collected acceleration data can be regarded as the vibration which the terrain induces to the robot's body. During data acquisition, the robot navigated over five different terrains (Fig. 2): asphalt, gravel, grass, paving, and clay (the surface of a boule court). To demonstrate that our method is not constrained to work at a certain driving speed, we varied the speed between 0.2, 0.4, and 0.6 m/s. In total, the dataset consists of more than 4000 patterns.

A compact, low-dimensional representation of the 100-sample-sized vibration segment $s = \{s_1, s_2, \ldots, s_{100}\}$ has to be chosen. Feature selection should be handled with care since high dimensional density estimation is known to be a non-trivial task suffering from the curse of dimensionality. In the context of Gaussian mixture models $k + k \cdot d + k\frac{d\cdot(d+1)}{2}$ free parameters have to be estimated for defining the k mixing coefficients, the mean vectors, and the covariance matrices. Here, k is the number of mixture components (classes) and d is the dimensionality of the data set. With an increase of the latter we also increase the possibility of getting stuck in a local minimum when applying the EM algorithm.

In this work, our vibration signature u consists of a subset of the features proposed in [18]: Given a raw vibration segment s, the feature vector u is defined as:

$$u = \{\ sgn,\ r_k,\ norm(s),\ \min(s),\ \max(s),\ std(s)\ \}^T,$$

where sgn denotes the number of sign changes in s, r_k is the autocorrelation r_k of s at lag $k = 1$, and $norm(s)$, $min(s)$, $max(s)$, and $std(s)$ denote the Euclidean norm, the minimum, the maximum and the standard deviation σ of s, respectively.

B. Experiments

Since some terrain transitions are easier to detect than other ones, creating an adequate test set must be handled with care. This is because the results depend on the order in which assembled terrain segments of varying terrain type are presented. We minimized this effect by arranging our test sets in a systematic manner: After data acquisition and feature extraction we obtain k vibration segment sets, one for each terrain type. From each of these sets, we draw a homogeneous terrain patch consting of δ consecutive vibration segments without replacement. Depending on the robot speed the assembled segments yield travel distances between $\delta \cdot 0.2\,m$ and $\delta \cdot 0.6\,m$. Then, homogeneous terrain patches of varying terrain types were grouped together yielding the final test set.

In total, two experiments were carried out to investigate the performance of the MRF-based clustering technique. In a first experiment, we generated test sets which consisted of homogeneous terrain patches of constant size δ. To analyze the temporally coherent clustering approach in situations of low-frequency and high-frequency terrain transitions, we systematically altered the homogeneous terrain patch size for varying test sets: Here, δ was chosen from the set $\{2^s\}$, $s \in [0; 5]$. In a further experiment, we generated a set of $\frac{32}{\delta}$ homogeneous terrain patches. Each homogeneous patch consisted of δ vibration segments acquired consecutively. From these sets we iteratively drew k random patches, one from each set, without replacement and assembled these patches using a random order. We repeated the drawing process until all sets of homogeneous terrain patches were empty. Finally, we iteratively applied the homogeneous terrain patch generation and patch drawing process with increasing δ. Starting from $\delta = 4$, the value of δ was increased by a factor of 2 in each step and reset to 4 if a homogeneous patch size of 32 was exceeded. The resulting test set represents a more realistic terrain setting with both high-frequency and low-frequency terrain transitions.

In both experiments, each vibration segment originated from a robot traversal at constant speed. Note that this is not a significant limitation since the speed is known at each time step and can thus be logged. After data acquisition the data is split according to the recorded speed and each data subset can be clustered separately.

C. Evaluation of Clustering

We evaluated the quality of a clustering by means of the classification performance. As classification performance measure we used the true positive rate (TPR). It is the ratio (measured in percent) between the number of correct class assignments for which the predicted terrain class equals the actual terrain class and the number of measurements contained in the data set. In this context, the predicted terrain class i is the one that belongs to the Gaussian mixture component i. Note, however, that a Gaussian mixture model is not unambiguous with regard to class assignment. Exchanging mixture component i by mixture component j we still have the same Gaussian mixture model in matters of unconditional probability estimation, but class assignment is now different from the unmodified model. Hence, we cannot generally expect that mixture component i truly represents class i. Instead, we have to find the permutation of Gaussian mixture model indices η which maximizes the likelihood that the permuted cluster index $\eta_c(i), c \in [1; k]$, represents terrain class i. Evaluating our results we did this by searching the permutation of mixture component indices which maximized the true positive rate.

