Temporally Coherent Terrain Discrimination Using Inertial Sensor Information

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard-Karls-Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Philippe Komma

aus Ostfildern-Ruit

Tübingen
2012

Tag der mündlichen Qualifikation:	17.10.2012
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Andreas Zell
2. Berichterstatter:	Prof. Dr. Hanspeter A. Mallot

"*The only way to find out how to do a PhD is to do one. Therefore all advice is useless.*"

Richard Butterworth

Abstract

Along with a growing number of applications in unstructured outdoor environments, mobile robots are faced with increased requirements with respect to their driving safety and operational demands. Issues arise due to varying terrain types, each possessing different navigability characteristics such as the degree of hazardousness. Hence, a safe robot traversal necessitates the classification of the present ground surface from sensor data. In this thesis, a retrospective approach based on tactile sensing has been considered for the terrain discrimination task. That is, the ground surface classification relies on sensor data being collected while the robot traverses the environment. Here, the sensor data is represented by means of preprocessed acceleration patterns which emanate from terrain-wheel-chassis interactions and directly relate to the mechanical properties of the ground surfaces.

The main contribution of this thesis comprises the integration of contextual information into the terrain classification procedure. Here, contextual information denotes the temporal dependencies between consecutive measurements which are likely to arise from the same terrain type. It is demonstrated that the modeling of these dependencies in a principled way results in a significant improvement of the classification performance. As for the underlying framework, the Bayes filter approach has been adopted and modified so as to requiring the posterior probability of individual terrain classifications only. Starting with a support vector machine as the base classifier, the latter technique is compared to other machine learning techniques which also provide class posterior estimates. In this context, the random forest and random ferns classifiers are employed as novel means for ground surface estimation. It further shows that the random ferns approach as well as several other classifiers benefit from a more compact acceleration signal representation which is based on the Mel-frequency cepstral coefficient extraction process.

The classification approach requires the terrain classes to be known a priori. Since this information might not be available in all domains, this thesis also addresses the problem of unsupervised learning. That is, given acceleration data acquired during the robot traversal a model is established which autonomously partitions the data instances into clusters such that observations with similar characteristics are assigned to the same cluster. Dissimilar measurements, on the other hand, are to be located in differing clusters. Analogous to supervised classification, the proposed model makes use of temporal coherences contained within subsequent observations. These temporal dependencies are incorporated by means of a Markov random field-based clustering approach which assumes that the class labels of nearby acceleration patterns are generated by prior distributions with similar parameters. Experimental results reveal the superiority of the temporally coherent approach in comparison with the one which does not exploit temporal dependencies. The problem of obtaining varying local optima after the clustering process is addressed in terms of a deterministic cluster model initialization scheme. It shows that the inclusion of temporal coherences within the initialization step increases the clustering performance.

Further contributions involve the introduction of a novel unsupervised feature selection approach which enables the identification of important characteristics of the acceleration signal. Finally, systematic means are presented which allow for the estimation of the number of terrain classes when this information is not provided.

Zusammenfassung

Einhergehend mit einer wachsenden Anzahl von Anwendungen in unstrukturierten Umgebungen steigen auch die Ansprüche an den mobilen Roboter bezüglich seiner Fahrsicherheit und Fahrwerkbeanspruchung. Probleme ergeben sich durch verschiedene Terraintypen, welche unterschiedliche Merkmale bezüglich der Navigierbarkeit wie beispielsweise dem Grad der Fahrunsicherheit aufweisen. Wie sich hieraus ableiten lässt, erfordert eine sichere Roboternavigation die Klassifizierung des gegenwärtigen Terrains anhand von Sensordaten. In dieser Arbeit wurde für die Terrainidentifikationsaufgabe ein retrospektiver Ansatz basierend auf taktilen Eingabesignalen gewählt. Das heißt, dass die Klassifikation auf Sensordaten beruht, welche während der Terraintraversierung aufgenommen werden. In diesem Zusammenhang werden die Daten durch vorverarbeitete Beschleunigungsmuster repräsentiert, welche aus Terrain-Rad-Fahrgestell-Interaktionen hervorgehen und direkt mit den mechanischen Eigenschaften des Untergrundes in Verbindung gebracht werden können.
Der wesentliche Beitrag dieser Arbeit besteht in der Einbindung von Kontextinformation in den Terrainvorhersageprozess. Hierbei bezieht sich der Term „Kontextinformation“ auf temporale Abhängigkeiten zwischen aufeinanderfolgenden Messungen, welche mit hoher Wahrscheinlichkeit von derselben Terrainklasse stammen. Es zeigt sich, dass durch ein methodisches Modellieren dieser Abhängigkeiten die Klassifizierungsleistung signifikant verbessert werden kann. Als zugrundeliegendes Wahrscheinlichkeitsmodell wurde der Bayes-Filter-Ansatz gewählt, welcher derart umformuliert wurde, dass die Ergebnisse des Filterprozesses allein aus den a posteriori Wahrscheinlichkeiten einzelner Terrainschätzungen abgeleitet werden können. Ausgehend von der Support Vektor Maschine als Basisklassifizierer wird dieser Ansatz mit anderen Verfahren des maschinellen Lernens verglichen, welche ebenso a posteriori Verteilungen bereit stellen. In diesem Zusammenhang werden die Random Forest und Random Ferns Ansätze als neue Methode zur Terrainklassifizierung vorgestellt und bezüglich der Klassifizierungsleistung verbessert. Weiterhin zeigt sich, dass die Random-Ferns-Technik und einige weitere Klassifizierer von einer kompakteren Beschleunigungsdatenrepräsentation profitieren, welche auf der Extrahierung von Mel-frequency cepstral Koeffizienten beruht.
Der Klassifikationsansatz benötigt zur Modellbildung Sensordaten von allen zu klassifizierenden Untergrundtypen im Voraus. Da diese Information nicht in allen Einsatzgebieten vorhanden sein muss, legt diese Arbeit einen weiteren Schwerpunkt auf unüberwachte Lernverfahren. Das heißt, dass bei gegebenen Daten einer Robotertraversierung ein Modell zu erstellen ist, welches die Sensordaten autonom in Cluster unterteilt, so dass Messungen mit ähnlichen Merkmalen den selben Clustern zugewiesen werden. Unähnliche Beobachtungen sollen andererseits in unterschiedliche Cluster platziert werden. Analog zur überwachten Klassifikation verwendet das vorgeschlagene Modell temporale Kohärenzen aus aufeinanderfolgenden Messungen. Diese temporalen Zusammenhänge werden mittels eines Markov-Random-Field-Ansatzes in den Clustervorgang integriert. Hierbei basiert das probabilistische Modell auf der Vermutung, dass die Klassenbezeichnungen von naheliegenden Beschleunigungsmustern aus a priori Verteilungen mit ähnlichen Parametern hervorgehen. Die resultierenden experimentellen Ergebnisse zeigen den Vorteil der temporal kohärenten Methode im Vergleich zum Alternativansatz auf, welcher nicht auf temporalen Abhängigkeiten beruht. Das Problem der randomisierten Initialisierung der Clustermodelle und der daraus resultierenden Varianz bei der Klassenzuweisung wird mittels eines deterministischen Initialisierungsprozesses gelöst. Auch in diesem Zusammenhang erweist sich die Einbindung temporaler Kohärenzen als vorteilhaft.

Ein weiterer Beitrag dieser Arbeit besteht in der Entwicklung einer neuen unüberwachten Methode zur Merkmalsselektion, welche die Identifikation wichtiger Charakteristika des Beschleunigungssignals ermöglicht. Schließlich werden Methoden vorgestellt und weiterentwickelt, welche die Anzahl der Terrainklassen abschätzen, wenn diese Information nicht gegeben ist.

Danksagung

Meine Dissertation wäre nicht so erfolgreich verlaufen, hätte ich keine so große Unterstützung von Außen erfahren. Beginnend bei meinem Doktorvater, Herrn Professor Zell, bei welchem ich mich vor allem dafür bedanken möchte, dass er mir sein Vertrauen schenkte und mir damit erst ermöglichte, meine Promotion an seinem Lehrstuhl aufzunehmen. Insbesondere seine anregenden Vorlesungen zum Thema neuronale Netze und evolutionäre Algorithmen weckten mein Interesse für das Gebiet der maschinellen Lernverfahren, welches zum Schwerpunkt meiner Dissertation wurde. Weiterer Dank gilt ebenso für die zahlreichen Konferenzbesuche sowie Einsätze bei Drittmittelprojekten. Letztere boten mir nicht nur die Möglichkeit, mein Wissen zu vertiefen, sondern auch mit anderen Forschern und Industriepartnern in Kontakt zu treten, was meiner Motivation zusätzlichen Auftrieb verlieh. Besonders sei an dieser Stelle das Projekt mit BorgWarner Beru Systems, geleitet und betreut von Herrn Dr. Wolfgang Wenzel und Herrn Dejan Kienzle, erwähnt, welches mir einen kleinen Ausblick in die Zeit nach der Promotion gewähren sollte. Dankend sei auch Herr Professor Hanspeter-Mallot genannt, welcher nicht nur den Zweitkorrektor meiner Arbeit darstellt, sondern vielmehr einen Eckpfeiler meines Studiums. Mit seiner freundlichen Art und interessanten Vorlesungen ebnete er den Weg für die kognitiven Forschungsaspekte während meiner Promotionszeit.
Besonderer Dank sei auch an mein engeres Umfeld verwiesen, welches mir stets ein gesundes Maß an Zerstreuung offerierte und unter anderem dafür verantwortlich war, mich nicht in den Alltagssorgen eines Promovierenden zu verlieren. Ob meine Lebensgefährtin Sanna Zimmermann, meine Eltern, Draha und Walter Komma, mein Bruder Christopher und auch meine Schwiegereltern, Riitta und Werner Zimmermann, alle trugen sie aufgrund dem hohen Maß ihrer Zuneigung und Unterstützung vor, während und nach meiner Promotionszeit einen nicht unwesentlichen Anteil an meinem persönlichen Werdegang.
Doch ist an dieser Stelle nicht nur familiärer, sondern auch kollegialer Dank auszusprechen. Zum einen an meine beiden Zimmergenossen Hannes Planatscher und Holger Franken für die wissenschaftlichen Auseinandersetzungen und nicht minder erwähnenswerten persönlichen Gespräche, Philipp Vorst als verlässlicher Korrekturleser und Diskussionspartner, Marcel Kronfeld und Sebastian Scherer für ihre hilfsbereite, konstruktive und vor allem kompetente Unterstützung bei Projekten und Übungsgruppen, Vita Serbakova als freundliche Unterstützung bei organisatorischen Angelegenheiten sowie Vorbereitungen auf unterschiedliche Projekttreffen, Klaus Beyreuther für sein engagiertes Arbeiten und Lösen zahlreicher Soft- und Hardwareprobleme und nicht zuletzt Christian Weiss, welcher mit seiner Forschung die Grundlage dieser Dissertation ebnete.

Contents

Nomenclature

$X \equiv \{x_i\}_{i=1}^{n} \equiv x_{1:n}$	set of input patterns
$Y \equiv \{y_i\}_{i=1}^{n} \equiv y_{1:n}$	set of true class labels
$\mathbb{I}$	indicator function
a_i	predicted class or cluster given input pattern x_i
c_j	class j
d	dimensionality of the input patterns, $x_i \in \mathbb{R}^d$
$dist$	travel distance
k	number of classes or clusters, respectively
n	number of instances contained in the data sets
$x_{i,j}$	jth feature of the ith input pattern
$H(X)$	entropy of random variable X
$MI(X,Y)$	mutual information between random variables X and Y
$\mathbb{E}(X)$	expected value of random variable X
AMI	adjusted mutual information index
ARI	adjusted rand index
BAYES	Bayes filter classification approach
BIC	Bayesian information criterion
DAS	DFT amplitude spectrum
EM	Expectation-maximization
GMM	Gaussian mixture model
KS-test	Kolmogorow-Smirnow-test
LDA	linear discriminant analysis
MCD	minimum covariance determinant
MFCC	Mel frequency cepstral coefficients
MRF	Markov random field

MSPRT	multi-hypothesis sequential probability ratio test
SBS	sequential backward selection
SFS	sequential forward selection
SO	single observation classification approach
SVM	support vector machine
TPR	true positive rate
UGV	unmanned ground vehicle

1. Introduction

1.1. General Overview and Motivation

Recently, a growing number of outdoor tasks for mobile robots such as planetary and rescue missions as well as agricultural assignments has emerged. Thereby, the main motivation of the various application domains is diverse. For example, planetary robotics is concerned with the search for life in our solar system and beyond. Particularly, researchers focused on Mars with regard to both the astrobiological question of whether life forms were present on this planet [EKL+02] and its ability of being colonized. In this context, a planetary unmanned ground vehicle (UGV) has to navigate in an unknown, hostile terrain, recognize and circumnavigate obstacles, and acquire samples from scientific targets [Ell08]. Environments of similar hostility can be found in the domain of disaster management, requiring a large number of heterogeneous and autonomous mobile robots. The necessity of considering the involved search and rescue tasks is given by natural disasters induced from seismic activities of the tectonic plates. For example, the Great Hanshi-Awaji earthquake which hit Kobe City in 1995 caused more than 6500 causalities, destroyed more than 80000 wooden houses, and damaged all infrastructures whose reconstitution costs were estimated at more than 1 billion US dollars. Likewise, the nuclear reactor incident of Fukushima in the recent past showed that catastrophes of historical proportions can befall us at any time and unexpectedly. Robots which are assigned to search and rescue tasks should not only be able to collect necessary information and provide physical support but also to fulfill their task reliably and robustly. Finally, autonomous robots employed in the field of agriculture aim at the reduction of energy resources and hence production costs in the long run. According to Blackmore et al. [BSR05], they "should have enough intelligence embedded within them to behave sensibly for long periods of time, unattended, in a semi-natural environment, whilst carrying out a useful task".
All given applications involve an increasing demand regarding a robot's driving behavior. Particularly in outdoor environments, an UGV is exposed to a variety of different terrain types. To enable a safe traversal of unknown terrain, the robot should adapt its driving style according to the present ground characteristics. While even surfaces with good traction allow a traversal at high speed, loose, slippery and bumpy surfaces are hazardous and require a reduction of the driving speed to avoid robot damages. In literature, hazards that are attributed to the ground surface are known as non-geometric hazards [Wil94].

1.1.1. Approaches for Terrain Hazardousness Estimation

The presence of hostile terrain types shows the necessity of a reliable terrain classification scheme. Thereby, the hazardousness of the ground surface can be inferred by at least two different approaches. One technique involves a direct estimation of terrain parameters like cohesion or slippage without knowing the exact terrain type the robot is driving on. Here, it is assumed that all control information can be directly inferred to adequately update the control settings of the robot. As mentioned in the context of wheel slip [AMH+06, WSG+09], however, it is sometimes difficult to determine a consistent measure of these quantities. Especially

in outdoor environments, terrain characteristics are likely to change frequently, resulting in numerous changes of the driving behavior.
In the classification approach, different terrain types are grouped into classes each representing a ground surface of a certain degree of hazardousness. Using sensor measurements, a model is generated which predicts the present class from the set of available classes. The classification approach assumes that the instances of each terrain class affect the driving behavior in a consistent manner. Hence, the choice of predetermined control settings is likely to yield the best vehicle performance with respect to a specific terrain. This dissertation focuses on proprioceptive classification algorithms which are based on the analysis of the interaction between the robot and the terrain. Thereby, the robot "senses" internal and external variations like wheel sinkage, induced acoustic noise, or wheel slippage. More commonly, proprioceptive algorithms focus on vehicle vibrations as originally proposed by Iagnemma et al. [ID02]. They showed that extracted signatures from vibration data provide enough information to distinguish between different terrain classes. Usually, accelerometers are used to record vibration signals during the robot traversal. The sensors can be attached at the wheels, the axes, or the body of the robot. The use of tactile feedback is motivated by findings in the animal kingdom. There, research showed that whiskers in animals are able to extract information about surface texture and shape [Vin12, Ley79]. For example, rats employ these sensors to compensate for their lack of adequate vision abilities. Note that the visual acuity of a rat is approximately 30 times less accurate than the one of humans [BJ79].

1.1.2. Terrain Classification in the Automotive Domain

In the automotive domain, the benefits of considering terrain-based modifications of control settings have been demonstrated by car manufacturers such as Land Rover and the Ford Motor Company. The first system, denoted as *Terrain Response*, was introduced for the Land Rover LR3 distinguishing five different terrain modes: general driving, grass/gravel/snow, mud and ruts, sand, and rock crawl [Van05]. Depending on the selected control mode, various settings of the vehicle system are changed such as the anti-lock braking system (ABS), the traction and stability control systems, the locking action of the differentials, the shift schedule of the transmission, and the throttle response of the engine in order to improve traction, steering, and fuel efficiency. Concerning the results, the application of the *Terrain Response* approach results in a 3%-17% improvement in zero to twenty mph acceleration on simulated ice surfaces and a reduction in stopping distance of up to 35% on mud, sand, and gravel terrains [Van05]. In a later car, the Land Rover LR4, additional terrain response features have been implemented. For example, the LR4 now includes a mode for "sand launching" which prevents the wheel spin when moving from stillstand [Har09]. A similar terrain-dependent control system, called *Terrain Management* has been developed by the Ford Motor Company. Integrated within the newly unveiled 2011 Ford Explorer, the control mode system also distinguishes between five different terrain types enabling features such as an increased slip and stability control in grass, gravel, and snow environments, among others.

1.1.3. Temporally Coherent Terrain Classification

The use of terrain-dependent control strategies in a real-world application illustrates the importance of ground surface classification techniques. It is noticeable that none of the existing approaches makes use of temporal coherences. Each terrain classification only considers the actual observation. However, it is very likely that the robot traverses over the same terrain type

for several time steps. Hence, not only actual sensor readings should influence the classification but also former ones. The development of a systematic means which filters the predictions of several times steps is addressed in this thesis. It is assumed that while temporal coherences show significant benefits when this assumption proves to be valid, issues occur in situations of fast system changes. Thus, the established technique must take the latter cases into account to not compromise the results negatively. In the context of terrain classification that means that the chosen approach must be both reactive and stable enough to detect fast terrain transitions and selective false classifications.
Further, supervised terrain classification can only be considered as a first step towards the complete autonomy of the robot. Problems occur if the autonomous vehicle is placed in environments containing completely or partially unknown ground surfaces. Here, the classification approach fails, since the robot is only able to predict the terrain types which were presented during training. More appropriately, UGVs should generate a model of the terrain characteristics on their own, assigning an acquired observation to a specific terrain class or cluster in later stages. Hence, beside the supervised classification approach mentioned above, learning in the unsupervised case is also taken into account for the problem of terrain understanding. 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 vibration 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 certain 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 instances 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. Similar to the supervised case, the use of temporal dependencies between consecutive measurements is advised 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. Starting from clustering with temporal constraints in which the number of clusters k is assumed to be known, several other aspects of the unsupervised learning task are taken into account including model initialization, feature selection, and model selection. The latter demonstrates the performance of clustering approaches when the number of clusters is not known a priori which can be regarded as a further increase of the degree of autonomy.

1.2. Related Work

In this section, the related work with regard to terrain discrimination is presented. Here, the term terrain discrimination comprises both the terrain classification and clustering task and is introduced to facilitate the notations. The objective of this summary is not only to provide a systematic hierarchical structuring of previous research but also to reveal the benefits and disadvantages of each approach. For the reason of clarity and compactness, this survey only

focuses on fundamental concepts. A more detailed overview of individual research is provided in the following chapters.

1.2.1. A Classification of Ground Surface Estimation Approaches

Terrain classification comprises the following two problem domains: first, the navigation task, which determines the path the robot is heading for, and second, the control task which answers the question of how the UGV reaches its destination. In outdoor environments, both questions are highly related, since the traversability characteristics of certain terrain types directly influence the robot traversal. Note that these issues differ in the context of indoor environments where the ground surface is assumed to be constant and the path is defined by obstacles.
Terrain classification with regard to navigation is addressed using forward-looking sensors which determine the terrain type prior to the robot traversal. Thereby, two approaches can be distinguished: traversability estimation and ground surface classification. Whereas the first approach only divides the present terrain into traversable and non-traversable sections in a binary manner, the second technique aims at classifying the underlying ground surface given a set of available classes. As for the sensors, camera and terrestrial laser scanner devices such as RADAR or LIDAR (LADAR) can be employed.

Image-based Terrain Classification

In the camera-based setting, an image is decomposed into smaller primitives where each of the primitives is assigned either a terrain class [FC97, AMHP07a, TC09] or a binary traversability characteristic [EHS$^+$07, VTL08, BBMG09]. Commonly, these primitives are represented in terms of pixels [THKS88] or patches [DVH04, PUH03, KSO$^+$06]. Pixels, however, are prone to noisy estimations, rendering the task of identifying homogeneous regions complicated [DKSM95]. Region primitives such as patches, on the other hand, enable the use of local image descriptors such as color distributions and texture statistics.

Ground Surface Estimation using Color Features The use of color features is motivated by the neurobiological findings that the human eye is able to differentiate thousands or even millions of color shades [Leo06] in comparison with only two-dozen shades of gray. As a consequence, several classification and indexing techniques arose which rely on color features only. Although color provides an efficient tool in these domains, color features suffer from a variety of deficiencies. These include changes in color perception in dependence of daytime, the weather condition, and observer position and orientation. For example, when considering a certain surface, the variation in the perceived color due to illumination and the object's reflectance properties can be larger than the one caused by two differing colors. Although color constancy approaches exist which aim at the reduction of these effects, these techniques are based on simplified assumptions concerning the illuminant and the object's reflective characteristics. For instance, they adopt a local lighting model only [Sch04] which renders them incapable of taking object inter-reflections into account. In an outdoor environment, however, the robot is faced with varying shading and weather conditions. Hence, the above-mentioned assumptions on which the color constancy techniques rely are likely to fail.

Texture Features for Terrain Classification Texture features, on the other hand, proved to be more robust with respect to illumination changes. Texture describes an intrinsic property of virtually all surfaces, such as wood, gravel, or PVC floors. Informally, it represents the

Figure 1.1.: Images acquired from a robot traversal at a speed of 0.6 m/s which show the loss of image structure due to motion blur artifacts.

distribution of gray-level variations or regular structural patterns within an image and captures the relationship to the surrounding environment. In comparison with color, texture is defined using a local area around a certain image pixel as opposed to the pixel itself. Furthermore, texture information is usually extracted by means of gray level images only and hence can be regarded as orthogonal to color-based techniques. Various approaches have been suggested for texture representation: Tuceryan et al. [TJ98] thereby distinguished between four classes: statistical [HSD73], geometric [Wil89, Asa99], model-based [Bes74, LP96], and signal processing methods [BCG90, MFM04]. Statistical techniques describe the spatial organization of the gray-level image. Based on the distribution of pixel intensities, several types of simple statistical features are determined. Among these features, the co-occurrence approach proved to be the most appropriate for terrain classification [OD92]. Geometric texture analysis techniques describe textures in terms of texture primitives and their spatial distribution. These primitives are obtained by applying image filters such as edge detection or morphological filters. A mathematical process describing the texture is generated for model-based texture analysis. Examples of these techniques include the random mosaic model [AR81] or Markov random fields [KH95, MC91]. Finally, signal processing approaches apply linear transforms, filters, or filter banks to the texture. In a succeeding step, energy measures are extracted from the preprocessed textures. The key idea of signal processing techniques is to transform the texture data into another representation which is better suited for classification. The transform approach is motivated by the finding that the most important information is contained in the low-energy frequency components of the image. High-frequency content, in contrast, only represents noise and can thus be omitted. Despite their use in many classification tasks, texture approaches have several shortcomings. First, they are sensitive to motion blur artifacts emanating during robot traversal at higher speeds. As shown in Figure 1.1, nearly all texture information is lost in the acquired image. Second, vision-based classifiers are only able to extract features from the topmost terrain surface which does not necessarily represent the load-bearing surface of interest. For example, fallen leaves and snow (cf. Figure 1.2) or stalks of straw occlude the actual ground surface and hence result in a false classification of traversability and, as a consequence, in an improper robot navigation.

Laser-based Terrain Classification

To circumvent the effects induced by partial or full occlusion, terrain classification schemes based on laser devices have been proposed. These devices are capable of directly sensing depth information in the form of unorganized 3-D point clouds. Labeled data can then be employed to determine three-dimensional features representing the inputs for predictive models. In the following, a trained predicted model is capable of distinguishing between load-bearing surfaces and vegetation [WCS05, WCS06, WKSB09], detecting obstacles [MMM00, MBC$^+$03], and

Figure 1.2.: Examples of non-trivial terrain classification due to the presence of occlusion caused by fallen leaves and/or snow.

terrain classification [VHKH04, LOCD09]. Beside the laser-based traversability approach of Wolf et al. [WSFB05] which divides the traversed path in navigatable and non-navigatable areas, researchers addressed the problem of classifying outdoor entities such as large tree trunks and ground surfaces. Vandapel et al. [VHKH04] therefore employed three-dimensional ladar data which is preprocessed by local point statistics. Their system enabled the discrimination between three classes including clutter to identify grass and tree canopy, linear structures to classify thin objects such as wires or tree branches, and ground surfaces to capture solid objects like ground terrain surfaces and rocks. While the classification technique of Vandapel et al. is able to identify both, long and mid-range environment entities, the laser stripe-based structured light sensor approach of Lu et al. [LOCD09] performs terrain classification at very close range at a distance of less than 1 m. Yet, it allows for a more detailed classification of present terrain types instead of outdoor entity classes. Note, however, that the laser stripe-based structured light sensor consists of both a laser and a camera yielding range and intensity information. Hence, the latter approach can be considered as a combined camera and laser technique similar to the ones presented by Rasmussen [Ras02] and Häselich [HALP11].
Along with their many benefits such as lighting independence or night-vision ability, the application of laser-based approaches remains difficult for at least four reasons [BL11]: first, the variable degree of resolution and sparsity of the data due to inevitable shadowing effects, second, the intrinsic heterogeneity and complexity of natural surfaces, and third, the large amount of data which is generated by modern terrestrial laser scanners. Finally, fallen leaves or a thin layer of snow which cover the complete ground surface result in occluding artifacts and misguide the driving behavior selection scheme.

Proprioceptive Ground Surface Estimation

The issue of covered terrain can be addressed by considering interactions between the robot and the ground surface. In this way, the UGV does not only “see” the current terrain but also “feels” it using proprioceptive sensors, similar to human perception during car navigation. Since terrain characteristics such as wheel slip or wheel sinkage are difficult to measure accurately, proprioceptive terrain classification usually relies on vehicle vibrations. This is mainly due to the ease of sensor evaluation yielding an estimate for the vibration strength. In the literature, whiskers are known to be an efficient solution for collecting information about

surface properties. The generated whisker deformation can be detected using transducers representing a device which converts one form of energy into another representation such as voltage. The quality of the latter is measured by the transducer's inherent characteristics with regard to sensitivity, resolution, noise, and bandwidth. Examples include simple electrical contacts [Wal50, Rus84], potentiometers to measure springy whiskers deflections [JZ96], strain gauges [PPM+07], load cells [SR04], Hall-Effect sensors [HAZ+06], microphones recording sound data [RDF96], and sensors detecting changes in the optical transmission of fiber optic whiskers [RSPX01]. For wheeled vehicles, the most frequently employed transducer is an accelerometer [BID05, CCJD+08, DMRCJ05, DMS+05, DMCJC08, DRM06a, OBWK06, WFZ06, WFSZ07, WSZ07, WZ08, WTZ08]. This type of sensor is more sensitive to rapid motion in comparison with the whisker device mentioned above. Another important aspect of accelerometer signals is their frequency response: Terrain discrimination is performed by spatial terrain signatures which arise from vehicle vibrations. The term spatial terrain signature denotes the features of the spatial frequency response of a certain terrain which distinguishes its frequency response from the one of other ground surfaces. Several approaches demonstrated the effectiveness of these signatures in various applications such as planetary rovers [BI05], autonomous ground vehicles [DMCJC08], and experimental unmanned vehicles [DMR08]. One drawback of vibration-based terrain classification schemes is that their sensitivity highly depends on the vehicle itself. A heavy chassis mounted on a suspension system in combination with air-filled tires can be regarded as a damped mass-spring system. This damping results in a (partial or complete) removal of high-frequency components of the acceleration data which yields an inferior discrimination performance as compared to vehicles with no suspension [GD09]. Further issues arise from the fact that a proprioceptive-based classification approach can only identify terrain which was already traversed by the robot.

Combining Vision- and Proprioceptive-based Terrain Classification Approaches

A combination of forward-looking and proprioceptive techniques was advised by Krebs et al. [KPS09]. While the former is used to predict the terrain type in front of the robot, the latter technique validates the prediction. Classifier fusion is achieved in terms of the AdaBoost [FS95] algorithm which autonomously selects the most appropriate vibration and vision characteristics. From these characteristics, weak classifiers are trained in a succeeding step whose linear combination yields a strong classifier. In another approach, Sarvadevabhatla [Sar06] established a relationship between visual features and proprioceptive ones as the robot navigates through unknown terrain. Therefore, a vibration-based classifier is employed to automatically adapt a visual feature-based classifier. This visual classifier is then able to predict terrain types in the distance. Note that the latter technique is an example of self-supervised learning. Here, the key idea is that one sensor provides the ground truth for learning a predictive model which relies on other sensor modalities. In other words, an a priori trained vibration-based terrain classifier provides the labels for the succeeding on-line visual model generation process. After labeled training data has become available, standard supervised classification techniques can be used to train the visual classifier. A similar approach to the one of Sarvadevabhatla has been proposed by Brooks et al. [BI07] using differing acceleration and visual features. Measured slip was adopted by Angelova et al. [AMHP07b] to train a vision system being able to recognize regions of potentially high slip in the context of Martian exploration. These approaches show the benefits of local terrain sensing in terms of vibration signatures resulting in an improvement of the overall system performance.

Terrain Classification using Incremental Machine Learning Techniques

The above-mentioned techniques follow a supervised classification paradigm in which a generative model is trained to distinguish between certain terrain classes whose number is known a priori. In outdoor environments, however, the robot is faced with an unknown number of terrain classes. This renders the identification of ground surfaces necessary which do not belong to a predefined set of classes. Brooks et al. pointed out [BI09] that the unnoticed traversal of novel terrain classes result in both a loss in mobility safety when classifying hazardous unknown terrain as a ground surface with safe characteristics and potentially missed scientific opportunity due to the disregard of scientifically interesting terrain types. This problem was addressed by Brooks et al. [BI09] and Weiss et al. [WZ08] in the context of novelty detection strategies. While the former researchers rely their approach on a two-class SVM, the latter ones employed a mixture model-based technique to identify novel terrain.

Unsupervised Ground Surface Estimation

Another class of learning problems is unsupervised learning in which the data set does not contain any truly labeled instance. Instead, a clustering algorithm has to establish an appropriate data subdivision assigning similar data instances into the same clusters while distributing dissimilar instances into differing clusters. In [GD08], Giguere et al. proposed a novel off-line vibration signature clustering framework exploiting time-dependency between consecutive vibration samples. Their system is based on the optimization of a user-specified classifier with respect to a given cost function which takes the temporal correlation of sensory measurements into account. Note that the application of temporal coherences is well founded in literature with respect to several other domains such as computer graphics [SYM10, SSM11, MBW08, DFR04], computer vision [RF03, MCW09], and robotics [BB89, CKW94, Bar01].

1.3. Thesis Outline and Research Objectives

The main focus of this dissertation is to integrate temporal coherences into the domain of terrain classification and clustering using proprioceptive sensors. As previously described, this is an important domain of research, since most current approaches rely the terrain type discrimination task on single observations only. The remainder of this dissertation is organized into nine chapters which discuss domain-specific problems along with their solutions. The objective of the second chapter is twofold. Besides introducing the autonomous robot, the proprioceptive sensor, and the characteristics of sensor observations, the basic elements of the pattern recognition and clustering approaches are presented. A more detailed description of these elements is provided in Chapter 3. Here, not only the model generation processes are taken into account but also the recall phase along with the assessment of the model quality. Starting from Chapter 4, the contributions of this thesis are presented. This chapter shows how temporal coherences can be efficiently integrated into the terrain classification framework in terms of a recursive Bayes filter. To render this approach applicable in situations of both low-frequency and high-frequency terrain changes, the Bayes filter is modified adaptively based on a history of terrain type estimates. The latter, however, were derived from support vector machines only, disregarding the capability of other classification techniques to provide these estimates. Chapter 5 thus considers other classifiers to be embedded into the Bayes filter prediction scheme, each featuring different characteristics. Furthermore, a novel preprocessing scheme is proposed which does not only decrease the time spent on training the model but also improves the qual-

ity of certain classifiers such as the Gaussian mixture-based technique. As demonstrated later, this finding will become important in the following chapters which address the unsupervised learning task.
Since training instances of each present terrain type might not be available a priori for some application domains, Chapter 6 considers the problem of unsupervised vibration data clustering when no training data is provided. Here, a Markov random field-based clustering approach is proposed taking the inherent temporal dependencies between consecutive measurements into account. As a further contribution, a general means is derived enabling the estimation of the model's most important parameter from the observed data. Motivated by the findings of Chapter 5 which reveal an advantage of compact feature representations over longer ones, a novel unsupervised feature selection technique is proposed and experimentally evaluated against current state-of-the-art methods. The feature selection scheme is based on a mutual information measure which has already been successfully applied in the supervised case [KZ09b]. Since the non-deterministic initialization procedure of the Markov random field-based clustering technique introduces an unwanted overhead during feature selection, several means of obtaining a deterministic model initialization scheme are presented in Chapter 7. It shows that, in this context, the newly proposed technique incorporating temporal constraints yields the best generalization performance. The experimental part of this thesis is concluded by a systematic comparison of techniques estimating the number of modes (clusters) in the data set when the number of clusters is not provided in advance (Chapter 9). To assess the performance of the resulting cluster models, two novel quality measures are proposed. The first one penalizes homogeneous, yet smaller clusters less in comparison with existing approaches while the other measure provides information about the splitting tendency of certain ground surfaces. Finally, Chapter 10 summarizes the contribution of this dissertation, gives concluding remarks and provides suggestions for future work on vibration-based terrain discrimination.

2. Terrain Identification Overview

In the following, the key elements of the terrain discrimination process are presented. Since both terrain classification and clustering techniques are considered herein differences and common building blocks have to identified and discussed. While the primary objective of the following chapter consists of providing an overview of the various stages within the discrimination pipeline, a more detailed description of the applied algorithms can be found in the next chapter.

2.1. Common Building Blocks

Although pursuing their own objectives, terrain classification and clustering feature structural intersections which are introduced in the following.

Data Acquisition To establish a model for terrain classification, training data has to be collected first. In this thesis, a RWI ATRV-Jr outdoor robot (cf. Figure 2.2(a)) was employed as experimental platform. Including the sensors, the robot has a dimension of $77.5 \times 62 \times 95\text{cm}^3$ referring to length, width, and height. The total weight of the robot is about 60 kg where 10 kg (17%) are assigned to the sensors. Equipped with a skid-steering drive, the velocities of the left and right wheels can be adjusted independently. Hence, the robot is able to rotate around its height axis without a translational kinetic component. Big pneumatic tires render a fast robot traversal possible even in rough outdoor terrains. The robot's maximal translational velocity is bounded by 1 m/s.
The body of the robot is split into two parts. In the lower part, the engines and batteries are located, whereas the upper part contains a 2 GHz Pentium M PC with 1 GB RAM. The latter allows for a real-time logging and processing of various sensor information. Besides odometry, vision, ultrasonic, laser range, and differential GPS sensors, an attitude and heading reference system (AHRS), the Xsens MTi[1] (Figure 2.2(b)[2]), is mounted on top of the robot. This device contains accelerometers, gyroscopes, and magnetometers providing three-dimensional sensor data. While the first sensor supports accelerations up to 5 g, the second sensor measures angular velocities at a resolution of ± 300 deg/s. Finally, the magnetometers yield 3D earth-magnetic field data at a scale of ± 750 mGauss.

Data Subdivision During robot traversal, vibration signals were acquired using accelerometers at a sampling rate of 100 Hz. This data stream was split into overlapping segments consisting of 128 samples where 28 samples comprised the overlap. In this way, each subset corresponds to 1.28 s of robot travel and enables a prediction frequency of 1 Hz.

Feature Extraction Feature extraction can be regarded as a decorrelation method. Thereby, a signal is transformed into another representation which is believed to have more beneficial properties compared to the original one. These methods include data transformation techniques such as the Fast Fourier Transform (FFT) which decomposes a signal into

[1] Xsens Technologies B.V., http://www.xsens.com/
[2] Source: cn.gcimg.net

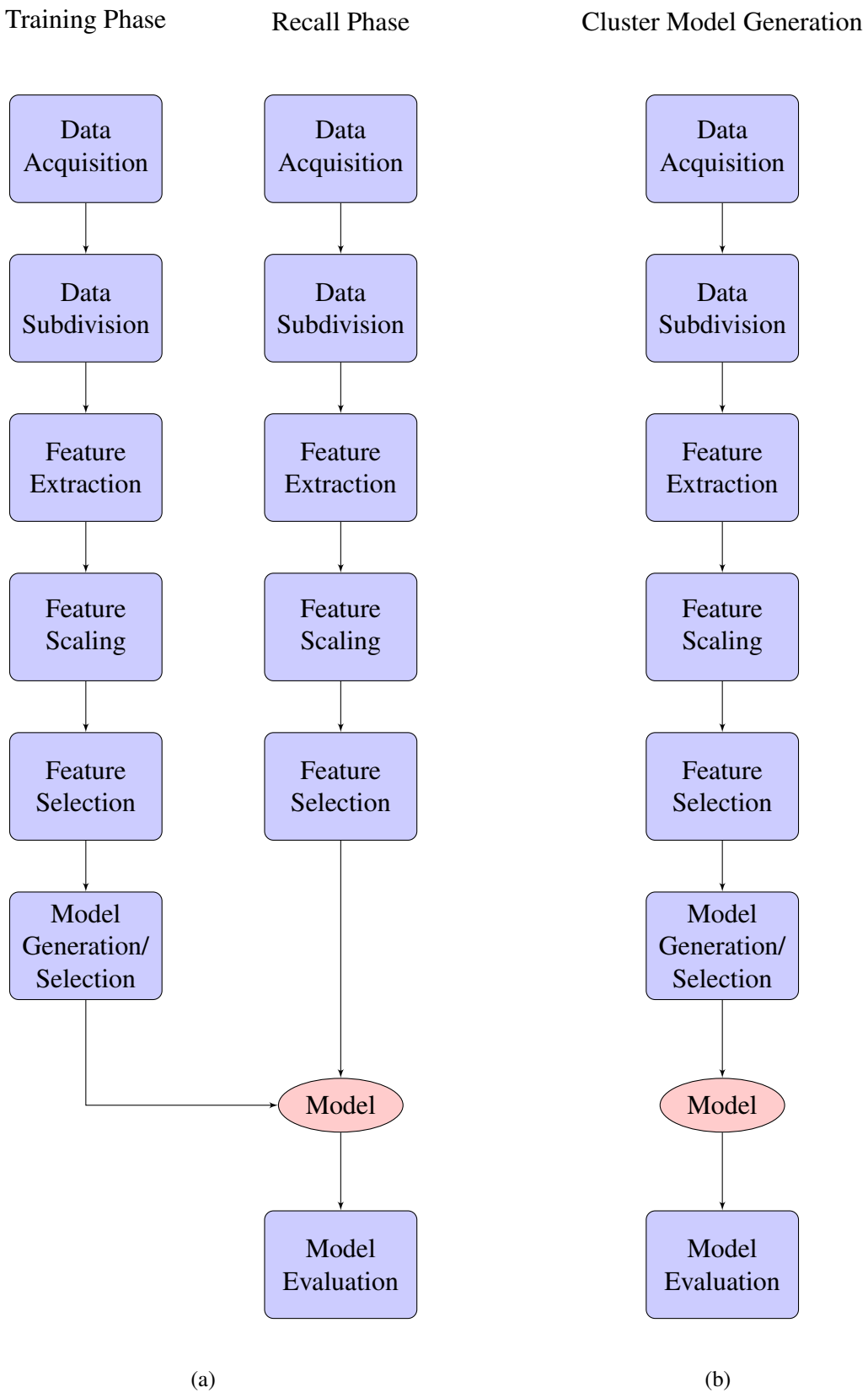
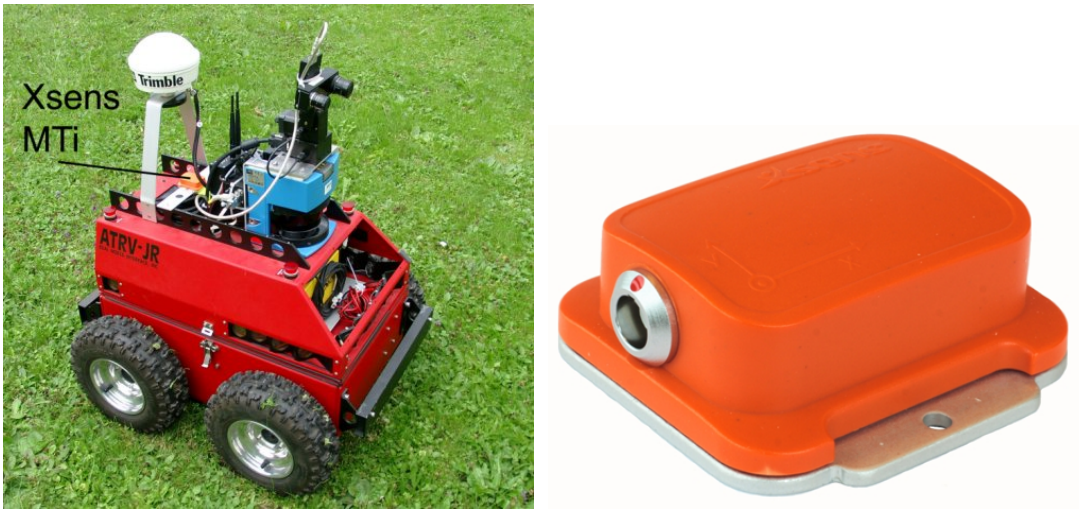


Figure 2.1.: A flow chart containing the building blocks of (a) terrain classification and (b) terrain clustering.



(a) (b)

Figure 2.2.: (a) The experimental platform used in this thesis and (b) the Xsens MTi inertial measurement unit (image source: www.xsens.com).

its constituent frequencies and dimensionality reduction methods. The latter map high-dimensional observations into a lower-dimensional subspace by generating a new feature set from a combination of the most important ones. The main drawback of this approach is that the original features lose their physical meaning and are thus hard to interpret. Two well-known techniques for dimension reduction are the principal component analysis (PCA) [Pea01] and the independent component analysis (ICA) [Com94].

Feature Selection Feature selection refers to the task of searching for a small set of features which (completely) explain the properties of the data. It differs from feature extraction in that the latter constructs new features by projecting the original feature set into a lower dimensional space. In contrast, feature selection techniques are based on the unmodified feature set and thus maintain the interpretability of the results. There are several potential benefits of using a reduced feature set such as the facilitation of data visualization and data understanding, the reduction of training and testing time, and the decrease in the influence of the curse of dimensionality. Moreover, noisy features can degrade the performance of a learning task and should thus be excluded during the model generation process.

Feature Normalization Data normalization is the most common form of preprocessing which ensures that all of the input variables are of order unity. This becomes the more important the larger the difference in their range is, because the magnitude of varying variables may not reflect their relative importance in determining the required outputs.