Given certain homogeneous terrain patch and neighbor set sizes, the classification performance is influenced by two further parameters: the initialization of the Gaussian mixture parameter set θ and varying terrain transition detection capabilities. We countervailed the effects of the latter by generating 20 test sets for given neighbor set and homogeneous terrain patch sizes. Here, each test set offered a permuted order of terrain classes.

In our experiments, we initialized the Gaussian mixture parameter set θ using the k-means algorithm. That is, the mixture model covariances are initialized with the covariances of the clusters found by the k-means algorithm and the mixing coefficients are set to the fractions of data points assigned to the respective clusters. For the MRF-based approach, these

fractions define the initial values of each prior distribution π_i. Since the k-means clustering can yield a different initial Gaussian mixture parameter set at each run due to its random initialization, the EM algorithm might terminate with varying solutions as well. Thus, we performed 20 reruns of each clustering (given a certain test set) using a random k-means setup for each run. Note, however, that relating to a single clustering both the original and the MRF-based clustering approach used the same initial parameter values.

V. Results and Discussion

In Table II, we present the maximum (max.) and average (avg.) true positive rates (TPR) along with the respective standard deviations (std.) for given neighbor set and homogeneous terrain patch (δ) sizes. For each velocity profile, the first row represents the results of the Gaussian mixture model-based clustering approach (gmm). The following rows represent the clustering performance of the first experiment when adopting the temporally constrained MRF-based approach. In the last row (var), the true positive rates of the second experiment are shown. Results which are derived from the homogeneous neighbor set size estimation technique (Sect. III-B.3) are presented in the last column.

Table II reveals that the MRF-based approach benefits from the use of temporal constraints when enough temporal dependencies are provided. Related to the maximum TPR the classification performance is the better, the larger is the homogeneous patch size (δ). We obtain the largest maximum true positive rates when the neighbor set size is smaller than, but similar to the homogeneous patch size. If the neighbor set size exceeds the homogeneous patch size the classification performance decreases and falls below the one of the GMM-based clustering approach. Here, the filtered priors (and posteriors) are significantly influenced by neighboring prior (and posterior) distributions which likely generate another label as compared to the one of the center observation.

For the averaged TPRs, we obtain similar trends with respect to an increase of the homogeneous terrain patch size. When varying the neighbor set size in situations of low-frequent terrain transitions (δ=32), however, the neighbor set size does not significantly influence the average TPR. These results suggest that, on average, a small set of 2 neighbors suffices to resolve ambiguous class assignments. Similar findings can be derived from the second experiment, which also shows an insensibility against varying neighbor set sizes.

If there is no or only a small amount of temporal coherence contained in the data set ($\delta \leq 2$), a neighbor set size of 2 results in an inferior classification performance. This case, however, can be identified using the homogeneous neighbor set size estimation technique presented in Sect. III-B.3. As shown in Table I, it reliably estimates a homogeneous neighbor set size $\overline{\mathcal{S}}$ of 0 which is equivalent to switching to the standard GMM-based clustering approach. For the remaining homogeneous terrain patch sizes, the estimated homogeneous neighbor set sizes yield average true positive rates which are superior or at least close to those of the ones obtained when applying the most appropriate neighbor set size. Comparing the average TPRs of the GMM-based clustering approach to those of the MRF-based approach applied with the homogeneous neighbor set size estimation technique, the latter increases the classification performance by up to 8.2%, 22.8%, and 14.3% for velocity profiles 1-3, respectively. Considering homogeneous patch sizes larger than 2, all differences in the TPRs proved to be statistically significant at a significance level of 5%.

TABLE I

The estimated homogeneous neighbor set size $\overline{\mathcal{S}}$ for varying terrain patch sizes (δ) averaged over all generated test sets.

patch size (δ)	vel. 1	vel. 2	vel. 3
1	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
2	0.0 ± 0.0	0.2 ± 0.6	0.0 ± 0.0
4	2.0 ± 0.0	2.0 ± 0.0	2.0 ± 0.0
8	2.7 ± 1.0	2.6 ± 0.9	3.9 ± 0.4
16	5.3 ± 1.0	4.0 ± 0.1	6.2 ± 0.6
32	8.1 ± 0.5	5.6 ± 0.8	9.5 ± 1.5
var	6.0 ± 0.0	4.4 ± 0.8	8.8 ± 1.1

As can be seen from the pseudo code in Sect. III-B.2 the application of the temporally coherent EM algorithm only introduces little additional overhead in comparison with the Gaussian mixture model EM approach. The additional overhead of determining filtered distributions is reduced by a decrease in the number of iterations required for meeting the convergence criterion (95 ± 64 iterations for the temporally coherent EM approach, vs. 116 ± 22 iterations for the standard Gaussian mixture model EM approach). Finally, we note that the run-time complexity of the temporally constrained EM algorithm has not been an issue in our experiments: The model generation process only took 3.9 s (± 3.8 s) on average using an Intel Core Duo 2.40 GHz CPU.