2.1.1. Classification of Vibration Data

Model Generation In the model generation phase, the model learns the correct assignment of classes j, $j \in [1,k]$ for each observation $x_i \in \mathbb{R}^d, i \in [1,n]$ of the training set $X \equiv \{x_i\}_{i=1}^n$. Hence, a trained model can be considered as a function f which maps a d dimensional input vector into the set of available classes, $f : x_i \in \mathbb{R}^d \rightarrow \mathbb{N}$. Note that all the classifiers presented in this work do not only provide a hard assignment, i.e. the class label, but also

posterior probabilities. These probabilities, $p(c = j|x_i)$ denote the probability of data instance x_i to belong to class j.

Model Selection Model selection describes the process of determining the free parameters for the selected classification technique. For example, when adopting a support vector machine with a radial basis function kernel this includes the specification of the kernel parameter σ, the dimensionality d of the input space, and the cost value C. These free parameters are usually determined using a grid search in conjunction with a n-fold cross validation.

Model Evaluation For model evaluation, unknown data has to be a applied to the trained model. In the context of terrain classification, the term "unknown" refers to novel observations which were collected from known ground surfaces rather than terrain classes which the robot did not traverse. As quality measure, the average true positive rate (TPR) has been adopted. The TPR is defined as the correct number of class assignments related to a certain terrain type averaged over the complete set of classes:

$$\mathrm{TPR} = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{n_j} \sum_{y_i \in \text{class } j} \mathbb{I}(y_i, a_i), \tag{2.1}$$

where n_j is the number of instances contained in class j and $\mathbb{I}$ is the indicator function returning the value of 1 if the true class of test instance i, y_i, equals the predicted class a_i and 0 otherwise. Equation (2.1) is superior to simply counting the number of correct predictions for all test instances in terms of increasing the TPR's informational value when the class frequencies are distributed non-uniformly. For instance, if the test set consists of 99 objects from class 1 but only a single object from class 2, a false prediction of the latter stays unnoticed if the remaining objects are correctly estimated. Using (2.1) for the definition of the TPR, however, class 1 and 2 are assigned true positive rates of 100% and 0%, respectively, yielding an average TPR of 50%.

2.1.2. Clustering of Vibration Data

Model Generation The objective of the clustering task is to divide a given set of objects or measurements into subgroups or clusters based on a similarity criterion. Thereby, the data instances have to be arranged in such a way that objects assigned to the same cluster should be similar whereas objects belonging to different clusters differ significantly. More formally, the clustering process can be defined as follows [JMF99]: given a set of data instances $X \equiv \{x_i\} \in \mathbb{R}^d, i \in [1,n]$, the objective of the clustering process is to partition X into k clusters such that for two data instances, $x_1 \in$ cluster u, $x_2 \in$ cluster $v, u, v \in [1,k]$, $\mathrm{dist}(x_1,x_2)$ is small if $u = v$ and large otherwise. Here, *dist* denotes an appropriate distance function. As a result, a clustering algorithm yields an injective mapping of data instance x_i to cluster j.

Model Selection Analogous to the model selection scheme for classification, model selection in relation to clustering involves the determination of free cluster model parameters. Note, however, that due to the absence of class labels a grid search approach along with cross validation is not possible in this context. Instead, the unknown parameters have to be inferred using statistical measures or Bayesian techniques.

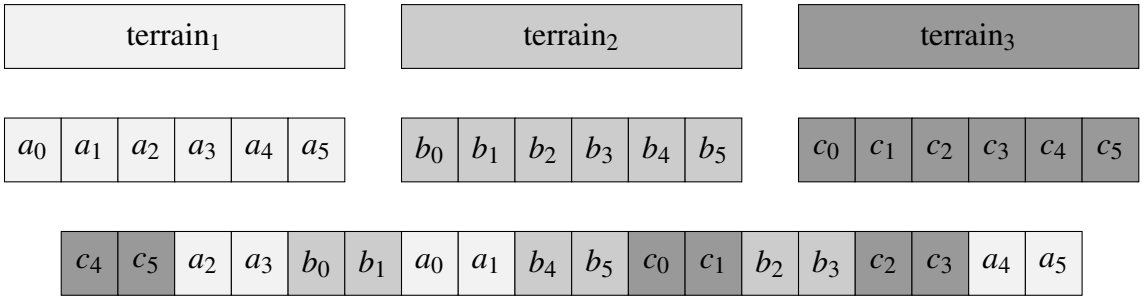


Figure 2.3.: The visualization of the artificial path generation process. Given the acceleration samples of a certain terrain class, these samples are first subdivided into smaller patches. Then, patches from varying ground surfaces are systematically concatenated to form the final path.

Model Evaluation In general, model evaluation for clustering is a harder problem in comparison with classification tasks, which is mainly a reason of the missing label problem. Further problems arise due to the potentially differing number of classes comprised in the reference and estimated clusterings. This renders the direct application of quality measures derived in the context of classification problems impractical. Note, however, that in this case, a correct terrain labeling along with the number of clusters k is available which enable the use of external cluster quality criteria. That is, based on the given labeling of data instances, a clustering with a pre-specified structure can be established representing the intuition about the clustering structure of the data set. This structure is then employed to evaluate the results of a certain clustering technique.
To demonstrate the effectiveness of the presented clustering approach, the evaluation is not only based on a single but on several quality measures, including pair counting of elements, information theory, and the above mentioned classification performance.

2.2. Experimental Data Sets

The paths employed for assessing the terrain identification quality emerged from both a contiguous robot traversal and an artificial composition of consecutive vibration segments. All measurements were collected within the Sand area next to the Department of Computer Science of the University of Tübingen after rainfall. Hence, the robot was faced with slippery grass and soil ground surfaces. For both data sets, the data 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 traversal speed and each data subset can be identified separately.

2.2.1. Natural Paths Including Three Terrain Classes

The paths generated from a (natural) contiguous robot traversal contain 3 terrain classes (cf. Figure 2.4(a)): paving, asphalt, and grass. Due to the contiguous recording technique, the data sets also include terrain transitions, i.e. vibration segments comprising samples from two distinct ground surfaces. Further, the paths are characterized by a small frequency of terrain transitions and a non-uniform distribution of terrain classes at a ratio of 0.56:0.27:0.17 with regard to asphalt, paving, and grass.

(a)

(b)

Figure 2.4.: Visualization of the terrain classes employed in the (a) 3 classes and (b) 5 classes experiments: (a) asphalt, paving, and grass and (b) indoor floor, asphalt gravel, grass, and clay, respectively.

2.2.2. Artificially Generated Paths Including Five Terrain Classes

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. This effect was minimized by establishing a second test environment in a systematic manner. Here, artificial paths have been preferred over natural ones as the former allow for a more detailed investigation of the effects of temporal coherences. Note that with a growing distance the robot navigates on the same terrain (which will be denoted as travel distance in the following), the amount of temporal dependencies is assumed to increase as well. To verify whether the algorithms are able to exploit the increased temporal coherences, paths have to be considered which contain a systematic variation of the travel distance. Since such paths which provide these characteristics are hard to find or simply do not exist, artificially generated paths had to be used. One disadvantage which arises from this approach is the absence of terrain transitions. It is important to stress, however, that the frequency of terrain transitions is much less as opposed to the frequency of non-transitions. Hence, it is assumed that the impact of these transitions on the performance of the terrain discrimination process is rather small. Furthermore, since the vibration segments which represent terrain transitions tend to be outliers with respect to the complete data set, they can be easily filtered out using standard outlier detection techniques.

In total, the artificially generated paths include five terrain classes shown in Figure 2.4(b): indoor floor, asphalt gravel, grass, and clay. The path generation process can be described as follows: After data acquisition and feature extraction, k vibration segment sets are obtained, one for each terrain type. From each of these sets, a homogeneous terrain patch consisting of δ consecutive vibration segments is drawn without replacement to yield a certain travel distance $dist$. Since the robot speed is varied among different experiments between $v_1 = 0.2\,\text{m/s}$ up to $v_3 = 0.6\,\text{m/s}$, δ is determined by $\delta = \lceil dist/v_i \rceil$. Then, homogeneous terrain patches of varying terrain types were grouped together yielding the final test set. In Figure 2.3, an example path generation process with a homogeneous terrain patch size of two vibration segments is depicted.

In total, two artificial path generation techniques were carried out to investigate the performance of the terrain discrimination approach. In the first setting, test paths were generated 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, the travel distance *dist* has been systematically altered for varying test paths: Here, *dist* was chosen from the set $\{2,4,8,16,32\}$ m.
In a further setting, a generated path is allowed to contain varying travel distances. Therefore, k random patches, one from each set, are iteratively drawn without replacement and then assembled in random order. Finally, the patch assembly process was repeated with an increasing travel distance. Starting from $dist = 2m$, the value of *dist* was increased by a factor of 2 in each step and reset to 2 m if a travel distance of 32 m was exceeded. The resulting test paths represent a more realistic terrain setting with both high-frequency and low-frequency terrain transitions.

3. Applied Techniques

This chapter introduces the technical details of data preprocessing, model generation, and model evaluation. Here, the general structure conforms to the one of the previous chapter rendering the discrimination between the classification and clustering tasks necessary. On the other hand, common processes of both framework are mutually discussed. Note that the main focus of this survey is on the presentation of techniques which are frequently used in the following chapters. Methods which are subject to a modification to fit into the terrain discrimination task are introduced later on.

3.1. Common Building Blocks

3.1.1. Feature Extraction

Fast Fourier Transform

Fourier analysis allows for an alternative description of a time-discrete signal. Instead of representing a signal as a function of time, the Fourier technique provides information about the energy content contained at different frequencies. It therefore decomposes a signal into a sum of sines and cosines of different frequencies. Given a periodic signal $f(x)$ of period 2π, f can be represented by the following infinite series:

$$f(x) = c_0 + \sum_{n=1}^{\infty} (a_n \cos(n \cdot x) + b_n \sin(n \cdot x)), \tag{3.1}$$

where c_0, a_n, and b_n are the Fourier coefficients defined by the integrals:

$$\begin{aligned} c_0 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx, \\ a_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(n \cdot x) dx, \\ b_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(n \cdot x) dx. \end{aligned}$$

Using complex notations and Euler's formula, the Fourier series can be expressed in an algebraically simpler form. Here, one can make use of the fact that the complex exponential $\exp(i\theta)$ satisfies $\exp(i\theta) = \cos(\theta) + i\sin(\theta)$. Thus, we have:

$$\cos(\theta) = \frac{1}{2}(\exp(i\theta) + \exp(-i\theta)), \tag{3.2}$$

$$\sin(\theta) = \frac{1}{2i}(\exp(i\theta) - \exp(-i\theta)). \tag{3.3}$$

$$\tag{3.4}$$

From (3.2) and (3.3) it can be shown that (3.1) can be reformulated as

$$c_0 + \sum_{n=1}^{\infty} (c_n \exp(inx) + c_{-n} \exp(inx)) = \sum_{n=-\infty}^{\infty} c_n \exp(inx),$$

where c_n is defined for all integers n by

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \exp(-inx) dx. \tag{3.5}$$

Note that the definition of the Fourier transform given above requires the underlying function $f(x)$ to be given analytically. In the context of terrain discrimination, however, the function f is represented in terms of discrete acceleration samples acquired equidistantly. This renders a reformulation of the Fourier series for time-discrete signals necessary. The respective transformation is known as the discrete Fourier transform (DFT) which maps a finite sequence of M numbers $\{f(x_k)\}_{k=0}^{M-1} \equiv \{g_k\}_{k=0}^{M-1}$ into a set of DFT coefficients $\{G_n\}$. The derivation of the respective formulas makes use of the fact that in the discrete case the integrals employed for the calculation of the Fourier series coefficients $\{c_n\}$ in (3.5) can be discretely approximated by Riemann sums.
For a positive integer M, we define $x_k = -\pi + 2\pi k/M$, for $k = \{0, 1, \ldots, M-1\}$ and let $\Delta x = 2\pi/M$. Then, the nth Fourier coefficient c_n of a function f is approximated by:

$$\begin{aligned} c_n &\approx \frac{1}{2\pi} \sum_{k=0}^{M-1} f(x_k) \exp(-i2\pi n x_k) \Delta x \\ &= \frac{\exp(-in\pi)}{M} \sum_{k=0}^{M-1} f(x_k) \exp(-i2\pi kn/M). \end{aligned} \tag{3.6}$$

Replacing $f(x_k)$ in (3.6) by g_k yields the approximation of f in terms of DFT coefficients $\{G_n\}$:

$$G_n = \sum_{k=0}^{M-1} g_k \exp(-i2\pi kn/M). \tag{3.7}$$

Note, however, that instead of applying (3.7) for determining the Fourier coefficients c_0, a_n and b_n, the Fast Fourier Transform (FFT) is used which transforms the signal $\{g_k\}_{k=0}^{M-1}$ in $O(M \log_2 M)$ operations.
Although the obtained DFT coefficients provide an effective representation of the underlying signal, a common approach is its description in terms of the power spectrum of its positive frequencies. Given a set of M coefficients $\{a_i, b_i\}_{i=0}^{M-1}$, the power spectrum is determined by:

$$c_i = \sqrt{a_i^2 + b_i^2}, i \in [0, M/2 - 1].$$

In the following, this feature representation is referred to as the DFT amplitude spectrum (DAS) descriptor.

Principal Component Analysis

Principle component analysis aims at finding m principle directions in the d-dimensional data set which contain the largest variance. These directions are defined using linear projectors $T \in \mathbb{R}^{d \times m}$, i.e., the matrix T projects the data onto the subspace spanned by the principal

directions. The determination of the projection matrix T can be mathematically formulated as a minimization of the following residuum-functional:

$$L(x_i, T, \mu) = \sum_{t=1}^{T} \|(x_i - \mu) - T \cdot T^T (x_i - \mu)\|_2^2, \tag{3.8}$$

where $\mu \in \mathbb{R}^d$ is the center vector defined as $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$. The residuum-functional describes the least-squares difference between the unmodified observation and its m-dimensional projection.
Equation (3.8) is equivalent to:

$$L = \sum_{i=1}^{T} (x_i - \mu)^T (I - T \cdot T^T)(x_i - \mu). \tag{3.9}$$

Since the projectors T are subject to the orthogonality condition $T^T \cdot T = I^{m \times m}$, (3.9) can be minimized analytically to find the optimal parameters for μ and T:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i,\ C = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T = T \cdot S \cdot T + O(\Lambda_{\min}).$$

Here, T is the matrix of m dominant eigenvectors and $S = diag(\lambda_1, \ldots, \lambda_m)$ denotes the m corresponding largest eigenvalues of the covariance matrix C. The diagonal matrix $\Lambda_{\min} = diag(\lambda_{m+1}, \ldots, \lambda_n)$ corresponds to the remaining spectrum. The above results show that the optimal value for μ is determined by the expected value of the data set. Furthermore, the optimal projection matrix T is given by the dominant eigenvectors of the data covariance matrix.

3.1.2. Data Normalization

In the following, a d-dimensional data set $X = \{x_i\}_{i=1}^{n}$, $x_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathbb{R}^d$ containing n instances is assumed. Further, $x_{i,j}$ denotes the jth feature (variable, component) of the ith instance and $x_{*,j}$ is the set of all n feature values with respect to a certain component j. That is, the latter set is represented by the jth column of the data matrix X.
Given these definitions, data normalization can be applied by transforming a specific variable j such that the jth column of the data matrix, $x_{*,j}$, has zero mean and a standard deviation of 1. This is achieved by determining the mean $\bar{x}_j$ and standard deviation σ_j for each variable $x_{*,j}$ followed by subtracting the mean $\bar{x}_j$ from each instance of variable $x_{*,j}$ and finally dividing each instance by the standard deviation σ_j.

$$\begin{aligned}
\bar{x}_j &= \frac{1}{n}\sum_{i=1}^{n} x_{i,j} \\
\sigma_j^2 &= \frac{1}{n-1}\sum_{i=1}^{n} (x_{i,j} - \bar{x}_j)^2 \\
\tilde{x}_{i,j} &= \frac{x_{i,j} - \bar{x}_j}{\sigma_j}
\end{aligned}$$

3.2. Classification of Vibration Data

3.2.1. Model Generation

Support Vector Machines

Based on the work of Burges [Bur98], this section presents a mathematical formulation of the hard-margin support vector machine (SVM) for binary, linear separable problems. That is, problems involving training data which can be correctly classified by the following linear function:

$$F(x_i) = w \cdot x_i - b, \tag{3.10}$$

where x_i is a d-dimensional real vector given from the data set $X = \{x_i\}_{i=1}^{n}$, y_i denotes the class of x_i and is assigned either -1 or 1. Further, w is the weight vector and b is the bias. Both values are determined during the training process of the SVM and are chosen such that the sign of $F(x_i)$ is either positive or negative depending on the class membership of x_i:

$$\begin{aligned} w \cdot x_i - b \geq 0 \qquad & \text{if } y_i = 1, \text{ and} \\ w \cdot x_i - b < 0 \qquad & \text{if } y_i = -1, \end{aligned}$$

which can be reformulated as:

$$y_i(w \cdot x_i - b) > 0, \forall (x_i, y_i) \in X. \tag{3.11}$$

The linear function F defines a hyperplane which separates the data into two subspaces. Yet, there is an infinite number of hyperplanes which satisfy (3.11). Hence, an additional constraint has to be defined which yields a unique solution. In the context of a SVM, a maximum-margin criterion is used. The margin denotes the distance between the hyperplane and its closest points. This constraint results in the following modification of (3.11):

$$y_i(w \cdot x_i - b) \geq 1, \forall (x_i, y_i) \in X. \tag{3.12}$$

Given that the data set is linearly separable, or equivalently, each instance satisfies (3.11), then the existence of (3.12) is also guaranteed. This is achieved by a rescaling of the weight vector w and bias b.
The distance of an instance x_i to the hyperplane is defined as $\frac{|F(x)|}{\|w\|}$. Hence, the margin becomes $\|w\|^{-1}$ due to the constraint $|F(\{x_c\})| = 1$, where $\{x_c\}$ denotes the set of the closest vectors to the hyperplane. These vectors are denoted as the support vectors.
Maximizing the margin $\|w\|^{-1}$ is equivalent to the minimization of $\|w\|$ resulting in the following constrained optimization problem:

$$\begin{aligned} \text{minimize } Q(w) &= \frac{1}{2}\|w\|^2 \\ \text{subject to } y_i(w \cdot x_i - b) &\geq 1, \forall (x_i, y_i) \in X \end{aligned} \tag{3.13}$$

The optimization problem can be solved using Lagrange multipliers [Min86] yielding a Lagrange function of the form:

$$J(w, b, \alpha) = \frac{1}{2} w \cdot w - \sum_{i=1}^{n} \alpha_i \{y_i(w \cdot x_i - b) - 1\}, \tag{3.14}$$

where the variables $\alpha \geq 0$ denote the Lagrange multipliers.

The solution of the constrained optimization problem is given by:

$$\begin{aligned} w^* &= \sum_i \alpha_i^* y_i x_i \text{ and} \\ b^* &= 1 - w^* \cdot x_i. \end{aligned} \tag{3.15}$$

Here, the set $\{\alpha_i^*\}$ denotes the optimal Lagrange multipliers determined after optimizing (3.14). Finally, the decision function of (3.10) becomes:

$$F(x) = \sum_i \alpha_i y_i x_i \cdot x - b \tag{3.16}$$

The optimization problem of the hard-margin SVM stated in (3.13) only has a solution if the data set X is linearly separable. For the linearly inseparable case, another SVM approach is considered, which is based on soft margins. Here, mislabeled data points are allowed while the objective of maximizing the margin is kept. Therefore, slack variables ξ are introduced which measure the degree of misclassification. The optimization problem for the soft margin SVM then becomes:

$$\begin{aligned} \text{minimize } Q(w,b,\xi_i) &= \frac{1}{2}\|w\|^2 + C \cdot \sum_i \xi_i \\ \text{subject to } y_i(w \cdot x_i - b) &\geq 1 - \xi_i, \forall (x_i, y_i) \in X \text{ and} \\ \xi_i &\geq 0 \end{aligned} \tag{3.17}$$

By including the slack variables ξ_i in (3.17), the misclassification of data instances is rendered possible, yet, the degree of misclassification is minimized along with the maximization of the margin. C denotes a user-defined parameter which is a trade-off between the margin size and the allowed classification error. Usually, C is obtained using a cross-validation [Koh95] approach.

A data set which is not linearly separable in its original space might be linearly separable when the input vectors are mapped into a high-dimensional feature space. In a second step, a SVM can be trained to find the hyperplane providing the maximum margin in the new feature space. Here, the separating hyperplane is a linear function in the transformed space but a non-linear one in the input space.
Given a non-linear mapping function from the d-dimensional input space to a higher dimensional feature space, $\varphi(x)$, the decision boundary in feature space is defined as:

$$w \cdot \varphi(x_i) - b = 0.$$

Using (3.15), the weight is determined by:

$$w = \sum \alpha_i y_i \varphi(x_i),$$

and hence, the decision function of (3.16) becomes:

$$F(x_i) = \sum_i \alpha_i y_i \varphi(x_i) \cdot \varphi(x) - b.$$

Note that in the latter equation, the mapping function occurs as the dot product $\varphi(x_i) \cdot \varphi(x)$. Instead of calculating this dot product directly, the Kernel trick is applied which replaces the dot product in feature space with a kernel function K in the original input space:

$$K(u,v) = \varphi(u) \cdot \varphi(v).$$

Since K can be determined in the input space, the time-consuming task of feature transformation can be skipped. As for the kernel function, the radial basis function $K(a,b) = \exp(-\gamma \cdot \|a-b\|^2)$ is used in this work.

The Estimation of Multi-Class Posterior Probabilities

In its original formulation, the SVM only yields binary classifications, i.e. decisions whether a class belongs to one class or to the other one. In many situations, however, not only the information about the class assignment is important but also the certainty of the decision. From a statistical viewpoint, the certainty of a decision can be represented by means of the posterior probability $p(c = 1|x_i)$ which denotes the probability of a certain measurement x_i to belong to class 1.
Platt [Pla00] proposed a sigmoid-fitting approach to transform (unthresholded) decision values into posterior probability estimates. Therefore, he used the following parametric form for the sigmoid:

$$p(c = 1|x_i) = (1 + \exp(Af_i + B))^{-1},$$

where f_i denotes the unthresholded output of the SVM model, and A and B are the parameters to be determined. To estimate the latter, Platt applies a maximum likelihood approach on the training set $\{f_i, t_i\}$, where t_i is defined as $t_i = (y_i + 1)/2$. In the maximum likelihood approach, the negative log likelihood of the training data is minimized with respect to the following cross-entropy error function:

$$-\sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i), \tag{3.18}$$

where

$$p_i = (1 + \exp(Af_i + B))^{-1}.$$

The minimization problem of (3.18) can be efficiently solved using a model-trust algorithm based on the Levenberg-Marquardt algorithm [Yua00].

This work focuses on the classification of multiple terrain types, rendering a multi-class classifier necessary. A general approach is to divide the multi-class problem into a number of binary classification tasks. In the case of a one-vs-one multi-class classifier system, $\frac{k \cdot (k-1)}{2}$ binary classifiers are required, one for each possible pair of classes. Then, a coupling approach can be employed which combines the posterior probability estimates to derive the posteriors p_i of each individual class i, $i \in [1,k]$.
More formally, let r_{kj} be the estimated posterior $\mu_{kj} \equiv p_k$, that is, the posterior probability of class k given a binary classifier. Here, the latter is only trained of data assigned to both class k and class j, respectively. Obviously, we have $\mu_{kj} = p_k/(p_k + p_j)$, since $p_k + p_j = 1$. In the approach of Wu et al. [WLW04], values for p_i are determined such that the resulting μ_{ij} are close to the binary estimates r_{ij}. Therefore, they establish the following quotients and identify them with the binary posterior probability estimates:

$$\frac{\mu_{ij}}{\mu_{ji}} = \frac{\frac{p_i}{p_i+p_j}}{\frac{p_j}{p_i+p_j}} = \frac{p_i}{p_j} \overset{!}{\approx} \frac{r_{ij}}{r_{ji}}.$$

From this, one can conclude that the smaller is the outcome of $r_{ji}p_i - r_{ij}p_j$, the smaller is the difference between μ_{ij} and r_{ij}. In their work, Wu et al. use a slightly modified term for the objective function which is defined as:

$$\min_p \sum_{i=1}^{k} \sum_{j,j\neq i} (r_{ji}p_i - r_{ij}p_j)^2 \text{ subject to } \sum_{i=1}^{k} p_i = 1.$$

It can be shown [WLW04] that the solution of this optimization formulation is obtained by solving a simple linear system.

3.3. Clustering of Vibration Data

3.3.1. Model Generation

K-means Clustering

The k-means algorithm is an iterative algorithm which minimizes the distance between each observation and its corresponding cluster center (centroid). Here, the distance can be defined in various ways, e.g. by means of the Euclidean distance, city-block, or hamming distances. Among these distance measures, the Euclidean distance is commonly used for k-means clustering. A Euclidean k-means algorithm minimizes the sum-squared-error (SSE) criterion defined as:

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in \text{cluster}_j} \|x_i - \mu_j\|^2,$$

where n_j denotes the number of instances contained in cluster j and μ_j is their respective mean, $\mu_j = \frac{1}{n_j} \sum_{x_i \in \text{cluster}_j} x_i$.
The k-means algorithm is initialized with k centroids which are chosen either deterministically or randomly. Then, the SSE criterion is minimized using the following two steps: First, each observation x_i is assigned to the nearest centroid j: $j = \arg\min_k \|x_i - \mu_k\|$. Second, the new locations of the cluster centroids are determined given the assignment of the first step. Both steps are repeated until convergence of the centroid locations.

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 $X \equiv \{x_i\}_{i=1}^n$. The probabilistic model is expressed as a linear combination of k basis functions $p(x_i) = \sum_{j=1}^k \pi_j p(x_i|c=j)$, where k denotes the number of components of the model, π_j is the mixing coefficient of component j, and $p(x_i|c=j)$ is the component likelihood. The latter defines the probability of a data point x_i to belong to a certain mixture component j. For Gaussian mixture models, the basis functions are given by Gaussian distribution functions with parameters $\{\mu_j, \Sigma_j\}$:

$$p(x_i|c=j) = \frac{1}{(2\pi)^{d/2} \left|\Sigma_j\right|^{1/2}} \cdot e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1} (x_i-\mu_j)}. \tag{3.19}$$

Here, μ_j denotes a d-dimensional mean vector and Σ_j is the positive definite $d \times d$ covariance matrix. The corresponding generative model is shown in Figure 3.1 where two neighboring

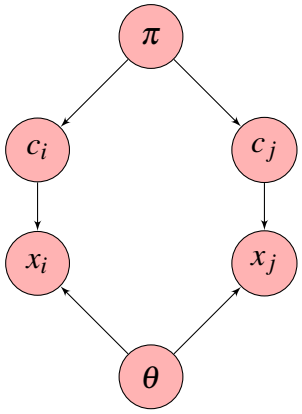


Figure 3.1.: The probabilistic graphical model of a standard Gaussian mixture model.

vibration segments i and j are presented. The model assumes a common prior distribution π which independently generates all vibration segment labels x_i.
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:

$$L_1^{(t)} = \sum_{i=1}^{n} \log p(x_i|\vec{\theta}^{(t)}), \tag{3.20}$$

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:

$$p(c = j|x_i) = \frac{p(x_i|c = j)\pi_j}{\sum_{l=1}^{k} p(x_i|c = l)\pi_l} \tag{3.21}$$

2. M-step:

$$\widehat{\mu}_j = \frac{1}{n\pi_j} \sum_{i=1}^{n} p(c = j|x_i)x_i \tag{3.22}$$

$$\widehat{\Sigma}_j = \frac{1}{n\pi_j} \sum_{i=1}^{n} p(c = j|x_i)x_i x_i^T - \widehat{\mu}_j \widehat{\mu}_j^T \tag{3.23}$$

$$\widehat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} p(c = j|x_i) \tag{3.24}$$

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 $1 - \frac{L_1^{(t-1)}}{L_1^{(t)}} < \varepsilon$. In this work, ε was chosen as $\varepsilon = 0.001$.
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, a certain class or cluster i is assigned to the mixture component i. Finally, the class (or cluster) belonging to a given data point x_l is chosen by selecting

Table 3.1.: The structure of a contingency table. Each entry n_{ij} denotes the number of observations which belong to both cluster U_i and cluster V_j. Further, the sets $\{a_i\}_{i=1}^{r}$ and $\{b_j\}_{j=1}^{c}$ are the row and column sums, respectively.

U/V	V_1	V_2	$\dots$	V_c	sum
U_1	n_{11}	n_{12}	$\cdots$	n_{1c}	a_1
U_2	n_{21}	n_{22}	$\cdots$	n_{2c}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_r	n_{r1}	n_{r2}	$\dots$	n_{rc}	a_r
sum	b_1	b_2	$\cdots$	b_c	$\sum_{ij} n_{ij} = n$

the mixture component j that maximizes the posterior probability $p(c = j|x_l)$. 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 thesis.

3.3.2. Evaluation

For comparing two different clusterings, a similarity measure has to be defined. In this section, four performance measures are presented which are all derived from a contingency table (Section 3.3.2), yet are based on different ideas: counting pairs of elements, information theory, and classification performance.

Contingency Table

The contingency table (Table 3.1) is a data structure which stores the information about the cluster overlap between two different clusters: given two clusterings U and V with r and c classes, the entries n_{ij} of the contingency table $C \in \mathbb{R}^{r\times c}$ denote the number of observations which belong to both cluster U_i and cluster V_j. Note that this approach assumes a hard clustering of observations, i.e., one observation is assigned to only a single cluster.

Using this contingency table, several cluster similarity indices can be defined.

The Rand Index and its Extensions The idea of the rand index is to count pairs of instances on which two clusterings agree or disagree. In total, there are $n \cdot (n-1)/2$ possible pairs each of which can be assigned to one of the following categories:

N_{11}: The number of pairs that are in the same cluster in both U and V.
N_{00}: The number of pairs that are in different clusters both in U and V.
N_{01}: The number of pairs that are in the same cluster in U but in different clusters in V.
N_{10}: The number of pairs that are in different clusters in U but in the same cluster in V.

The rand index only makes use of the former two cases which intuitively can be regarded as indicators of agreement between U and V. It is defined as:

$$RI(U,V) = \left(N_{00} + N_{11} / \left(\begin{array}{c} N \\ 2 \end{array} \right) \right).$$

Note that the rand index is bounded by 0 and 1. It is assigned a value of 1 if the two clusterings are identical, and 0 if no pair of instances appear either in the same cluster or in different clusters in both clusterings.
The issues related to the rand index are that its outcome for two random partitions is not constant (e.g. 0) and converges to one as the number of clusters increases. Hubert et al. [HA85] addressed these problems by modeling the partition generation process using a generalized hypergeometric distribution. Under this probability model, the expected value of the rand index can be determined when drawing two random partitions which contain the original number of clusters and instances in each. Hubert et al. used this expected value to define a modified rand index corrected for chance:

$$\begin{aligned} \text{ARI} &= \frac{\text{RI} - \mathbb{E}(\text{RI})}{\max(\text{ RI }) - \mathbb{E}(\text{RI})} \\ &= \frac{\Sigma_{ij}\binom{n_{ij}}{2} - \left[\Sigma_i \binom{a_i}{2} \Sigma_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\Sigma_i \binom{a_i}{2} + \Sigma_j \binom{b_j}{2}\right] - \left[\Sigma_i \binom{a_i}{2} \Sigma_j \binom{b_j}{2}\right] / \binom{N}{2}}, \end{aligned}$$

where each n_{ij}, a_i and b_i are derived from the contingency table. The ARI index is upper bounded by 1 and is assigned the value of 0 if the index equals its expected value.

Information Theoretic Measures

Adjusted Mutual Information The mutual information (MI) is a non-parametric measure of relevance which can be derived from information theory. The MI of two random variables X and Y is a measure of how X and Y depend on each other. It can be defined from the entropy $H(.)$:

$$\text{MI}(X,Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X),$$

where $H(Y|X)$ is the conditional entropy of Y given X. It measures the loss of uncertainty of Y when X is known. If X and Y are independent, then $H(X,Y) = H(X) + H(Y)$, $H(Y|X) = H(Y)$ and hence $\text{MI}(X,Y) = 0$. For a continuous random variable X, the mutual information corresponds to the Kullback-Leibler distance between the joint distribution and the product of the marginals:

$$\begin{aligned} \text{MI}(X,Y) &= \text{KL}(p(X,Y)||p(X)p(Y)) \\ &= \int\int p(X,Y) \ln\left(\frac{p(X)p(Y)}{p(X,Y)}\right) dXdY \end{aligned}$$

In the context of clustering, mutual information can be used to determine the amount of shared information between two clusterings. To render the mutual information index applicable, the notions of entropy and joint probability have to be defined in the clustering domain. Given a clustering U, the probability of a randomly chosen instance of the data set to belong to cluster U_i is defined as $p(i) = \frac{|U_i|}{n}$. Hence, the entropy of a clustering U can be calculated as:

$$H(U) = -\sum_{i=1}^{r} p(i) \log p(i).$$

Similarly, the entropy of another clustering V is determined by $H(V) = -\sum_{j=1}^{c} \tilde{p}(j) \log \tilde{p}(j)$, where $\tilde{p}(j) = \frac{|V_j|}{n}$. Finally, the mutual information between two clusterings U and V is defined as:

$$MI(U,V) = \sum_{i=1}^{r} \sum_{j=1}^{c} p(i,j) \log \frac{p(i,j)}{p(i) \cdot \tilde{p}(j)},$$

where the joint probability $p(i,j)$ denotes the probability that an observation belongs to cluster U_i in U and to cluster V_i in V:

$$p(i,j) = \frac{|U_i \cup V_j|}{n}.$$

The mutual information defines a metric on the space of all clusterings, yet, it is not bounded which renders an interpretation of the outcome difficult. A bounding can be established, however, by using the entropies of the specific clustering. This is because

$$MI(U,V) \leq \min(H(U), H(V)).$$

Strehl et al. [SG03] proposed a normalization scheme in terms of the geometric mean. There, they calculated the normalized mutual information by:

$$NI(U,V) = \frac{MI(U,V)}{\sqrt{H(U)H(V)}}.$$

For the NI index, we have $0 \leq NI(U,V) \leq 1$ with $NI(U,V) = 1$ for $U = V$ and $NI(U,V) = 0$ if for all $i, 1 \leq i \leq k$, and for all $j, 1 \leq j \leq l$, we have $p(i,j) = 0$ or $p_{i,j} = p(i) \cdot p(j)$.
The (normalized) mutual information measure suffers from the same issue as the unadjusted rand index of Section 3.3.2: the outcome when applying the mutual information measure to two random clusterings differs from 0 and is not constant for varying random clusterings. Similar to the approach of Hubert et al. [HA85], Vinh et al. [VEB09] adjust the mutual information index by incorporating the expected value of the mutual information between two random clusterings. As model for randomness, they also chose the generalized hypergeometric distribution in which clusterings are generated randomly with the constraint to have a fixed number of clusters and observations in each cluster.
As shown in [VEB09], the expected mutual information can be calculated as:

$$\begin{aligned} \mathbb{E}\{MI(M)|a,b\} \quad = \quad & \sum_{i=1}^{r} \sum_{j=1}^{c} \sum_{n_{ij}=(a_i+b_j-n)}^{\min(a_i,b_j)} \frac{n_{ij}}{n} \log\left(\frac{n \cdot n_{ij}}{a_i b_j}\right) \hookleftarrow \\ & \frac{a_i! b_j! (n-a_i)! (n-b_j)!}{n! n_{ij}! (a_i - n_{ij})! (b_j - n_{ij})! (n - a_i - b_j + n_{ij})!}. \end{aligned}$$

Given the expected mutual information, Nguyen et al. proposed the following adjusted mutual information index:

$$AMI(U,V) = \frac{MI(U,V) - \mathbb{E}\{MI(M)|a,b\}}{\sqrt{H(U)H(V)} - \mathbb{E}\{MI(M)|a,b\}}.$$

The AMI index is assigned a value of 1 if the two clusterings are identical, and 0 if the mutual information between the two clusterings equals its expected value.

Classification Performance

Given external class information $\{y_i\}_{i=1}^{n}$ for each data instance x_i along with the number of classes k contained in the data set, the true positive rate performance measure of Section 2.1.1 can be applied. Problems arise due to the inherent permutation symmetry of clustering algorithms which can be stated as follows: even if two distinct clusterings yield exactly the same instance grouping the labels may be arbitrarily permuted. To overcome this problem, the distance function of (3.25) is considered determining the summed difference between the real Y and estimated Y^* target label distribution:

$$d(Y,Y^*) = \min_{\pi} \sum_{i=1}^{N} \mathbb{I}(y_i - \pi(y_i^*)), \tag{3.25}$$

where π denotes a permutation of labelings which bijectively assigns each class of Y^* a certain class of Y. The optimal and hence chosen permutation is the one which minimizes the distance $d(Y,Y^*)$. Determining the optimal permutation either requires the complete enumeration of possible permutations or adopting more elaborate techniques such as the Hungarian method for solving the minimum weighted perfect bipartite matching problem [Kuh55, Mun57].

4. Terrain Classification using Temporal Coherences

4.1. Introduction

The objective of the work presented in this chapter is to predict the terrain type the robot is navigation on using vibration data. Therefore, a machine learning model has to be learned which establishes the assignment between current sensor measurements and the ground surface. For the machine learning model, a variety of classifiers can be employed such as support vector machines [WFZ06], linear discriminants [BID05] and probabilistic neural networks [DRM06b]. Incorporating all these classifiers in a comparative survey, Weiss et al. [WFSZ07] reported the support vector machine in conjunctions with a radial basis function kernel to yield the best classification performance. All the presented approaches have in common that the terrain classification is only based on a single observation and hence disregard the temporal coherences contained in succeeding observations. Given that the robot navigates on a certain terrain type for a longer period of time during the robot traversal, it is very likely that the ground surface does not change from one time step to the next. This assumption is based on the observation that the probability of a terrain transition is significantly smaller in comparison with the one that the terrain type remains the same.

4.1.1. Sequential Pattern Classification

The use of sequential patterns to improve the prediction quality of classifiers leads to the domain of sequential supervised learning [Die02]. Given n training examples $\{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{N}^+$, the learning problem is defined as the generation of a valid classifier h which predicts a new label sequence $Y = h(X)$ from the input sequence X. Similar, but yet distinct tasks include time-series prediction and sequence classification. The objective of the former technique is to predict the $t+1^{st}$ elements of a sequence $\{y_1, \ldots, y_t\}$, possibly given further features or covariates $\{x_1, \ldots, x_t\}$. From this definition, two key differences between time-series prediction and sequential supervised learning arise. First, in the latter task, the complete set of observations x_i is available whereas for time-series prediction, only a prefix of this sequence is present. Second, during the recall phase of sequential supervised learning, any of the target labels y_i is given but predicted by the machine learning model instead.
In the following, several techniques are briefly described which address the problem of sequential supervised learning.

4.1.2. Sliding Window Techniques

The first technique is the sliding window method. Here, the current target label y_t is predicted using a temporal window which contains the last k observations $\{x_{t-k+1}, \ldots, x_t\}$. For example, Coyle et al. [CCL10] used a sliding window approach to establish an update rule for switching

control modes of an outdoor robot. Thereby, the decision was based on past and present classifications and an appropriate window size was found using empirical data. Note, however, that this window size selection scheme is likely to fail whenever the observed frequency of terrain transitions during training does not reflect the one during the recall phase. Furthermore, the predictive model of Coyle et al. only allows for the use of two terrain classes which is inappropriate for most environmental settings.
The sliding window approach has the advantage that a window classifier can be trained using any classical supervised learning algorithm. Yet, it does not provide effective means to exploit existing correlations between nearby target labels. These correlations can only be incorporated using nearby observations. Relationships among target labels, however, which are independent from observations cannot be taken into account. Recurrent sliding windows account for these issues as they also include the last k terrain estimates, $\{y_{t-k}, \ldots, y_{t-1}\}$ as inputs for the prediction process. Analogous the non-recurrent window approach, its recurrent extension benefits from a variety of classifiers which can be adopted for the training task without further modifications. Note, however, that the last statement is only valid if the true labels are employed during model generation. A recurrent window approach applied to two-dimensional data has been proposed by Vega [Veg05]. There, a spatially coherent classifier was established being able to differentiate between five distinct types of plants. Training and test data were extracted from Synthetic Aperture Radar and Airborne Thematic Mapper images provided by the Feltwell data set [SR95, GRB00]. Vega reported an increase in prediction performance of 3% up to 10% depending on the employed classifier model.

4.1.3. Hidden Markov Models

The hidden Markov model (HMM) [Rab90] is an example of a generative model as it describes a means how the observations x_i and target labels y_i are generated. Formally, the hidden Markov model is a representation of the joint distribution $p(x,y)$. Two probability distributions are required to define the probabilistic model: first, the transition distribution $p(y_t|y_{t-1})$ representing the probability of moving from state y_{t-1} to state y_t, and second, the observation distribution $p(x|y)$ reflecting the relationship between the observations to the hidden target labels. Note, that both distributions are assumed to be stationary, i.e. constant for all considered time steps. Further model simplifications include the assumed independently and identically distributed nature of the observations and the Markov property between target labels. Whereas the former assumption states that each observation is generated independently conditioned on y, the latter Markov property only models the relationships between consecutive target labels. Both simplifications give indications why hidden Markov models are often a poor model of the process generating the data. Yet, several authors successfully adopted HMMs in the domain of terrain classification. In [WSFB05], Wolf et al. proposed a technique for generating three-dimensional data from two-dimensional laser scans. On these maps, the respective regions were divided into navigable and non-navigable regions using hidden Markov models. After both steps, a Markov random field-based segmentation approach was applied to increase the classification performance. In a later work [WS08], Wolf et al. extended the idea of structural mapping by means of incorporating semantic properties into the map. This enabled the generation of maps which did not represent metric occupancy only but also other features such as navigability or the degree of activity. Thereby, the authors addressed the semantic mapping problem using both hidden Markov models and support vector machines. They concluded that both techniques yield similar results in terms of mapping quality. A terrain classification method derived from gait bounce and gait roll measures was proposed by Larson et al. [LDV05]. Their approach is

based on the assumption that the spatio-temporal patterns of these signals can directly be used to estimate the characteristics of the terrain. In their work, a meta-classifier based on discriminant analysis and hidden Markov models was employed to extract the spatio-temporal patterns of the gait-bounce signal.

4.1.4. Conditional Random Fields

To overcome the limitations of the HMM, the conditional random field (CRF) [LMP01] has been suggested. It differs from a HMM by establishing a model of the conditional probability distribution $p(y|x)$ rather than $p(x,y)$. Hence, informally, CRFs does not try to explain the process generating the observations, but to predict the target labels given the observations. Thereby, the relationship between succeeding target label pairs y_{t-1} and y_t is modeled using a Markov random field conditioned on the observations. In other words, the influence of adjacent target labels is determined by the observations. Since conditional random fields became a popular tool in object recognition [QCD05] or segmentation [JWL+06] tasks, this technique was also applied to the domain of terrain classification. For example, Verbeek et al. [VT07] introduced a CRF-based labeling model taking both local and global features into account. There, global features are defined to represent terrain characteristics gathered over the whole or at least large sections of a given image. The advantage of their CRF model is that it does not require pixelwise-labeled training data, rendering the generation of the training data less time-consuming. Wang et al. [WZTL10] employed CRF models for the generation of long-range terrain perception in outdoor environments. Their approach does not only consider local region features but also spatial dependencies between different regions during classification.

4.1.5. The Proposed Bayes Filter Approach

Despite the many benefits of conditional random fields, their inference is known to be a tedious task rendering the training of CRFs expensive. For this reason, a novel technique is presented in this chapter which does not require any further information but the posterior probability estimates of individual observations. These posteriors are then recursively applied to a Bayesian filter enabling the integration of a history of terrain estimates into the prediction framework. Starting from the original Bayes filter formulation [Jaz70], several modifications had to be considered for enabling its use in the domain of terrain classification [KWZ09b]. As a further contribution, an adaptive extension to the proposed Bayes filtering approach is introduced which renders its application possible in situations of high dynamic range including both high-frequency and low-frequency terrain changes [KWZ09a]. Finally, further improvements could be achieved by embedding the filtering technique into the multi-class sequential decision process of [BV94].

4.2. Embedding Temporal Dependencies

The incorporation of temporal coherences renders the modification of the general terrain class prediction scheme (cf. Section 2.1.1) necessary. Figure 4.1 highlights the differences between the single observation-based ground surface estimation approach of Weiss et al. [WSZ07] and the one which relies on temporal dependencies [KWZ09a]. Note that in this figure, the processes of data subdivision, feature extraction, and feature scaling have been summarized into a single procedure denoted as data preprocessing. Further note that no feature selection scheme has been applied.

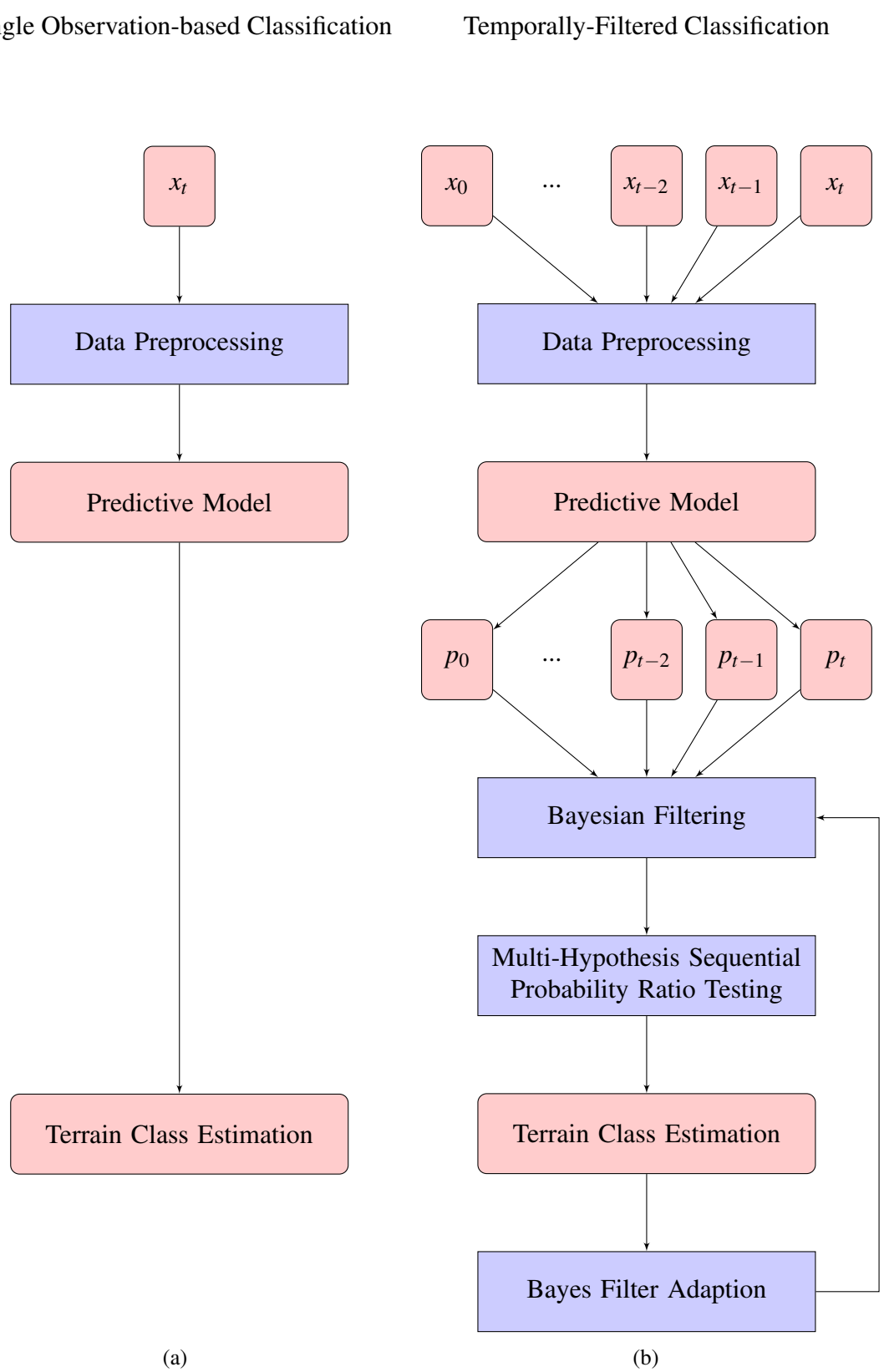


Figure 4.1.: A flow chart depicting the differences between a classification approach based on (a) single observations and (b) filtering a history of ground surface predictions. Here, p_t denotes the posterior distribution at time t.

From Figure 4.1 it can be derived that the single observation and the temporally coherent approaches differ by two key elements. First, when using the latter technique, not only one but all measurements up to the current one are considered for the terrain estimate. Second, the filtered prediction scheme does not rely on hard class assignments only, but takes the class posteriors into account. These class posteriors are then applied to a Bayesian filter to obtain temporally-filtered posterior probability estimates. At this time, an evaluation of the filtered class posteriors can be made yielding the final terrain class prediction. In this approach, however, the result is thresholded in terms of a multi-hypothesis sequential probability ratio test (MSPRT) to further reduce the number of (potentially) erroneously detected terrain transitions. Thresholding avoids a change of the terrain estimate in situations where the filtered maximum a posteriori probability $p(c = c_{\max}|X), c_{\max} = \arg\max_j p(c = j|X)$ indicates a terrain transition, yet $p(c = c_{\max}|X)$ does only differ insignificantly from the class posterior of another class $\neq c_{max}$. Finally, the filtered and thresholded terrain estimation is employed to adapt the dynamics of the Bayes filter. This aims at enabling its use in situations of both high-frequency and low-frequency terrain changes.

4.3. Applied Techniques

The method overview of Section 4.2 introduces two key techniques which have been adopted for filtering terrain class estimates: Bayesian filtering and the multi-hypothesis sequential probability ratio test (MSPRT). Both techniques are presented in the following subsections.

4.3.1. Bayesian Filtering

A Bayes filter allows for an estimation of a dynamic system's state from noisy observations. In this section, the key elements of Bayes filtering are summarized. A detailed description is provided in [TBF05].
When using Bayes filters, the state of a dynamic system at a certain time t is represented by a random variable c_t. Its uncertainty is denoted by a probability distribution over the state space c_t. Given $t+1$ sensor readings, $\{x_i\}_{i=0}^{t} \equiv x_{0:t} \equiv X$, the estimated target distribution is denoted by $p(c_t|x_{0:t})$.
Applying Bayes' rule, the posterior $p(c_t|x_{0:t})$ can be decomposed in the following way:

$$\begin{aligned} p(c_t|x_{0:t}) &= \frac{p(x_t|c_t, x_{0:t-1})p(c_t|x_{0:t-1})}{p(x_t|x_{0:t-1})} \\ &= \alpha_t p(x_t|c_t, x_{0:t-1})p(c_t|x_{0:t-1}), \end{aligned}$$

where $p(x_t|x_{0:t-1}) = \alpha_t^{-1}$ is a normalizing constant. Assuming that observations are distributed i.i.d., that is, given the current state past observations are independent of present ones, we have:

$$p(c_t|x_{0:t}) = \alpha_t p(x_t|c_t)p(c_t|x_{0:t-1}), \tag{4.1}$$

where $p(x_t|c_t)$ denotes the likelihood function or measurement probability and $p(c_t|x_{0:t-1})$ denotes the predictive distribution representing the current state estimate given past observations. Equation (4.1) quantifies the correction applied to c_t due to current sensor data. The predictive distribution is obtained by marginalizing over the previous state:

$$\begin{aligned} p(c_t|x_{0:t-1}) &= \int p(c_t, c_{t-1}|x_{0:t-1})dc_{t-1} \\ &= \int p(c_t|c_{t-1}, x_{0:t-1})p(c_{t-1}|x_{0:t-1})dc_{t-1}. \end{aligned}$$

Here $p(c_t|c_{t-1},x_{0:t-1})$ is the transition probability describing the system dynamics. $p(c_{t-1}|x_{0:t-1})$ denotes the posterior distribution from the previous time step. Bayes filters model the dynamic system by a first-order Markov process assuming that the information provided by the current state at time t suffices to predict future states without considering past observations. This yields:

$$p(c_t|x_{0:t-1}) = \int p(c_t|c_{t-1})p(c_{t-1}|x_{0:t-1})dc_{t-1}. \tag{4.2}$$

Equations (4.1) and (4.2) provide a recursive formulation of the posterior state distribution which only depends on previous and current observations. To determine the posterior probability recursively, an initial probability distribution $p(c_0) \equiv p(c_0|x_0)$ has to be defined. It is either initialized with prior knowledge about the initial state or is uniformly distributed if no prior knowledge exists.

4.3.2. Multi-Hypothesis Sequential Probability Ratio Testing

In the following, an ordered sequence of $t+1$ observations $X \equiv \{x_i\}_{i=0}^{t}$ along with their respective class memberships $Y \equiv \{y_i\}_{i=0}^{t} \in [1,k]$ are assumed. A sequential decision strategy S is a sequence of decision functions $S = \{S_1, S_2, \ldots\}$ which assign either a class j or an undecided state $\sharp$ to each observation: $S_t : (x_0,\ldots,x_t) \rightarrow \{1,\ldots,k,\sharp\}$. Here, undecided means that a class membership decision cannot be made from the first $t+1$ measurements. Starting from $t=0$, the strategy is evaluated sequentially until a class assignment can be performed.
A decision strategy is characterized in terms of the expected decision time $\hat{T}_S$ defined as

$$\hat{T}_S = \mathbb{E}(T_S(x))$$

and the error rate α_S which is the probability of an incorrect decision:

$$\begin{aligned} \alpha_S &= \sum_i \alpha_S^i \\ \alpha_S^i &= \sum_{j \neq i} \alpha_S^{ij} p(c_t = j) \qquad (4.3) \\ \alpha_S^{ij} &= p(S(x_{0:t}) = i | c_t = j), i \neq j. \qquad (4.4) \end{aligned}$$

Here, (4.3) denotes the probability of incorrectly assigning class m and (4.4) is the probability of erroneously assigning an instance from class j to class i. Using these definitions, the following decision criterion can be established:

$$S^* = \arg\min_s \hat{T}_S \text{ s.t. } \alpha_S^i \leq \alpha^i, i \in [1,k] \tag{4.5}$$

with nominal error rates $(\alpha_1,\ldots,\alpha_k)$ or

$$S^* = \arg\min_s \hat{T}_S \text{ s.t. } \alpha_S \leq \alpha, i \in [1,k] \tag{4.6}$$

with the nominal error rate α.

In contrast to the binary MSPRT, termed as SPRT, there is no optimal multi-class sequential decision strategy minimizing (4.5) or (4.6). Although approaches based on the truly optimal recursive Bayesian test exist [Tar89, Zac71, BG70], these solutions are impractical in most cases due to their complexity. Other techniques consider *ad hoc* tests which repeatedly apply a

two class SPRT in a pair-wise manner [EG91]. For most of these sequential tests, however, the bounds with regard to the truly optimal strategy are either very loose or simply do not exist.
In [BV94], a generalization to the SPRT was introduced which approximates the much more involved optimal test under Bayesian assumptions. For the parameter set $0 < A_1, \ldots, A_k < 1$, the following multi-class strategy was proposed:

$$S_t(x_{0:t}) = \begin{cases} i & \text{if } p(c_t = i|x_{0:t}) > \frac{1}{1+A_i} \\ \sharp & \text{otherwise} \end{cases} \tag{4.7}$$

Assuming all $A_i < 1$ the procedure is well-defined as there is only one i for which $p(c_t = i|x_{0:t}) > 1/(1+A_i) > \frac{1}{2}$, since the probabilities must sum to one.

In the context of terrain classification, the MSPRT approach can be adopted by first determining the filtered posterior probability $p(c_t|x_{0:t})$ in each step. If this probability exceeds a given threshold $1-\alpha$, the terrain estimate is changed to the class j which maximizes $p(c_t = j|x_{0:t})$. Otherwise, the terrain estimate remains the same. In the applied experiments, α was found experimentally using a separate data subset which was independent from the training set. Several paths with varying terrain transition frequencies were generated in a systematic manner (cf. Section 2.2) and the threshold α^* was chosen which maximized the obtained true positive rate (Section 2.1.1).
From (4.7), it can be derived that the filtering of posterior probabilities is the key element of the MSPRT technique. Hence, in the following section, two approaches are introduced which allow for their robust estimation. In this context, robust means to filter out selective misclassifications while preserving the detection of true terrain transitions.

4.4. Bayesian Filtering Applied to Terrain Class Estimation

This section considers the modifications which have been made to enable the use of Bayesian filtering and multi-hypothesis sequential probability ratio testing in the context of supervised terrain classification. Starting with a reformulation of the Bayes filter which relies on an observation's posterior probability point estimate instead of its likelihood, a general means of estimating the MSPRT threshold is presented.

4.4.1. Adaption of the Bayes Filter Formulation

In this context, the state vector comprises the class number $i \in [1,k]$, where k is the number of terrain classes to discriminate. By this coding scheme, a discrete set of k different states is obtained describing the dynamic system. The random variable c_t representing the state vector reveals the uncertainty with which the robot navigates on a certain terrain type. Preprocessed vibration data recorded by accelerometer sensors provide the observations.
To apply Bayes filtering to the problem of vibration-based terrain classification, three probability distributions have to be specified: an initial probability distribution $p(c_0)$ which denotes the probability at which the robot resides on a certain terrain type at time $t = 0$, the measurement probability $p(x_t|c_t)$ defining the likelihood that the vibration data measurement x_t can be observed navigating over a certain terrain type c_t, and the state transition probability $p(c_t|c_{t-1})$ denoting the probability that the robot moves from terrain type $c_{t-1} = j$ to terrain type $c_t = i$.

The initial probability distribution $p(c_0)$ For the derivation of $p(c_0)$, there is no information available that the robot is placed on a specific terrain type at time $t = 0$. Hence, $p(c_0)$ is assumed to be uniformly distributed.

The measurement probability $p(x_t|c_t)$ The distribution $p(x_t|c_t)$ can be learned from training examples using parametric or non-parametric density estimators [SS04]. Note, however, that the extracted feature vector generated from sensor readings has 64 dimensions. This poses a problem, since density function estimation of a high-dimensional random variable is a non-trivial task suffering from the curse of dimensionality. For this reason, another approach is adopted to represent the likelihood function. The key idea is to express the measurement probability in terms of the (estimated) posterior probability, $p(c_t = i|x_t)$, provided by machine learning classifiers. Note that in contrast to the probability density function of x_t, estimates for $p(c_t|x_t)$ are provided by certain classifiers like neural networks and support vector machines with only little additional costs.
Applying Bayes' rule to $p(x_t|c_t)$, we have:

$$p(x_t|c_t) = p(c_t|x_t)\frac{p(x_t)}{p(c_t)}. \tag{4.8}$$

The term on the right hand side of (4.8) now depends on the classifier posterior probability and the marginal probability of random variables c_t and x_t, respectively. Given no prior knowledge about the marginal probability of a certain terrain instance, $p(c_t)$ is modeled as a uniform distribution, i.e., it is assigned the value of $\kappa_1 = 1/k$ for all terrain classes. For $p(x_t)$, the complete Bayes filtering formulation is considered which assigns a probability value to a certain terrain class i, given sensor measurements $x_{0:t}$:

$$p(c_t = i|x_{0:t}) = \alpha_t p(c_t = i|x_t)\frac{p(x_t)}{\kappa_1}p_{\mathrm{pr}}(c_t = i), \tag{4.9}$$

where $p_{\mathrm{pr}}(c_t)$ is the predictive distribution:

$$p_{\mathrm{pr}}(c_t = i) = \sum_j p(c_t = i|c_{t-1} = j)p(c_{t-1} = j|x_{0:t-1}).$$

Note that the integral of (4.2) has become a sum, since we have a discrete set of possible states. From (4.9) we see that $p(x_t)$ is constant for all i and can thus be merged with the constant α to give a new normalizing constant. Introducing $\alpha_t^* = \alpha_t \frac{\kappa_2}{\kappa_1}$ with $\kappa_2 = p(x_t)$ yields the final Bayes filter formulation:

$$p(c_t = i|x_{0:t}) = \alpha_t^* p(c_t = i|x_t)p_{\mathrm{pr}}(c_t = i). \tag{4.10}$$

The state transition probability $p(c_t|c_{t-1})$ The transition probability $p(c_t = i|c_{t-1} = j)$ describes the probability of moving from state $c_{t-1} = j$ into state $c_t = i$. Given k states, k^2 probabilities have to be defined. These values are stored in a square matrix which is denoted as the transition matrix M with elements $m_{ij} \equiv p(c_t = i|c_{t-1} = j)$. The matrix diagonal elements m_{ii} represent the probabilities that the system remains in its current state whereas the non-diagonal elements $m_{ij}, i \neq j$ denote the probability of a system state change from state j to state i.
In this work, the derivation of the transition matrix is based on the control mode switching approach proposed by Coyle et al. [CCL10]. There, a transition to terrain type i is assumed if the number of terrain estimates for class i, n_i, divided by the total number of terrain predictions within a time window of w observations is larger than a predefined threshold η, i.e.

$f = \frac{n_i}{w} > \eta$. Note that the approach presented in this work differs from the one of [CCL10] in two vital aspects. First, the proposed technique does not rely the terrain prediction directly on the determined fraction f but f is employed to model the probability $p(c_t = i|c_{t-1} = i)$, i.e. the probability of remaining in the current state:

$$p(c_t = i|c_{t-1} = i) = f = \frac{n_i}{w}.$$

To avoid the zero-frequency problem [WB91], a slightly modified version of the probability distribution derivation scheme is adopted which is based on the Laplace estimation:

$$p(c_t = i|c_{t-1} = i) = \tilde{f} = \frac{n_i + 1}{w + k}.$$

Informally, the Laplace estimate increases the absolute frequencies of the observed class counts by one for each class. As an effect, the probability of remaining on a certain terrain class differs from 0 at all time steps. This choice has been experimentally verified in the conducted experiments.

Second, the number of considered ground surface estimates w, i.e. the window size, is chosen dynamically according to the present terrain characteristics. Therefore, several cases have to be distinguished. Given the current state $c_{\max}$, that is, the maximum of the filtered class posterior distribution at time t along with the state of the previous time step $\tilde{a}_{t-1}$, the window size w is either reset to one or divided in half at each iteration whenever the latest two terrain states $c_{\max} \neq \tilde{a}_{t-1}$ do not match. Here, all but the current entry of the window are rejected if the MSPRT indicates a change of the present terrain type. This is because enough evidence for a terrain transition is gathered and hence terrain estimates representing the previous terrain type can be safely removed. In the case of an unsuccessful MSPRT, the window size is halved keeping the most recent $\lceil w/2 \rceil$ ground surface estimates. Note that a change of the system state is caused by either a true or an erroneously detected transition. The aim of the window bisection technique is to establish a tradeoff between the loss of temporal dependencies in the case of a true change in the terrain class and the preservation of temporal coherences in situations of wrong system transitions. In a similar spirit, the window size is also halved whenever the previous and the current state remain the same, but the current unfiltered prediction a_t indicates a change of the present ground surface. In so doing, a possible system state transition is taken into account in terms of decreasing the degree of temporal dependencies. If none of the above-mentioned conditions is fulfilled, the probability of a terrain transition is assumed to be low which, in turn, allows for the use of a larger amount of temporal coherence. This is realized by incrementing the window size by a value of 1.

If the underlying structure of the terrain is unknown, each transition to one of the other terrain types can occur with equal probability. Hence, transition matrix elements m_{ij} with $i \neq j$ are assigned the constant value $\frac{1-m_{ii}}{k-1}$, such that $\sum_i m_{ij} \equiv \sum_i p(c_t = i|c_{t-1} = j) = 1$. The assignment of constant values to the non-diagonal elements of the transition matrix is based on the fact that typically no information is provided about the terrain characteristics in unknown environments. If, however, additional information such as impossible terrain transitions exists, the transition matrix provides an elegant means of incorporating this information into the Bayesian prediction framework.

To summarize, the elements of the transition matrix m_{ij} are given by

$$p(c_t = i|c_{t-1} = j) = \begin{cases} \frac{n_i+1}{w+k} & i = j \\ \frac{1-p(c_t=i|c_{t-1}=i)}{k-1} & otherwise \end{cases}. \tag{4.11}$$

Pseudo code for the complete Bayesian filter formulation is provided in listing 1.

Algorithm 1 A Bayesian filtering framework for coupling posterior distributions.

Require: $\{a_i\}_{i=1}^{n}$: the unfiltered terrain class estimates, k: the number of terrain classes
Ensure: $\{\tilde{a}_i\}_{i=1}^{n}$: the filtered class posteriors

1: $w = 1$
2: $\tilde{a}_1 = a_1$
3: **for** $t = 2 \rightarrow n$ **do**
4: For each class j, determine the relative frequency of terrain estimates in a given window:

$$n_j = \frac{\sum_{t^*} \mathbb{I}(\arg\max_l p(c_{t-t^*} = l | x_{t-t^*}), j)}{w}, j \in [1,k], t^* \in [0, w-1]$$

5: Update the entries of the transition matrix:

$$p(c_t = i | c_{t-1} = j) = \begin{cases} \frac{n_i+1}{n+k} & i = j \\ \frac{1-p(c_t=i|c_{t-1}=i)}{k-1} & otherwise \end{cases}$$

6: For each class, calculate the filtered, but unnormalized, class posteriors:

$$\tilde{p}(c_t = i | x_{0:t}) = p(c_t = i | x_t) \sum_j p(c_t = i | c_{t-1} = j) p(c_{t-1} = j | x_{0:t-1})$$

7: Normalize the filtered class posteriors such that $\sum_i p(y_t = i | x_{0:t}) = 1$:

$$p(c_t = i | x_{0:t}) = \frac{\tilde{p}(c_t = i | x_{0:t})}{\sum_j \tilde{p}(c_t = j | x_{0:t})}$$

8: Determine the class which maximizes the posterior along with the respective probability:

$$\begin{aligned} c_{\max} &= \arg\max_j p(c_t = j | x_{0:t}) \\ p_{\max} &= p(c_t = c_{\max} | x_{0:t}) \end{aligned}$$

9: Calculate the filtered prediction $\tilde{a}_t$ according to the flow chart depicted in Figure 4.2.
10: **end for**

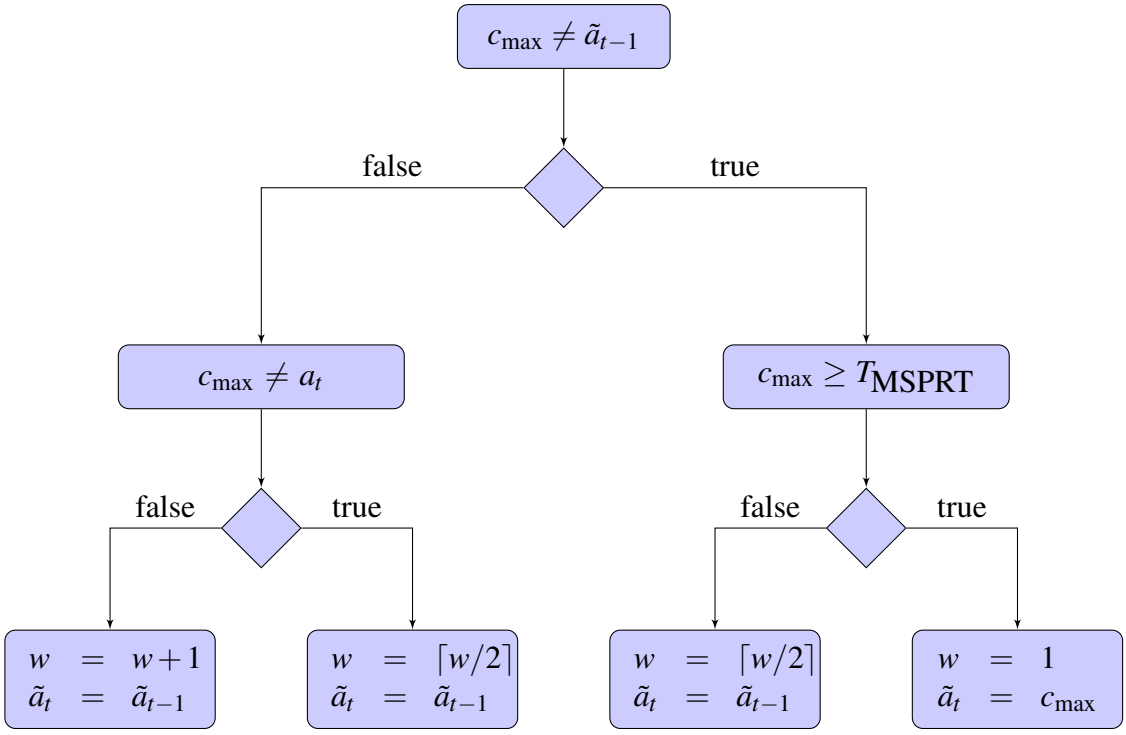


Figure 4.2.: Flow chart depicting the determination of the filtered prediction $\tilde{a}_t$ using the Bayes filtered estimate $c_{\max}$ and the most recent single classification a_t at time step t. Further w denoted the size of the current window and T_{MSPRT} is the threshold parameter for the multi-hypothesis sequential ratio test.

4.4.2. MSPRT Parameter Estimation

The threshold for the MSPRT is obtained using training data. To derive an unbiased estimate, the training set is therefore further refined into a reduced training set and a threshold estimation set. While the training set is employed exclusively for training the machine learning model, the threshold estimation set is used for determining the MSPRT threshold. Here, the subdivision is performed by assigning $2/3$ of the instances of each terrain class to the reduced training set whereas the remaining measurements constitute the threshold estimation set.
The actual threshold is determined using a one-dimensional grid search. Given a candidate threshold $T^*_{MSPRT} \in [0.51, 0.96]$ with step size 0.05, the Bayes filter is applied to a set of class posteriors. These class posteriors are obtained when applying the instances of the threshold estimation set to the trained model. The history of individual posterior probabilities represents a robot traversal over varying terrain types with certain terrain transition characteristics. After obtaining the filtered posterior probability distributions, the outcomes are evaluated in terms of the true positive rate. Finally, from the set of candidate thresholds, the threshold T_{MSPRT} is selected which maximizes the TPR.
As for the presented path, the distinction has to be made with respect to the 3 and 5 classes experiments. In the 3 classes experiment, the measurements were grouped according to their class membership. For each group, $2/3$ of all instances were assigned to the training set and the remaining ones to the threshold estimation set. These subsets were concatenated over all classes to yield the final training and threshold estimation set, respectively. The 5 classes experiment follows the class membership grouping process and the 2:1 subdivision scheme of the training and threshold estimation set. Yet, the observation concatenating technique of Section 2.2.2 has

been applied to the generated groups with a certain terrain transition profile. Modifying this transition profile among different paths from $d = 2m$ up to $d = 32m$ increases the probability of obtaining a threshold which is adequate for a range of travel distances.

4.5. Experimental Results

4.5.1. Experimental Setup

In this subsection, the choice of all remaining parameters and settings which have been employed during the experiments are presented. These include the SVM hyperparameters, the definition of the Bayes filter transition matrix, training and testing data subset partitioning, and the quality assessment procedure.

SVM Hyperparameter Selection The optimal values for the standard deviation σ of the RBF kernel as well as for the soft margin parameter C have been determined by a grid search. Each candidate parameter vector on the grid (σ, C) was evaluated by 5-fold cross-validation. As SVM implementation, the LIBSVM [CL05] library has been adopted.

Defining the Bayes Filter Transition Matrix In this work, three types of transition matrices have been implemented:

- **Static transition matrix** Following the choice of Grundmann et al. [GFB10], the elements of the transition matrix are set to $1/k$, where k denotes the number of classes. Thus, both the probability of moving to any other terrain class and the probability of remaining on the same one become equally likely.
- **Transition matrix determined by a window of constant size** Here, the determination of the transition matrix relies on the adaptive filter approach introduced in Section 4.4.1. Yet, while the adaptive technique uses a window of varying size w, the constant window approach always considers the same number of predictions to estimate the transition matrix diagonal, i.e. $w = \text{const}$. In the following experiments, w was chosen from the set $\{2, 4, 8, 16, 32\}$.
- **Transition matrix using an adaptive window approach** Using the adaptive window technique of Section 4.4.1 (denoted as *adapt* in the following), no user-defined parameters have to be specified.

Training and Testing Data Set Generation The paths employed for evaluating the proposed Bayesian filter represent natural paths containing three classes and artificially generated ones with five classes. For the latter, the path generation scheme of Section 2.2.2 has been adopted yielding robot traversals with either constant or varying travel distances, respectively. Note that certain terrain transitions are easier to detect than other ones. Hence, the classification results depend on the order in which assembled terrain segments of varying terrain type are presented. For the five classes experiment, this effect was minimized by randomly permuting this order and averaging the classification results determined after 50 reruns of a particular experiment.

Evaluation For the result evaluation, the average true positive rate introduced in Section 2.1.1 has been employed.

4.5.2. Results and Discussion

The discussion starts with the 3 classes experiments representing robot traversals which consist of continuously acquired vibration data and hence include natural terrain transitions. In Table 4.1 both the results of the single observation approach (SO) and the Bayes filter approach (BAY) are shown for varying velocity profiles (0.2 up to 0.6 m/s) when applying the constant window size (2-32), the adaptive window size (adapt) and the constant transition matrix determination schemes. In comparison with the single observation approach, the temporally coherent classification technique yields an improvement of the average true positive rate of 10.7% at most with respect to all velocity profiles and 7.2%, 10.7%, and 6.0% for velocity profiles 1-3 (0.2, 0.4, and 0.6 m/s), respectively. Concerning the constant window approach it can be stated that the filtered classification performance is rather independent from the chosen window size. That is, with a minimum derivation of 1.5%, 2.2%, and 1.1% for velocity profiles 1-3 the obtained average true positive rates do not change significantly when altering the window size between 2 and 32. For the const approach, the obtained results are always worse in comparison with the other transition matrix estimation techniques. Yet, the outcomes of the constant transition matrix determination scheme significantly outperform the ones of the single observation approach. Choosing the window size adaptively yields adequate results for all velocity profiles. Although the adaptive method does not result in the best classification performance for a certain test case, average true positive rates are obtained which are close to the best ones or at least are among the best three TPRs.
This observation is confirmed in Figure 4.3 where the results with regard to a certain window size are averaged over velocity profiles 1-3. Figure 4.3 also reveals the trend of favoring larger window sizes. Note that the size of the window should correlate with the amount of estimated temporal dependencies within the data set. The larger the latter, the more terrain predictions can be considered during the filtering process and hence, the larger has the window size to be chosen. As there are only 4 terrain transitions contained in the 3 classes experiments, a large amount of temporal coherences can be assumed which, in turn, validates the trend of selecting larger window sizes. Comparing the optimal window sizes for the constant window approach and the average window sizes of the adaptive approach for a given traversal path (Table 4.2), an approximate correspondence can be observed. This shows that the adaptive approach correctly determines the relationship between the amount of temporal dependencies contained in the present path and the window size to be chosen.

Table 4.1.: Classification performance in terms of the true positive rate [%] for the 3 classes experiments with respect to varying velocity profiles (vel), filter sizes, and the unfiltered single observation (SO) and Bayes filtered (BAY) classification approach.

vel	approach	filter size						
		2	4	8	16	32	const	adapt
0.2 m/s	SO	88.4	88.4	88.4	88.4	88.4	88.4	88.4
	BAY	94.1	94.7	**95.6**	95.0	95.4	92.4	95.1
0.4 m/s	SO	86.8	86.8	86.8	86.8	86.8	86.8	86.8
	BAY	95.4	95.2	95.9	97.4	**97.5**	94.7	96.3
0.6 m/s	SO	92.1	92.1	92.1	92.1	92.1	92.1	92.1
	BAY	97.1	**98.1**	98.0	98.0	97.1	96.4	98.1
average	SO	89.1	89.1	89.1	89.1	89.1	89.1	89.1
	BAY	95.5	96.0	96.5	**96.8**	96.7	94.5	96.5

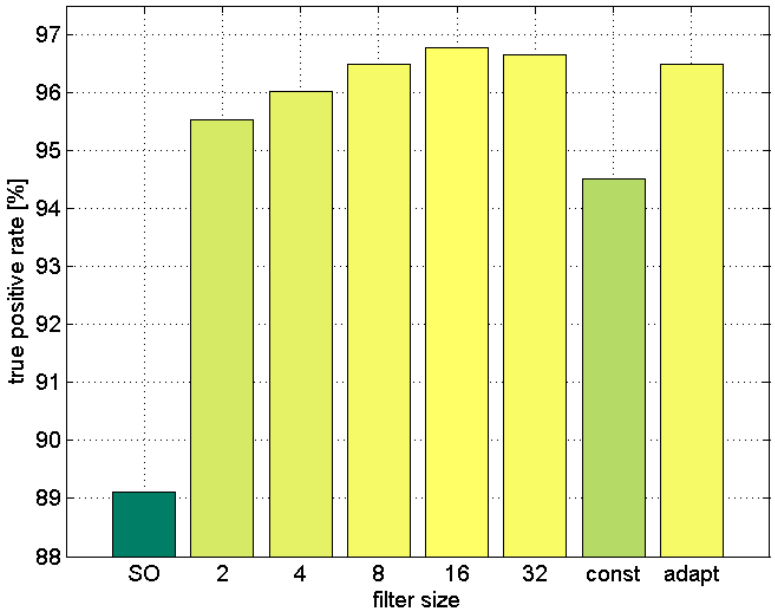


Figure 4.3.: The visualization of the results of the 3 classes experiments with respect to varying filter sizes when averaging the outcomes over the complete set of velocity profiles.

Table 4.2.: The average filter sizes for the 3 classes experiments with respect to varying velocity profiles (vel).

vel	classifier
	svm
0.2 m/s	25.4
0.4 m/s	19.9
0.6 m/s	11.8

The presentation of the five classes experiments is similar to the one of the three classes experiments. The key difference is the introduction of varying travel distances either related to an individual path (var) or a set of paths each featuring a single travel distance *dist*, with $dist \in \{2,4,8,16,32\}$.
The use of the temporally coherent classification scheme yields significant improvements in terms of the true positive rate. Depending on the chosen velocity profile and travel distance, a maximum increase of classification performance by 9.7% can be observed in comparison with the single observation approach. If the differing velocity profiles are considered individually, the overall improvement of TPR is 9.7%, 5.1%, and 6.2% for velocity profiles 1-3, respectively.
In contrast to the three classes experiments, the Bayes filter favors smaller window sizes in the 5 classes experiments for establishing the transition matrix. This is an expected behavior, since now the terrain transitions appear more frequently reducing the amount of temporal dependencies within the present paths. As an effect, the window size has to be reduced to not consider outdated measurements which represent another terrain type. Note that with increasing travel distance the amount of temporal dependencies rises as well. Hence, when using a temporal filter, the filtered classification results are expected to increase alike. As shown in Table 4.3, the true positive rates follow this assumption.
Similar to the 3 classes experiments, the adaptive transition matrix generation approach yields results which are close to the best ones (cf. Figure 4.4). One reason for this characteristic is shown in Table 4.4. Here, the average window size with regard to the travel distance is presented. The results reveal that the adaptive Bayes filter approach is able to establish a correct relation between the present travel distance and the estimated filter size.

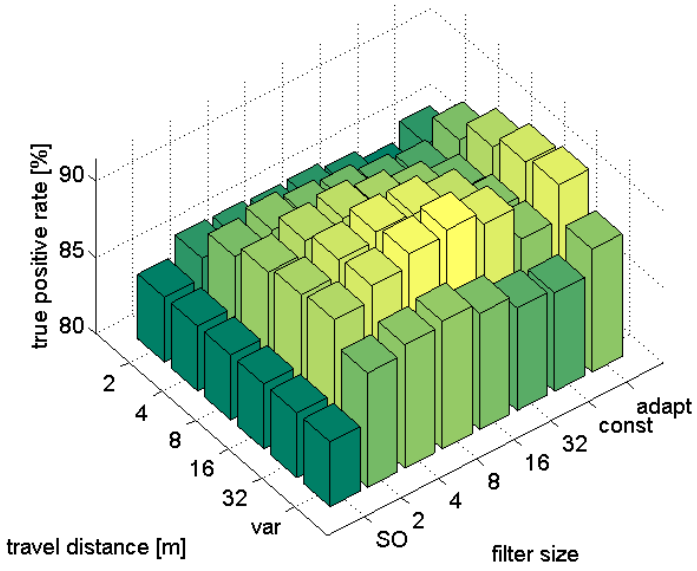
To assess the effects of the filtering process on the state transition behavior, Figure 4.5 depicts two robot traversals of the 5 classes experiment at a robot speed of 0.2 m/s. Here, correct (blue bars) and erroneous (red bars) transitions are presented with respect to unfiltered and Bayes filtered classification along with the reference transition sequence. Thereby, a true transition is defined as a state change from any terrain type of time step $t-1$ to the correct terrain type of time step t. If the system changes into an incorrect state representing a wrong ground surface the respective transition is denoted as an erroneous one. Furthermore, a gray bar in Figure 4.5 indicates situations in which no terrain transition occurs but the system erroneously remains in the wrong state.
Figure 4.5(a) shows a positive example of Bayes filtered classification. In comparison with the single observation approach, the temporally coherent classification scheme requires less state transitions which is due to the correct filtering of erroneous predictions. Further, the filtered transition sequence reveals a fast state switching behavior of the Bayes filter, since almost all state transitions are processed without any delay. The sole exception occurs at time step $t = 91$ where the system erroneously remains in its current state and finally migrates into the correct one not until the third prediction. Note, however, that the wrong prediction of the single observation approach at time step $t = 91$ contributes to the delayed detection of the state change.
Finally, Figure 4.5(b) provides an example where the Bayes filter fails at correcting erroneous predictions. Beginning from time step $t = 41$, the single observation approach yields wrong results over a period of 6 terrain estimates. Since the filtered classification technique is based on these individual predictions, there is much evidence for the system indicating the traversal over a wrong ground surface. This results in the belief of staying on a certain terrain type which differs from the true one. Note, however, that the number of erroneous terrain transitions is smaller in comparison with the single observation approach preventing the system mechanics from an increased exposure.

Table 4.3.: Classification performance in terms of the true positive rate [%] of the 5 classes experiments with respect to varying velocity profiles (vel), travel distances (dist), filter sizes and the unfiltered (SO) and Bayes filtered (BAY) classification schemes.

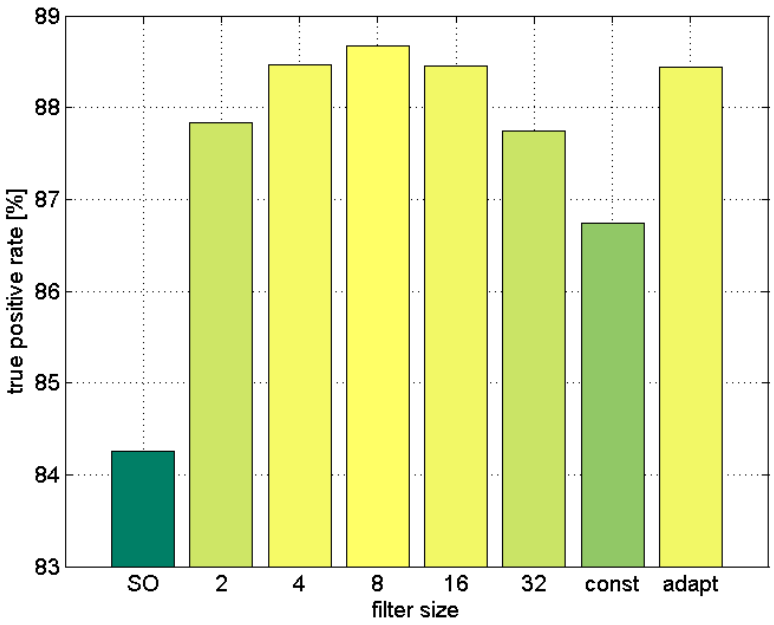
vel	dist	filter size						
		2	4	8	16	32	const	adapt
0.2 m/s	SO	75.2	75.2	75.2	75.2	75.2	75.2	75.2
	2	77.2	77.8	**78.0**	77.3	76.3	75.8	77.3
	4	80.2	81.2	**81.3**	81.2	79.3	78.7	80.9
	8	81.3	**82.7**	82.7	82.6	81.4	79.4	82.1
	16	81.8	83.5	83.7	**83.7**	83.2	80.0	83.0
	32	82.2	83.5	84.0	**84.8**	82.8	80.1	83.4
	-1	79.0	79.8	**80.0**	79.0	77.8	77.3	79.5
0.4 m/s	SO	89.5	89.5	89.5	89.5	89.5	89.5	89.5
	2	90.2	89.9	89.5	90.2	**90.3**	89.6	90.3
	4	92.0	**92.2**	91.4	91.3	91.2	91.5	91.8
	8	92.7	93.2	**93.3**	92.8	92.3	92.8	93.0
	16	93.2	93.7	**94.2**	93.9	93.4	93.3	93.9
	32	93.4	94.0	**94.6**	94.5	94.4	93.5	94.3
	-1	92.3	92.6	**92.8**	92.4	91.7	92.3	92.6
0.6 m/s	SO	88.1	88.1	88.1	88.1	88.1	88.1	88.1
	2	**89.2**	88.8	88.6	88.7	88.3	87.4	88.9
	4	90.4	**90.6**	89.9	89.4	89.6	88.9	90.2
	8	91.2	91.8	**92.3**	91.1	90.9	89.9	92.1
	16	91.6	92.5	93.4	**93.4**	92.1	90.5	93.1
	32	91.8	92.7	94.0	**94.2**	93.8	90.7	93.5
	-1	91.2	91.9	**92.3**	91.5	90.6	89.8	92.1
average	SO	84.3	84.3	84.3	84.3	84.3	84.3	84.3
	2	**85.6**	85.5	85.4	85.4	85.0	84.3	85.5
	4	87.6	**88.0**	87.5	87.3	86.7	86.4	87.6
	8	88.4	89.2	**89.4**	88.8	88.2	87.4	89.1
	16	88.8	89.9	**90.4**	90.3	89.5	87.9	90.0
	32	89.1	90.1	90.9	**91.2**	90.3	88.1	90.4
	-1	87.5	88.1	**88.4**	87.6	86.7	86.4	88.1

Table 4.4.: The average filter sizes for the 5 classes experiments with respect to varying velocity profiles (vel) and travel distances (dist).

dist	velocity		
	0.2 m/s	0.4 m/s	0.6 m/s
2	6.0	4.9	4.2
4	8.6	7.3	5.7
8	10.4	10.7	8.7
16	12.0	15.0	12.6
32	12.9	18.3	15.9
var	7.4	9.7	9.3

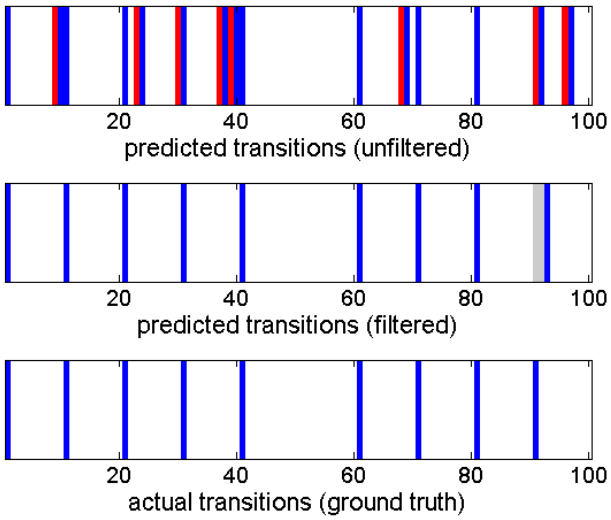


(a)

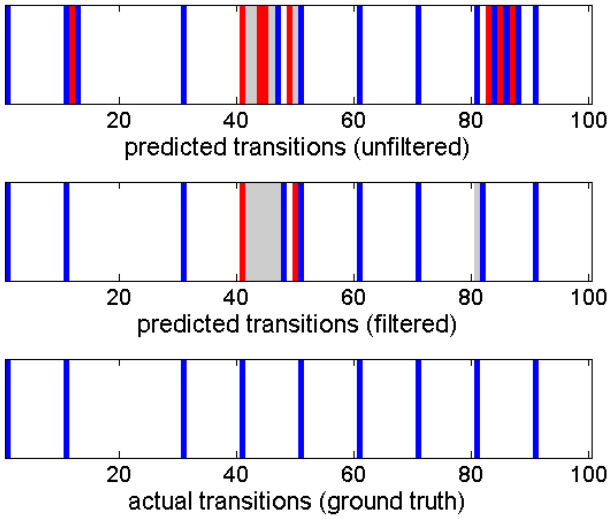


(b)

Figure 4.4.: The visualization of the results of the 5 classes experiments with respect to (a) varying travel distances and filter sizes when averaging the outcomes over the complete set of velocity profiles and (b) varying filter sizes when averaging the outcomes over the complete set of velocity profiles and travel distances.