VI. Conclusion and Future Works

This work focused on the clustering of different terrain types using vibration data. We therefore adopted the framework of Markov random fields which exploits temporal dependencies between consecutive observations. Extensive tests using a variety of terrain transition frequencies demonstrated the efficiency of the MRF model in situations of both low- and high-frequency terrain changes. Depending on the chosen driving speed of the robot and the frequency of terrain transitions we obtain an absolute increase in the true positive rate of up to 23.2%. We further derived a general approach which yields an estimate of the amount of temporal coherence contained in the data set. It also provides a reliable indication of the absence of temporal dependencies.

In further works, we will concentrate on unsupervised clustering approaches which do not require the number of classes to be known a priori. Referring to Markov random fields this can be realized by infinite MRFs which were recently introduced in [3]. Alternatively, we can adopt the technique of PG-means [5]. Since the latter approach is also based on Gaussian mixture model clustering it might benefit from temporally coherent mixture models as well.

TABLE II

THE MAXIMUM (MAX.) AND AVERAGE (AVG.) TRUE POSITIVE RATES ALONG WITH THE CORRESPONDING STANDARD DEVIATIONS (STD.) DEPENDING ON THE NEIGHBOR SET AND HOMOGENEOUS TERRAIN PATCH SIZES (δ).

			neighbor set size														
			2			4			8			16			estimated		
vel. 1			max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.
		gmm	66.6	64.4	± 5.3	66.6	64.4	± 5.3	66.6	64.4	± 5.3	66.6	64.4	± 5.3	66.6	64.4	± 5.3
	patch size	1	28.9	27.4	± 0.9	27.5	26.0	± 0.7	25.1	24.3	± 0.5	25.4	24.4	± 0.5	66.6	64.2	± 6.0
		2	54.0	50.4	± 2.0	48.6	43.0	± 2.5	33.1	29.9	± 1.2	26.9	25.7	± 0.7	66.6	64.7	± 5.4
		4	71.4	68.0	± 2.4	70.0	67.3	± 2.4	60.9	56.4	± 2.5	34.9	30.0	± 1.4	70.6	68.2	± 2.3
		8	73.1	70.2	± 2.3	75.9	69.8	± 2.6	72.9	66.7	± 2.7	63.6	58.4	± 3.0	74.1	69.4	± 2.5
		16	76.8	72.4	± 2.4	77.9	72.2	± 2.9	80.7	73.0	± 4.1	76.5	70.5	± 3.5	76.7	72.6	± 2.7
		32	77.6	71.1	± 2.8	78.3	71.8	± 3.7	77.6	71.0	± 3.5	80.0	70.9	± 4.2	80.3	71.3	± 4.2
		var	74.9	72.5	± 2.2	74.4	72.2	± 2.1	75.4	72.5	± 2.7	77.7	72.5	± 4.7	74.6	72.4	± 2.2
			2			4			8			16			estimated		
vel. 2			max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.
		gmm	51.2	48.4	± 0.8	51.2	48.4	± 0.8	51.2	48.4	± 0.8	51.2	48.4	± 0.8	51.2	48.4	± 0.8
	patch size	1	26.9	24.6	± 0.7	25.7	23.6	± 0.6	23.8	22.7	± 0.4	24.1	22.8	± 0.4	48.9	48.5	± 0.7
		2	50.3	42.5	± 4.3	44.9	35.6	± 4.3	29.1	25.7	± 1.1	25.2	23.4	± 0.6	50.6	48.2	± 1.0
		4	62.6	57.7	± 2.4	62.6	57.2	± 2.1	56.4	50.0	± 4.1	33.7	27.6	± 1.8	62.3	57.7	± 2.1
		8	67.9	62.0	± 2.6	68.9	62.5	± 2.8	68.2	61.4	± 2.9	60.9	54.7	± 3.0	67.4	62.3	± 2.5
		16	74.1	67.3	± 3.6	76.1	67.4	± 3.6	78.1	67.7	± 4.2	75.6	66.1	± 4.0	74.6	67.4	± 3.9
		32	78.5	70.2	± 2.8	79.0	70.8	± 3.0	81.1	71.6	± 3.4	83.4	71.0	± 3.6	78.9	71.2	± 3.0
		var	73.4	63.9	± 2.9	69.9	64.1	± 2.9	73.8	65.0	± 3.4	71.5	65.7	± 2.9	70.1	64.7	± 2.9
			2			4			8			16			estimated		
vel. 3			max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.	max.	avg.	std.
		gmm	57.9	57.4	± 0.2	57.9	57.4	± 0.2	57.9	57.4	± 0.2	57.9	57.4	± 0.2	57.9	57.4	± 0.2
	patch size	1	26.0	25.1	± 0.5	24.1	23.5	± 0.4	23.7	22.5	± 0.5	23.9	23.1	± 0.4	57.4	57.4	± 0.0
		2	45.9	39.8	± 2.3	36.6	34.3	± 1.0	28.2	27.0	± 0.6	25.1	24.3	± 0.5	57.4	57.4	± 0.0
		4	61.1	55.7	± 1.4	61.5	54.4	± 1.9	48.6	41.8	± 3.7	30.1	28.1	± 1.0	62.6	56.2	± 1.7
		8	69.3	62.8	± 1.6	68.9	63.1	± 1.9	68.5	59.5	± 2.5	55.7	50.0	± 2.4	67.4	62.4	± 2.1
		16	71.0	65.6	± 2.0	72.4	67.0	± 1.5	73.0	66.6	± 2.0	68.6	61.8	± 2.8	72.0	67.2	± 1.5
		32	73.9	70.7	± 1.5	72.7	69.9	± 1.4	76.5	71.6	± 1.5	77.2	72.0	± 1.6	74.1	71.6	± 1.4
		var	67.8	64.2	± 2.0	67.6	65.6	± 0.7	70.7	65.6	± 2.2	66.9	62.9	± 2.3	70.7	65.5	± 2.1