(a)



(b)

Figure 4.5.: The visualization of a simulated robot traversal over (a) 3 terrain classes and (b) 5 terrain classes. The blue bars indicate a correct transition, that is, a transition into a correct state. In this case, it is irrelevant whether the previous state represented the true terrain type. Red bars, on the other hand, reveal an erroneous belief that the ground surface changed from one time step to the next. Finally, the bars are colored gray when the system remains in an erroneous state.

4.6. Conclusion

In this chapter, a novel means of terrain classification has been proposed, which does not rely its current prediction on a single measurement only but uses a history of predictions instead. The temporal filtering is achieved in terms of a Bayes filter, whose original formulation had to be modified to fit into the proposed classification scheme. Using the modified formulation, the Bayes filter only makes use of a set of recent class posterior estimates which are then coupled in a systematic manner. As the coupling process does not need the generation of an additional coherent model, the filtered predictions are efficiently determined enabling the filter's use on robots with low computational and memory capacities as well.
In total, three approaches have been suggested to estimate the free parameter of the Bayes filter which is represented by the transition matrix. Here, window-based approaches resulted in the best classification performance and proved to be rather insensitive against a variation of the chosen window size. Further, a proposed adaptive technique which automatically adjusted the window size according to a history of terrain classifications provided near-optimal results in terms of the true positive rate. This enables the use of the Bayes filter in situations of both high-frequency and low-frequency terrain changes. To conclude, the application of the suggested Bayes filter formulation yielded a maximum absolute increase of the true positive rate of more than 10%.

5. Classifier Selection for Temporally Coherent Terrain Class Estimation

5.1. Introduction

The success of the Bayes filter approach presented in the last chapter highly depends on the quality of posterior probability estimates. In the ideal case, a large posterior probability should be assigned to the class the respective observation belongs to and a small posterior otherwise. For the estimation of posterior probabilities, the SVM classifier was employed. Note, however, that the original formulation of the SVM does not provide a formalism to estimate these probabilities directly. Instead, given a one-vs-one multi-class classification scheme, $\frac{n \cdot (n-1)}{2}$ binary classificators are evaluated and the respective classification scores are converted into pairwise probability estimates. These estimates are then coupled in a pairwise manner to yield the final posterior probability for each class. Any assumption within the prediction chain which does not reflect the underlying model generating the data set will increase the probability of observing inadequate posterior estimates. Thus, it remains questionable whether the support vector machine provides the best filtered classification performance. Furthermore, according to the *No Free Lunch Theorem* [WM95], there are no context-independent reasons to favor one classification method over another. This renders a systematic comparison of different posterior probability estimation techniques necessary with respect to their inclusion into the Bayes filter approach presented in the last chapter.

In the context of unfiltered terrain classification, the problem of classifier selection has already been addressed by two authors. Weiss et al. [WFSZ07] compared different classifiers including the support vector machine, multi-layer perceptrons, decision trees, the naïve Bayes classifier and the k-nearest neighbor approach for the terrain discrimination task. They concluded that a support vector machine employing a radial basis function kernel yielded the largest generalization performance. Later, Coyle et al. [CC08] refined these experiments by also adopting a principal component feature extraction scheme along with other feature representations. In contrast to the previously mentioned approach, the results presented by Coyle et al. do not reveal a superiority of the RBF-based SVM classifier over other approaches. For example, a support vector machine using a polynomial kernel and a Parzen window estimator performed comparably well in terms of the generalization performance.

Note, however, that both comparisons only evaluate the hard assignments of each observation to a certain class. The degree of reliability of each prediction is not taken into account. Since the success of the Bayes filter approach is highly correlated with the quality of posterior probability estimates, the experiments of this chapter focus on this issue [KZ09a]. As a further contribution, the random forest [Bre01] and random ferns classifiers [OCLF10] are introduced into the domain of terrain classification. The application of the former is motivated by both its success in vision-based terrain classification approaches [KKBZ11, KKZ11] and its almost parameter-free usage. Note that only a single user-defined parameter, the number of employed decision trees, has to be provided. The performance of the random ferns approach, on the other hand, is determined by the choice of its base binary classifier which renders a thorough

investigation of base classifier selection necessary. The final contribution of this work comprises the introduction of a novel acceleration segment preprocessing scheme which reduces the dimensionality of the input vector from 64 to 6 frequency components. As shown in the result section, the proposed preprocessing technique yields a significant improvement of the classification performance for several classifiers.

5.2. Applied Classifiers

In this section, all classifiers are briefly described which have been embedded into the Bayes filter classification approach. Therefore, it is explained how posterior probabilities $p(c = j|x_i)$ can be determined for each class j under consideration.

5.2.1. k-Nearest Neighbor Classification

k-nearest neighbor classification (KNN) [CD07] determines the set of the k nearest neighbors contained in a training set to a testing instance x_i. Then, the frequency of occurrence of each class in the neighbor set is calculated. The class with the largest frequency becomes the predicted class for the testing instance x_i. The posterior probability $p(c = j|x_i)$ is defined as the ratio between the number of occurrences of class j in the neighbor set, n_j, and the number of considered neighbors k, $p(c = j|x_i) = \frac{n_j}{k}$.

5.2.2. Multi-layer Perceptrons

The multi-layer perceptron (MLP) [Bis95] is an instance of an artificial neural network. It consists of artificial neurons which are interconnected in a well-defined manner. These neurons are arranged in three different layers: in an input layer, a hidden layer, and an output layer. When applying an input x_i to the network input, the *nhidden* neurons of the hidden layer perform a weighted sum of the input components: $net_l = \langle w_l, x_i \rangle$, $l \in [1, nhidden]$. Here, net_l denotes the net activation of neuron h_l and w_l is the weight vector determining the specific contribution of each input component to the final sum. An activation function f_{act}, typically chosen as $f_{act} = tanh(net_l)$, is then applied to each net activation to obtain the final output for the neurons of the hidden layer. The determination of the net activation of the output neurons is equivalent to the ones of the hidden layer except that we do not add weighted input coefficients but weighted activations of the hidden neurons. For classification problems, the activation function of the output neurons is replaced by the softmax function which takes the form $f_{act} = \exp(net_m)/\sum_{m'} \exp(net_{m'})$, where net_m is the net activation for output neuron m. Each output neuron represents a certain class to discriminate. The predicted class is the one which is represented by the neuron with the maximum activation. It can be shown [Bis95] that the activations can directly be interpreted as posterior probabilities.
Training the network involves the optimization of the adaptive network parameters, the weights and biases, with respect to a given error function E. For classification, the cross-entropy error function is employed which has the form:

$$E = -\sum_{i=1}^{n} \left\{ y_i \ln \frac{a_i}{y_i} + (1 - y_i) \ln \frac{1 - a_i}{1 - y_i} \right\},$$

where $\{y_i\}$ denotes the set of given training labels and $\{a_i\}$ is the set of predicted class labels for each training instance x_i. Several approaches have been advised to obtain a minimization

of E, e.g. the Newton, the Conjugate Gradients, and the Levenberg-Marquardt approaches. In this work, the quasi-Newton approach presented in [Nab02] has been adopted which does not require second-order information but uses first-order derivative information of the error function to determine increasingly accurate approximations of the inverse Hessian matrix.
Neural networks trained with iterative gradient-based methods tend to learn the Fourier components of the target function in a low-frequency to high-frequency order. Hence, stopping at an appropriate point during training, the network does not learn the high-frequency noise content contained in the training signals. This point can be identified by using a validation set which is disjunct from the training and test sets. While the training error will always decrease, the validation error will decrease to a minimum and then begins to rise again as the network is being overtrained. Ideally, training stops when the minimum validation error is reached to obtain a network with best generalization behavior. In this work, the validation set of size $valRatio \cdot n$ is chosen deterministically by means of a k-means clustering (cf. Section 3.3.1) with KKZ initialization [KKZ94]. The KKZ algorithm, named after its inventors Katsavounidis, Kuo, and Zhang, is an example of a distance-based initialization scheme. In each step of the greedy algorithm, the data instance is determined which has the maximum distance from its nearest cluster center already contained in the initial cluster set. Then, the chosen instance is included as novel entry into the set of initial cluster centers. A deterministic approach was chosen instead of a random one, since the former outperformed the latter in various experiments.

5.2.3. Probabilistic Neural Networks

Probabilistic neural networks (PNN) [Spe90] are another instance of artificial neural networks. In the training phase, scaled training patterns are inserted into a matrix W_c, $c \in [1,k]$, according to the class c they belong to. Each row of W_c represents a single pattern. The scaling is performed such that the L_2 norm of each training instance equals to one. In the recall phase, the same scaling is applied to the test vector x_i. For each class c, the inner product between each pattern w_i of the weight matrix W_c and the query x_i is determined yielding the net activation $net_{c,i}$. The net activations are non-linearly transformed using the activation function $f_{act}(net_{c,l}) = \exp((net_l - 1)/\sigma^2)$, where σ is a model parameter defining the size of the Gaussian window. For each class, the sum over all transformed net activations is determined, $s_c = \sum_l f_{act}(net_{c,l})$, and the predicted class becomes the one which maximizes s_c. Given that the probability of each class is distributed uniformly, the posteriors $p(c = j|x_i)$ are then defined as $p(c = j|x_i) = (n_j^{-1} s_j)/(\sum_l n_l^{-1} s_l)$, where n_j is the number of training instances for class j.

5.2.4. Gaussian Mixture Model Classifiers

In a Gaussian mixture model, each class j is represented by its own Gaussian distribution: class $j \sim N(\mu_j, \Sigma_j)$, where the mixture model parameters $\theta_j = \{\mu_j, \Sigma_j\}$ represent the mean of the class instances $x_i, i \in [1,n_j], x_i \in$ class j and the corresponding covariance matrix, respectively. Given the posterior probability of each individual class $p(c = j)$, the probability density function of the random variable X can be defined as $p(x_i|\theta) = \sum_{m=1}^{k} p(c = j) p(x_i|\theta_m)$ with $p(x_i|\theta_m)$ defining the likelihood distribution. The latter represents the density of each component and is assumed to be Gaussian distributed:

$$p(x_i|\theta_m) = \frac{|\Sigma_m|^{-1/2}}{(2\pi)^{d/2}} \exp\left\{ -\frac{1}{2}(x - \mu_m)^T \Sigma_m^{-1} (x - \mu_m) \right\}.$$

Applying the Bayes rule, the class posterior $p(c = j|x_i)$ is obtained by $p(c = j|x_i) = \frac{p(x_i|c=j)p(c=j)}{p(x_i)}$.

5.2.5. Random Forests

Decision Trees have shown their applicability in various classification tasks. Yet, predictive models which have been generated with this approach tend to overfit the data and hence do not generalize well. Random forests try to overcome these problems by injecting randomness into the tree generation procedure and by combining the output of n_T randomized trees into a single classifier.
The trees are established by recursively bisecting the data set into smaller subsets at each inner node R_i. As splitting criterion for each node R_i, the Gini-index is employed which is defined by:

$$I_G(i) = \sum_{j=1}^{k} \hat{p}_{ij}(1 - \hat{p}^{ij}),$$

where k is the number of classes to discriminate and $\hat{p}^{ij}$ denotes the probability of observing a measurement of class j given all instances provided in node R_i. That is, $\hat{p}^{ij} = \frac{n_j}{n_i}$, where n_j denotes the number of measurements belonging to class j and n_i is the total number of observations in node R_i. At each splitting step, the remaining data is separated into two distinct subspaces using a random feature subset of size m, where m is typically chosen by $m = \sqrt{d}$. Then, the best split on these features is used to bisect the node into two subnodes, R_{c_1} and R_{c_2}. In this context, the best split is defined to be the subdivision which maximizes the decrease in the Gini-index:

$$\Delta I_G(i) = I_G(i) - \hat{p}^{c_1} I_G(c_1) - \hat{p}^{c_2} I_G(c_2).$$

The splitting procedure is recursively applied until a maximum tree depth is reached. For decision trees, pruning or recursion depth limitation techniques have to be applied to prevent overfitting. Random Forests classifiers, however, grow trees of maximum depth without performing subsequent pruning steps.

After tree generation, each leaf node stores several instances along with their respective class membership. The latter can be adopted to assess posterior probabilities as described in the following paragraph.
In the relative class frequency approach, the posterior probability distribution is obtained by averaging the relative class frequencies of all members of the ensemble:

$$p(c = k^*|x_i) \approx F(\{t_1, \ldots, t_{n_T}\}, x_i, k^*) = \frac{1}{n_T} \cdot \sum_{j=1}^{n_T} \frac{f(t_j, x_i, k^*)}{\sum_{l=1}^{k} f(t_j, x_i, k_l)},$$

where $f(t_j, x_i, k^*)$ denotes the number of estimation examples which belong to class k^* and which are assigned to the same leaf as instance x_i in the jth tree t_j. Here, the term *estimation examples* is employed to stress that this set is only used for the estimation of posterior probabilities and hence does not have to be identical to the training set in general. In this work, however, the approaches of [Bre96] and [Bre01] have been followed which suggest to choose the estimation examples to be identical to the original set of training examples.
To motivate the second posterior estimation technique, the following example is considered where two classes, A and B, are given along with the leaf node of a single tree for which the class posterior estimation is based on a single instance $x, x \in$ class A. Further, a second leaf node is provided containing instances of both classes, A and B. Here, the estimated posterior $p(A|x) = 1$ assigned to the former node will always be larger in comparison with $p(A|x)$ of the latter one, independently of the number of examples and the respective class distribution.

A common solution to this problem is to modify the observed class frequencies. In the Laplace estimation approach, the relative frequencies are adjusted by adding the value of 1 to the number of observed estimation examples for each class in each leaf:

$$p(c = k^*|x_i) \approx L(\{t_1,\ldots,t_{n_T}\},x_i,k^*) = \frac{1}{n_T}\cdot\sum_{j=1}^{n_T}\frac{1+f(t_j,x_i,k^*)}{k+\sum_{l=1}^{k}f(t_j,x_i,k_l)}.$$

Here, the increment by one represents a regularization term which behaves as a uniform Dirichlet prior [Bis06] over feature values. If an instance assigned to a specific leaf node is not encountered during training, the inclusion of the additional terms assign a non-zero value to the corresponding probability.
The last posterior estimation technique is based on averaging the unweighted votes of the members of an ensemble. Here, each member votes for a single, i.e. the most probable class:

$$p(c = k^*|x_i) \approx V(\{t_1,\ldots,t_{n_T}\},x_i,k^*) = \frac{1}{n_T}\cdot\sum_{j=1}^{n_T}\mathbb{I}\left(\max_{k'}\left(t_j(x_i,k')\right),k\right),$$

where $t_j(x_i,k')$ returns the estimated probability of x_i belonging to class k' according to the jth tree t_j.

During the recall phase, the test pattern traverses each random tree until a leaf node is reached. The posterior distributions assigned to the respective nodes are then averaged over all members of the ensemble. Finally, the class which maximizes F, L, or V is chosen to be the classification result of the test pattern.

5.2.6. Random Ferns

Random ferns [OCLF10] consist of an ordered set of n_F binary tests, yielding either 0 or 1, respectively. Hence, when applying an observation to each fern, a bitstring of n_F values is obtained. Further, this bitstring is divided into $\lceil n_F/n_B\rceil$ substrings of length n_B. Depending on the applied observation, each substring is assigned to one of 2^{n_B} possible states $\in [0,2^{n_B}-1]$. For each state, a posterior distribution $p(c = j|x_i,s_l)$ has to be established during model training which defines the probability of observing a certain class j given the observation x_i and state $s_l, l \in [0,2^{n_B}-1]$. In the recall phase, each substring of the initial bitstring is evaluated and the obtained posterior distributions are averaged over all ferns. In the following, details of the binary tests and posterior distribution calculations are provided.

Binary Tests

In the present classification framework, several binary tests have been implemented, ranging from simple feature magnitude comparisons to both random and deterministic linear classifiers.

Feature Magnitude Comparison In the original paper [OCLF10], Özuysal et al. propose a simple binary test whose outcome only depends on the magnitude of two features $x_{i,u}$ and $x_{i,v}$:

$$b_j(x_{i,u},x_{i,v}) = \begin{cases} 1 & \text{if } x_{i,u} < x_{i,v} \\ 0 & otherwise \end{cases},$$

where j denotes the index of the fern under consideration. Further, the feature indices u and v are randomly selected and stored for their later use in the recall phase.

Random Linear Classifier The second binary test is represented in terms of a linear classifier. This type of predictive model aims at dividing the data space into two distinct subspaces using a separating hyperplane which is formally described by

$$f(x) = w^T x + b.$$

The class decision is made by assigning each instance fulfilling $f(x) > 0$ to the first class and the others to the remaining class:

$$b_j(x_i) = \begin{cases} 1 & \text{if } w^T \cdot x_i + b \leq 0 \\ 0 & otherwise \end{cases} \tag{5.1}$$

The function f is referred to as decision function and the separating hyperplane defined by $f(x) = 0$ represents the decision boundary. For choosing the model hyperparameters w and b, Bosch et al. [BZM07] suggested a random approach: w is defined to be a random vector having the same number of components as the observation x_i. Similar to the random forest approach, only a subset of features is considered for the node test. This is realized by assigning n_f components a random value which is uniformly drawn from the interval $[-1,1]$. The remaining components are set to zero. The value of b is assigned a random value as well. It is chosen randomly between 0 and the distance of the furthest instance $x_i, i \in [1,n]$, from the origin.
Since not all random tests are appropriate in terms of class separation, r candidate binary tests are initialized and evaluated using the following entropy criterion:

$$\Delta E = -\sum_i \frac{|Q_1|}{|Q|} H(Q_1),$$

where Q is the complete set of instances applied to test b_j and Q_1 is the set of examples which are assigned to the first subspace according to the given test b_j. Further, $H(Q)$ is the entropy $H(Q) = -\sum_{j=1}^{k} p_j \log_2(p_j)$ with p_j denoting the fraction of examples in Q belonging to class j and $|.|$ being the size of the respective set. Using the entropy criterion ΔH, the binary test which maximizes ΔH is selected from the candidate set.

Posterior Probability Calculation

In the random ferns approach, a set of binary tests of size n_B are grouped together. Hence, when applying an observation to this group, a bitstring s_l, $l \in [1,n_F]$ of length n_B is obtained. This bitstring is a representative of the observation, yet this representation is not unambiguous in the sense that instances from varying classes can be transformed into the same bitstring. Given the training set, however, the probability $p(c = j|s_l)$ can be determined by counting the instances of a class j which are mapped to a certain bitstring s_l and dividing this number by the total number of training examples which are mapped to s_l. In [BZM07], Bosch et al. used this distribution to approximate the posterior $p(c = j|x_i)$. They therefore determined a class posterior $p_l(c = j|x_i)$ for each fern l and averaged the outcomes over the complete set of ferns. Due to the averaging process, this posterior probability estimation technique is denoted as the average class posterior approach in the remainder of this chapter.

5.3. Revisiting Random Ferns and Feature Extraction

When considering the results of Section 5.4.2, at least two issues can be noticed. First, the inferior classification performance of the random ferns classifier with respect to the other classification approaches, and second, the large amount of time spent during the model selection procedure. Solutions which address both issues are discussed in the following subsections.

5.3.1. Improving the Performance of the Random Ferns Classifier

Concerning the random ferns classifier, its prediction capability highly depends on the performance of the chosen binary test. Thus, the substitution of the binary tests presented in Section 5.2.6 by a more adequate one is likely to improve the true positive rate. In the following, two candidate tests are introduced: the binary support vector machine with a linear kernel and the Fisher linear discriminant.

liblinear Classifier Instead of choosing random hyperplanes for data separation, this test uses a deterministic approach for calculating n and b. Therefore, the results of the training process of a linear support vector machine is employed, i.e. n_F linear classifiers are trained in a one-vs-all scheme. Here, classifier i assigned to fern i distinguishes between class $((i-1) \text{ MOD } k)+1$ and the remaining classes. The data instances representing positive and negative examples are randomly subsampled from the training set such that the size of both sets, n_P, is equal. This assures the avoidance of unwanted effects caused by an uneven class distribution. Analogous to the previous testing scheme, only a subset of n_f features are taken into consideration for the classification task.

Fisher Linear Discriminant Here, the conversion of the multi-class problem into several binary ones is applied analogously compared to the previous liblinear approach. Yet, another instance of linear discrimination is employed which is denoted as the Fisher linear discriminant. This binary test defines the optimal separating hyperplane to be the one which preserves the data's variance best when the latter is projected into a one-dimensional space. Therefore, the following cost function is maximized:

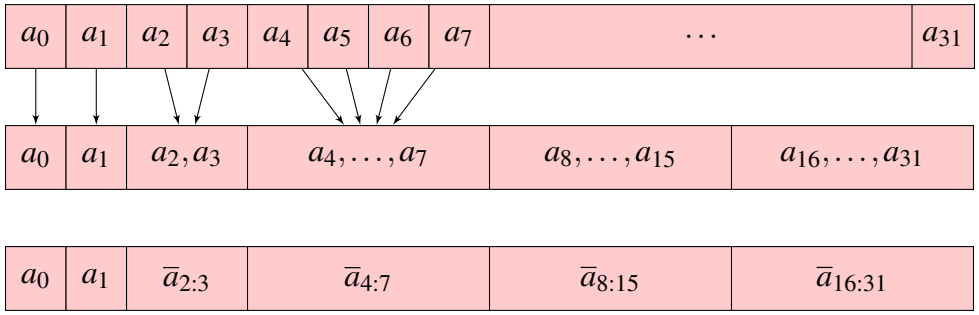
$$J(w,\mu_x,\mu_y,\Sigma_x,\Sigma_y) = \frac{w^T(\mu_x-\mu_y)(\mu_x-\mu_y)^T w}{w^T(\Sigma_x+\Sigma_y)w} = \frac{(w^T(\mu_x-\mu_y))^2}{w^T(\sigma_x+\sigma_y)w},$$

where X and Y denote two random variables, and μ_x, Σ_x and μ_y, Σ_y are the mean and the covariance of X and Y, respectively. Informally, the maximization of J is achieved by choosing a weight vector w such that the projected means of both classes are distant from each other while minimizing the variance within the classes. A discriminant which maximizes J can be calculated as

$$w^{\text{norm}} = (\Sigma_x+\Sigma_y)^{-1}(\mu_x-\mu_y),$$

yielding the maximum Fisher discriminant ratio

$$(\mu_x-\mu_y)^T(\Sigma_x+\Sigma_y)^{-1}(\mu_x-\mu_y) = \max_{w\neq 0} J(w,\mu_x,\mu_y,\Sigma_x,\Sigma_y).$$



(a)

Figure 5.1.: The visualization of the MFCC-like feature extraction scheme. First, individual features are binned based on a logarithmic scale and second, the set of all features contained in a specific bin are averaged to form the final feature descriptor.

5.3.2. Low-Dimensional Fingerprints of Vibration Patterns

The acceleration pattern's amplitude spectrum has been reported to adequately represent a vibration signal. Yet, consisting of 64 dimensions, it gives rise to problems for classification techniques which suffer from the curse of dimensionality. This renders a more compact representation of the vibration signal necessary. The proposed vibration fingerprint has only 6 dimensions. It is based on the Mel-frequency cepstral coefficients (MFCC) approach which has gained attention in the audio signal processing domain. The descriptor is established by first subdividing the audio data into frames which represents a block of audio samples acquired consecutively. This block is then transformed into the Fourier domain using a discrete Fourier transform (DFT) and the corresponding amplitude spectrum is determined. After calculating the logarithm of the latter, scaling and smoothing operations are applied to the transformed DFT coefficients. The smoothing process is performed by assigning each spectral component a certain bin, where the number of bins is (significantly) smaller than the number of spectral components. Thereby, the number of transformed DFT components assigned to a specific bin rises with increasing frequency. This choice is based on the findings that lower frequencies are perceptually more important than higher frequencies. Smoothing is now accomplished by means of averaging the set of transformed DFT components within each bin, where averaged values become the new bin's representative. In a final step, a decorrelation transform is applied to these binned and averaged features to reduce the dependencies among them.
The proposed fingerprint for vibration signals adopts the essential elements of the MFCC approach consisting of frame subdivision, Fourier transformation, amplitude spectrum determination, spectral binning, and spectral smoothing operations. Whereas the first three steps of the MFCC-like vibration segment transform equal the original vibration signal processing pipeline introduced in Section 2.1, the binning and smoothing steps are novel means which aim at the reduction of the overall feature dimensionality. Analogous to the MFCC descriptor, the proposed fingerprint also assigns higher importance to the spectral components with lower frequencies. This is based on the assumption that high-frequency content of the vibration signal only represents noise which is induced by smaller rocks or bumps on the ground surface. As these factors are frequently observed on a variety of terrain types, the resulting high-frequency parts of the signal do not indicate a certain terrain type and hence play only an inferior role in the discrimination task.
In this implementation, the binning scheme is defined as follows (cf. Figure 5.1): In total, there are 6 bins being labeled from 0 to 5. Given a subset of 32 preprocessed DFT coefficients

$\{a_i\}_{i=0}^{31}$ which represent the lower energy content of the vibration signal, the first two components, a_0 and a_1, are unalteredly inserted into the feature set, i.e. are assigned to bin 0 and 1, respectively. Starting from the third bin, bin i contains 2^{i-1} entries including the spectral components 2^{i-1} up to $2^{i-1}-1$. Note that by this means, the preprocessed DFT components a_{32} up to a_{63} remain unconsidered. This choice relies on experimental results which showed superior clustering characteristics in comparison with the inclusion of the high-frequency components. Further note that these findings support the assumption that the importance of spectral components decreases with increasing energy content. In a final step, the MFCC-like vibration segment representation is established by substituting the set of frequency components of each bin by their respective average.

5.4. Experimental Results

5.4.1. Experimental Setup

The general experimental setup of embedding a variety of classifiers into the proposed Bayesian classification framework follows the one based on the support vector machine which was presented in the previous chapter. Hence, only the key differences between both experimental settings are detailed in the following.

Choosing the Transition Matrix Estimation Scheme Instead of considering various transition matrix estimation schemes individually, the selection of the latter was integrated into the MSPRT threshold selection technique. That is, given a variety of generated test paths, this procedure does not only provide an adequate MSPRT threshold but also the optimal transition matrix estimation technique along with its corresponding parameter. Thereby, the set of transition matrix estimation scheme candidates comprised the constant, adaptive window size, and constant window size approaches where the latter employed a window of size w, with $w \in \{2,4,8,16,32\}$.

Vibration Segment Fingerprint In contrast to the previous chapter where only the DFT amplitude spectrum has been employed for the representation of the vibration signal, the following experiments also consider the MFCC-like vibration segment fingerprint.

Model Selection The optimal hyperparameters for the various classification schemes have been found using grid-search. The following listing provides the set of all free parameters along with lower and upper bounds which were defined prior to training the respective model. For an explanation of the hyperparameters, it is referred to the technical section given above.

- **Support Vector Machine**
  - *kernel K* — radial basis function kernel
  - *gamma* γ — 10^{-4} up to 10^{1}, $\log_{10}$ sampling, 32 samples
  - *cost C* — 10^{-2} up to 10^{2}, $\log_{10}$ sampling, 16 samples
- **Multi-Layer Perceptron**
  - *optimizer* — Quasi-Newton optimization
  - *valRatio* — 10%
  - *nhidden* — 8 up to 24, linear sampling, 5 samples

- **k-Nearest Neighbor**
  - *k* — 1 up to 32, linear sampling, 32 samples
- **Probabilistic Neural Network**
  - *sigma* σ — 10^{-3} up to 10^{2}, $\log_{10}$ sampling, 64 samples
- **Gaussian Mixture Model**
  - none
- **Random Forest**
  - n_T — 25 up to 250, linear sampling, 10 samples
  - *posterior estimation* — voting or voting + Laplace approximation
- **Random Ferns**
  - n_F — 64 up to 640, linear sampling, 10 samples
  - n_B — 2^2 up to 2^4, $\log_2$ sampling, 4 samples
  - n_P — 25 up to 250, linear sampling, 10 samples
  - *liblinear solver* — multi-class SVM by Crammer and Singer [CS02]
  - *r* — 100
  - *nodeConstruction* — entropy (applies to the random linear classifier binary test)
  - *binTest* — compare or linear discr. or liblinear or LDA
  - *prior* — Dirichlet prior
  - *posterior estimation* — voting or average leaf posterior

5.4.2. Results and Discussion

Due to the multitude of aspects to investigate, the section comprising the results and discussion is subdivided into several entities. The first investigation addresses the choice of the binary test and the posterior probability calculation scheme with regard to the random ferns classifier. The second section provides a more detailed discussion of class posterior derivation in the context of the random forest technique and the random ferns classifier. Given the best parameters for the random forest and random ferns approaches by means of the evaluation of the latter two experiments, these ground surface estimation techniques are compared with a set of classifiers previously adopted in the context of supervised terrain classification. All presented results relate to the evaluation of the 3 and 5 classes experiments with regard to both feature extraction schemes, the DFT amplitude spectrum approach (DAS) and Mel-frequency cepstral coefficients-like preprocessing (MFCC).

Evaluating the Binary Tests for the Random Ferns Approach

Table 5.1 shows the classification performance of the random ferns approach with regard to the 3 and 5 classes experiments when varying the underlying binary test, the chosen preprocessing technique, and the posterior probability estimation scheme. Here, the presented classification quality measure denotes the mean true positive rate obtained after averaging the results of the single observation approach over all three velocity profiles.
From Table 5.1 it can be derived that the Fisher-LDA provides the most appropriate binary test followed by the liblinear, the LDA, and the comparison techniques. The 3 classes experiments using DAS preprocessing represent the sole exception of the given ranking as the liblinear binary test outperforms the Fisher-LDA test in this case.
Concerning the posterior probability estimation technique, it can be stated that the voting

Table 5.1.: Classification performance in terms of the true positive rate [%] of the random ferns classifier for the 3 and 5 classes experiments with respect to varying preprocessing techniques (preproc) and binary tests when applying the unfiltered single observation classification scheme.

#classes	preproc	binary test							
		compare		LDA		liblinear		Fisher-LDA	
		voting	post	voting	post	voting	post	voting	post
3	DAS	34.3	58.0	34.6	64.9	81.9	**88.3**	87.7	87.1
	MFCC	40.0	77.2	37.6	80.4	77.1	81.5	82.5	**84.5**
5	DAS	17.7	72.6	18.9	59.1	75.3	78.7	78.1	**79.4**
	MFCC	16.7	71.1	21.1	44.9	69.9	75.6	74.9	**77.4**

method performs significantly worse than the average class posterior technique in all considered cases but the 3 classes experiment in conjunction with the DFT amplitude spectrum signal processing strategy. Note that when comparing both posterior probability estimation techniques, the decrease in the classification performance is more distinct for the comparison and LDA binary tests as opposed to the liblinear and Fisher LDA approaches.
To keep the presentation of the results clear, the following discussion focuses on the most appropriate binary test only which is represented by the Fisher LDA approach. In contrast, as the average true positive rate does not differ significantly with regard to the voting and average posterior probability estimation techniques, the discussion of both techniques is continued in the next subsection.

A Comparison of varying Terrain Classification Schemes

The presentation of the results of the applied terrain classification approaches is based on the one of the previous chapter. That is, the classification performance is provided for both the single observation approach and the temporally coherent Bayes filter technique. Thereby, the prediction quality is given in terms of the average true positive rate.
Regarding the 3 classes experiments, the TPR is shown for each individual velocity profile as well as for the average determined over all considered velocity profiles. The tables which refer to the results of the 5 classes experiments denote the classification performance of both the single observation approach and the Bayes filter technique using various travel distances which range from 2 m up to 32 m. Furthermore, the mean TPR is presented which is obtained by averaging all results but the ones of the single observation approach in a column-wise manner. For the random forest and random ferns classifiers, each column of Tables 5.2 up to 5.3 denote the classification performance with respect to a certain posterior probability determination scheme. Concerning the random ferns technique, this includes class posterior estimation methods based on voting (vot) and voting applied with the Laplace correction (lap). As for the posterior probability determination techniques of the random ferns classifier, results for the voting method (vot) and the average class posterior strategy (prob) are presented. Finally, it has to be noted that all the above-mentioned parameter settings have been considered with respect to the DAS and MFCC-like feature extraction techniques.

Results of the Random Forest Classifier At first, the results of the random forest and random ferns approaches are discussed. Note that in this context, the main focus is on the evaluation of differing classifier properties rather than the effects of temporal coherences, varying preprocessing schemes, or differing velocity profiles.
Table 5.2 shows the classification performance of the random forest classifier for the 3 classes experiment. Referring to the DAS preprocessing scheme and the mean true positive rates obtained after averaging over all velocity profiles, the voting technique is superior to the combined voting/Laplace approximation method (86.7% vs. 85.4% for the single observation approach and 96.1% vs. 94.5% for the Bayes filter technique). When adopting the MFCC-like preprocessing strategy on the other hand, the voting/Laplace approximation method outperforms the voting-only technique: 85.1% vs. 85.4% (single observation) and 93.9% vs. 94.7% (Bayes filtering), respectively.
For the 5 classes experiments, the results reveal a more general trend with respect to the class posterior estimation technique. Considering the outcomes obtained after averaging the TPR of all but the single observation experiment, the voting approach yields larger true positive rates than the voting/Laplace approximation technique: 84.5% vs. 82.7% for DAS preprocessing and 83.0% vs. 81.5% for the MFCC-like preprocessing scheme.
To conclude, due to the good performance of the random forest approach related to the 5 classes experiments, the following discussion only considers the voting posterior probability estimation technique.

Results of the Random Ferns Classifier For evaluating the results of the random ferns classifier with respect to the 3 classes experiments, the mean true positive rates are considered. The latter are obtained after averaging the classification outcomes over all 3 velocity profiles. From Table 5.2 it can be derived that the voting strategy represents the most effective class posterior estimation technique. On the other hand, when varying the class posterior determination technique for the MFCC-like preprocessing scheme, the average class posterior approach outperforms the voting-based strategy.
Referring to the 5 classes experiments and the mean true positive rates which represent the average TPR over all but the single observation experiment, the results with respect to a certain posterior probability estimation procedure have to be discussed separately in dependence of the chosen feature extraction technique: As for the DFT amplitude spectrum-based preprocessing method, the voting and average posterior probability approaches perform similarly. In contrast, both techniques yield differing true positive rates where the average posterior probability method significantly outperforms the voting strategy.
As shown during the discussion of the random ferns approach, there is no general trend of a certain posterior probability determination scheme to outperform all other techniques in the context of the 3 classes experiments. Hence, the choice of the posterior estimation method to be considered in the following discussion is based on the 5 classes experiments. Here, the outcomes suggest the use of the average posterior probability technique for estimating the class posteriors.

Table 5.2.: The true positive rates [%] of the random forest and random ferns classifiers for the 3 classes experiments with respect to varying preprocessing schemes (preproc), velocity profiles (vel), and posterior derivation techniques (the voting (vot) and the combined voting/Laplace approximation (lap) techniques and the voting (vot) and the average class posterior approaches, respectively) when applying the unfiltered single observation (SO) and Bayes filtered (BAY) classification schemes.

preproc	vel	approach	random forest		random ferns	
			vot	lap	vot	post
DAS	0.2 m/s	SO	**83.2**	83.1	**83.3**	82.0
		BAY	**96.0**	90.9	**96.5**	93.5
	0.4 m/s	SO	87.9	**88.6**	**87.5**	87.2
		BAY	96.6	**97.1**	**96.7**	96.4
	0.6 m/s	SO	**88.9**	84.6	**91.1**	90.1
		BAY	**95.6**	**95.6**	**97.8**	96.4
	average	SO	**86.7**	85.4	**87.3**	86.4
		BAY	**96.1**	94.5	**97.0**	95.4
MFCC	0.2 m/s	SO	83.1	**84.5**	80.0	**84.4**
		BAY	91.1	**91.9**	**96.5**	95.9
	0.4 m/s	SO	**90.3**	89.7	**85.7**	85.2
		BAY	96.4	**97.6**	95.1	**98.3**
	0.6 m/s	SO	81.7	**82.1**	**79.7**	77.2
		BAY	94.1	**94.5**	**94.6**	94.2
	average	SO	85.1	**85.4**	81.8	**82.3**
		BAY	93.9	**94.7**	95.4	**96.1**

Table 5.3.: The true positive rates [%] of the random forest and random ferns classifiers for the 5 classes experiments when averaging the outcomes over the complete set of velocity profiles. For an explanation of abbreviations, the latter table is referenced.

preproc	dist	random forest		random ferns	
		vot	lap	vot	post
DAS	SO	**81.1**	**81.1**	78.8	**79.0**
	2	**78.5**	75.6	70.5	**72.5**
	4	**83.4**	81.5	77.5	**78.5**
	8	**85.9**	84.1	**81.3**	**81.3**
	16	**87.1**	86.1	**83.3**	82.6
	32	**87.8**	87.1	**84.1**	83.3
	var	**84.1**	81.6	**79.2**	79.1
	average	**84.5**	82.7	79.3	**79.5**
MFCC	SO	78.6	**78.9**	75.6	**77.2**
	2	**78.6**	75.8	63.7	**71.5**
	4	**82.3**	79.0	71.7	**76.3**
	8	**84.3**	83.0	76.1	**79.4**
	16	**85.1**	84.6	78.5	**81.0**
	32	**85.6**	85.5	80.0	**81.4**
	var	**82.3**	81.0	73.3	**77.5**
	average	**83.0**	81.5	73.8	**77.9**

Table 5.4.: Classification performance of the single observation (SO) and Bayes filtering schemes (BAY) in terms of the true positive rate [%] for the 3 classes experiments when varying the preprocessing scheme (preproc), velocity profile (vel.), and underlying classifier.

preproc	vel	approach	classifier						
			svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	0.2 m/s	SO	**88.4**	82.5	73.9	79.1	77.0	83.2	82.0
		BAY	95.6	86.6	73.9	92.4	80.7	**96.0**	94.3
	0.4 m/s	SO	86.8	87.1	82.0	85.4	51.9	**87.9**	87.2
		BAY	**97.5**	91.0	83.5	87.9	53.6	96.6	96.5
	0.6 m/s	SO	**92.1**	85.0	79.7	84.4	33.3	88.9	90.1
		BAY	**98.1**	86.4	79.7	90.0	33.3	95.6	96.4
	average	SO	**89.1**	84.9	78.5	83.0	54.1	86.7	86.4
		BAY	**97.1**	88.0	79.0	90.1	55.9	96.1	95.8
MFCC	0.2 m/s	SO	**85.1**	82.4	82.4	84.0	82.0	83.1	84.4
		BAY	92.4	87.0	88.5	**96.0**	89.0	91.1	95.9
	0.4 m/s	SO	85.4	84.4	82.6	82.7	88.1	**90.3**	85.2
		BAY	96.2	85.7	97.3	85.9	96.3	96.4	**98.3**
	0.6 m/s	SO	79.9	70.9	70.9	79.8	78.0	**81.7**	77.2
		BAY	**96.5**	72.4	80.1	94.0	94.8	94.1	94.2
	average	SO	83.5	79.2	78.6	82.2	82.7	**85.1**	82.3
		BAY	95.0	81.7	88.6	91.9	93.4	93.9	**96.1**

Embedding various Classification Techniques into the Bayesian Prediction Framework The discussion now focuses on a comparison of terrain classifiers with respect to their ability to exploit temporal dependencies within the ground surface prediction framework. Furthermore, the performance of both preprocessing schemes is evaluated: the DAS descriptor comprising a set of 64 spectral coefficients and the compact MFCC-like descriptor which consists of only 6 components.

Table 5.4 shows the classification performance of the single observation and Bayes filter approaches with regard to the 3 classes experiments in dependence of the chosen classifier, velocity profile, and preprocessing scheme. To facilitate the following discussion, the mean prediction quality is provided as well which denotes the average true positive rate over all considered velocity profiles. Note that the latter measure is depicted in Figure 5.2. For the single observation approach, it shows that the SVM classifier performs best followed by the random forest, random ferns, multi-layer perceptron, probabilistic neural network, and k-nearest neighbor classification techniques. The Gaussian mixture model yields the worst classification performance being not able to generate an appropriate model of the underlying data. Note that the same ranking can be observed for the temporally coherent Bayes-filter approach. When applying the latter, the classification performance is always larger in comparison with the single observation approach, where the SVM, PNN, random forest, and random ferns benefit the most from the inclusion of temporal dependencies. Here, the true positive rates increase by 8.0%, 7.1%, 9.4%, and 9.4%, respectively. Averaged over the complete set of classifiers, the Bayes filter approach results in an absolute improvement of 5.6% TPR. Referring to the MFCC-like preprocessing scheme, an average increase of 9.6% TPR is obtained which turns out to be larger in comparison with the standard DAS feature extraction approach. Note that all but the MLP classifier raise the classification performance by at least 8.8% and 13.8% at maximum.