REFERENCES

[1] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.

[2] C. A. Brooks and K. Iagnemma. Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics*, 21(6):1185 – 1191, 2005.

[3] S. Chatzis and G. Tsechpenakis. The infinite hidden Markov random field model. In *IEEE International Conference on Computer Vision (ICCV 2009)*, pages 654–661, Kyoto, Japan, October 2009.

[4] A. Diplaros, N. Vlassis, and T. Gevers. A spatially constrained generative model and an EM algorithm for image segmentation. *IEEE Transactions on Neural Networks*, 18(3):798–808, 2007.

[5] Y. Feng and G. Hamerly. PG-means: learning the number of clusters in data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 393–400. MIT Press, Cambridge, MA, 2007.

[6] S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[7] P. Giguere and G. Dudek. Clustering sensor data for terrain identification using a windowless algorithm. In *Proceedings of Robotics Science and System (RSS)*, pages 25–32, Zürich, Switzerland, June 2008.

[8] K. Iagnemma and S. Dubowsky. Terrain estimation for high-speed rough-terrain autonomous vehicle navigation. In *Proceedings of the SPIE Conference on Unmanned Ground Vehicle Technology IV*, Orlando, FL, USA, 2002.

[9] P. Komma, C. Weiss, and A. Zell. Adaptive Bayesian filtering for vibration-based terrain classification. In *IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan*, pages 3307–3313, May 2009.

[10] P. Komma and A. Zell. Posterior probability estimation techniques embedded in a Bayes filter for vibration-based terrain classification. In *7th International Conference on Field and Service Robots (FSR 2009)*, MIT, Cambridge, Massachusetts, USA, May 2009.

[11] W. J. Krzanowski, editor. *Principles of multivariate analysis: a user's perspective*. Oxford University Press, Inc., New York, NY, USA, 1988.

[12] M. H. Law, A. Topchy, and A. K. Jain. *Clustering with soft and group constraints*, volume 3138 of *LNCS*. Springer, January 2004.

[13] J. Modestino and J. Zhang. A Markov random field model-based approach to image interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(6):606–615, 1992.

[14] B. Nelson and I. Cohen. Revisiting probabilistic models for clustering with pair-wise constraints. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 673–680, New York, NY, USA, 2007. ACM.

[15] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[16] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[17] C. Ward and K. Iagnemma. Speed independent vibration-based terrain classification for passenger vehicles. *Vehicle System Dynamics*, 47(9):1095–1113, September 2009.

[18] C. Weiss, H. Fröhlich, and A. Zell. Vibration-based terrain classification using support vector machines. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 4429–4434, Beijing, China, October 2006.

[19] B. H. Wilcox. Non-geometric hazard detection for a mars microrover. In *Proceedings of the NASA Conference on Intelligent Robots in Field, Factory, Service and Space*, pages 675–684, Houston, TX, 1994.

[20] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, January 2001.