The evaluation of both preprocessing schemes yields a differing ranking with respect to the single observation and Bayes filter approaches. For the former, there are several classification techniques such as the PNN, GMM, random forest, and random ferns classifiers which yield similar true positive rates in comparison with the SVM approach where the random forest classifier even outperforms the support vector machine on average. Another ranking is established for the Bayes filter technique in conjunction with the application of the MFCC-like descriptor. Note, however, that the general trend with regard to the classification performance for the varying methods is maintained: while the SVM, PNN, GMM, random forest, and random ferns classifiers provide true positive rates of more than 91.1%, the multi-layer perceptron and k-nearest neighbor classifiers increase the classification error by 14.4% and 7.5%, respectively. These findings denote the decrease in the TPR with respect to the best classification technique which is represented by the random forest approach. Further note the good performance of the GMM classifier. In comparison with the DAS descriptor, the more compact feature vector results in an increase in the true positive rate of 28.6% and 37.5% referring to the single observation and Bayes filter approaches. This is an important finding as the clustering technique of Chapter 6 is based on Gaussian mixture models and hence benefits from the novel representation of acceleration signals. The superiority of the MFCC-like descriptor in the context of Bayesian filtering can also be observed for other classifiers such as the KNN, PNN, and random ferns techniques. Note, however, that the increase in the true positive rate for the latter two classification techniques is generally not as significant as for the GMM technique.

Concerning the 5 classes experiments and the DAS descriptor, the SVM classifier yields the largest true positive rates. This statements holds true for both the single observation approach (84.3%) and the Bayes filter technique (88.7%) on average. With regard to the Bayes filter scheme and the averaged classification performance measure, the remaining classifiers can be ranked in decreasing order of classification performance as follows: First, the random forest technique (84.5%), then random ferns (80.2%), probabilistic neural networks (79.9%), multi-layer perceptrons (79.6%), the k-nearest neighbor technique (79.4%), and the Gaussian mixture model classifier (76.5%). The respective ranking related to the single observation approach is nearly identical with the only exception of the MLP and PNN classifiers which exchange their positions.
Comparing the increase in the true positive rates of the single observation approach and the averaged ones of the temporally coherent technique, an average improvement of 3.2% TPR is obtained. On the other hand, the average difference of the classification performance between the single observation technique and the Bayes filter method at a travel distance of 32 m is 5.0%. In general, the findings that the true positive rate rises with an increase of the travel distance are valid for each classifier. Already at a travel distance of 2 m, the temporally coherent approach yields a better classification performance than the single observation technique. The only exceptions occur when adopting the random forest and random ferns methods. Here, the Bayes filter approach outperforms the single observation method starting at a travel distance of 4 m.
On the other hand, the random forest approach does not suffer from a decrease in the classification performance when adopting the MFCC-like feature extraction strategy. Here, an increase of the TPR is obtained already at a travel distance of 2 m. This is in contrast to the random ferns classifier where the terrain classification scheme benefits from temporal dependencies starting at a travel distance of 4 m. The general trend of a rising TPR with increasing travel distance is confirmed for the 5 classes experiments as well. In comparison with the single observation approach, the SVM and GMM classifiers benefit the most from the inclusion of temporal coher-

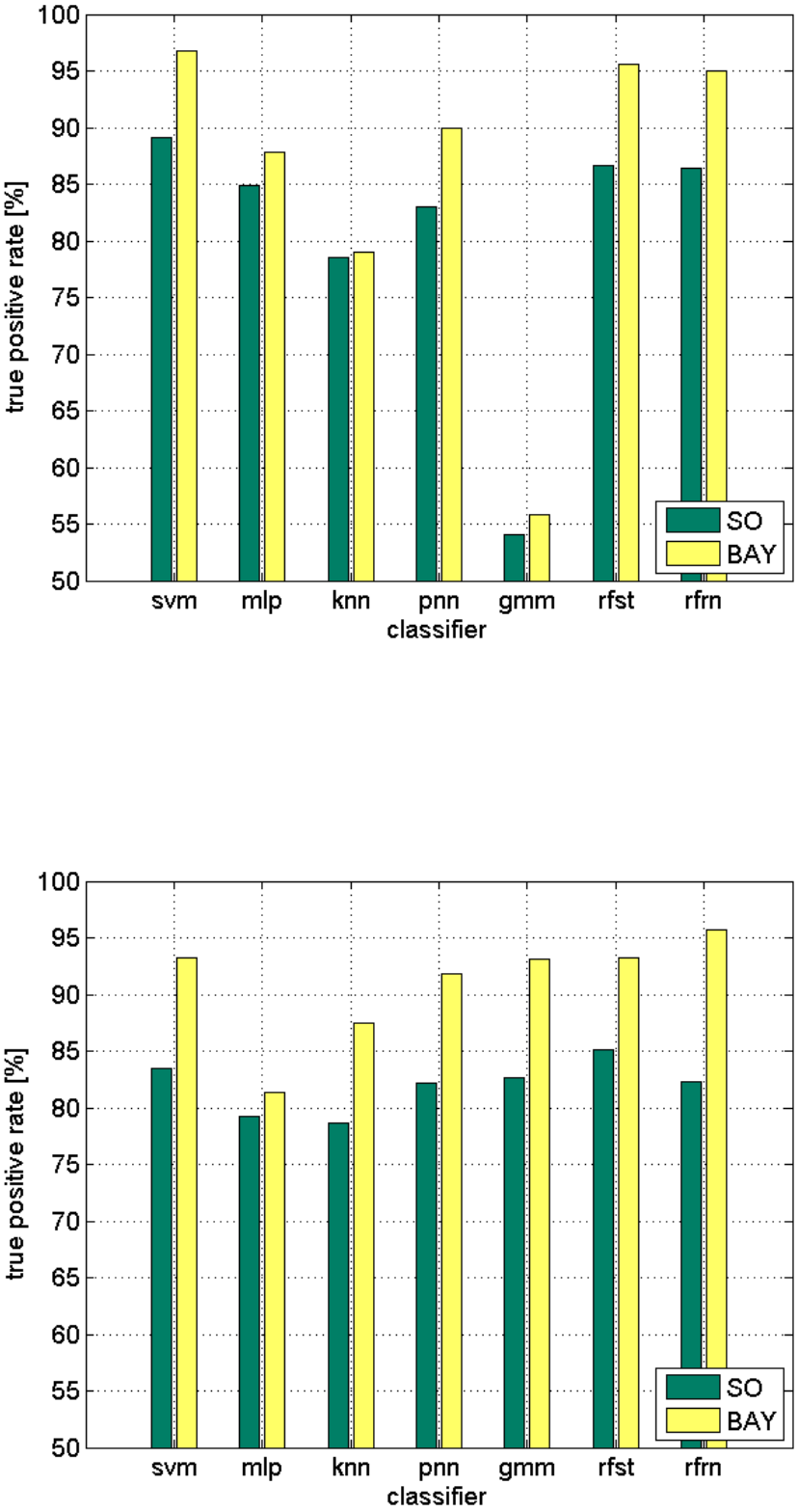


Figure 5.2.: The visualization of the true positive rates [%] for the 3 classes experiments and (a) the DAS and (b) MFCC-like feature extraction schemes with respect to varying classifiers when averaging the outcomes over the complete set of velocity profiles.

Table 5.5.: Classification performance in terms of the true positive rate [%] of the single observation (SO) and Bayes filtering schemes for the 5 classes experiments with respect to varying preprocessing schemes (preproc), travel distances (dist.), and terrain classifiers when averaging the outcomes over all velocity profiles.

preproc	dist	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	SO	**84.3**	77.3	75.5	75.7	73.8	81.1	79.0
	2	**85.4**	78.8	75.7	77.3	75.6	78.5	74.8
	4	**87.5**	79.4	78.4	79.3	76.3	83.4	79.2
	8	**89.4**	79.8	80.2	80.5	76.6	85.9	81.5
	16	**90.4**	80.0	81.2	81.1	76.9	87.1	82.6
	32	**90.9**	80.0	81.5	81.4	77.0	87.8	83.2
	var	**88.4**	79.6	79.2	79.5	76.5	84.1	79.8
	average	**88.7**	79.6	79.4	79.9	76.5	84.5	80.2
MFCC	SO	78.7	75.8	78.4	76.3	**79.3**	78.6	77.2
	2	**80.2**	77.8	79.4	77.9	80.1	78.6	74.3
	4	82.2	79.7	**82.7**	79.8	82.6	82.3	78.0
	8	84.3	80.7	**84.8**	81.3	84.3	84.3	79.9
	16	85.2	81.3	**85.6**	82.0	85.1	85.1	81.0
	32	85.7	81.5	**86.3**	82.3	85.5	85.6	81.2
	var	**83.4**	79.9	83.0	80.3	83.3	82.3	78.5
	average	83.5	80.2	**83.6**	80.6	83.5	83.0	78.8

ences providing a maximum absolute increase of the TPR of 7.0%, respectively. Averaged over all applied classifiers, the classification error decreases by 6.3% when comparing the results of the single observation technique with the ones of the temporally filtered classification scheme at a travel distance of 32 m. Referring to the mean results obtained after averaging the TPR over all but the single observation technique, the KNN classifier yields the best classification performance (83.5%). Similar findings can be inferred for the SVM, GMM, and random forest techniques where the classification error is increased only insignificantly by at most 0.6%. The remaining approaches yield worse but still acceptable true positive rates which deviate from the maximum one by at most 4.8%. Noticeable is the good performance of the GMM approach in conjunction with the MFCC-like preprocessing scheme compared to the higher dimensional DAS descriptor. Since the same characteristics can be observed for the 3 classes experiments, the MFCC-like feature extraction scheme is an adequate choice when the underlying model is represented by a Gaussian mixture. Yet, as shown in Table 5.5, several other classification schemes such as the MLP, KNN, and PNN techniques benefit from a more compact representation of acceleration signals.

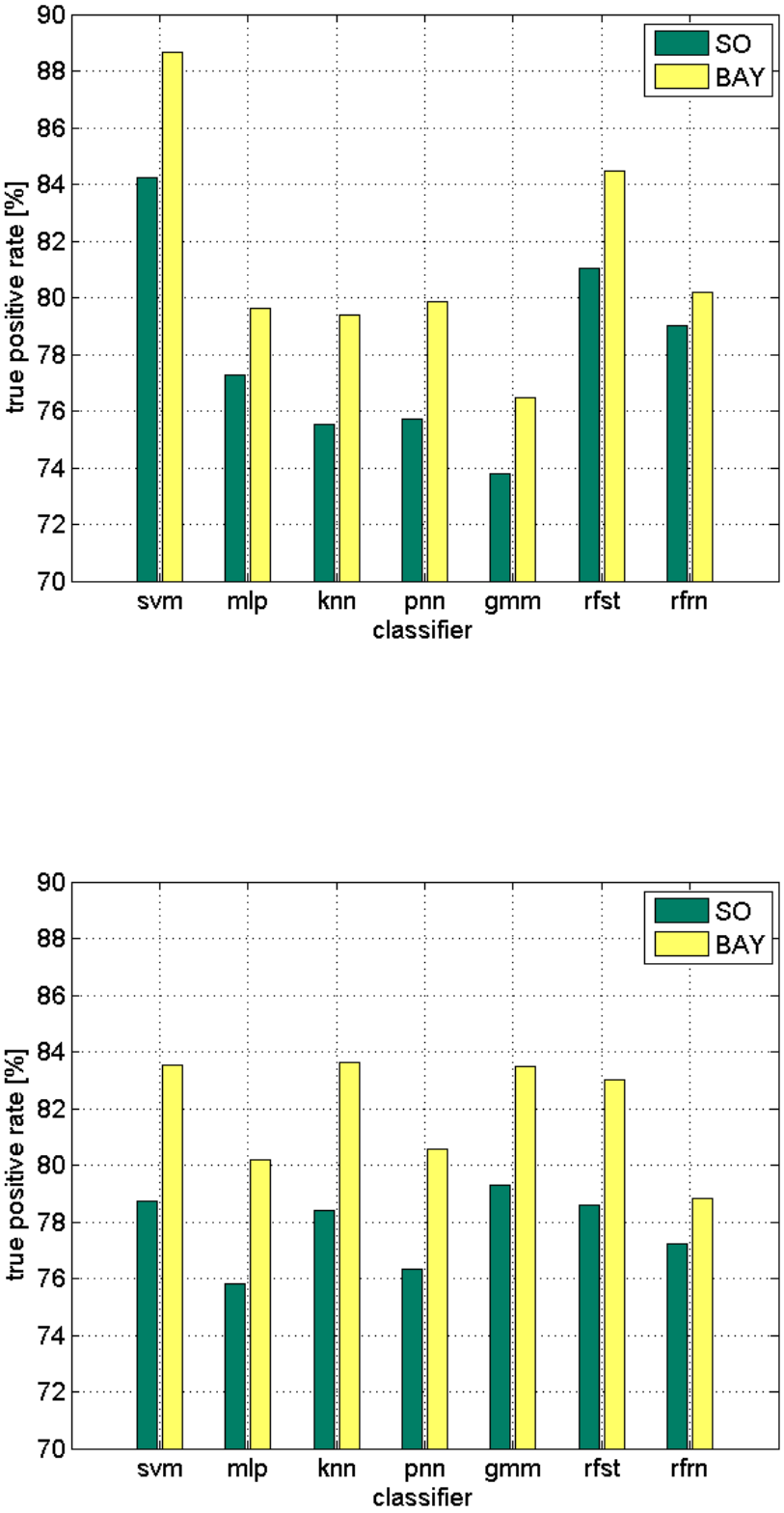


Figure 5.3.: The visualization of the true positive rates [%] for the 5 classes experiments and (a) the DAS and (b) MFCC-like feature extraction schemes with respect to varying classifiers when averaging the outcomes over the complete set of velocity profiles and travel distances.

Investigating other Characteristics of the Adopted Classifiers Classifier selection should be handled with care, since each approach has different characteristics. KNNs and PNNs belong to the class of lazy learning techniques. That is, all computations are delayed until a prediction query is requested. On the one hand, this renders a time-demanding training phase unnecessary which is advantageous if the underlying phenomenon changes frequently. On the other hand, all patterns have to be available at run-time which might pose a problem if storage is limited. Given that the acquired training set consists of n samples, storage requirements are $O(n \cdot d)$, where d is dimensionality of a training instance. Furthermore, if the computing capacity is constrained in the recall phase, the desired prediction frequency might not be accomplished due to a large set of training patterns. For example, when using the KNN classifier, a naïve approach involves $O(n)$ distance calculations to determine the k-nearest-neighbors. Although accelerating data structures like M-trees [Yia93] exist, high-dimensional nearest-neighbor search is known to be a non-trivial task, suffering from the curse of dimensionality. To decrease the effects of the latter, the MFCC-like preprocessing scheme can be employed reducing the feature set size from 64 to 6 dimensions. As shown in Tables 5.6(a)-5.7(c), the application of the MFCC-like feature descriptor does not only result in an increase of the TPR for several classifiers but also decreases the model training and testing time as well as the model size. This renders the MFCC-like feature extraction scheme attractive for mobile robots with low computational capacities and storage resources.
MLP and SVM classifiers typically provide compact models, resulting in a fast prediction performance (cf., e.g. [SDK+11]). The models generated by the support vector machine, however, are larger in comparison with the ones of the multi-layer perceptron. This is because the SVM model is not sparse, i.e. too many data points are defined as support vectors. To overcome this issue, model complexity reduction schemes can be applied which significantly decrease the number of support vectors while conserving the classification performance. One example of SVM complexity reduction strategies is the approach of Downs et al. [DGM02] which has also been successfully adopted in the context of engine state prediction [KZ09b]. With regard to training, multi-layer perceptrons are computationally much more demanding compared to support vector machines. Although both methods iteratively try to minimize a given error function, the minimization process in the context of support vector machines is based on solving a convex optimization problem. For the latter, an efficient approach denoted as sequential minimal optimization [Pla98] exists in contrast to the standard optimization techniques used for training neural networks (e.g. [CKL96]). Yet, it has to be noted that the time spent on choosing a classifier with a good generalization behavior is significantly increased by the model selection process which has to consider a sufficiently large set of candidate model parameter settings.
With regard to the classification performance, the random forest approach yields similar results in comparison with the SVM classifier. This is remarkable, since the latter technique represents the best terrain classification technique in the context of DAS preprocessing. Hence, a separate discussion is rendered necessary which focuses on other important properties of both classifiers and aims at facilitating the choice of a certain technique with respect to a given application domain. Averaging the results over the 3 and 5 classes experiments and both applied preprocessing schemes, the model generation process of the random forest approach takes 2.6 times longer than the one of the SVM classifier. Model evaluation, however, is 40% faster when the former classification technique is employed. This renders the random forest technique appropriate when run-time complexity becomes an issue. Finally, the memory requirements of the random forest approach exceed the one of the SVM classifier by one magnitude. Note, however, that storage requirements are significantly reduced as a large fraction of the consumed memory is required as auxiliary structure when determining the class posteriors. These

auxilary structures are calculated once at run-time and thus do not have to be stored on disc. Moreover, lossless compression schemes [KFDB07] can be applied to the model, enabling a further reduction of the required disc space.
Considering memory requirements, the random ferns approach generates models which occupy 3.5 times more space in memory as opposed to the random forest classifier. Furthermore, when comparing the other two characteristics, the duration of the random ferns model generation process exceeds the one of the random forest classifier by a factor of 1.7 where the duration of the model evaluation step differs by a magnitude. Summarizing these characteristics, the application of the random ferns technique is inferior to the random forest approach. Note, however, that the random ferns classifier yielded appropriate results in the context of the 3 classes experiments. Furthermore, it is characterized by an efficient implementation whose core system can be represented by approximately 20 lines of Matlab code. Finally, since most of the involved calculations can be expressed in terms of scalar products, the Matlab code is easily portable to graphical processing units.
Tables 5.6(a) up to 5.7(c) summarize the key characteristics of the proposed classifiers for the 3 and 5 classes experiments: for the best classification model, both the training time using data contained in one fold of a 5-fold cross validation scheme and the respective memory requirements (measured in kB) are presented. Further, Tables 5.6(b) and 5.7(b) show the average testing time for a single query. All run-time analyses were performed on a Pentium D 3.0 GHz desktop PC. For the storage considerations, each floating point number was represented as *double*, each 8 bytes in size.

Table 5.6.: (a) Model training times [s], (b) model evaluation times [ms], and (c) model sizes [kB] for varying classifiers with respect to the 3 classes experiments and the DAS and MFCC-like preprocessing schemes. These results reflect the outcomes obtained for the model with the best classification performance when averaging over all 5 folds of the 5-fold cross-validation technique.

(a)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	1.943	124.084	**0.035**	0.057	0.125	4.462	4.707
MFCC	0.767	21.045	**0.023**	0.027	0.059	1.169	3.892

(b)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	0.460	**0.025**	2.189	0.313	0.158	0.155	3.238
MFCC	0.141	**0.019**	0.535	0.227	0.069	0.102	2.390

(c)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	545.03	**11.46**	465.34	458.96	162.90	9341.91	1027.57
MFCC	45.02	2.86	50.58	44.20	**2.04**	6640.59	44901.42

Table 5.7.: (a) Model training times [s], (b) model evaluation times [ms], and (c) model sizes [kB] for varying classifiers with respect to the 5 classes experiments and the DAS and MFCC-like preprocessing schemes. These results reflect the outcomes obtained for the model with the best classification performance when averaging over all 5 folds of the 5-fold cross-validation technique.

(a)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	1.213	78.678	**0.075**	0.204	0.372	4.131	6.109
MFCC	0.435	8.856	**0.055**	0.208	0.538	1.546	4.504

(b)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	0.390	**0.086**	1.684	0.480	0.524	0.214	3.031
MFCC	0.160	**0.096**	0.662	0.280	0.680	0.230	4.634

(c)

preproc	classifier						
	svm	mlp	knn	pnn	gmm	rfst	rfrn
DAS	340.94	**11.17**	251.72	248.39	97.89	5032.30	14598.70
MFCC	23.16	2.80	27.58	24.24	**1.37**	4311.17	26908.32

5.5. Conclusion

The work in this chapter focused on the integration of various classifiers into the temporally coherent classification framework. This is possible by methods of calculating the required posterior probability estimates for the SVM classifier, the multi-layer perceptron, the k-nearest neighbor approach, probabilistic neural networks, the Gaussian mixture model classifier, random forests and random ferns. A thorough comparison including all mentioned classification schemes revealed that all techniques benefit from the inclusion of temporal coherences. Depending on the number of classes to discriminate, the selected preprocessing technique, and the length of the path which a robot has to navigate before the terrain type changes, an increase in the true positive rate by 13.8% at maximum can be observed. Providing other characteristics of each classifier such as the time spent on training and testing the respective model as well as memory requirements facilitates the choice of an appropriate terrain classification scheme in a variety of application domains.
As a further contribution, two classifiers have been introduced in the context of ground surface estimation: the random forest and the random ferns approaches. For both techniques, the problem of choosing an adequate posterior probability estimation strategy has been addressed. While random forests yield the best classification performance in conjunction with a voting scheme, the evaluation of the random ferns classification results advises an averaging process over individual posterior estimates.
Concerning the classification performance, the random forest technique achieves similar true positive rates as the SVM approach. The evaluation of the generated random forest classifier, however, turns out to be 40% faster than the one of the SVM method. The random ferns classifier, on the other hand, is characterized by a good performance with respect to the 3 classes experiments, an efficient implementation, and a straightforward portability on programmable graphics hardware.
The final contribution of this chapter is the introduction of a novel preprocessing scheme which adopts the basic elements of the Mel-frequency cepstral coefficients feature extraction scheme. Its application significantly decreases the component count from 64 to 6 entries while remaining and even improving the classification performance of various classifiers. Especially the Gaussian mixture model-based terrain estimator benefits from the applied dimensionality reduction. The resulting classification models outperform the ones which have been trained using the 64 dimensional feature representation. These findings are important, since the techniques presented in the following chapters make intensively use of Gaussian mixtures.

6. Markov Random Field-based Clustering of Vibration Data

6.1. Introduction

Classical clustering algorithms such as k-means or mixture-based clustering are based on the independent processing of observations. Hence, relationships between succeeding observations cannot be incorporated into the clustering process. Note, however, that measurements from the same terrain class tend to cluster together and thus introduce temporal constraints, which can be exploited by the clustering model. Despite this additional information, only Giguere et al. [GD08] adopted a temporally coherent terrain clustering technique. In their approach, they established a cost function to find an optimal clustering model with respect to its parameter set. The applied cost function also incorporates temporal dependencies of vibration signatures as it constrains the posterior distributions of consecutive terrain estimates to be similar. Later, Giguere et al. extended their work to be applied to two-dimensional data [GDP$^+$09]. Therefore, they divided the two-dimensional problem into two one-dimensional ones, considering the sequence of images in a horizontal and vertical manner. The main drawback of both approaches is the time spent on minimizing the proposed cost function. 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 [SBHHW04]: 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 [NC07]. 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 [LTJ04] 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 [GG87] and [Bes86], 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 denoising [Bar09] or image interpretation [MZ92]. The main advantage of MRF models is that they 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 image segmentation tasks where an image is partitioned into multiple descriptive segments this additional assumption increases the probability of achieving a better image analysis. This is because small sensor noise-related errors can be detected during the segmentation process and are subsequently removed in an efficient manner.
In this chapter, a model is described which exploits spatial dependencies among neighboring terrain patches instead of pixel arrays [KZ10b]: the applied MRF model [DVG07] assumes that the class labels of observations are generated by prior distributions which share similar pa-

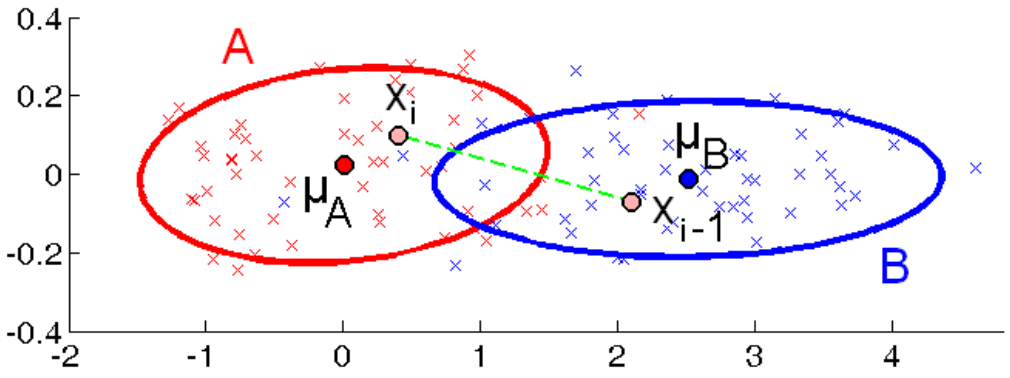


Figure 6.1.: Two overlapping classes arising from two normal distributions.

rameters for neighboring observations. As shown in the result section, 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, a general means of estimating this neighbor set size from the given data is derived.

6.2. Temporally Coherent Clustering

6.2.1. Motivation

Since temporally close measurements are likely to belong to the same cluster, the inclusion of this side information into the above-mentioned Gaussian mixture model-based clustering process is advised. To motivate a new EM formulation which takes temporal dependencies into account, Figure 6.1 is referred. There, two clusters A and B are depicted representing two distinct classes. Both measurements x_{i-1} and x_i belong to the same terrain class B, although, in this example, they are assigned to two different clusters. This is because the posterior $p(A|x_i)$ is larger than the posterior $p(B|x_i)$.
Following the E-step of the EM algorithm by applying (3.21), the new cluster centers μ_A and μ_B are determined by the weighted mean of all data points: the larger the posteriors $p(A|x_l), l \in [1,n]$, the larger is the contribution of each data point x_l to the new cluster center μ_A. The same applies to $p(B|x_l)$ and cluster center μ_B. In the provided example, the measurement x_i erroneously influences the calculation of the new cluster center μ_A more than the one of μ_B, although x_i belongs to cluster B. By increasing $p(B|x_i)$ and hence decreasing $p(A|x_i)$, since $p(A|x_i) = 1 - p(B|x_i)$, we finally obtain a solution which is closer to the underlying data distribution.
The modification of the posterior probabilities is achieved by considering not only the current measurement x_i for their calculation, $p(c = j|x_i)$, but a history of measurements $X = \{x_i\}_{i=1}^{n}$ instead: $p(c = j|X)$. When the robot acquires several recent measurements on the same terrain type, the probability of staying on this terrain type aggregates over several time steps. Measurements which intrinsically belong to another cluster but to the same terrain class j (cf. x_i in the given example) thus obtain a larger posterior probability for class j as their posterior is filtered with the ones of former measurements. To embed temporally filtered posterior probabilities into the EM algorithm described in Section 3.3.1, the posteriors $p(c = j|x_i)$ determined by (3.21)-(3.24) of the E-step are replaced by $p(c = j|X)$. Then, the model parameter estimates

are updated in the M-step using the modified posterior probabilities. Thereby, the temporal filtering of class posteriors can be applied using the adaptive Bayes filter approach of Section 4.4.1.
Experiments provided in [KZ10a] which focus on the performance of the temporally coherent clustering scheme demonstrate the effectiveness of the proposed technique. Note, however, that the likelihood function of 3.3.1 is not guaranteed to converge when replacing the estimated class posteriors by their filtered representatives. To overcome this issue, another approach based on Markov random fields is presented in the following section. This method is based on a similar key idea, as varying probability distributions are filtered over time for the inclusion of temporal dependencies.

6.2.2. Clustering Using Markov Random Fields

The Gaussian mixture model-based clustering model presented in Section 3.3.1 does not exploit temporal a priori dependencies between vibration segments, since the prior π_k is independent from the vibration segment index i. In the context of vibration segment clustering, 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, the generative model of Diplaros et al. [DVG07] was applied. 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.
The latter is enforced by including a penalty term $p(\pi|\beta)$ to the log-likelihood function of (3.20). This term penalizes neighboring pixels with different priors:

$$L_2 = L_1 - p(\pi|\beta).$$

To model the joint density over vibration segment priors, the following Besag approximation is used [Bes74, Bes86]:

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

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:

$$\pi_{N_i} = \sum_{l\in N_i, l\neq i} \lambda_{il}\pi_l. \tag{6.1}$$

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):

$$\log p(\pi_i|\pi_{N_i},\beta) = -\beta\left[\mathrm{KL}(\pi_i||\pi_{N_i}) + H(\pi_i)\right]. \tag{6.2}$$

$\mathrm{KL}(\pi_i||\pi_{N_i})$ denotes the Kullback-Leibler (KL) divergence between π_i and π_{N_i} and is defined as $\mathrm{KL}(\pi_i||\pi_{N_i}) = \sum_{j=1}^{k} \pi_{ij} \log \frac{\pi_{ij}}{\pi_{N_ij}}$. 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, the neighbors are constrained 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, these priors are not expected to be distributed uniformly.
Instead of optimizing (6.2) directly, the following approximation of the conditional density is employed [DVG07]:

$$\log p(\pi_i|\pi_{N_i},\beta,s_i) \approx -\beta\left[\mathrm{KL}(s_i||\pi)+\mathrm{KL}(s_i||\pi_{N_i})+H(s_i)\right], \tag{6.3}$$

where $\{s_i\}$ is an auxiliary set of distributions. Replacing (6.2) by (6.3) turns the constrained optimization problem into an efficient one which can be solved using the EM algorithm described below. 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 (6.3), a data-dependent penalty term $\mathscr{P}_d$ is introduced, which incorporates useful domain knowledge. Therefore, the posterior distributions are constrained to be similar among neighboring vibration segments and to be as informative as possible:

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

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):

$$\begin{aligned} L_2(\theta,\pi,s,q) = \quad & \textstyle\sum_i \quad \Big[\log \sum_j p(x_i|j,\theta)\pi_{ij} \hookleftarrow \\ & -\beta \quad [\mathrm{KL}(s_i||\pi_i)+\mathrm{KL}(s_i||\pi_{N_i})+H(s_i)] \hookleftarrow \\ & -0.5 \quad [\mathrm{KL}(q_i||p_i)+\mathrm{KL}(q_i||p_{N_i})+H(q_i)]\Big]. \end{aligned} \tag{6.4}$$

6.2.3. Estimating the Model Parameters

The parameter set $\{\theta,s,q\}$ is estimated using an EM algorithm which maximizes the energy of L_2 by coordinate ascent: in the E-step, θ and π are fixed and L_2 is maximized over s and q. In the M-step, s and q are fixed and L_2 is maximized over θ and π. Pseudo code for the EM algorithm is provided below. For a complete derivation of the respective formulas, it is referred to the work of Diplaros et al. [DVG07]. Note the similarity between the EM formulation for training the MRF-based generative model and the one for Gaussian mixtures (cf. Section 3.3.1). 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.

6.3. MRF-based Vibration Signature Clustering

6.3.1. Filtering Prior and Posterior Probabilities

The determination of the mixture distributions over the priors (π_{N_i}), the posteriors (p_{N_i}), and the auxiliary set (q_{N_i}) require the definition of both the mixing weights λ_{ij} and the neighborhood size. Note that the evaluation of (6.1) is equivalent to a convolution operation $\pi_{*j}\odot\lambda$, for each mixture component j. In this 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, modified versions (i.e., the center coefficient was set to zero) of a box, tent, quadratic, cubic, and a Gaussian filter were applied. Although relevant differences in the

Algorithm 2 The temporally constrained EM algorithm

1: Initialize the parameter vector θ (including the priors $\{\pi_i\}$) using the k-means algorithm (see Section 6.4.1).
2: **E-step**: Determine posterior probabilities p_i using the current estimates of θ and $\{\pi_i\}$:

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

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 L_2 using (6.4)
10: **if** convergence of L_2 **then**
11: terminate.
12: **else**
13: goto step 2.
14: **end if**

results were expected when adopting the varying filters, these differences did not prove to be statistically significant. Hence, only the results of the Gaussian filter are presented in the result section which tend to yield slightly better results in comparison with the other filters. Further, the standard deviation of the Gaussian filter was chosen to be equal to half of the neighbor set size.

6.3.2. Choosing an Appropriate Neighbor Set Size

The neighbor set size should be selected data-dependently: 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. 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, the following technique is proposed to estimate the neighbor set size from the acquired sensor data: first, a k-means clustering of the vibration signals is performed. This clustering provides a broad assignment of each vibration segment x_i to one of the terrain classes. Then the homogeneous neighbor size $\mathscr{S}$ is determined for each vibration segment x_i which is defined as:

$$\begin{aligned} \mathscr{S}(x_i) &= \arg\max_l \ \mathscr{T}(x_i, \{x_m\}),\ m \in [i-l, i+l] \backslash \{i\}, \\ \mathscr{T}(x_i, V) &= \begin{cases} 0, & \exists v_o \in V,\ v_o \text{ is an outlier w.r.t. } N_{\mu_j, \Sigma_j} \\ |V|, & \text{else} \end{cases} \end{aligned}$$

In other words, the homogeneous neighbor set size $\mathscr{S}$ of a vibration segment x_i is the maximum number of contiguous vibration segments which are located symmetrically around observation x_i and belong to the same terrain class j as vibration segment x_i with a certain probability. Since the terrain class j of x_i is represented by a multivariate Gaussian distribution with mean μ_{c_j} and covariance matrix Σ_{c_j}, the latter verification is rendered possible by using the Mahalanobis distance-based outlier test of [RL87], pp. 224. This test is based on the fact that the Mahalanobis distances between the instances of class j and the cluster center μ_j are approximately χ^2_d-distributed with d degrees of freedom [Krz88]. Here, d denotes the dimensionality of a vibration segment x_i. The test defines a given vibration segment v_o as outlier with respect to class c_j if the squared Mahalanobis distance between v_o and the cluster center μ_{c_j} 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, the homogeneous neighbor set size of the whole data set, $\overline{\mathscr{S}}$, is defined to be the mean neighbor set size averaged over all vibration segments x_i, $i \in [1,n]$, mapped to the next multiple of 2. For the last free parameter β, the suggestion of [DVG07] is followed, i.e. β is set to 0.5. Experimental results proved this choice to be valid.
Note that the proposed testing scheme does not only provide a means to estimate the frequency of terrain changes but also enables the partition of the data set in segments of similar terrain transition frequencies. That is, the data set can be recursively split until the variance of the individual neighbor set sizes contained in the smaller segments falls below a certain threshold. Then, each subsegment can be clustered separately whereas each clustering is performed with a more adequate neighbor set size.

6.4. Experimental Results

6.4.1. Experimental Setup

This section considers the remaining aspects of the GMM-based clustering and evaluation schemes which have been employed during the experiments. These include the vibration segment preprocessing approach, the GMM cluster center initialization technique, the cluster model evaluation criteria, and the path generation technique.

Vibration Segment Preprocessing As shown in the result section of Chapter 5, the Gaussian mixture model-based classifier favors a compact, low-dimensional representation of the 100-sample-sized vibration segment. The vibration segment representation 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, the possibility of getting stuck in a local minimum rises as well when applying the EM algorithm. Motivated by the success of the MFCC-like descriptor in the context of Bayes filtered terrain classification, this descriptor is now adopted to the problem of unsupervised classification.

Cluster Center Initialization In the context of cluster center initialization, the discrimination between the Gaussian mixture model technique and the Markov random field model approach has to be made. For GMM-based clustering, two techniques have been considered: random and k-means initialization. The former method randomly selects k instances from the data set as initial cluster centers and sets the starting values for the class prior probabilities to $1/k$. Using the k-means initialization technique, a preceding k-means clustering step has to be applied. The mixture model covariances are then 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.
Experimental results revealed that the latter two initialization schemes result in inadequate models representing inferior local optima of the likelihood criterion when applied to Markov random field-based clustering. Employing the model obtained from the GMM clustering process as initial estimate of the MRF clustering solution yielded significantly better results with respect to all quality criteria under consideration. Note, however, that the GMM model has to be modified to fit into the structure defined by the Markov field approach. This is because the latter technique comprises an individual prior on a per-observation basis rather than on a per-cluster basis. In this implementation, the GMM model is converted by assigning the GMM prior of cluster j to the jth component of the MRF observation prior.

Cluster Model Evaluation Criteria In contrast to the latter two chapters which based their evaluation on the true positive rate only, a total of three cluster model evaluation schemes has been considered: the true positive rate, the adjusted rand index and the adjusted mutual information criterion (cf. Section 3.3.2). By this means, a more general conclusion can be obtained examining the obtained results under various perspectives such as information theoretic, combinatoric, and descriptive aspects. Note that the respective tables and figures which show the outcomes when appying the adjusted rand index and the adjusted mutual information criterion are moved into the appendix to keep the presentation of the results clear.

Table 6.1.: Estimated filter size for the 3 classes experiments with respect to varying velocity profiles.

vel	estimated filter size
0.2 m/s	78.0
0.4 m/s	46.0
0.6 m/s	42.1

Path Generation Scheme The path generation procedure follows the one of the latter two chapters including the 3 classes experiments representing natural paths and the 5 classes experiments which are characterized by paths containing a varying amount of temporal coherence.
Since both the random and the *k*-means initialization schemes are likely to yield a different initial Gaussian mixture parameter set for each single path due to their random components, the EM algorithm might terminate with varying solutions as well. Thus, 50 reruns of each clustering have been applied with regard to a certain, i.e. constant, path using a novel random and *k*-means setup for each trial.

6.4.2. Results and Discussion

Results of the 3 Classes Experiments

Table 6.2 shows the clustering performance in terms of the true positive rate for the GMM and MRF approaches, random and *k*-means initialization schemes, and velocity profiles 1-3 (0.2-0.6 m/s). Further, the last four columns denote the true positive rates when averaging the results over all robot driving speeds. In comparison with the GMM clustering technique, the temporally coherent approach in conjunction with the *k*-means (random) initialization scheme yields a maximum absolute improvement of the TPR by 14.8% (13.7%), 9.3% (9.4%), and 13.8% (12.8%), for velocity profiles 1-3 and 12.4% (11.9%) on average. Thereby, using *k*-means for cluster center initialization always results in a larger clustering performance in comparison with the random technique.
Regarding the filter size parameter of the MRF model, Table 6.2 reveals that larger filter sizes yield a better clustering model. This is an expected behavior, since the natural paths of the 3 classes experiments comprise a large amount of temporal dependencies as they contain a small number of transitions only. Hence, large filter sizes are required to adequately exploit the contained temporal coherences.
As shown in Table 6.1 which present the estimated filter size for the respective paths, the filter size derivation technique is capable of correctly selecting a large filter size. As an effect, the obtained results when applying the MRF clustering approach along with the proposed filter size estimation scheme represent the superior ones or are at least close the best ones.
Finally, Figure 6.2 visualizes the clustering performance with respect to the true positive rates obtained when employing the *k*-means initialization technique and averaging the results over the three robot driving speeds.
Tables A.1 and A.2 (cf. Section A.1.1) show the results when the clustering performance measure is substituted by the adjusted rand index (ARI) and adjusted mutual information (AMI) quality measures. There, the same findings can be derived in comparison with the TPR quality measure with respect to varying velocity profiles, cluster center initialization, and the estimated filter sizes. Note, however, that the relative improvements of the MRF approach with regard to GMM clustering are significantly larger in comparison with the TPR performance measure.

Table 6.2.: True positive rate [%] for the 3 classes experiments with respect to varying velocity profiles, mixture model initialization schemes, and filter sizes when adopting the GMM-based and the temporally coherent MRF-based clustering techniques.

vel	init	approach	filter size					
			2	4	8	16	32	$\overline{\mathscr{S}}$
0.2 m/s	random	gmm	82.7	82.4	82.3	82.3	82.3	82.3
		mrf	91.4	92.1	93.8	95.1	96.0	**96.0**
	k-means	gmm	82.3	82.3	82.3	82.3	82.3	82.3
		mrf	91.7	93.2	94.8	96.1	97.0	**97.1**
0.4 m/s	random	gmm	87.2	86.2	87.1	85.9	87.0	87.0
		mrf	92.0	91.1	93.4	94.3	96.2	**96.4**
	k-means	gmm	87.7	87.7	87.7	87.7	87.7	87.7
		mrf	92.3	92.5	93.9	96.3	**97.0**	97.0
0.6 m/s	random	gmm	67.3	65.7	65.6	64.6	66.9	64.2
		mrf	73.5	72.2	73.5	74.0	**79.7**	76.4
	k-means	gmm	72.0	70.3	71.6	72.4	72.0	72.8
		mrf	79.9	78.8	84.4	**86.2**	85.1	85.0
average	random	gmm	79.0	78.1	78.3	77.6	78.7	77.8
		mrf	85.6	85.1	86.9	87.8	**90.6**	89.6
	k-means	gmm	80.7	80.1	80.5	80.8	80.7	80.9
		mrf	88.0	88.1	91.0	92.8	**93.1**	93.0

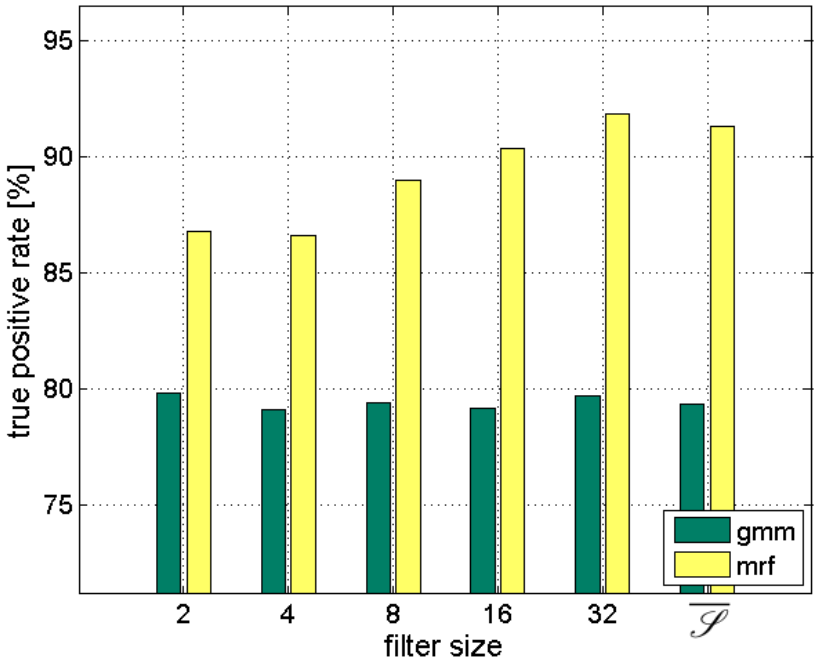


Figure 6.2.: True positive rate [%] for the 3 classes experiments and *k*-means model initialization with respect to varying filter sizes when averaging the outcomes over the complete set of velocity profiles.

Table 6.3.: Estimated filter size for the 5 classes experiments with respect to varying velocity profiles and travel distances.

vel	estimated filter size for distance					
	0 m	2 m	4 m	8 m	16 m	var
0.2 m/s	0.0	7.8	13.4	18.6	30.1	23.6
0.4 m/s	0.0	4.0	6.6	13.3	22.2	16.4
0.6 m/s	0.0	2.9	4.7	8.7	14.4	11.8

Results of the 5 Classes Experiments

Table 6.2 shows the results of the 5 classes experiments with respect to the true positive rate when the travel distance is systematically varied. Here, the paths employed in a certain experiment represent artificially generated ones providing a differing amount of temporal dependencies between varying experiments. For the reason of clarity, the clustering performance is averaged over all velocity profiles. Further, the average true positive rate over all travel distances is provided for each initialization scheme.
Considering the GMM and the MRF approaches, the latter always outperforms the former, starting at a travel distance of 2 m. The maximum increase of the true positive rate is 9.2% (9.4%) and 5.4% (9.2%) on average using k-means (random) model initialization. Whereas the best results of smaller travel distances are obtained for smaller filter sizes, larger filter sizes have to be chosen when increasing the travel distance. As the filter size is assumed to be a function of the present amount of temporal dependencies, this characteristic has been expected. At a travel distance of 0, which denotes the situation of a terrain transition after each measurement, only the adaptive filter size estimation technique yields adequate results. This is because in this case no temporal coherences are provided. Techniques, however, which rely on the presence of temporal dependencies between succeeding observations are likely to fail, since their key assumption is not met. As presented in Table 6.3, the adaptive filter size estimation technique is able to detect these cases and determines a filter size of 0. Note, that using this parameter for the Markov random field technique, the MRF clustering scheme turns into a standard GMM approach disregarding temporal coherences. Table 6.3 further reveals that the filter size is autonomously adjusted according to the present travel distance. This enables an adequate utilization of temporal dependencies by considering an appropriate number of neighboring observations.
Finally, in contrast to the 3 classes experiments, only small differences between the random and k-means initialization schemes can be observed.
Figure 6.3 depicts the increase in the true positive rate when using the MRF approach with respect to the GMM-based clustering technique. Here, the relationship between the travel distance, filter size and the obtained clustering performance is visually highlighted.
Analogous to the 3 classes experiments, Tables A.3 and A.4 and Figures A.3 and A.4 (cf. Section A.1.2) present the clustering performance when exchanging the TPR clustering quality measure by the adjusted rand index and the adjusted mutual information quality criterion. It shows that all performance measures yield similar characteristics with respect to the travel distance, filter size, and clustering quality relationship, varying cluster center initialization schemes and the adaptive filter size selection behavior.

Table 6.4.: True positive rate [%] for the 5 classes experiments with respect to varying velocity profiles, mixture model initialization schemes, and filter sizes when adopting the GMM-based and the temporally coherent MRF-based clustering techniques.

init	dist	filter size					
		2	4	8	16	32	$\overline{\mathscr{S}}$
random	gmm	73.9	74.1	73.7	73.5	73.9	74.0
	0	28.5	29.8	27.6	26.6	26.0	**72.2**
	2	77.2	**78.2**	74.2	65.5	51.5	77.8
	4	79.0	**80.7**	79.8	76.1	67.1	80.6
	8	80.1	80.9	80.5	80.9	77.6	**82.2**
	16	80.1	80.2	80.9	82.2	**84.1**	83.3
	var	79.7	79.9	81.8	81.0	81.8	**83.4**
	average	70.8	71.6	70.8	68.7	64.7	**79.9**
k-means	gmm	73.4	74.1	73.1	74.1	73.6	73.0
	0	28.2	30.0	27.3	26.6	26.0	**73.6**
	2	76.2	**77.2**	74.5	66.0	51.1	77.2
	4	77.9	**78.8**	78.3	75.9	65.9	78.8
	8	78.6	79.3	**80.9**	80.1	76.8	78.9
	16	79.5	**83.2**	79.4	82.8	83.2	82.2
	var	79.0	78.9	78.7	**82.5**	80.9	80.0
	average	69.9	71.2	69.9	69.0	64.0	**78.4**

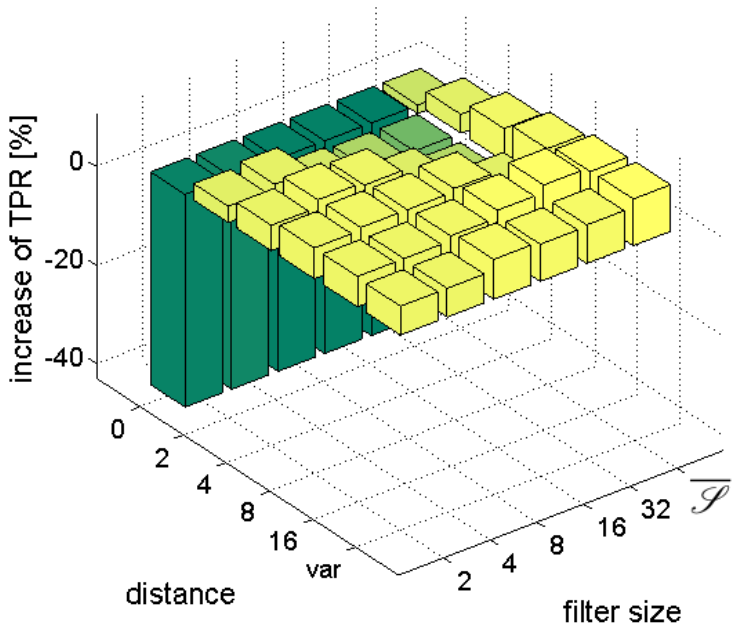


Figure 6.3.: Increase in the clustering performance in terms of the true positive rate for the 5 classes experiments and *k*-means model initialization with respect to varying filter sizes when averaging the outcomes over the complete set of velocity profiles and travel distances.

6.5. Conclusion

The work proposed in this chapter focused on the clustering of different terrain types using vibration data. Therefore, the framework of Markov random fields has been adopted which exploits temporal dependencies between consecutive observations. Thorough 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 number of present terrain classes, the chosen driving speed of the robot and the frequency of terrain transitions, an absolute increase in the true positive rate of up to 14.8% is obtained. The application of two other quality measures, the adjusted rand index and the adjusted mutual information criterion, which both resulted in similar findings in comparison with the true positive rate emphasizes the adequate performance of the proposed technique. As second contribution, a general approach was derived which yields an estimate of the amount of temporal coherence contained in the data set. Furthermore, this also provides a reliable indication of the absence of temporal dependencies.

7. Temporally Coherent Initialization of Gaussian Mixture Models

7.1. Introduction

In this thesis, a density-based clustering approach using Gaussian mixture models (GMMs) is considered. For deriving the parameters of the GMM the expectation maximization (EM) algorithm has become the most widely applied technique. Although the EM algorithm guarantees a convergence of the respective parameters within a finite number of iterations, the obtained parameter set is likely to not represent a global optimum of the likelihood function. Instead, the algorithm terminates at a local optimum which can be arbitrarily worse than the global optimum. To circumvent the local minimum problem, it is a common practice to restart the EM algorithm repeatedly, each time being initialized with a different set of starting parameters. In this way, each run is likely to explore different regions of the solution space, decreasing the probability of terminating in the same local optimum. By evaluating the objective function, i.e. the likelihood function, at the end of each run, the best clustering solution can be defined as the one whose parameter set maximizes the likelihood function.
Applying the EM algorithm multiple times, however, results in a waste of computational resources, since most of the calculations do not contribute to the final result. Moreover, in the context of unsupervised feature selection, a non-deterministic approach is not desirable, since it is not clear whether an unsatisfying clustering result emanates from an inappropriate choice of initial GMM parameters or from the feature subset itself. To decrease the effects of the former, initial parameters have to be determined deterministically which yield a reasonable local optimum of the likelihood function. For example, Bishop [Bis95] suggests to employ the results of k-means clustering as the initial GMM parameter set. Note, however, that due to its random initialization scheme, the k-means algorithm becomes non-deterministic as well, hence shifting the problem of random characteristics from the Gaussian mixture approach to its initializer only. Although a globally optimal solution for the k-means algorithm (as well as for the EM algorithm) can be obtained, this problem has shown to be NP-hard [GJW82]. That is, the k-means algorithm has to be applied taking each possible data subset of size k as the initial cluster centers into account. An approximate technique denoted as global k-means clustering was proposed by Likas et al. [LVV03]. They chose a greedy technique inserting one additional cluster center from one step to the next. Given a solution for the clustering problem with $k-1$ clusters, n k-means reruns are performed. In each rerun, one of the n data instances is employed as the kth initial cluster. Since the optimal solution for the one cluster setting is known[1], a deterministic outcome of the k-means clustering scheme is obtained. Although the authors demonstrated the effectiveness of their approach, the computational complexity allows its use for small datasets only.
Another technique was adopted by Celeux et al. [BCG03]. In their approach, they employed

[1] Here, the cluster center is the mean of all data instances and the respective covariance matrix is the covariance of the complete data set.

several short runs of a slightly modified EM algorithm initialized with random starts. The modifications consist of relaxing the convergence criterion and introducing a limit for the maximum number of EM iterations. After all EM runs have terminated, the solution is selected for initialization of the full EM algorithm which provides the largest likelihood. Similar to the global k-means algorithm, this approach is computationally expensive. Furthermore, it suffers from a random component introduced by the random initialization scheme.
The technique, which is presented in this chapter, is based on principal direct divisive partitioning (PDDP). The PDDP initialization scheme is an instance of divisive algorithms. This is, it starts with a single cluster which contains all data points. In the following, this cluster is recursively split into smaller subsets. Thereby, the algorithm selects those subclusters to split which have the largest variance. The actual splitting process is performed with regard to a hyper-plane which is orthogonal to its leading principal component. Recently, this work was adopted by several researchers and a variety of extensions have been proposed. For example, Kruengkrai et al. [KSI03] introduced an additional refinement step of the novel cluster centers after the splitting process. There, the adjustment was realized by a local 2-means clustering only considering the data contained in the cluster to split. Furthermore, Kruengkrai et al. employed the BIC criterion as a measure when to stop the splitting process. Later, Zeimpekis et al. [ZG03] proposed a 2^l-way division of each (sub)cluster instead of a 2-way (binary) data subdivision in each splitting step. This was achieved by considering not only the leading but the first l eigenvectors sorted according to the amount of explained variance. These eigenvectors define l hyperplanes which intersect at the origin and divide the space into 2^l orthants to which the respective data points are assigned. Similarly, Nilsson [Nil02] also makes use of the first l eigenvectors adopted in a binary splitting scheme. In this work, a binary splitting tree is employed whose nodes contain the index of the eigenvector which defines the actual splitting process. Given a tree depth of s, there are l^s possible splitting combinations to determine. This renders the binary splitting tree approach more and more computationally demanding as l increases. Finally, Tasoulis et al. [TT08] proposes a novel strategy to determine the position of the splitting hyperplane. Therefore, they consider the distance of two consecutive data instances previously projected onto the leading eigenvector. The pair having the maximum distance is then used to define the separating hyperplane. Furthermore, this maximum distance is used to choose the next subcluster to split. This choice is based on the assumption that a large distance is an indicator for multi-modality.
The approach presented in the following section is also based on principal direct divisive partitioning. Yet, it introduces several modifications to the original technique. The first one refines the new cluster centers after splitting a subcluster. Similar to the approach presented by Kruengkrai et al., a local 2-way clustering is applied to the contained subdata. In contrast to their approach, the 2-means approach is replaced by either a Gaussian Mixture model or a Markov random field model with two components. The latter approach enables the direct inclusion of temporal coherences into the initialization step of the chosen cluster model. Second, a direct application of the PDDP algorithm was not possible due to the impact of outliers. These outliers led to inadequate initial conditions of the clustering model which, in turn, resulted in ill-conditioned covariance matrices in the further course of the EM algorithm. Hence, a KNN-based outlier removal technique [AP05] was adopted prior to the initialization process. As a further contribution, the outlier removal technique was extended to enable the autonomous estimation of its most important parameter.

7.2. Applied Techniques

7.2.1. PCA-based Data Partitioning

The PCA-based data partitioning scheme splits the data into two subclusters. These subclusters, in turn, can be recursively subdivided into smaller clusters until the desired number of cluster centers is obtained. Here, two issues have to be clarified: which (sub)cluster is subject to a division and how the actual splitting operation is performed. The PCA-based data partitioning scheme solves the first problem by considering the likelihood of each (sub)cluster j: since the aim of GMM clustering is to maximize the data log-likelihood, an adequate heuristic is to split the cluster j providing the minimum log-likelihood L_j. The latter is defined by:

$$\begin{aligned} L_j = \ln p(X) &= \sum_{x_i \in \text{ cluster } j} (\ln \pi_j - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_j| - \frac{1}{2}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)) \\ &= \sum_{j=1}^{k}\left(n_j \ln\frac{n_j}{n} - \frac{dn_j}{2}\ln(2\pi) - \frac{n_j}{2}\ln|\Sigma_j| - \frac{dn_j}{2}\right), \end{aligned}$$

where π_j denotes the prior probability, μ_j is the mean, and Σ_j is the covariance of mixture component j, respectively. Since the determination of L_j requires the calculation of the determinant of the covariance matrix for each component j, another, less computationally demanding heuristic is suggested. This heuristic splits the cluster with the largest-within sum-squared-error SSE_j, defined as:

$$SSE_j = \sum_{x_i \in \text{ cluster } j} \|x_i - \mu_j\|^2.$$

After selecting the cluster j to split, the data points belonging to cluster j have to be assigned to one of the subclusters. The PCA-based data partitioning approach therefore projects these points to a one dimensional subspace. Projected observations which reside to the left of α_j, the projected mean of the considered data, are assigned to the first cluster and the remaining points are assigned to the other one. To motivate the choice of the projection vector which maps the data point into a one dimensional subspace, it has to be noted that with an increase in the likelihood the determinant of the covariance matrix $|\Sigma_j|$ decreases. Hence, the choice of the direction that contributes most to $|\Sigma_j|$ provides an adequate candidate for data splitting. In [SD07], it is shown that this direction is determined by the eigenvector which is associated with the largest eigenvalue. This is because $|\Sigma|$ is the product of all eigenvalues. Pseudo code for the PCA-based data partitioning is presented in Listing 3.

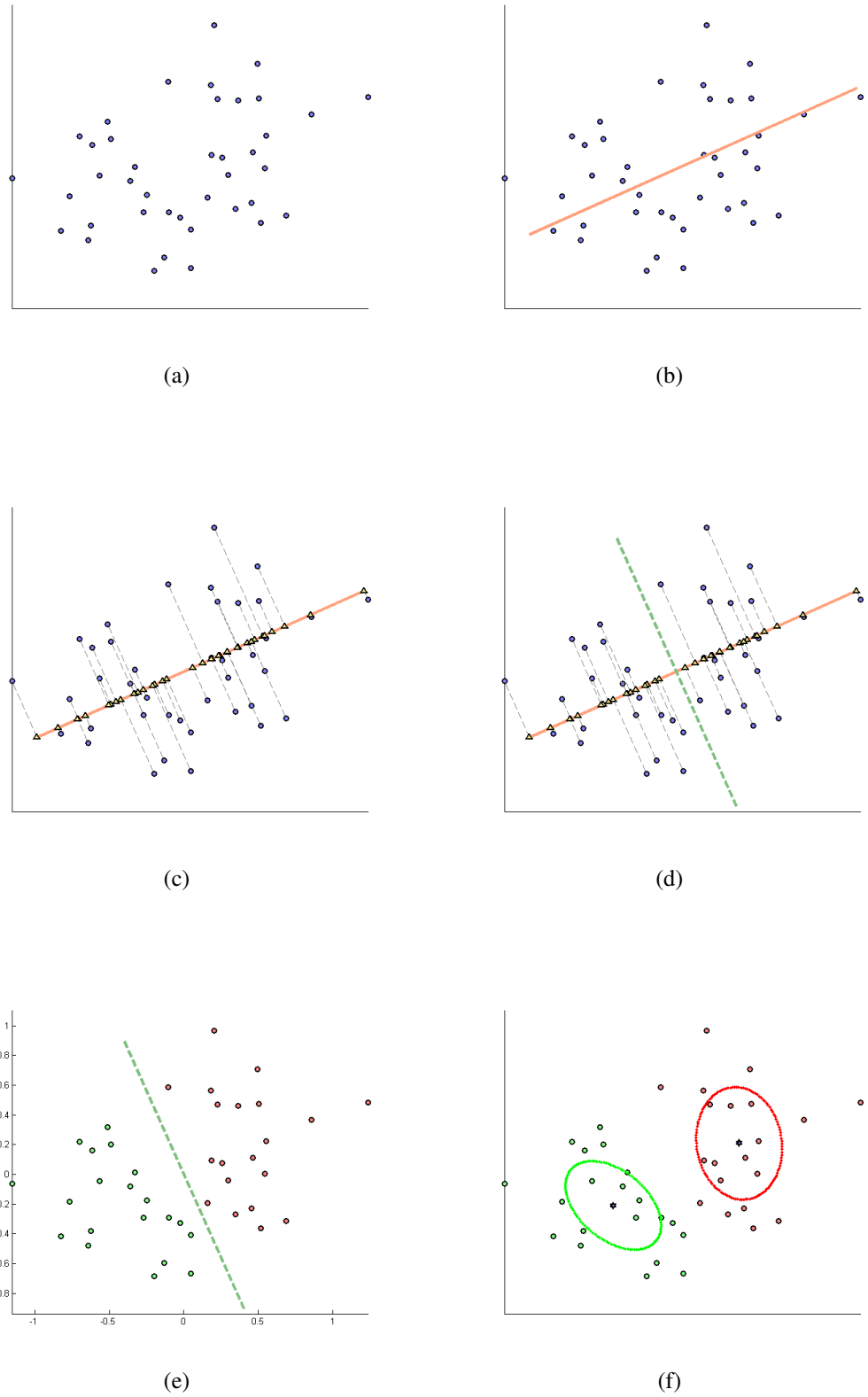


Figure 7.1.: Visualization of the cluster splitting process: (a) the data instances under consideration, (b) the direction defined by the main principal component, (c) the projection onto the one-dimensional subspace defined by the main principal component, (d) the decision boundary, (e) the assignment according to the established decision boundary, and (e) the result of a preceding 2-means step using the previously determined data assignment.

Algorithm 3 PCA-based Data Partitioning

1: $k = 1$
2: **repeat**
3: Given k clusters, select the cluster j with minimum SSE_j
4: Determine $\{X|x_i \in \text{ cluster } j\}$ and project X to the largest principal component axis defined by the data subset X. In the following, x_i^* is denoted as the projection of x_i onto this principal component axis.
5: Cluster j is split into two subclusters, cluster $k+1$ and cluster $k+2$ as follows: let α_j be the projected mean of X. For any $x_i \in X$, if $x_i^* \leq \alpha_j$ then x_i is assigned to cluster $k+1$ and to cluster $k+2$ in the other case.
6: Insert clusters $k+1$ and $k+2$ into the set of clusters and remove cluster j from this set.
7: $k = k+1$
8: **until** k equals the desired number of clusters.

7.2.2. Outlier Detection Using the Minimum Covariance Determinant

In [Rou85], Rousseeuw proposed the minimum covariance determinant (MCD) technique yielding robust estimates of the data's location and scatter. The key idea of the MCD method is to derive a subset of h observations from the data which result in a classical covariance matrix having the minimum determinant. The average of these h points is then referred to as the MCD estimate of location $\hat{\mu}_0$ and their covariance matrix represents the MCD estimate of scatter $\hat{\sigma}_0$. Further improvements of these estimates can be obtained by assigning weights to each observation and determining weighted variants of location $\hat{\mu}_w = (\sum_{i=1}^n w_i x_i) / (\sum_{i=1}^n w_i)$ and covariance matrix $\hat{\sigma} = (\sigma_{i=1}^n w_i (x_i - \hat{\mu}_w)$. Here, the weight w_i equals 1 if the robust Mahalanobis distance $\sqrt{(x_i - \hat{\mu}_0)^T \hat{\sigma}_0{}^{-1} (x_i - \hat{\mu}_0)}$ is smaller than the threshold $\sqrt{\chi^2_{p,0.975}}$ and 0 otherwise.
The value of h is a user-defined parameter determining the robustness of the MCD estimator. This is because the choice of h directly influences the breakdown value, i.e. the smallest fraction of observations which have to be corrupted to render the respective location and scale estimates unusable. For the MCD approach, the breakdown value is given by $(n-h+1)/n$, reaching its maximum if $h = \lfloor (n+p+1)/2 \rfloor$. Rousseeuw advises to choose h close to $0.5n$ if the data set is highly contaminated, $0.75n$ is recommended for a higher finite-sample efficiency. Note that the exact calculation of the MCD estimate requires the evaluation of all $\binom{n}{h}$ subsets of size h. An approximate approach, the FAST-MCD estimator, was proposed in [RD99] enabling the efficient calculation of the minimum covariance determinant. Its main component is the concentration step (C-step) defined as follows. First, an initial h-subset $H_1 \subset \{x_i\}_{i=1}^n$ of size $|H_1| = h$ is drawn. Given H_1, its empirical mean $\hat{\mu}_1$ and covariance matrix $\hat{\sigma}_1$ are determined. The latter estimates are then employed to calculate the respective Mahalanobis distances defined as

$$d_1(i) = \sqrt{(x_i - \hat{\mu}_i)^T \hat{\sigma}_1^{-1} (x_i - \hat{\mu}_i)}, i \in [1, n].$$

The h observations with minimum distances $d_1(i)$ constitute the elements of the succeeding h-subset H_2. Based on this new subset, the respective mean $\hat{\mu}_2$ and covariance matrix $\hat{\sigma}_2$ are computed. Rousseeuw showed that

$$\det(\hat{\sigma}_2) \leq \det(\hat{\sigma}_1),$$

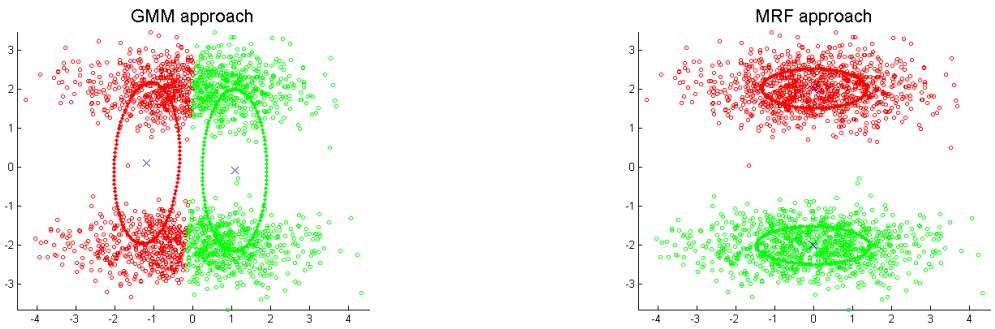
thus, each C-step results in a new h-subset with a lower covariance determinant. The gradually decrease in the data's covariance is referred to as a concentration process, hence the name concentration or C-step. These C-steps are iteratively applied until $\det(\hat{\sigma}_i) = 0$ or $|\det(\hat{\sigma}_{i-1}) - \det(\hat{\sigma}_{i-1})| < \varepsilon$. The convergence of succeeding determinants is guaranteed within a finite number of steps, since the set of possible h-subsets is finite. Yet, the converged determinant value may represent a local minimum of the MCD objective function only. To obtain an approximate global solution, the FAST-MCD algorithm can be run with varying initial subsets H_1. Then, succeeding C-steps are applied to each of the subsets and the solution is kept which provides the lowest determinant.
For choosing the initial H_1-subsets, a random $(p+1)$-subset R is drawn and both its mean $\hat{\mu}_R$ and covariance $\hat{\sigma}_R$ are calculated. The H_1 subset then consists of the h observations yielding the h minimum Mahalanobis distances $d_R(i) = \sqrt{(x_i - \hat{\mu}_R)^T \hat{\sigma}_R^{-1} (x_i - \hat{\mu}_R)}$. Note that by choosing an initial $(p+1)$-subset instead of a regular h-subset, the probability is increased to obtain an outlier-free set of observations. The random nature of the initial R-subset selection scheme renders the FAST-MCD algorithm non-deterministic. Even if the sampling follows a deterministic rule, its outcome still depends on the order in which observations appear in the data set. To obtain a fully deterministic approach, Hubert et al. [HRV11] proposed the DetMCD algorithm which provides similar robustness characteristics while remaining the permutation invariance. There, the permutation invariance is achieved by starting from seven well-chosen initial estimates along with a successive application of C-steps until convergence. Among others, the initial estimates include the Spearman correlation matrix, the spatial sign covariance matrix, and the raw covariance estimate derived from the OGK estimator [MZ02]. The complete enumeration of initial H_1-subset derivation technique is listed in [HRV11] and [TV10].
The FAST-MCD approach can also be applied in a univariate setting [RL87]. Here, the derived estimates represent the (weighted) mean and variance of the h-subset with the smallest variance. The computational complexity reduces to $O(n \cdot \log n)$ by considering contiguous h-subsets and by exploiting the fact that in this case, the respective mean and variance can be efficiently determined in a recursive manner.

7.3. Improving PCA-based Data Partitioning

Primal experiments using a PCA-based data partitioning scheme yielded inferior results in comparison with random initialization schemes such as random initial center selection or a k-means-based initialization technique. Yet, improvements of the clustering performance could be observed when adopting the temporally coherent MRF clustering approach of Chapter 6 not only for the actual clustering process but also for cluster center initialization. The key elements and the motivation of using an initialization scheme which makes use of temporal dependencies are presented in the following subsection. Further improvements are obtained when applying an outlier removal technique prior to initialization. Thus, in a succeeding subsection, the employed outlier removal technique is introduced along with proposed modifications to estimate its most important parameter.

7.3.1. Temporally Coherent Data Partitioning

Algorithm 7.2.1 shows that the inclusion of temporal coherences can be realized within two stages of the divisive initialization framework: in the data projection and in the cluster bisection steps. The latter comprises the task of partitioning the (sub)cluster into two halves and



(a) (b)

Figure 7.2.: The clustering result when adopting a clustering technique with 2 components which (a) disregards and (b) exploits temporal dependencies.

assigning each data instance to one of the generated subclusters. Considering data projection, temporal dependencies can be considered by finding a projection matrix W such that temporally close neighbors are spatially close to each other in the projected data space. Techniques which allow for this kind of transformations include the locality preserving projection technique [HN03] and the neighborhood preserving projection approach [KS07]. Note, however, that these techniques did not provide adequate results when applied to the present problem of cluster center initialization and hence are not further addressed.
Instead, the integration of temporal dependencies into the cluster bisection process is proposed. After projecting the data onto the leading eigenvector defined by the data subset under consideration, each data instance is assigned to one of the generated subspaces. Thereby, the corresponding subspace is defined by the sign of the respective projected data instance. After the assignment step, there are two clusters defined determined by their mean and covariance matrix. The latter two parameters are employed to initialize a further clustering which aims at refining the present solution. For the cluster refinement step, a variety of techniques is available. Yet, the use of a temporally coherent clustering scheme is advised. This is motivated by Figure 7.2 where a data set with two classes is depicted. When applying the original PCA-based initialization scheme, the leading eigenvector and hence the projection direction becomes the one parallel to the y-axis. As shown in Figure 7.2(a), this results in a data subdivision which erroneously bisects the instances of each class, respectively. A clustering approach, however, which makes use of temporal constraints yields the correct clustering (cf. Figure 7.2(b)). Based on the assumption that a better clustering is obtained for non-artificial data sets as well, the original cluster bisection process is extended by a succeeding temporally coherent MRF clustering step with two components. This technique is denoted as the PCA-MRF approach in the following.

7.3.2. A Combined k-NN and MCD approach for Outlier Removal

The outlier detection technique adopted in this chapter is based on the outlier definition of Angiulli et al. [AP05]. There, the accumulated distances from each data instance x_i to its k nearest neighbors $x_l, l \in [1,k]$ are considered. Angiulli et al. refers to these accumulated distances of a certain data point x_i as weight $w_i = \sum_l d(x_i,x_l), x_l \in \mathcal{N}$, where $\mathcal{N}$ denotes the k neighborhood of x_i. Sorting the data instances according their weight w_i, outliers can then be determined by choosing the n data points with maximum weight. Here, n denotes the number of expected outliers in the data set and is a used-defined value. Note that, in general, the latter

number cannot be selected a priori but has to be estimated from the data set instead. Since the original approach of Angiulli does not provide means for the estimation of the expected number of outliers n, a technique for its derivation is introduced. This technique is based on the assumption that the weights w_i of inliers follow a Gaussian distribution $N(\mu,\Sigma)$. Outliers, on the other hand, have weights arising from a different distribution hence representing outliers with respect to $N(\mu,\Sigma)$. By estimating the parameters of the inliers' Gaussian distribution by means of robust mean and variance estimation, outliers can be detected using the univariate MCD method introduced in Section 7.2.2.

7.4. Experimental Results

7.4.1. Experimental Setup

Cluster Center Initialization For the cluster center initialization, not only the proposed PCA-MRF-based partitioning scheme (PCA-MRF) has been implemented but also the original PCA-based technique (PCA). The latter represents the reference when evaluating the novel temporally coherent partitioning method. Furthermore, to demonstrate the superiority of temporally coherent clustering within the cluster refinement step, the involved MRF clustering with two components is replaced by alternatives which do not take temporal dependencies into account, namely k-means clustering with $k = 2$ (PCA-kmeans) and GMM-based clustering with two components (PCA-GMM). Finally, to establish the connection between this and the previous chapter, the random and the k-means initialization schemes (random and kmeans, respectively) have also been included into the set of initialization procedure candidates. This is because they are commonly proposed as the default initialization technique (cf., i.e. [Bis95]). Note, however, that the latter two approaches have been implemented for the reason of completeness, although they do not yield deterministic results and hence do not match the scope of this research.
Almost all introduced initialization schemes are parameter-free. The sole exception is the PCA-MRF-based partitioning technique which requires the neighbor set size $\overline{\mathscr{S}}$ to be defined. Experiments revealed the superiority of choosing $\overline{\mathscr{S}} = 2$. That is, only the direct neighbors of an instance are considered for establishing the temporal coherences which guide the clustering process.
To render the cluster center initialization robust against outliers, an outlier removal technique was applied prior to the actual initialization process. For the outlier removal technique, the combined k-NN and MCD approach of Section 7.3.2 has been adopted. The corresponding parameter k was experimentally found and set to $k = 16$. Note, however, that experiments revealed an insensitiveness against this parameter in the range of $k \in [8,32]$.

Vibration Segment Preprocessing The aim of the research presented in this chapter is to develop an adequate cluster initialization technique which can then be applied to the feature subset selection task introduced in the next chapter. Thereby, the set of available features comprises all components of the DFT amplitude spectrum which are extracted from a block of 128 acceleration samples. To access the performance of varying cluster center initialization schemes, it is thus beneficial to consider differing subsets of the complete feature set as input representations of the preprocessed vibration pattern. Since the application of all possible, i.e. 2^{64}, subsets is intractable due to the large computational complexity, a sampling of the complete subset space has been performed. Each subset candidate of this sampling contains the first m

spectral components where m was chosen from the set $M = \{8, 16, 24, 32, 40, 48, 56\}$. Note that the complete set of features, i.e. $m = 64$, resulted in invalid clusterings for the random and k-means initialization procedures in all conducted replications. Hence, the decision was made to remove the latter case from the list of chosen subset candidates. Finally, the last feature representation is given in terms of the MFCC-like descriptor being an instance of feature extraction rather than feature selection. Since, however, the following two chapters make use of this feature representation, its performance in the context of cluster center initialization is important to be pointed out as well.

Clustering Model The clustering model consists of the temporally coherent Markov random field-based technique. Furthermore, for determining the neighborhood set size, the autonomous neighborhood set size estimation technique of Section 6.3.2 has been employed.

Cluster Model Evaluation Criteria Since the varying cluster quality criteria which have been adopted in the previous chapter turned out to be quite similar with respect to their outcome, only one quality measure is employed in this chapter. The selected quality criterion is the true positive rate which is the most descriptive one among the three.

Path Generation Scheme The following experiments employ the same path generation scheme as introduced in Section 2.2. This includes both the 3 classes experiments providing naturally generated paths as well as the 5 classes experiments representing paths with varying travel distances. As the only difference, the travel distance of 0 was removed for the 5 classes experiments. This is because a travel distance of 0 yields an MRF model being identical to a Gaussian mixture-based clustering model.

7.4.2. Results and Discussion

Results of the 3 Classes Experiments

Table 7.1 shows the clustering performance of the initialization experiments with 3 classes when averaging the results over velocity profiles 1-3 (0.2, 0.4, and 0.6 m/s). Here, the table comprises the results which are obtained when applying the MFCC-like feature extraction and the low-energy feature selection schemes, where the latter employs a feature subset of size 8 up to 56 spectral components. Furthermore, the mean outcomes of all GMM and MRF experiments are presented, averaging the true positive rates over all feature extraction and feature selection approaches. Note that the latter outcomes are depicted in Figure 7.1. Referring to these averaged results, the following ranking can be established: first, the PCA-MRF approach 93.7% (84.4%), followed by kmeans 92.7% (83.1%), PCA-GMM 91.7% (82.3%), PCA-kmeans 91.6% (81.8%), PCA 90.4% (80.8%), and finally *random* 89.1% (77.7%) where the first number denotes the true positive rate of the MRF clustering technique and the second number in brackets is the outcome of the GMM clustering result. Hence, the same ranking applies to both the GMM and MRF clustering approaches. These results advise the use of an additional clustering step for determining the new cluster centers in the cluster refinement procedure as the respective techniques outperform the PCA approach on average. In this context, the temporally coherent PCA-MRF initialization technique is superior to all other subdivision techniques which do not incorporate temporal dependencies into the refinement process.
The maximum TPR for MRF clustering is obtained when applying the MFCC-like preprocessing scheme. This is notable, since the same feature extraction technique does not provide the best GMM clustering performance. Instead, using the latter technique, the most appropriate clustering is obtained in terms of selecting a subset of the amplitude spectrum. Other feature subsets result in a differing clustering performance depending on the size of the respective subset: for both, the GMM and MRF approaches, the true positive rate rises until a feature set size of 32 and then worsens with increasing set size. There are at least two reasons for this behavior. First, the larger the set size, the larger is the number of DFT amplitude components with high-frequency content. As the latter is assumed to only contain the noisy parts of the signal, the high-frequency components do not contribute to a good clustering but worsen the clustering performance instead. Second, with an increasing set size, the dimensionality of the input vectors rises as well. This, in turn, increases the probability of obtaining a model parameter set which represents a local optimum only. Note that these findings confirm the choice of the MFCC-like descriptor which rejects the 32 spectral components representing the high-frequency part of the signal. These findings related to the decrease of the clustering performance starting with a set size of 40 DFT amplitude components are valid for all but the PCA-MRF approach. Here, the true positive rates remain approximately the same up to a feature set size of 56 spectral components. As shown in the next chapter, this characteristic is beneficial for backward feature selection techniques which start their search for an appropriate feature subset with the complete set of features.

Table 7.1.: True positive rate [%] of the 3 classes experiments when varying the clustering model, the number of input dimensions, the feature extraction scheme, and the model initialization technique, averaged over all velocity profiles.

model	#feat	initialization technique					
		random	kmeans	PCA	PCA-kmeans	PCA-GMM	PCA-MRF
gmm	0	77.8	80.9	75.2	76.4	79.9	**82.7**
	8	**69.4**	68.8	69.0	68.8	69.0	68.6
	16	79.2	83.5	76.7	**84.8**	84.3	84.3
	24	81.9	87.2	**88.0**	87.1	87.1	87.2
	32	83.0	88.9	89.2	88.9	89.4	**89.7**
	40	80.0	87.4	88.2	87.4	**88.7**	88.3
	48	77.7	86.0	85.0	86.1	86.7	**87.2**
	56	72.9	81.9	75.3	75.0	73.3	**87.2**
	average	77.7	83.1	80.8	81.8	82.3	**84.4**
mrf	0	89.6	93.0	88.3	88.1	93.6	**97.1**
	8	**81.9**	81.2	81.8	81.2	81.8	81.1
	16	91.9	94.3	85.6	95.4	**95.5**	95.4
	24	92.2	95.4	95.4	95.4	**95.5**	95.4
	32	92.3	**95.3**	95.2	**95.3**	**95.3**	**95.3**
	40	90.8	94.9	**95.0**	94.9	**95.0**	**95.0**
	48	89.1	94.8	95.1	95.0	94.8	**95.2**
	56	85.0	92.6	86.9	87.5	82.1	**95.0**
	average	89.1	92.7	90.4	91.6	91.7	**93.7**

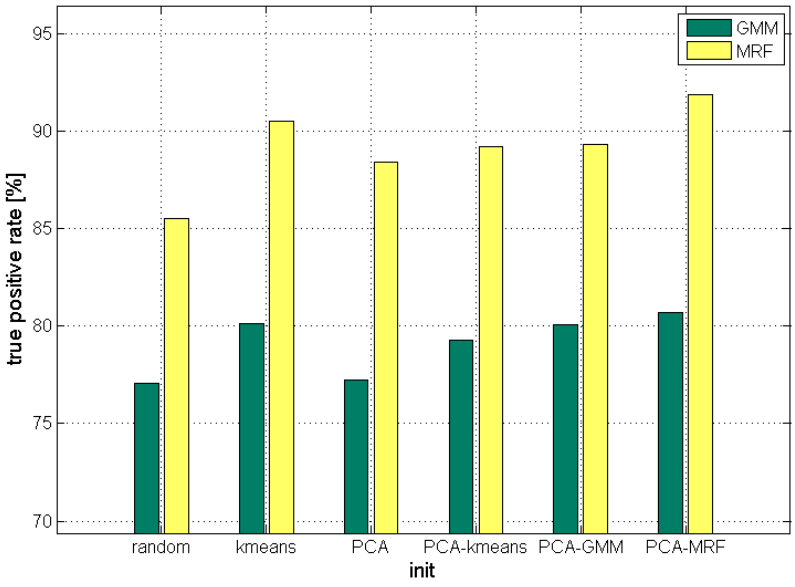


Figure 7.3.: The true positive rate [%] with respect to the 3 classes experiments and varying model initialization techniques when averaging the outcomes over all clustering models, descriptor lengths, and feature extraction schemes.

Results of the 5 Classes Experiments

The structure of Table 7.2 presenting the results of the 5 classes experiments follow the one of the 3 classes experiments. The respective outcomes were obtained after averaging the true positive rates over both velocity profiles 1-3 and travel distances of 2, 4, 8, and 16 m, and combinations of those values, respectively. Again, the clustering performance of the feature selection and feature extraction schemes are provided along with their respective mean (average). Establishing a ranking based on the latter measure, the different approaches can be organized in decreasing clustering performance as follows: PCA-MRF 79.4% (73.8%), PCA-GMM 78.8% (72.9%), PCA-kmeans 78.3% (72.2%), kmeans 77.1% (71.6%), PCA 74.7% (68.9%), and finally random 74.0% (68.0%), where the first number denotes the true positive rate of the MRF clustering technique and the second number in brackets is the GMM clustering result. Note, that this ranking is the same for both the GMM and MRF clustering approaches. These findings reveal that an additional clustering step for determining the new cluster centers in the cluster refinement procedure positively effects the succeeding progress of the clustering task. Further improvements are achieved by incorporating temporal coherences into the cluster refinement procedure by means of a MRF-based clustering approach with two components.
Considering the performance of the feature selection and feature extraction techniques, the MFCC-like descriptor yields the largest true positive rates for both the GMM and MRF clustering methods. The feature selection strategy performs best at a feature subset size of 24 with the sole exception of the random initialization scheme. Larger feature sets result in a decreased clustering performance supporting the assumption of the higher-frequency spectral components to be irrelevant.

Table 7.2.: True positive rate [%] of the 5 classes experiments when varying the clustering model, the number of input dimensions, the feature extraction scheme, and the model initialization technique, averaged over all velocity profiles.

model	#feat	initialization technique					
		random	kmeans	PCA	PCA-kmeans	PCA-GMM	PCA-MRF
gmm	0	74.3	72.9	71.5	80.2	73.1	**80.2**
	8	69.1	69.7	62.3	60.5	**70.0**	69.1
	16	70.3	**71.8**	69.8	68.9	70.1	70.4
	24	70.3	73.4	74.2	76.6	76.1	**76.7**
	32	67.8	73.1	69.4	74.3	74.9	**75.1**
	40	65.3	71.7	69.4	**74.0**	73.6	73.2
	48	63.8	70.6	70.5	72.6	**73.7**	73.5
	56	62.8	69.5	63.9	70.9	71.9	**72.4**
	average	68.0	71.6	68.9	72.2	72.9	**73.8**
mrf	0	81.4	79.4	78.8	87.5	79.5	**87.5**
	8	77.4	78.4	71.8	70.8	78.7	**78.8**
	16	75.8	**77.0**	75.9	74.4	76.0	75.8
	24	75.5	80.0	79.6	**83.8**	83.4	82.5
	32	73.0	77.5	71.1	78.8	79.6	**80.0**
	40	70.9	76.1	72.8	**79.0**	78.2	77.3
	48	69.5	74.7	77.0	76.7	**78.5**	77.3
	56	68.3	73.8	70.5	75.2	**76.7**	76.2
	average	74.0	77.1	74.7	78.3	78.8	**79.4**

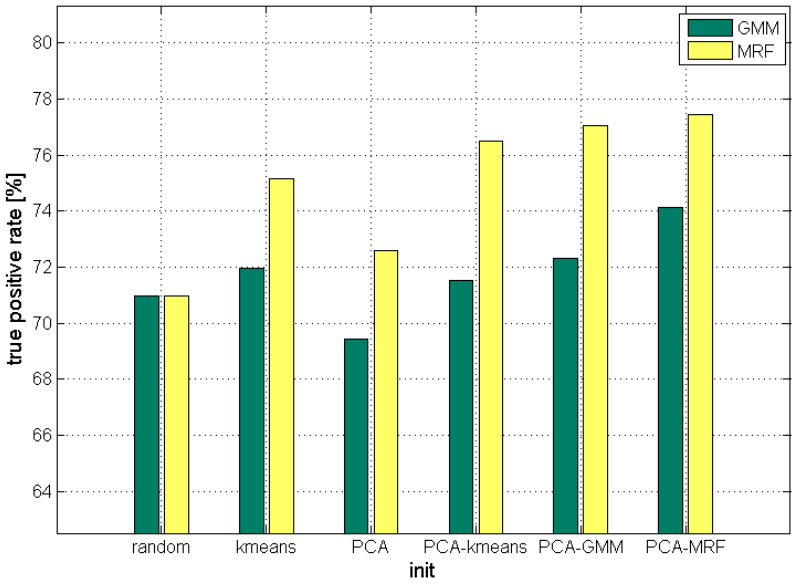


Figure 7.4.: True positive rate [%] of the 5 classes experiments when varying the clustering model, the number of input dimensions, the feature extraction scheme, and the model initialization technique, averaged over the complete set of velocity profiles.

7.5. Conclusion

This chapter focused on the development of a deterministic scheme for cluster center initialization in the context of Gaussian mixture model-based clustering. The analysis was rendered necessary, since the previously considered initialization techniques, a random and a k-means approach, are characterized by random components. These random components are likely to yield clustering models with varying performance due to the chance of ending up in differing local optima after the model generation.
The proposed technique is based on a hierarchical partitioning scheme. Starting with a single cluster, the data is recursively divided into two subclusters until a desired number of clusters is reached. Here, the cluster is split by means of projecting the data onto the leading eigenvector and assigning each instance to one of the generated subspaces according to the sign of its corresponding projection. The novel cluster center of each subspace is then defined as the mean vector determined over the respective instance subset. Experimental results suggest the use of a succeeding refinement step to adjust the obtained cluster centers. Here, the refinement step is realized in terms of a clustering technique with 2 components. As for the applied clustering scheme in the refinement process, the best clustering performance was achieved by means of a Markov random field-based technique. The MRF-based clustering scheme yielded an improvement of 3.3% TPR for the 3 classes experiments and 4.7% TPR for the 5 classes experiments on average in comparison with the original PCA-based partitioning approach.

8. Feature Selection for Vibration-based Terrain Clustering

8.1. Introduction

Feature selection describes the task of reducing the dimensionality of the given data. Its main objective is to represent the data feature set by a smaller subset without compromising the original data characteristics. Along with the reduction of the computational costs, feature selection is likely to improve the performance of a predictive model, rendering it less prone to overfitting, increasing its resistance towards noise, and maintaining its discriminative capabilities. This is because different features often contain a certain degree of dependency or redundancy among each other. Their removal does not affect the prediction performance significantly, since all necessary information is already contained within other features. Hence, in recent years, the problem of redundant feature removal became an active research area. Although the main focus has been devoted to feature selection in supervised learning tasks, a growing interest in the unsupervised clustering community has emerged recently. Note that in contrast to classification problems, where the variable selection problem can be clearly stated as the search for an attribute set yielding the highest classification performance, this task is less well-defined for cluster analysis. This is mainly a matter of missing class information which could guide a specific selection scheme. Generally, the unsupervised feature selection task can be divided into three categories: filter, embedded and wrapper techniques. Methods from the first category choose a certain feature subset from the original data space. Here, the selection is based on evaluation criteria which are independent from the clustering algorithm. Since no clustering model has to be generated and evaluated, filter techniques are computationally more efficient than embedded and wrapper methods. For the evaluation critera, researchers focused on information-based metrics [LSLZ09] as they provide an adequate measurement of quantifying the uncertainty of a feature. For example, Søndberg-Madsen et al. [SMTn03] defined the relevance of a given feature by scoring the dependence between the feature under consideration and the remaining features. There, pairwise dependent scores were determined in terms of both their mutual information and their mutual prediction capability, respectively. Another filter technique was proposed by Yen et al. [YCL10]. Their technique employed an eigen-decomposition to rank the linear dependency among different features sequentially, removing those features which could be represented best by the remaining ones. Further, Yen et al. showed that their approach is similar to removing features which contribute the most to the principal components with the smallest eigenvalue.

Embedded methods combine the feature selection and model generation task as they aim at incorporating knowledge about the specific structure of the clustering algorithm. Furthermore, this combined approach involves that it is neither possible to separate the clustering from the feature selection process nor to replace the clustering technique by another one. Several researchers addressed the problem of embedded feature selection methods in the context of Gaussian mixture model-based clustering. Law et al. [LFJ02] therefore defined salient features as those which are capable of describing the data with a multi-modal distribution and modes which

can be adequately represented by Gaussian components. In this context, they represent saliency in terms of probability. Given a certain feature, data instances are considered independent of a certain mixture component up to a determined probability. Feature saliency is then defined as the complement of this probability. To allow both the detection of appropriate feature subsets and the correct number of clusters, Law et al. adopted the minimum message length (MML) criterion. As the latter approach was derived by means of several assumptions and simplifications, Constantinopoulus et al. [CTL06] extended this technique using a variational framework which introduces a Bayesian approach for mixture learning.
In a feature selection wrapper approach, the performance of a certain feature subset is estimated using performance criteria which are derived from an established clustering model. That is, the variable selection scheme is wrapped around the clustering technique under consideration. Several performance criteria can be considered. For example, Dy et al. [DB04] considered the scatter separability and maximum likelihood obtained after the clustering process. In another work, Figueiredo et al. [FJL03] focus on the change in the posterior probability estimates when removing certain features. There, it is assumed that the removal of unnecessary features has a less significant impact on the posterior probability estimates in comparison with the removal of useful ones. Although wrappers provide a systematic means for feature subset selection, they suffer from at least two drawbacks: at first, wrapper techniques are tightly coupled with a predefined clustering algorithm. That is, feature subsets which result in an appropriate clustering with respect to a certain clustering technique do not have to be optimal when exchanging the clustering technique. Second, the requirement to run the clustering algorithm with any provided feature subset renders the wrapper technique time-consuming and often intractable for large scale problems.
Recently, researchers aimed at combining the benefits of both the filter and wrapper approaches. Here, the common strategy is the use of an iterated step-wise procedure. In a clustering step, a hypothetical partition is established generating the base for the successive relevance determination step in which the features are scored for relevance. Hruschka et al. [HCHE05] therefore employed the k-means algorithm for clustering in conjunction with a Bayesian filter which determines the relevance of each feature. For the Bayesian filter, Hruschka et al. adopted the Markov blanket framework [Pea88] in which a feature set can be replaced by a smaller subset if the latter is able to represent the original set completely. In another approach, Xing et al. [XK01] ranked the features according to their intrinsic discriminability, relevance to the hypothetic partitions, and irredundancy to other relevant features. This ranking is then employed to select the features which are to be used in a following clustering step.
From the results of Chapter 5 it can be derived that a compact vibration signal descriptor outperforms the higher-dimensional one which is based on the signal's amplitude spectrum in the context of Gaussian mixture model-based classification. The definition of the MFCC-like descriptor, however, relied on the assumption that the most relevant spectral components are the ones representing the lower-frequency subbands of the signal. To the best knowledge of the author, no experiments were carried out which support this assumption. Hence, it remains unclear, whether the choice of the proposed binning strategy in the context of the MFCC-like descriptor is an appropriate one. The work presented in the following chapter aims at clarifying this issue under the perspective of unsupervised learning: given 64 spectral components representing the elements of the amplitude spectrum of a 128 sample-sized vibration signal, systematic approaches are discussed, compared, and extended to find an adequate feature subset from the set of spectral components. As presented in the result section, the best feature selection approach is a hybrid technique combining a filter and wrapper methodology.

8.2. Applied Feature Selection Techniques

Given a d-dimensional data set $X = \{x_i\}_{i=1}^{n}, x_i \equiv F = \{f_j\}_{j=1}^{d}$, the aim of the feature selection task is to choose a subset S from F, $S \subset F$, which represents the data set best with regard to the clustering performance. Due to the presence of noise in certain frequency bands of the vibration signal, not all spectral components are believed to contribute equally well to the final clustering result. Hence, an adequate feature selection scheme is characterized by the capability of maintaining "good" features while identifying and removing noisy features. The following subsections discuss two popular methods for feature selection including filter-based and wrapper-based techniques.

8.2.1. Filter-based Feature Selection

In filter-based feature selection, each feature f_i is assigned a certain degree of relevance. In a succeeding step, the features are rearranged according to the relevance criterion in decreasing order and the final feature vector comprises the first n elements from the resorted feature list. Since this assignment is independent from the employed clustering algorithm, filter methods provide an effective and general means of feature selection. The novel unsupervised filtering scheme presented in this work is based on a mutual information (MI) filter. Details of how the mutual information between two random variables can be estimated are provided in the following section.

Mutual Information The mutual information index has received increased attention due to its strong theoretical background in information theory. For its derivation, the Kraskov MI estimator $MI^{(1)}$ [KSG04] is employed which is based on entropy estimation using k-nearest neighbor statistics. Given normed spaces X and Y with norms $\|.\|_x$ and $\|.\|_y$, a new space $Z = X \times Y$ is defined. For each $z_i \in Z$, $z_i = (x_i, y_i)$, its norm $\|z_i\|_Z$ is determined by:

$$\|z_i\|_Z = \max\left\{\|x\|_X, \|y\|_Y\right\}.$$

Given $k \in \mathbb{N}$, we define $\varepsilon(i)/2$ as the distance from z_i to its k-th nearest neighbor. Further, we denote by $n_x(i)$ the number of points x_j with $\left\|x_i - x_j\right\|_X \leq \varepsilon(i)/2$ and by $n_y(i)$ the number of points y_j with $\left\|y_i - y_j\right\|_Y \leq \varepsilon(i)/2$. Then the MI can be estimated by:

$$MI^{(1)}(X,Y) = \psi(k) - \langle \psi(n_x + 1) + \psi(n_y + 1) \rangle + \psi(N),$$

where $\langle \ldots \rangle = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\{\ldots(i)\}$ is an averaging operation both over $i = 1, \ldots, N$ and over all realizations of the random samples. $\psi(x)$ is the digamma function recursively defined as:

$$\psi(x+1) = \begin{cases} \gamma, & \text{for } x = 0 \\ \psi(x) + \frac{1}{x} & \text{otherwise} \end{cases},$$

where γ is the Euler-Mascheroni constant.

8.2.2. Wrapper-based Feature Selection

In the wrapper-based feature selection approach, the quality of a feature subset can directly be assessed in terms of the underlying clustering technique rather than a cluster model independent measure. Given a candidate feature subset $C \subset F$, a cluster model is trained using C. Then, the

generated model is evaluated by means of an internal criterion J yielding the performance of the feature subset C. In a final step, the candidate feature subset C is defined to be the optimal one which maximizes the chosen internal criterion J. The trivial solution to wrapper-based feature selection is to perform an exhaustive search over all possible feature subsets. However, there are 2^d feature subsets in a d-dimensional data set and thus a brute force search is computationally infeasible. Two approximations of the brute force search which only explore a fraction of the overall search space are the sequential forward feature selection and the backward search feature selection approaches. Both techniques are introduced in the following subsections.

Sequential Forward Feature Selection

Formally, the feature selection task is defined as a search problem of finding a subset S of m features among the complete feature set F with respect to a chosen internal measurement $J(S)$. Sequential forward feature selection starts with an empty set which is iteratively enlarged by (at least) one feature. The new feature is determined among the remaining features, i.e. features which are not included in the currently selected feature set such that the criterion J is maximized or minimized. The forward selection terminates when a feature insertion step does not result in an improvement of the criterion J or the feature subset reaches its predefined size.

Backward Search Feature Selection

In a backward search strategy, one or more features are iteratively removed from the feature subset. Starting with the complete set of features, (at least) one feature is removed after each iteration. Similar to the sequential forward feature selection approach, the candidate feature sets $\{C_i\}$ are evaluated according to a criterion $J(C_i)$. Yet, in this context, a candidate feature set C_i is decomposed of the feature (sub)set from the previous iteration S^{t-1} minus a candidate feature c_j with $c_j \in S^{t-1}$, that is $C_i = S^{t-1} \setminus c_j$. The chosen feature being removed from S^{t-1} becomes the optimal one with respect to J. An instance for the criterion J is presented in the following paragraph.

Backward Search based on the Change of Posterior Distributions The wrapper technique of Figueiredo et al. [FJL03] considers the change in the posterior estimates when removing a certain feature. Here, the key assumption is that the class posteriors are altered less significantly by conditionally independent features in comparison with relevant features when they are removed from the feature set. In this context, the change of posterior probabilities denotes the deviation of the class posteriors with respect to the original model and the one which operates on the smaller feature subset.
For the derivation of the respective formulas, the E-step of the EM-algorithm is reconsidered. There, the posterior estimates are determined by

$$p(c = m|x_i, \theta) \equiv w_{i,m} \propto \hat{\pi}_m p(x_i|\hat{\theta}_m). \tag{8.1}$$

In the following, it is assumed that the complete feature set can be divided into a useful f_U and non-useful f_N feature set. Furthermore, the non-useful features are independent from the useful ones and the distribution of the former is the same for all clusters. This yields

$$p(f_i|\theta_U, \theta_N) = p(f_{i,N}|\theta_N) \sum_{m=1}^{k} \pi_m p(f_{i,U}|\theta_{m,U}), \tag{8.2}$$

where f_i is equivalent to the set of features contained in x_i. Further, θ_N denotes the set of parameters characterizing the distribution of the non-useful features, and $\theta_U = \{\theta_{1,U}, \ldots, \theta_{k,U}\}$ is the set of parameters characterizing the mixture distribution of the useful features. Considering the m-th mixture component and inserting (8.2) into (8.1), we have

$$w_m = \frac{\pi_m p(f_U|\theta_{m,U}) p(f_N|\theta_N)}{\sum_{j=1}^{k} \pi_j p(f_U|\theta_{j,U}) p(f_N|\theta_N)} = \frac{\pi_m p(f_U|\theta_{m,U})}{\sum_{j=1}^{k} \pi_j p(f_U|\theta_{j,U})}.$$

Informally, the latter equation reveals that the posterior probability of a certain observation can be represented in terms of useful features only.
To clarify the relationship between posterior probability calculations and feature selection, the notion of conditional independence has to be introduced. Applying the definition of Koller et al. [KS96], a feature subset f_N is denoted as irrelevant if it is conditionally independent from the labels c, given a useful feature set f_U. Formally, this can be expressed as

$$p(c|f) = p(c|f_U, f_N) = p(c|f_U).$$

From the latter equation the following implication can be made

$$p(c|f_U, f_N) = p(c|f_U) \Rightarrow \mathrm{KL}\left(p(z|f_U, f_N) \| p(c|f_U)\right) = 0, \tag{8.3}$$

where KL denotes the KL divergence between the probability distributions $p(c|f_U, f_N)$ and $p(c|f_U)$. Equation (8.3) shows that the removal of non-useful features from the set of useful features does not alter the posterior probability distribution and hence results in a KL divergence of 0.
To obtain a criterion for the performance of a candidate feature set, Figueiredo et al. propose to average this measure over the feature space given by the training samples. Note, however, that in the case of unsupervised learning, the true labels c cannot be inferred. Instead the expected values $W = \{w_{i,m}\}_{i=1,m=1}^{n,k}$ are employed to estimate the sample-based feature usefulness measure. The latter is based on the assumption that W was obtained using the full feature set and a clustering model with parameter vector $\hat{\theta}$. Further, the set $V = \{v_{i,m}(N)\}_{i=1,m=1}^{n,k}$ is defined denoting the expected label values which are obtained when using the features in the corresponding useful subset $U = F \setminus N$ only. Thus, for $v_{i,m}(N)$, we have:

$$v_{i,m}(N) = \hat{\pi}_m p(c_{i,U}|\hat{\theta}_{m,U}) \left(\sum_{j=1}^{k} \hat{\pi}_j p(c_{i,U}|\hat{\theta}_{j,U}) \right)^{-1}.$$

Then, Figueiredo et al. determine the "non-usefulness" of a feature by:

$$J(N) = \frac{1}{n} \sum_{i=1}^{n} \sum_{m=1}^{k} w_{i,m} \log \frac{w_{i,m}}{v_{i,m}(N)}, \tag{8.4}$$

with $J(N)$ being the smaller the larger is the conditional independence between y_N and the expected class labels given y_U. Hence, the aim of the feature selection task is to find the largest subset of non-useful features N such that $J(N)$ equals zero. If no feature subset exists which fulfills this condition, feature selection can be continued by moving the feature f_i from the useful to the non-useful feature set such that $J(N \cup f_i)$ is minimum.

8.3. Feature Subset Evaluation Criteria

The filter-based and wrapper-based feature selection approaches require the definition of the relevance criterion and evaluation measurement function J, respectively. The adopted techniques are introduced herein. Furthermore, the specific issues related to feature selection in the unsupervised case are addressed.

8.3.1. A Mutual Information-based Feature Selection Technique

A Combined Filter-Wrapper Approach

Motivated by the success of the mutual information feature selection strategy of Komma et al. [KZ09b] in the domain of supervised learning, the mutual information criterion is now considered to estimate the relevance of features with respect to the estimated labels generated by the Gaussian mixture. Problems arise due to the lack of class labels required for assessing the degree of statistical relevance. Analogous to the approach of Figueiredo et al. [FJL03], the estimated class labels obtained after the clustering process are considered instead. Given the set of estimated class labels, the mutual information-based feature selection scheme can be summarized as follows:

1. Perform the clustering given the data instances each consisting of the feature subset S.
2. Determine the estimated labeling of the data instances.
3. Calculate the mutual information between each feature and the estimated class labels.
4. Select the first m spectral components with the largest mutual information value.

Note, however, that the presented algorithm violates the definition of a filter-based feature selection technique. This is because a certain clustering model has to be defined which determines an estimate of the class labels rendering this approach dependent on the employed clustering method. Hence, the introduced algorithm can be regarded as a combined filter-wrapper technique rather than a true filter approach.

Embedding the Mutual Information Filter into a Backward Feature Selection Scheme

Considering the latter approach, the selection of the m spectral components with the largest mutual information value can equivalently be regarded as removing $d - m$ features from the feature set at once. Alternatively, it is also possible to apply the feature reduction process by iteratively decreasing the feature count by a predefined number. In so doing, the following backward feature selection scheme is obtained: starting with the complete set of features, step 1-3 of the above-introduced feature selection technique are applied. Instead of processing the forth step, r spectral components with the smallest mutual information value are removed from the current feature set. Then, these steps are iteratively applied until the current feature set reaches a user-specific size. To the best knowledge of the author, this approach has not been reported in literature and hence provides a novel means for the feature selection task in the unsupervised case.

8.3.2. Wrapper-based Techniques

Wrappers and Sequential Forward Search

In this work, candidate feature subsets for wrapper-based feature selection have been established using sequential forward selection (SFS) and sequential backward selection (SBS), either inserting or removing one feature at each iteration. Concerning SFS, the active feature subset A starts with the empty set, $A = \emptyset$, while the open feature set O consists of the complete set of available features, $O = F$. In each iteration, a single feature $o_i, o_i \in O$, is taken from the open set and concatenated with the active feature subset to form the candidate feature set $C = A \cup o_i$. This candidate feature subset is evaluated using a quality criterion function $J(C)$ yielding a measure for the performance of C. Repeating these steps for all entries of the open set, the optimal feature subset C_{opt} of size $|A|$+1 is defined to be the one providing the maximum $J(C)$. The corresponding optimal feature o_{opt} is then moved from the open feature set into the active one and the whole process is iteratively repeated until a desired size of the active feature set is reached.
Using different quality assessment functions $J(C)$, the evaluation of a feature subset follows the same scheme. At first, a clustering model is generated using the candidate feature subset C. Then, the clustering is employed to assess the feature subset quality. As for the quality assessment functions, a variety of criteria have been considered:

Sum of Squared Errors The sum of squared errors is the optimization criterion used during k-means clustering. It is defined by the sum of squared Euclidean distances d between the instances of a cluster j and the respective cluster center μ_j summed over the complete set of clusters $\in [1,k]$:

$$J_{\text{SSE}}(C) = \sum_{j=1}^{k} \sum_{x_i \in \text{ cluster } j} d(x_i, \mu_j).$$

Data Negative Log Likelihood This criterion defines the negative log likelihood of the data set given the cluster model with model parameters θ where the definition of the likelihood is given in (3.20) of Section 3.3.1:

$$J_{\text{LL}}(C) \equiv L = \sum_{i=1}^{n} \log p(x_i|\theta).$$

Temporally Coherent Data Likelihood In comparison with the previous criterion $J_{\text{LL}}(C)$, the data likelihood estimation technique of (6.4) introduced in Section 6.2.2 also incorporates temporal dependencies. Here, it is employed as subset quality criterion as:

$$J_{\text{MRF}}(C) = L_2 \equiv L_{\text{MRF}}(\theta, \pi, s, q) = \sum_i \Big[\log \sum_j p(x_i|c=j,\theta)\pi_{ij} \tag{8.5}$$

$$-\beta \left[\text{KL}(s_i||\pi_i) + \text{KL}(s_i||\pi_{N_i}) + H(s_i) \right] \tag{8.6}$$

$$-0.5 \left[\text{KL}(q_i||p_i) + \text{KL}(q_i||p_{N_i}) + H(q_i) \right] \Big]. \tag{8.7}$$

Temporally Coherent Cost Function In [GD08], Giguere et al. proposed the following cost function to find an optimal classifier with respect to its parameter set θ:

$$J_{\text{cost}}(C) \equiv \text{cost}(\theta) = \sum_{j=1}^{k} \frac{\sum_{i=1}^{n-1}(p(c=j|x_{i+1},\theta) - p(c=j|x_i,\theta))^2}{var(p(c=j|X,\theta))^2}, \tag{8.8}$$

where k denotes the number of terrain classes $i, i \in [1,k]$ to cluster and n is the number of data points $x_i, i \in [1,n]$. Equation (8.8) "strike[s] a balance between minimizing variations of classifier posterior probabilities over time, while simultaneously maintaining a wide distribution of posterior probabilities" [GD08].

Number of Incoherent Pairs The number of incoherent pairs criterion is derived from the temporally coherent cost function by considering the number of terrain transitions, i.e. situations in which the estimated class label $\arg\max_j p(c=j|x_i)$ differs from the label $\arg\max_j p(c=j|x_{i+1})$ which are required for traversing the path given by $\{x_i\}_{i=1}^n$. Hence, the proposed criterion is defined as:

$$J_{\text{NIP}}(C) = \sum_{i=1}^{n-1} 1 - \mathbb{I}\left(\arg\max_j p(c=j|x_{i+1},\theta), \arg\max_j p(c=j|x_i,\theta)\right).$$

Mutual Information The mutual information SFS approach makes use of the estimated labels $\{a_i\}_{i=1}^n$ generated after the clustering process. This labeling is employed to determine the mutual information between each candidate feature o_j from the open set O and the estimated class labels. The quality assessment function $J_{\text{MI}}(C)$ based on the mutual information criterion is then given by:

$$J_{\text{MI}}(C) \equiv J_{\text{MI}}(A \cup o_j) = -\text{MI}\left(\{a_i\}, \{o_{i,j}\}\right),$$

where $\{o_{i,j}\}$ denotes the set of realizations of feature j given n data instances. Finally, it has to be noted that all the introduced criteria $J_*(C)$ have been selected such that a decrease in the value of $J_*(C)$ indicates a rise in the assumption of observing a feature subset candidate which provides adequate clustering characteristics. In other words, the optimal feature subset of size C is defined to be the one which minimizes the respective cost function $J_*(C)$.

Wrappers and Sequential Backward Search

For the sequential backward search (SBS), the active feature set is initially set to the complete feature set F, $A = F$. Note that no open feature set O is required in the context of a sequential backward search. In each iteration, a candidate feature subset is generated by removing one feature $r_i, r_i \in A$ from the active set, $C_i = A \setminus r_i$, which is then applied to a quality criterion function $J(C_i)$. Considering each element of the active set a_i as a candidate for being removed, a ranking of these elements can be established with increasing value of $J(C_i)$. Given that a small value of $J(C)$ indicates the estimated insignificance of feature a_i with regard to the clustering task, the first m features of the ranking are removed from the active set. Repeating these steps until the active set reaches a predefined limit yields the final feature subset.
As for the quality criterion $J(C_i)$, the posterior-based "non-usefulness" function of (8.4) is advised yielding:

$$J_{\text{post}}(C) = \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{k} w_{i,m} \log \frac{w_{i,m}}{v_{i,m}(N)},$$

with v and w being posterior distributions as described above. Since $J_{\text{post}}(C)$ is the smaller the more irrelevant a feature subset is assumed, the feature subset selection scheme aims at the minimization of $J_{\text{post}}(C)$ at each iteration.

8.3.3. Feature Transformation

Several feature transformation techniques have been adopted to further assess the performance of the above-mentioned feature selection strategies including low energy, low dimensionality, MFCC-like, and principal component analysis approaches. Details of the considered techniques are presented in the following.

Low Energy Feature Selection The low energy feature selection approach (LE) is not a true instance of feature transformation techniques. This is because the selected feature subset consists of the first m spectral components while no composition of the latter is performed. Since the low energy feature selection scheme can be realized in terms of a $d \times m$ matrix

$$T = \{t_{i,j}\}, \text{ with } t_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

being multiplied with the data matrix X, $X \in \mathbb{R}^{n \times d}$, $X \cdot T$, this type of feature selection is assigned to the group of feature transformation techniques.

Low Dimensional Feature Transform The low dimensional feature transform (LD) is the only transform which is based on the unmodified 128 element vibration segment s instead of its 64-dimensional spectral representation. Here, the proposed feature vector x consists of a subset of the features originally advised in [WFZ06]: given a raw vibration segment s, the feature vector x is defined as:

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

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

logbinning The logbinning technique (LB) applies the MFCC-like feature transform as described in Section 5.3.2.

Principal Component Analysis Using principal component analysis (PCA), the resulting feature transform achieves both a reduction of the instance dimensionality and a linear feature decorrelation. Given the normalized data matrix $\tilde{X} \in \mathbb{R}^{n \times d}$, i.e. a matrix whose columns have a mean of 0 and a standard deviation of 1, the PCA yields a matrix U whose columns contain the eigenvectors e_i of $\tilde{X}$. The actual transformation is applied by first sorting the eigenvectors in decreasing order of their corresponding eigenvalues. Then, the first m eigenvectors are chosen from the reordered set which explain the largest amount of variance. Denoting this set by $\{e_i\}_{i=1}^{m}$ and arranging them column-wise in a matrix $E \in \mathbb{R}^{d \times m}$, the PCA transform is formally defined as $\tilde{X} \cdot E$.

8.4. Experimental Results

8.4.1. Experimental Setup

Both, the varying criteria for feature selection and the set of feature extraction schemes have been experimentally evaluated. The general setup with regard to vibration segment preprocessing, cluster model generation, clustering evaluation, and path generation schemes along with technique-specific parameters which have been applied during the experiments are presented in the following.

Vibration Segment Preprocessing With regard to the applied feature selection techniques, the set of candidate features comprises all 64 DFT amplitude spectrum components. For the feature extraction approaches, all techniques described in Section 8.3.3 have been adopted. Note that all generated vibration segment representations consist of a feature subset of size 6. This choice was made because of the good performance of the MFCC-like descriptor which also has a dimensionality of 6. Hence, the objective of this work is not only to find a valid feature subset with beneficial clustering characteristics but also to determine a subset of spectral components which contains as much features as the MFCC-like descriptor while providing the same clustering performance.

Clustering Model The clustering model was established by means of the MRF clustering technique proposed in Chapter 6.2.2. Its corresponding parameter was determined by the neighbor set size estimation technique (Section 6.3.2). To obtain a deterministic result for each path, the PCA-MRF partitioning scheme (Section 7.3.1) with a neighbor set size of 2 has been employed as cluster center initialization technique. Prior to the initialization step, outliers were detected and removed using the outlier removal approach of Section 7.3.2. Note that while the PCA-MRF partitioning scheme was based on the outlier-processed data set, the actual clustering considered the whole set of data instances.

Cluster Model Evaluation Criteria Analogous to the previous chapter, the true positive rate has been employed for accessing the clustering performance.

Path Generation Scheme The following experiments are based on robot traversals over three and five terrain classes. The considered paths correspond to those that have been adopted in the experiments of the previous chapters including natural paths as well as artificially generated ones (cf. Section 2.2). Since the main interest of this chapter is not to examine the influence of temporal dependencies on the clustering performance but to study the quality of varying feature selection and extraction techniques, the set of generated paths for the 5 classes experiments has been reduced. That is, only paths are considered which contain varying travel distances from 2 m up to 32 m, respectively.

8.4.2. Results and Discussion

Tables 8.1(a)-8.1(c) show the results of the 3 and 5 classes experiments for the applied feature selection techniques. Here, the clustering performance is given in terms of the true positive rate. Beside the presentation of the clustering quality measure at various robot driving speeds, the average over all velocity profiles is denoted. Further, Tables 8.1(a)-8.1(c) show the average over all trials of the 3 and 5 classes experiments at the bottom of the respective tables. Using

the latter average measure, the following ranking for the evaluation criteria of the forward feature selection technique (Table 8.1(a)) can be established: J_{fMI} (81.3%), J_{ICP} (80.7%), J_{MRF} (77.4%), J_{cost} (72.7%), and finally J_{SSE} (70.9%). The J_{fMI} is characterized by the second best results for the 3 classes experiments and the best results for the 5 classes experiments. Problems only arise with respect to robot traversals over 3 terrain classes at a speed of 0.2 m/s. The J_{ICP} criterion performs best with regard to the 3 classes experiments, yet fails to select an appropriate feature set for the 5 classes experiments. Considering the J_{cost} measure, the contrary characteristics can be observed. Here, the J_{cost} criterion yields the worst results for the 3 classes experiments with regard to all other criteria but the best ones for the 5 classes experiments. When applying the J_{MRF} approach, an appropriate clustering performance for the 3 classes experiments can be observed, yet only a mediocre one in the context of robot traversals over 5 terrain classes. The use of the J_{SSE} measure cannot be advised, since it yields true positive rates below average for all but 2 experiments.
The next part of the discussion considers the backward feature selection schemes (Table 8.1(b)). Regarding the J_{bMI} criterion, it does not show a significant sensitivity against the number of features which are removed from the active set in each iteration. With respect to the averaged results provided at the end of Table 8.1(b), the optimal number of features to be removed from the active set is 8. Note, however, that the results of the $J_{bMI_{56}}$ approach where all but 6 features are removed in a single iteration still yields an adequate clustering behavior. Furthermore, the true positive rates which are obtained with the $J_{bMI_{56}}$ criterion outperform the ones of the best forward feature selection approaches. The other backward feature selection scheme using the J_{post} criterion results in significantly worse outcomes in comparison with all instances of the J_{bMI} approach.
The ranking of the clustering performance with respect to varying feature extraction techniques is well-defined. The logbinning method performs better than PCA-transformed spectral components (92.9% vs. 87.5%), followed by the low energy (77.9%), and the low dimension descriptor (73.7%) approaches. Here, the number in brackets denotes the averaged results over both the 3 classes and 5 classes experiments. The logbinning technique provides the largest true positive rates or at least true positive rates which are close to the best ones. The PCA approach performs well for the 3 classes experiments, yet it does not yield an appropriate clustering model for robot traversals over 5 terrain classes at a robot speed of 0.2 m/s. The results obtained when adopting the low energy method and the low dimensional descriptor are diverse. As some experiements reveal their good performance with respect to the generation of an appropriate clustering model, other experiments do not confirm these findings.
Comparing the best approaches of varying feature selection and extraction techniques among each other, it can be stated that the feature extraction scheme outperforms backward feature selection. The latter, in turn, provides better results in comparison with the applied forward feature selection methods. Given this ranking, the alternative use of feature selection techniques instead of feature extraction methods becomes questionable. Note, however, that feature selection is superior to feature extraction under the following two point of views: computational complexity and descriptiveness. Concerning the computational complexity, it has to be noted that the logbinning and PCA feature extraction approach require the complete set of DFT coefficients to be determined in a first step. Only then, the final descriptor can be determined. In contrast, the feature selection approaches presented in this chapter choose a subset of 6 spectral component features. Hence, a reduced FFT can be applied which only determines the DFT coefficients under consideration, resulting in a significant decrease of computational complexity. The second benefit of feature selection is its descriptiveness. Since a novel feature obtained after the principal component analysis represents a linear combination of spectral components,

the original meaning which is assigned to the unmodified features gets lost. That is, in the context of terrain clustering, no conclusions with respect to the frequency-clustering performance correlation can be made. In contrast, feature selection approaches enable this kind of data interpretation as they remain the original features.

Table 8.1.: True positive rate [%] of the 3 and 5 classes experiments for varying velocity profiles when adopting (a) forward search feature selection, (b) backward search feature selection, and (c) feature extraction strategies.

(a)

#classes	vel	feature selection technique				
		SSE	MRF	cost	ICP	MI
3	0.2 m/s	61.6	73.1	66.8	**94.2**	70.7
	0.4 m/s	71.0	95.4	77.4	**97.7**	97.0
	0.6 m/s	94.1	94.7	68.4	**97.9**	96.2
	average	75.6	87.7	70.9	**96.6**	88.0
5	0.2 m/s	55.3	**66.7**	66.2	60.6	64.2
	0.4 m/s	**83.4**	69.4	82.1	72.1	77.1
	0.6 m/s	60.3	65.2	75.6	61.4	**82.4**
	average	66.3	67.1	**74.6**	64.7	74.6
average		70.9	77.4	72.7	80.7	**81.3**

(b)

#classes	vel	feature selection technique					
		MI_1	MI_2	MI_4	MI_8	MI_{56}	$post_8$
3	0.2 m/s	91.5	91.2	91.2	93.1	**96.3**	72.4
	0.4 m/s	95.8	95.2	95.2	94.6	95.6	**96.1**
	0.6 m/s	94.9	**97.0**	95.3	95.3	86.5	94.6
	average	94.1	**94.5**	93.9	94.3	92.8	87.7
5	0.2 m/s	60.8	64.6	65.3	**68.2**	63.3	55.6
	0.4 m/s	86.3	**92.6**	92.5	92.2	85.9	76.5
	0.6 m/s	80.2	79.4	79.4	79.4	**81.1**	67.9
	average	75.8	78.9	79.1	**79.9**	76.8	66.7
average		84.9	86.7	86.5	**87.1**	84.8	77.2

(c)

#classes	vel	feature extraction technique			
		LE	PCA	LD	LB
3	0.2 m/s	94.6	96.7	63.4	**97.1**
	0.4 m/s	82.9	95.7	90.3	**97.0**
	0.6 m/s	57.2	**97.6**	61.0	97.3
	average	78.2	96.7	71.6	**97.1**
5	0.2 m/s	**84.0**	64.2	61.9	82.0
	0.4 m/s	88.6	89.4	78.7	**93.7**
	0.6 m/s	60.0	81.3	86.7	**90.3**
	average	77.5	78.3	75.8	**88.6**
average		77.9	87.5	73.7	**92.9**

8.5. Conclusion

This chapter addressed the problem of feature selection in the context of unsupervised vibration segment clustering. Varying feature selection techniques have been examined including approaches based on sequential forward search and sequential backward search. For both subset search strategies, novel means have been proposed which are based on mutual information statistics. While the forward search uses the mutual information measure as quality criterion to evaluate the performance of a candidate subset in a wrapper-style manner, the proposed mutual information backward search combines the filter and wrapper methodologies. That is, given a set of candidate features, a clustering model is established using this subset. Then, the labels which are generated according to the clustering are used to determine the mutual information between each feature and the estimated labels. In a final step, a certain number of features is removed from the current set and the process is repeated until a given number of features is reached.
Experimental results showed the superiority of the mutual information feature subset quality criterion in comparison with all other candidates of the same feature subset search strategy. Comparing the clustering performance obtained when applying the proposed feature selection techniques with those of feature extraction methods, the former approaches yield competitive results in terms of the true positive rate while conserving the interpretability of the selected feature subset.

9. Estimating the Number of Mixture Components

9.1. Introduction

The clustering techniques in the previous chapters are based on the assumption that the number of terrain types (clusters) is known a priori. Note, however, that this assumption gives rise to problems if an adequate estimate of the cluster count can be obtained using prior knowledge, yet, the robot does not traverse all these terrain types. As a consequence, vibration data representing the same ground surface will be assigned to different clusters although they belong together. Furthermore, it is not always possible to generate a valid estimate of the present cluster count. Hence, autonomous approaches should be considered which try to determine whether a given cluster count k represents the data well or another cluster model should be chosen by altering the number of clusters.
Recently, several researchers have addressed the problem of automatically selecting a valid k. The estimated cluster count is either obtained by splitting or merging existing clusters, or performing both splitting and merging operations in an alternating way. The following techniques all have in common that they start from a minimum number of clusters and greedily increase the cluster size by one if the actual clustering model explains the data set worse than the extended model. For example, Tibshirani et al. [TWH01] introduces the gap statistic comparing the likelihood of a generated model with the distribution of the likelihood of models trained on data drawn from a null distribution. Although this approach works quite well for the selection of k when there is a small number of clusters contained in the data set, Feng et al. showed that the Gap statistic becomes worse with increasing cluster count [FH07]. A hierarchical technique was proposed by Pelleg et al. [PM00]. Starting from a cluster which contains all data instances, this cluster is recursively split into two subclusters whenever a better fit to the data is obtained. As splitting criterion, the Bayesian information criterion (BIC) [Sch78] is applied. The statistical significance of the BIC, however, has been questioned in [HE04]. There, it is claimed that the BIC is likely to generate overfitted models by generating too many cluster centers if the data does not emanate from a strictly spherical Gaussian distribution. Another hierarchical approach was introduced by Hamerly et al. [HE04]. There, the decision whether to split a cluster or not is based on the Anderson-Darling test [AD52]. The Anderson-Darling test is a statistical test which verifies if the data contained in a subcluster appears to be Gaussian distributed. If this is not the case, the cluster is bisected to improve the overall Gaussian fit. The g-means algorithm does not assume spherical clusters and provides good results if the true clusters are separated well. Yet, in the case of overlapping clusters as this is the case for vibration data, the g-means algorithm is reported to worsen its ability in finding a valid cluster count [FH07]. To alleviate the problem of overlapping clusters, Feng et al. [FH07] present the pg-algorithm. Similar to the g-means technique, it uses a statistical hypothesis test to decide whether a given cluster count represents the data set well. In contrast to the g-means algorithm, the pg-means technique performs this test on the entire model and not on a subcluster level. Feng et al. show the effectiveness of their approach in difficult cases as well, including non-Gaussian data, over-

lapping and eccentric clusters, and high dimensional data.
The term "clustering stability" refers to another group of techniques for assessing a valid cluster count estimate. These methods are based on the assumption that clustering algorithms generate stable clusterings with an appropriate k and unstable clusterings if k is chosen inappropriately. Yin et al. [YH09] extended this approach by introducing the measure of hierarchical stability. There, each model component is tested for stability and recursively split until a stability criterion is met. Similar to the latter approach, Ghorbani et al. [GO10] also employ a stability-based technique to detect whether to split an existing cluster. Yet, the splitting decision is based on an outlier detection technique assigning identified outliers to new centroids. Furthermore, Ghorbani et al. also allow for the fusion of proximate clusters if they are assumed to be generated from the same underlying process.
In the following section, the problem of extracting the cluster count from the present data is applied to the problem of vibration segment clustering. On the one hand, the novel contribution consists of a systematic comparison of the above-mentioned approaches. Here, the discussion focuses on the ability of the respective algorithms to cope with overlapping clusters as they are present in this domain. Furthermore, two external criteria are introduced which allow for an assessment of the clustering performance with respect to both robot traversal safety and the number of unnecessary driving mode transitions. This was rendered necessary as the cluster performance measures of Section 3.3.2 neither provide an intuitive evaluation of the generated cluster model nor an adequate penalization strategy of cluster refinements, i.e. the recursive separation of data instances which actually belong to the same terrain type.

9.2. Applied Techniques

Determining the number of components in a Gaussian mixture which is required for an appropriate representation of the data set is an important issue. If the number of components is chosen too large, the mixture model overfits the data. Selecting too few components, on the other hand, yields a model which is unable to represent the data well. Recently, researchers proposed several systematic means of establishing an adequate estimate of the component count such as the BIC-criterion, x-means, g-means, pg-means clustering, and consensus clustering techniques. An overview of these methods is provided in the following subsections.

9.2.1. The Bayesian Information Criterion

In literature, a variety of component count estimation techniques can be found which are Bayesian-motivated. Examples include the Bayesian information criterion (BIC) [Sch78, FR98], the integrated completed likelihood (ICL) [BCG00], the Akaike information criterion (AIC) [Aka74], and the deviance information criterion (DIC) [SBCL02]. Assessing the performance of the different approaches in a comparative study, Steele et al. [SR09] showed the superiority of the BIC approach with respect to the model selection problem for Gaussian mixture models. The Bayesian information criterion is a measure of evaluating the probability of a model M_j with parameter vector θ from a set of candidate models $\left\{M_j\right\}_{j=1}^{l}$ given the data set instances $X = \{x_i\}_{i=1}^{n}$, i.e. $p(M_j|X)$. Since there is no general means of accessing the model posterior $p(M_j|x_i)$, this probability is determined by applying the Bayes theorem:

$$p(M_j|X) = \frac{p(M_j)p(X|M_j)}{p(X)}.$$

Assuming the prior probability of a model, $p(M_j)$, to be equal for each model and rejecting the likelihood of X, it can be derived that the model posterior is proportional to the probability that the data is generated by the model M_j, $p(X|M_j)$. Note that the validity of the rejection of the data likelihood $p(X)$ relies on the fact that $p(X)$ is equal for each model. The probability of observing the data set X given the model M_j is determined by:

$$p(X|M_j) = \int p(X, \theta_j, M_j) p(\theta_j|M_j) d\theta_j,$$

which can be estimated using the Laplace approximation [Raf95, Rip96] as follows:

$$\log p(X|M_j) \approx \log p(X|\hat{\theta}_j, M_j) - \frac{1}{2} d(M_j) \log n.$$

Here, $\hat{\theta}_j$ denotes the maximum likelihood estimate of θ_j, $d(M_j)$ is the number of free parameters in the model M_j, and n is the number of training patterns. Using this result, Fraley et al. [FR98] define the BIC value of the model M_j given the training data set X as:

$$BIC(M_j, X) = 2 \log p(X|\hat{\theta}_j, M_j) - d(M_j) \log n.$$

The latter equation states that the larger the BIC value is, the larger is the evidence for the model. Related to the selection of the number of components, that means that the component count k has to be chosen which maximizes the BIC value.

9.2.2. x-Means Clustering

The x-means clustering algorithm [PM00] is a hierarchical approach. That is, it starts with a certain number of clusters which are then recursively refined during the clustering process. The refinement is achieved by splitting one existing cluster into two different subclusters. Therefore, it has to be determined whether a cluster has to be split and how the actual splitting operation is performed.

The x-means algorithm addresses the latter problem by replacing the centroid of a candidate cluster by two children and moving them a distance apart proportional to the size of the region in opposite directions along a randomly chosen vector. Then, the k-means algorithm with $k = 2$ is applied to the instances which are assigned to the candidate cluster. The resulting centroids become the centers of the newly created subclusters.

For choosing the cluster to split, the local BIC score of a candidate cluster and the BIC scores of its children which are generated when splitting the candidate cluster are determined. Since the x-means algorithm is based on the k-means algorithm for which the identical spherical Gaussian assumption holds, the BIC score $\mathrm{BIC}(M_j)$ of a single cluster j with respect to the complete k-component model is defined as:

$$\begin{aligned} \mathrm{BIC}(M_j) &= L_j(X) - \frac{p_j}{2} \cdot \log R, \\ L_j(X) &= -\frac{R_n}{2} \log(2\pi) - \frac{R_n \cdot d}{2} \log(\sigma^2) - \frac{R_n - k}{2} + R_n \log R_n - R_n \log R. \end{aligned}$$

Here, R denotes the number of data points of the whole data set, R_j is the number of points which belong to cluster j, d denotes the dimensionality of the data set, and p_j is the sum of the free parameters of the model. In this context, p_j is determined by $p_j = k - 1 + d \cdot k + 1$, for $k - 1$ class probabilities, $d \cdot k$ center coordinates, and one variance estimate. To determine

the local BIC score comprising only the candidate cluster or its children, both BIC(M_j) and BIC($M_{j,1}$) + BIC($M_{j,2}$) have to be calculated where BIC($M_{j,i}$) denotes the local BIC score of the ith child of candidate cluster j. The calculation of the 2-component BIC is based on the fact that the log-likelihood of the data instances which are assigned to the considered clusters is the sum of the log-likelihoods of the individual clusters. Further note that for the calculation of the local BIC score, R has to be replaced by the number of elements contained in the considered subclusters.
Finally, a candidate cluster is split if the sum of the local BIC scores of the child clusters is larger than the BIC score of their parent.

9.2.3. g-Means Clustering

The g-means algorithm [HE03] is similar to the x-means clustering approach in terms of splitting existing clusters. Yet, it employs another splitting criterion which subdivides a given cluster if the cluster's data instances do not follow a Gaussian distribution. For this test, Hamerly et al. adopted the Anderson-Darling statistic [AD52] which is a normality test based on the empirical cumulative distribution function (ECDF). This test is carried out in a one-dimensional subspace of the original input data since a univariate normality test is known to be a simpler task in comparison with its multivariate counterpart [Kaf03]. Given the data points $\{x_i\}_{i=1}^{n}$ assigned to cluster j, the following two alternative hypotheses can be formulated:

- $H_0 : \{x_i\}$ are sampled from a Gaussian distribution.
- $H_1 : \{x_i\}$ are not sampled from a Gaussian distribution.

Depending on whether H_0 is accepted or rejected, the cluster under consideration is either kept or split, respectively.
As previously noted, the Anderson-Darling test is a one-dimensional test statistic. Yet, the vibration patterns are high-dimensional rendering it necessary to project the data instances into a lower-dimensional space. Therefore, a direction has to be defined on which the data points are projected. The g-means algorithm therefore considers the two new cluster centers which emerge from the candidate cluster after splitting. The direction which is assumed to be important for the clustering task is chosen as the vector which connects the new cluster centers. For placing the new centers of the child clusters, Hamerly et al. used a deterministic approach in which the cluster center locations are determined by means of a k-means clustering algorithm ($k = 2$) applied to the data instances of the candidate cluster. The k-means algorithm is initialized with the cluster center of the candidate cluster c_i shifted by $\pm m$ units along a certain direction $\vec{v}$, $\hat{c}_{1,2} = c_i \pm m \cdot \vec{v}$. Here, the direction v is determined by the leading eigenvector with eigenvalue λ with respect to the data subset assigned to the candidate cluster. Finally, m is defined as $m = v\sqrt{2\lambda/\pi}$.

9.2.4. pg-Means Clustering

The projected Gaussian-means (pg-means) clustering approach [FH07] is based on a similar idea in comparison with g-means, yet the former can also cope with non-Gaussian and overlapped data sets. The term *projected Gaussians* is derived from the use of projections which are applied to both the clustering model and the data during hypothesis testing. Note that the projection of the complete model and data set is the key difference to the x-means and g-means approaches where the statistical tests are performed on a cluster-based level.

pg-means adopts Gaussian mixture models for clustering which are trained using the expectation maximization algorithm. Starting with $k = 1$ classes, the number of classes is increased by one in each iteration until enough evidence is gained that the observed data was generated by the current model. To verify the latter condition, the Kolmogorov-Smirnov (KS) test is applied which determines whether a sample $\{x_i\}_{i=1}^n$ originates from a specific distribution. Here, the respective hypotheses are formulated as follows:

- $H_0 : \{x_i\}$ are sampled from a mixture of Gaussians with k components.
- $H_1 : \{x_i\}$ are not sampled from a mixture of Gaussians with k components.

Since the KS-test is based on one-dimensional data only, both the Gaussian mixture model and the data set have to be projected into a one-dimensional space. In the approach of [FH07], the authors therefore adopt the technique of random projections [Das00].
In the following paragraph, both the random projection technique and the Kolmogorov-Smirnov test are explained in more detail.

Model and Data Projection

To verify the model fitness for a given value of k, the *pg*-means algorithm projects both the model and the data set into a one-dimensional space. Data projection yields several benefits: first, a mixture of Gaussians remains a mixture of Gaussians after the projection and second, testing the model fitness in one dimension is effective, efficient, and less involved than a test in higher dimensions.
In the following, a data set $X = \{x_i\}_{i=1}^n$ from a single Gaussian cluster j with distribution $X \sim N(\mu_j, \Sigma_j)$ in d dimensions is assumed. Here, μ_j denotes the $d \times 1$ mean vector and Σ_j is the $d \times d$ covariance matrix. Given a projection vector p of unit length, i.e. $\|p\| = 1$, the projection $\tilde{x}_i$ of a data instance x_i is given by $\tilde{x}_i = p \cdot x_i$. Further, the projected data set $\tilde{X}$ follows again a Gaussian distribution with mean μ'_j and standard deviation σ'_j, i.e. $\tilde{X} \sim N(\mu'_j, \sigma'_j)$ with $\mu'_j = p^T \mu_j$ and $\sigma'_j = p^T \cdot \Sigma_j \cdot p$. Applying the projection vector to the data set X and each component of the mixture, a one-dimensional representation of the original data and mixture model, respectively, is obtained. Using both projections, empirical and estimated cumulative distribution functions (CDFs) are determined which are then applied to the model fitness test described below.
For the choice of projection vectors, a random projection approach was considered [Das00]. Here, n_p different projection vectors $p_i \sim N(0, 1/dI)$ are generated randomly having approximately unit length in high dimensions. Feng et al. provide statistical justification of choosing $n_p \approx -2.62 \log(\varepsilon)$, where ε denotes the probability of only observing "bad" projections which are not capable of separating the cluster means.

The Kologorov-Smirnov Test and the Lilliefors Test

After data projection, the univariate Kolmogorov-Smirnov test [Mas51] is applied to verify the clustering model for a specific k. The KS test statistic is defined as

$$D = max_l |F(\tilde{x}_l) - S(\tilde{x}_l)|,$$

which is the maximum absolute difference between the true cumulative distribution function (CDF) $F(X)$ and the empirical cumulative distribution function (ECDF) $S(X)$. Along with the test statistic D, two other parameters are necessary to completely define the KS-test: the

significance level α and the critical value. The significance level α defines the probability of the α error, i.e. the erroneous rejection of H_0 if H_0 is valid. Further, the critical value is the threshold the value of the test statistic D is compared to. If the test statistic D is larger than the critical value, H_0 is rejected.
One issue with the KS-test is its requirement for a fully specified true CDF $F(X)$ which is, however, not available in this domain. To solve this problem, a modification of the KS-test, the Lilliefor's test [Lil67], is adopted. This test allows for the use of a CDF estimated from the data set by modifying the critical values. The test hypotheses, on the other hand, remain the same. Feng et al. employed the following analytic approximation for deriving the critical values which also takes test repetitions with regard to the Bonferroni adjustment [Bon35] into account:

$$\alpha = \exp\left(-7.01 D_{cv}^2(\tilde{n}+2.78) + 3.00 D_{cv}(\tilde{n}+2.78)^{1/2} - 1.22 + \frac{0.97}{\tilde{n}^{1/2}} + \frac{1.68}{\tilde{n}}\right), \quad (9.1)$$

$$\tilde{D}_{cv} = \begin{cases} D_{cv} & \text{for } n \in [5,100] \\ D_{cv} \cdot (n/100)^{0.49} & \text{otherwise} \end{cases}, \quad (9.2)$$

where n is the sample size, $\tilde{n}$ denotes the saturated sample size, $\tilde{n} = \min(100, n)$, and α is a user-defined confidence level. Furthermore, $\tilde{D}_{cv}$ represents the approximated critical value. Note that, in its original form, the Lilliefor's test is intended for the correction of the critical values for CDFs which arise from estimated univariate Gaussian distributions. In [FH07], Feng et al. showed, however, that these corrections also apply to multivariate Gaussian distributions and distributions from mixtures of Gaussians as well when they are projected to a one-dimensional space.

9.2.5. Mixture Component Count Estimation Using Consensus Clustering

Consensus clustering [MTMG03, SG03] provides a means of unifying the knowledge obtained from several clustering processes. Thereby, varying knowledge can be generated by subsampling the present data set, employing data from differing sensors, or altering the clustering technique. Given a set of clustering solutions, consensus clustering techniques aim at integrating the obtained data labelings into a final clustering which has beneficial characteristics in terms of robustness and quality. Vinh et al. [VEB09] adopted this framework in the domain of estimating the number of clusters k in a data set. Their method is based on the assumption that a valid k minimizes the discrepancy between the clusterings obtained by different algorithms or different runs of a single algorithm. Given a candidate value for k, $\hat{k}$, along with a set of b clustering solutions, $U_k = \{U_i\}_{i=1}^b$, Vinh et al. define the consensus index (CI) of U_k as:

$$\mathrm{CI}(U_k) = \sum_{i<j} \mathrm{AM}(U_i, U_j),$$

where the agreement measure AM is an appropriate clustering index. In other words, the consensus index CI quantifies the average agreement between all pairs of clustering solutions in the clustering set U_k. The larger the value of CI, the larger is the stability of the clustering process and hence the larger is the probability that k was chosen accurately. Finally, the optimal cluster count is the one which maximizes the consensus index.

9.3. Mixture Component Count Estimation for Vibration Fingerprint Clustering

The BIC-criterion and the *x*-means, *g*-means, and *pg*-means clustering techniques can be applied to any data set without modifications. Yet, the consensus clustering techniques requires a definition of how varying clustering results, i.e. differing labelings of the data instances, are obtained. In this approach, varying clustering solutions are generated by altering the representation of the vibration data. The employed representations are discussed in the following subsection.

9.3.1. Diversifying the Clustering Input Data Set

The results of the previous chapter reveal that varying vibration data representations yield a differing clustering performance with respect to the classification error. Since the discrepancy in the classification error is an effect of diverse clustering solutions, the selection of varying vibration data representations for the generation of differing clustering solutions is proposed. As for the representations, the mutual information backward feature selection and several feature extraction techniques have been considered. In comparison with the feature extraction techniques presented in the last chapter, three additional approaches have been considered: PCA-MI, MI-binning, and PCA-MI-binning.

PCA-MI The PCA-MI approach follows the mutual information-based approach introduced in Section 8.3.1. The key difference is the process of generating the initial model being responsible for assigning a mutual information score to each variable. Instead of relying the model generation process on unmodified 64-dimensional DFT amplitude spectra, a lower dimensional representation of these input vectors is employed. This lower dimensional representation of the spectral components is obtained by PCA transforming the input signals using a set of six eigenvectors which explain the largest amount of variance.

MI-binning The MFCC-like preprocessing approach of Section 5.3.2 is based on the assumption that the importance of the spectral components decreases with increasing frequency. In contrast, the proposed MI-binning technique determines this importance statistically by calculating the mutual information between each spectral component and the determined set of labels analogously to the mutual information-based feature selection approach of Section 8.3.1. Then, the spectral components are rearranged in order of decreasing mutual information. Applying the MFCC-like binning technique of 5.3.2 with the same logarithmic scale, a set of bins is obtained each containing the more elements the less is the amount of mutual information assigned to its members. Replacing these members by their average yields the final 6-dimensional representative of the vibration pattern.

PCA-MI-binning Note that the determination of the initial clustering model for estimating the mutual information score in the MI-binning approach relies on the complete 64-dimensional DFT amplitude spectrum. Substituting these input patterns by a lower dimensional representation results in the PCA-MI-binning technique. The actual dimensionality reduction process was applied by means of PCA using a set of six eigenvectors with the largest eigenvalues.

(a) (b)

Figure 9.1.: (a) The correct and (b) the erroneous subdivision of a 2-class data set. Although the solution of (b) can be regarded as a refinement of the true clustering an evaluation of the ARI and AMI performance measures only yield values of 0.57 and 0.58, respectively.

9.3.2. Evaluation of the Clustering Results

Evaluation in the Context of Traversal Safety

The adjusted rand index and the adjusted mutual information clustering quality measures introduced in Section 3.3.2 can also be applied when the cluster counts k^{true} and k^{est} of the true and estimated clustering solutions differ. Note, however, that in the context of robot traversal safety, it is arguable whether the determination of the true clustering is of primary interest. To support this claim, a clustering of an artificial two terrain setting is depicted in Figure 9.1. Whereas one terrain class is correctly represented by a single cluster A, the other terrain class is separated into two distinct subclusters B_1 and B_2 (cf. Figure 9.1(a)). As an effect, the ARI and the AMI indices penalize the solution shown in the second image too strongly although the estimated clustering only represents a refinement of the true one. In the given example, the evaluation by means of the ARI and AMI indices yields outcomes of 0.57 and 0.58, respectively. To alleviate this effect, an aggregation step is proposed prior to the cluster evaluation. In this aggregation step, all clusters are merged whose majority of instances belong to the same class in relation to the true data labeling. In this context, the class majority for a given cluster l is defined in terms of the cluster purity which denotes the maximum relative frequency of a certain class j, $\{x_i \in \text{class } j | x_i \in \text{ cluster } l\}$, with respect to the total number of instances $|X|$ of cluster l:

$$\text{Purity}(\text{cluster } l) = \max_j \frac{\{x_i \in \text{class } j | x_i \in \text{ cluster } l\}}{|X|}$$

The class providing the majority of instances within cluster C_l is given by:

$$\text{Majority}(\text{cluster } l) = \arg\max_j \frac{\{x_i \in \text{class } j | x_i \in \text{ cluster } l\}}{|X|}.$$

Here, $|\cdot|$ denotes the number of elements contained in the respective set and class j is the true label of instance x_i. The agglomeration process starts by labeling the instances of the generated clusters according to their true class membership. Then, for each class j the set of clusters $\mathscr{C}_j \subset \{\text{cluster } 1, \ldots, \text{cluster } m\}$ is determined for which $\text{Purity}(\text{cluster } i) > 0.95$ and $\text{Majority}(\text{cluster } i) = \text{class } j$, $i \in [1, m]$. Note that while each class j represents a class from the reference data set, cluster i denotes an estimated cluster obtained after model generation.

For the reason of clarity, the set of class labels of the reference data set is referred to as $\pi = \{\pi_i\}_{i=1}^{k}$ and the set of cluster labels corresponding to the estimated clustering with k^{est} clusters is denoted by $\rho_i = \{\rho\}_{i=1}^{k^{\text{est}}}$. Given the set of clusters $\mathscr{C}_j$, the instances of the majority class j are assigned a new and unique label $\pi_j \notin \pi$, $\pi_j \notin \rho$. Finally, the label of all other instances is remapped to the one generated by the clustering process, that is, to a label contained in the ρ set.

Evaluation in the Context of Driving Mode Transition Reduction

If the main objective is the reduction of necessary driving mode transitions rather than robot traversal safety, the quality of an appropriate cluster count estimation procedure becomes an important issue. These situations arise when all of the expected terrain classes are assumed to be non-hazardous, yet, each terrain type requires its own specific driving style. In this context, the probability of an unnecessary driving mode transition rises with an increasing refinement of a certain terrain cluster, that is, the oversegmentation of a larger cluster representing a single terrain into several smaller ones. The driving mode transition is termed unnecessary, since in this case, the set of smaller clusters represent the same terrain type along with the same ground surface characteristics and hence does not require a change in the driving behavior. To account for a too detailed refinement of the presented terrain, an external measure is proposed which is based on the cluster purity index introduced above. This external measure quantifies the number of clusters in which vibration signatures from a certain terrain class represent the majority with respect to all other classes of the reference data set. Formally, this measure is defined as:

$$\text{Split}(\text{class } j) = \sum_{i=1}^{k^{\text{est}}} \mathbb{I}(\text{Majority}(\text{cluster } i), \text{class } j),$$

where k^{est} denotes the cluster count of the estimated clustering and class j is the jth class of the reference data set.

9.4. Experimental Results

9.4.1. Experimental Setup

The choice of the experimental setup was based on the results of the previous chapters, considering the techniques which yielded the best clustering performance. That is, for all cluster count estimation approaches which rely on the outcomes of an initial clustering step, the temporally coherent MRF clustering scheme along with the 2-component PCA-MRF cluster center initialization technique have been adopted. The corresponding parameter of the initialization technique, the neighbor set size, was assigned a value of 2 as advised in the latter two chapters. If not stated otherwise, the MFCC-like descriptor was chosen as vibration segment representation providing the best clustering performance with regard to the considered feature selection and feature extraction methods (cf. Chapter 8). And finally, the employed paths correspond to those which have been adopted in the experiments of the previous chapter including robot traversals over 3 and 5 classes, respectively.
In this work, all the cluster count estimation techniques introduced in Section 9.2 have been implemented, including the BIC approach, x-means clustering, g-means clustering, pg-means clustering, and the consensus clustering techniques. The choice of the respective parameters and implementation details are presented in the following paragraphs.

BIC There are no parameters assigned to the Bayesian information criterion.

***x*-means** Pelleg et al. [PM00] do not provide a detailed description of the cluster splitting process leaving the concrete implementation of their algorithm to the user. In this work, a candidate cluster is bisected by first selecting two instances of the data subset in a random manner yielding u and v. The splitting axis is then defined as the straight line passing through the selected points. Given the normalized orientation of this line $\vec{o}$ along with the standard deviation σ of the candidate cluster, two novel points are determined by $p_{1,2} = \vec{m} \pm 4\sigma\vec{o}$ where $\vec{m}$ denotes the center between u and v. In the following, $p_{1,2}$ are employed as initial cluster centers for a succeeding k-means clustering step with $k = 2$. The clustering which arises from the latter step represents the outcome of the cluster bisection procedure.

***g*-means** The g-means approach requires the significance level α to be specified. The value α denotes the probability of erroneously rejecting the hypothesis that the data instances around the center are sampled from a Gaussian distribution. For its derivation, it has to be noted that the g-means cluster count estimation procedure is based on multiple tests hence demanding a Bonferroni adjustment [Mil91]. Following the suggestion of Hamerly et al. [HE03], α was set to $\alpha = 0.0001$.

***pg*-means** For projecting the Gaussian mixture model to a one-dimensional subspace, a total of 18 projection directions has been considered. Thereby, these directions where sampled at random as detailed in Section 9.2.4. Further, the Lilliefors test was applied at a significance level of $\alpha = 0.001$, hence determining the required critical value D_{cv} by means of (9.2).

consensus clustering The implemented consensus clustering technique employs the differing preprocessing schemes of Section 9.3.1 to generate a variety of clustering solutions. For the agreement measure AM, both the adjusted rand index and the mutual information criterion have been adopted resulting in the ARI consensus clustering ($cons_{ARI}$) and AMI consensus clustering ($cons_{AMI}$) approaches. Each clustering applies the respective steps as explained in the introduction given above.

Regarding the evaluation of the obtained data partitionings, not only the number of estimated clusters was taken into account but also the quality of the obtained clusterings. As quality criteria, the adjusted rand index and the mutual information measure have been employed. Note that the use of the true positive rate is not applicable any more, since the number of clusters of the reference and estimated partitions may vary. In the result section, both criteria are presented denoting the clustering performance of the generated models before and after the aggregation procedure (Section 9.3.2). The term "generated models" does not necessarily relate to the models which are established during the cluster count estimation procedure. For the BIC and pg-means approaches, it proved to be beneficial to disregard temporal dependencies during the generation of the model which estimates the number of clusters k. In a second step, a temporally coherent MRF clustering was adopted with k representing its hyperparameter. The same applies to the consensus clustering techniques which do not establish a single clustering but multiple data partitions instead. As for the vibration segment representation, the MFCC-like descriptor has been used in this context. The outcomes of the x-means and the g-means algorithms, on the other hand, which claim to yield an appropriate clustering by themselves, are applied unalteredly to the proposed evaluation schemes.

9.4.2. Results and Discussion

Results of the 3 Classes Experiments

Table 9.1 shows the results of the cluster number estimation experiments along with the clustering performance which is obtained when evaluating the generated partitions. For the evaluation process, paths are considered which contain a total of three terrain classes and which are traversed at velocity profiles 1-3, ranging from 0.2 m/s up to 0.6 m/s. As clustering performance criteria, both the adjusted rand index and the adjusted mutual information measures have been adopted. Here, the discrimination is made between the original clustering and a clustering with a succeeding aggregation step as described in Section 9.3.2. Further, the last 6 rows denote the results when averaging the outcomes over all robot driving speeds. With regard to the latter measure, the consensus clustering terrain count estimation technique yields the best clustering performance for both the adjusted rand index and the adjusted mutual information criterion. Here, the estimated cluster count does not vary with respect to the chosen evaluation criterion ($cons_{ARI}$, $cons_{AMI}$) yielding equivalent outcomes when evaluating the respective clustering models. The second best cluster number derivation method is the BIC technique, followed by the *pg*-means, *g*-means, and *x*-means approaches. It is noticeable that the obtained clustering of the *g*-means approach is significantly worse than the one provided by the *BIC* technique although both estimation approaches yield the same number of clusters. This is because the latter technique performs a temporally coherent MRF clustering after the cluster number estimation step. For the *g*-means method, on the other hand, the data partition is kept which arises during the estimation procedure.
The cluster aggregation technique is only applied to a robot driving speed of 0.2 m/s for the BIC, *x*-means, and *g*-means approaches. The largest decline in the estimated cluster count is obtained for the *x*-means technique. Here, this number decreases from 23 to 10 cluster. The results indicate that the respective approaches recursively subdivide a pure cluster rather than generating subclusters which contain various terrain types. In this context, a pure cluster is defined as a cluster which contain elements of the same terrain at a majority of 95%.
Table 9.2 presents the number of clusters in which the considered terrain classes represent the bare majority. It reveals that at driving speeds of 0.2 m/s and 0.4 m/s, the *x*-means, *g*-means, and *pg*-means techniques show the characteristics of subdividing the cluster which contains instances from the asphalt class. Further, for velocity profile 0.2 m/s and terrain count estimation technique *x*-means the grass cluster is subdivided into 16 subclusters. This indicates that the distribution of grass instances does not follow a Gaussian distribution with spherical shape.

Table 9.1.: The estimated cluster count along with the clustering performance in terms of the ARI and AMI quality measures for the 3 classes experiments when varying the velocity profile. Here, the results before (none) and after (applied) the cluster aggregation step are presented.

vel	aggregation	criterion	k estimation technique					
			bic	x-means	g-means	pg-means	cons_{ARI}	cons_{AMI}
0.2 m/s	none	k^{est}	5	23	5	4	**3**	**3**
		ARI	0.34	0.12	0.24	0.51	**0.91**	**0.91**
		AMI	0.42	0.22	0.34	0.57	**0.87**	**0.87**
	applied	k^{est}	4	10	4	4	**3**	**3**
		ARI	0.52	0.17	0.27	0.51	**0.91**	**0.91**
		AMI	0.51	0.26	0.37	0.57	**0.87**	**0.87**
0.4 m/s	none	k^{est}	4	4	4	4	**3**	**3**
		ARI	0.62	0.38	0.37	0.62	**0.91**	**0.91**
		AMI	0.58	0.38	0.40	0.58	**0.87**	**0.87**
	applied	k^{est}	4	4	4	4	**3**	**3**
		ARI	0.62	0.38	0.37	0.62	**0.91**	**0.91**
		AMI	0.58	0.38	0.40	0.58	**0.87**	**0.87**
0.6 m/s	none	k^{est}	**3**	4	**3**	2	**3**	**3**
		ARI	**0.93**	0.31	0.36	0.37	0.91	0.91
		AMI	**0.89**	0.29	0.38	0.39	0.86	0.86
	applied	k^{est}	**3**	4	**3**	2	**3**	**3**
		ARI	**0.93**	0.31	0.36	0.37	0.91	0.91
		AMI	**0.89**	0.29	0.38	0.39	0.86	0.86
average	none	k^{est}	4	10.33	4	3.33	**3**	**3**
		ARI	0.63	0.27	0.32	0.50	**0.91**	**0.91**
		AMI	0.63	0.30	0.37	0.51	**0.87**	**0.87**
	applied	k^{est}	3.67	6	3.67	3.33	**3**	**3**
		ARI	0.69	0.28	0.33	0.50	**0.91**	**0.91**
		AMI	0.66	0.31	0.38	0.51	**0.87**	**0.87**

Table 9.2.: The number of clusters in which a certain terrain type represents the bare majority when varying the velocity profile along with the cluster count estimation technique. All results denote the outcomes of the 3 classes experiments.

vel	terrain	k estimation technique					
		bic	x-means	g-means	pg-means	cons_{ARI}	cons_{AMI}
0.2 m/s	asphalt	2	6	2	2	1	1
	pavement	2	1	1	1	1	1
	grass	1	16	2	1	1	1
0.4 m/s	asphalt	2	2	2	2	1	1
	pavement	1	1	1	1	1	1
	grass	1	1	1	1	1	1
0.6 m/s	asphalt	1	2	1	1	1	1
	pavement	1	1	1	0	1	1
	grass	1	1	1	1	1	1

Results of the 5 Classes Experiments

The presentation of the results of the 5 classes experiments follows the one of the 3 classes experiments. Concerning the averaged clustering performance of the respective approaches presented in the latter rows of Table 9.3, the *pg*-means approach yields the best outcomes with respect to both the adjusted rand index and the adjusted mutual information criterion. The best performing techniques of the 3 classes experiments, $cons_{ARI}$ and $cons_{AMI}$, are positioned third place excelled by the BIC cluster number estimation approach. This statement proves to be valid for both techniques with respect to their adopted quality criteria, $cons_{ARI}$ and $cons_{AMI}$, respectively. The *g*-means and *x*-means techniques denote the cluster count estimation techniques providing the lowest clustering quality.
For robot driving speeds above 0.2 m/s, a general trend of overestimating the cluster count size can be observed. This overestimation is distinctive most when adopting the *x*-means clustering technique and less noticeable for the *g*-means method. The BIC and *pg*-means approaches, on the other hand, yield cluster counts which are close to but still larger than the correct ones. The consensus clustering derivation schemes are characterized by underestimation for all velocity profiles. Note that this is likely to decrease the robot safety during the robot traversal, since the aggregation of varying terrain types into a single cluster prevents their distinguishability. As one of these ground surfaces might increase its hazardousness due to a change in weather conditions, both terrain types are nevertheless processed in the same manner. This either results in an over- or underestimation of the present robot traversal safety and hence an inappropriate velocity profile is chosen. Note that an underestimation of the present terrain type count can also be observed for all but the *x*-means and *pg*-means techniques at a robot driving speed of 0.2 m/s.
Concerning the aggregation technique, it is applied in 7 out of 18 experiments. The decline in the number of estimated clusters, however, is rather insignificant. This is a reason of the class number derivation techniques themselves yielding outcomes which are close to the real ones. The only exception is the *x*-means technique which significantly overestimates the number of clusters. Referring to Table 9.4, this overestimation originates mainly from the grass terrain class. In addition, the clusters containing asphalt instances and less significantly PVC floor instances are partitioned in smaller subclusters as well. With regard to both consensus clustering techniques, the asphalt class and clay terrain classes are underrepresented for the 0.2 m/s velocity profile while the PVC floor instances do not obtain a majority with respect to their assigned cluster in the 0.6 m/s experiments.

Table 9.3.: The estimated cluster count along with the clustering performance in terms of the ARI and AMI quality measures for the 5 classes experiments when varying the velocity profile. Here, the results before (none) and after (applied) the cluster aggregation step are presented.

vel	aggregation	criterion	k estimation technique					
			bic	x-means	g-means	pg-means	cons_{ARI}	cons_{AMI}
0.2 m/s	none	k^{est}	**4.67**	23	4	6	3	3
		ARI	0.47	0.15	0.38	**0.48**	0.42	0.44
		AMI	0.56	0.30	0.43	**0.58**	0.49	0.51
	applied	k^{est}	**4.67**	9	4	5.58	3	3
		ARI	0.47	0.34	0.38	**0.51**	0.42	0.44
		AMI	0.56	0.37	0.43	**0.58**	0.49	0.51
0.4 m/s	none	k^{est}	7.50	24	16	7	**5**	**5**
		ARI	0.74	0.26	0.34	0.74	**0.84**	**0.84**
		AMI	0.74	0.39	0.44	0.75	**0.85**	**0.85**
	applied	k^{est}	6.08	14	13	5.50	**5**	**5**
		ARI	0.84	0.51	0.49	0.84	**0.84**	**0.84**
		AMI	0.82	0.49	0.50	0.84	**0.85**	**0.85**
0.6 m/s	none	k^{est}	**5**	18	7	**5**	3.58	4
		ARI	**0.82**	0.29	0.56	**0.82**	0.61	0.68
		AMI	**0.82**	0.41	0.57	**0.82**	0.65	0.71
	applied	k^{est}	**5**	11	7	**5**	3.58	4
		ARI	**0.82**	0.40	0.56	**0.82**	0.61	0.68
		AMI	**0.82**	0.46	0.57	**0.82**	0.65	0.71
average	none	k^{est}	**5.72**	21.67	9	6	3.86	4
		ARI	0.67	0.23	0.43	**0.68**	0.63	0.65
		AMI	0.71	0.37	0.48	**0.72**	0.66	0.69
	applied	k^{est}	**5.25**	11.33	8	5.36	3.86	4
		ARI	0.71	0.42	0.48	**0.72**	0.63	0.65
		AMI	0.73	0.44	0.50	**0.75**	0.66	0.69

Table 9.4.: The number of clusters in which a certain terrain type represents the bare majority when varying the velocity profile along with the cluster count estimation technique. All results denote the outcomes of the 5 classes experiments.

vel	terrain	k estimation technique					
		bic	x-means	g-means	pg-means	cons_{ARI}	cons_{AMI}
0.2 m/s	PVC	1	2	1	1	0.83	0.92
	asphalt	0.67	2	1	1.58	0.17	0.08
	gravel	0.50	1	0	0.92	0.83	1
	grass	1	16	1	1.42	1	1
	clay	1.50	2	1	1.08	0.17	0
0.4 m/s	PVC	1	5	3	1	1	1
	asphalt	1.50	8	5	1	1	1
	gravel	1	2	2	1	1	1
	grass	2	7	3	2	1	1
	clay	2	2	3	2	1	1
0.6 m/s	PVC	1	2	1	1	0.33	0.25
	asphalt	1	2	1	1	0.67	0.75
	gravel	1	1	1	1	1	1
	grass	1	9	1	1	1	1
	clay	1	4	3	1	0.58	1

9.5. Conclusion

In this chapter, several techniques have been thoroughly examined for autonomously estimating the number of clusters contained in a given data set. This problem is important to consider, since many clustering schemes such as the k-means and Gaussian mixture model-based techniques require this parameter to be specified a priori.
The results obtained after evaluating the generated models in terms of the adjusted rand index and the adjusted mutual information measure did not prove to be consistent. In a series of experiments with 3 classes, a consensus clustering scheme outperformed all other techniques, correctly determining the true cluster count at various robot driving speeds. Here, the cluster count estimation scheme is based on establishing a sequence of differing clustering solutions at a fixed k. In a succeeding step, an average consensus index is derived over all pairs of generated labelings. The estimated number of classes then becomes the k^{est} which maximizes the consensus index. In another series of experiments, the paths represented robot traversals over 5 terrain types. Here, the best cluster count estimation method is based on a combined projection-testing approach. Given a certain Gaussian mixture model with k components, this model along with the instances of the data set are projected into a one-dimensional space. Then, the obtained empirical cumulative distribution function of the projected data is tested against the cumulative distribution function of the projected model in terms of a one-dimensional Lilliefors test. The chosen cluster count k^{est} is the minimum k for which the Lilliefors test passes with regard to a given number of random projections. Although both the consensus clustering technique and the pg-means approach yielded adequate clusterings with respect to the 3 classes and 5 classes experiments, respectively, these findings were not confirmed when adopting these algorithms in the other experimental setting. For example, if the pg-means technique is applied in the 3 classes experiments, both under- and overestimation of the correct number of terrain types occurred. The use of the consensus clustering scheme in the 5 classes experiments, on the other hand, resulted in a significant underestimation of the true cluster count in 2 of 3 experimental settings.
As further contribution, two quality measures were introduced to evaluate the performance of the resulting cluster models. The first one penalizes the refinement of a larger cluster into smaller subclusters less in comparison with existing approaches. This quality measure was motivated by the fact that in the context of robot traversal safety, a cluster refinement does not increase the degree of hazardousness. Note that in this case, each subcluster represents a single terrain type which requires a well-defined driving style. This is in contrast to non-uniform subclusters which comprise instances from varying terrain classes and hence agglomerate driving styles of differing characteristics. With regard to the driving style transition frequency, however, a possible cluster refinement becomes an issue as each subcluster may be assigned a varying driving behavior. Hence, a second quality measure was proposed providing information about the splitting tendency of certain ground surfaces.

10. Conclusion

10.1. Thesis Summary

This thesis addressed the problem of terrain discrimination using inertial sensors in the context of supervised and unsupervised learning. As for the inertial measurements, acceleration data acquired during robot navigation has been employed which are known to provide varying characteristics depending on the traversed terrain class. For both domains, the main contribution comprises the inclusion of temporal coherences which are contained in succeeding measurements. The developed techniques are based on the assumption that these measurements were acquired on the same terrain type assigning a smaller probability to terrain transitions and a larger one to situations in which the ground surface does not change from one measurement to the next.

For the task of terrain classification where labeled training data is available, previous research focused on point estimates only, considering the ground surface with the maximum posterior probability. In contrast, the proposed technique takes the distribution of class posteriors over the complete set of terrain types into account. These class posteriors are then filtered over time such that the current prediction does not only depend on a single measurement but a history of observations. The actual filtering process is realized by means of a Bayes filter which allows for a recursive terrain class discrimination with regard to all measurements acquired from the beginning of the robot traversal up to the current time. The application of the original Bayes filter formulation introduces the problem of high-dimensional density estimation, which is known to be a non-trivial task due to the curse of dimensionality. To overcome this issue, a Bayes filter reformulation was proposed which substituted the likelihood of single observations by the respective class posteriors. This modification facilitates the overall filtering process, since these class posteriors are already available after each point estimate of the ground surface. Experimental results using data which were collected during a robot traversal showed the superiority of the proposed temporally coherent prediction scheme in comparison with a single observation approach where the ground surface estimate relies on an individual measurement only. These experiments comprised both natural paths of robot traversals over 3 classes and artificially generated ones with 5 classes where preprocessed blocks of acceleration signals were concatenated in a systematic manner.

Note that the Bayes filter approach only relies on the estimation of class posteriors of individual measurements. Since varying classification techniques provide means of deriving the latter probabilities, different class posterior estimation approaches were embedded into the Bayesian prediction framework. A thorough investigation showed that all considered classifiers benefit from the inclusion of temporal dependencies. As a further contribution, the random forest classifier and the random ferns classification techniques have been adopted in the domain of terrain discrimination. Since the calculation of posterior probabilities represent the key element of Bayesian filtering, special care has been taken with regard to a classifier's capability of determining appropriate class posteriors. In this context, a succeeding posterior probability calibration step for the random ferns approach has been proposed. Using the calibration method, the posterior probabilities of true positives are assigned larger values, hence increasing the con-

fidence of correct predictions. Finally, the investigation also comprised the introduction of a novel preprocessing scheme which borrows its key characteristics from the Mel-frequency cepstrum approach. Using this kind of preprocessing, the dimensionality of the obtained feature vector is decreased to 6 in comparison with the 64 dimensions of the DFT amplitude spectrum feature extraction scheme which was previously considered for the terrain discrimination task. With regard to the experimental results, several classifiers benefited from the shortened feature descriptor such as the Gaussian mixture classifier approach. The latter finding is notable as the machine learning models described in later chapters of this thesis were represented in terms of Gaussian mixtures.

The second part of this work focused on terrain discrimination in the domain of unsupervised learning. Compared to the supervised case, there are no labeled instances available which can be used for the generation of a predictive model. Instead, a clustering approach aims at autonomously grouping acquired acceleration signals into clusters such that observations with similar characteristics are assigned to the same cluster while observations with varying characteristics are positioned in different clusters. For this task, a graphical model approach based on Markov random fields has been considered which is also capable of integrating temporal dependencies. Here, temporal coherences are exploited by constraining the distributions of class priors and posteriors to be similar within a neighborhood of given size. To estimate the latter, a general means was proposed which relies on multivariate outlier detection techniques. It showed that the neighborhood size estimation technique yields appropriate clustering results in the case of both high-frequency and low-frequency terrain changes. Furthermore, it is able to identify situations where there is no temporal coherence provided. Comparing the temporally coherent Markov random field approach with a Gaussian mixture model-based clustering technique, the former significantly outperforms the latter in terms of clustering performance. Note that for assessing the clustering performance, a total of three quality measures have been adopted: measures based on pair counting, mutual information and classification performance. The feature extraction investigations of Chapter 5 revealed the superiority of a more compact representation of acceleration signals in the context of Gaussian mixture model generation. Note, however, that the Mel-frequency cepstrum-based feature extraction scheme results in a loss of interpretability as the novel features are determined by an averaging process of spectral frequency amplitude magnitudes each representing varying energy content. Hence, there is no possibility to recover the contribution of each frequency subband to the quality of the clustering process. The latter information, however, is necessary to infer whether a good clustering emanates from smaller or larger amplitudes contained in the acceleration signal. To overcome this problem, varying feature selection approaches were introduced which selected a subset of spectral components each representing a certain frequency content. Experimental results revealed that a novel approach based on mutual information significantly outperformed all other considered techniques and yielded comparable results with regard to the Mel-frequency cepstrum-based vibration signal representation scheme. This approach relied on a backward search strategy removing a certain number of feature candidates at each iteration. For the removal, those features are considered which have the minimum amount of mutual information with the set of estimated target labels.

The clustering model employed in this work was based on Gaussian mixture models. One issue related to this kind of clustering techniques is that the final model parameters after training depend on their initialization. Each initial choice might result in another outcome representing a varying local optimum with respect to the criterion being optimized during the cluster model training process. Note that this problem complicates the feature selection task: given a clustering model of bad performance it is impossible to state whether the improper model behavior

emanates from an inappropriate model initialization or an unfavorable feature subset. Hence, this thesis also focused on a deterministic means of initializing Gaussian mixture model-based clustering techniques. The proposed model is a hierarchical one which recursively bisects the data in a top-down manner. After choosing the cluster to split and a separating hyperplane which divides the cluster into two subspaces, the cluster instances are assigned to one of the generated subclusters. The respective means of the two instance sets are then defined as the novel centers of the clustering solution replacing the center of the divided cluster. Experiments showed that the clustering model benefits from a further cluster center refinement process realized in terms of a 2-means clustering step. Several techniques have been considered for the latter where the application of a temporally coherent clustering approach yielded the largest improvements in the clustering quality.
The final investigation focused on the estimation of terrain classes contained in an unlabeled data set. This work proved to be necessary as Gaussian mixture model-based clustering approaches require the number of classes to be known a priori. The complexity of this problem can be inferred from the experimental results as no approach was able to estimate the terrain class count correctly in all situations: while the Bayesian information criterion performs well for the 5 classes experiments but results in an overestimation of the terrain class count in the 3 classes experiments, an approach based on consensus clustering correctly determined the number of classes in the 3 classes experiment but underestimated this number when the robot traversal paths contained 5 classes. For the latter technique, the consensus was established using the outcomes from varying models which are based on the same MRF clustering approach but varying preprocessing schemes. Thereby, the set of preprocessing techniques consisted of differing feature extraction strategies all providing a feature representation with 6 dimensions. As a further contribution, two novel quality assessment criteria have been proposed which evaluate the generated clustering with regard to traversal safety and driving mode transition frequency.

10.2. Outlook

Besides the investigation of the effects on incorporating temporal dependencies into a classification and clustering framework, this thesis provides means of increasing the autonomy of mobile outdoor robots in comparison with existing approaches. The augmented autonomy renders it unnecessary to train a discriminative terrain model a priori as the preprocessed acceleration patterns are distinctive enough to generate an appropriate clustering model by their own without the need of labeled instances. It is arguable whether a mobile robot requires the information about the actual terrain type it is navigating on. More importantly, a machine learning model has to group instances which provide the same characteristics with regard to the degree of hazardness. As shown in the experimental sections of this work, the generated clusterings follow the natural distribution of preprocessed acceleration signals. Note, however, that this thesis sees itself as a waypoint to the full degree of autonomy of a mobile robot rather than having reached this goal already. This is because the presented techniques require the complete robot traversal being finished at the time of model generation. An incremental model generation process would be preferable allowing for the estimation of traversal safety at the time of data acquisition. The approaches of Weiss et al. [WZ08] and Bouveyron [Bou10] can be regarded as a first step into this direction, yet, both approaches start with a known set of classes which is extended during the robot traversal. An interesting method is advised by Arandjelović et al. [AC05] proposing a Gaussian mixture model-based framework which is incrementally updated

by one instance at a time. Note, however, that, so far, there is no experimentally-supported evidence that this method can also be applied in the domain of acceleration signal clustering.
Another issue of the inertial measurement-based approaches is that the data is acquired using an expensive sensor which has a scheduled price of more than 1000€. This renders the application of the proposed techniques impossible in low-cost domains. It remains unclear, however, whether the full resolution provided by the Xsens MTi sensor is necessary to enable the terrain discrimination task. For example, the Wii Motion Plus sensor[1] offers similar facilities at a price of approximately 14€. Hence, further research should focus on the choice of adequate, yet economic sensors to increase the application range of inertial measurement-based terrain discrimination techniques.

[1] http://www.nintendo.com/wii/console/accessories/wiimotionplus

A. Appendix

A.1. Further Results of the MRF-based Clustering Approach

A.1.1. Results of the 3 Classes Experiments

Table A.1.: Adjusted rand index for the 3 classes experiments when adopting the GMM-based and the MRF-based clustering techniques.

vel	init	approach	filter size					
			2	4	8	16	32	$\overline{\mathscr{S}}$
0.2 m/s	random	gmm	0.53	0.52	0.52	0.52	0.52	0.52
		mrf	0.74	0.77	0.82	0.86	0.88	**0.89**
	k-means	gmm	0.52	0.52	0.52	0.52	0.52	0.52
		mrf	0.75	0.79	0.84	0.88	0.91	**0.91**
0.4 m/s	random	gmm	0.63	0.62	0.63	0.61	0.63	0.63
		mrf	0.75	0.74	0.79	0.83	0.88	**0.89**
	k-means	gmm	0.64	0.64	0.64	0.64	0.64	0.64
		mrf	0.76	0.77	0.81	0.88	0.90	**0.90**
0.6 m/s	random	gmm	0.32	0.31	0.31	0.30	0.31	0.30
		mrf	0.44	0.42	0.45	0.45	**0.57**	0.51
	k-means	gmm	0.38	0.37	0.38	0.39	0.39	0.39
		mrf	0.55	0.53	0.64	**0.68**	0.67	0.67
average	random	gmm	0.49	0.48	0.49	0.48	0.49	0.48
		mrf	0.64	0.64	0.69	0.71	**0.78**	0.76
	k-means	gmm	0.51	0.51	0.51	0.51	0.51	0.52
		mrf	0.68	0.70	0.76	0.81	0.83	**0.83**

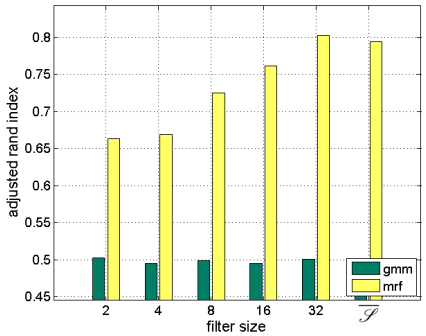


Figure A.1.: Adjusted rand index (ARI) for the 3 classes experiments and *k*-means model initialization when averaging the outcomes over the complete set of velocity profiles.

Table A.2.: Adjusted mutual information index for the 3 classes experiments with respect to varying velocity profiles, mixture model initialization schemes, and filter sizes when adopting the GMM-based and the temporally coherent MRF-based clustering techniques.

vel	init	approach	filter size					
			2	4	8	16	32	$\overline{\mathscr{S}}$
0.2 m/s	random	gmm	0.55	0.55	0.55	0.55	0.55	0.55
		mrf	0.72	0.74	0.79	0.84	**0.85**	0.85
	k-means	gmm	0.55	0.55	0.55	0.55	0.55	0.55
		mrf	0.73	0.76	0.81	0.85	**0.87**	0.87
0.4 m/s	random	gmm	0.63	0.62	0.63	0.61	0.63	0.63
		mrf	0.73	0.72	0.77	0.80	0.85	**0.85**
	k-means	gmm	0.64	0.64	0.64	0.64	0.64	0.64
		mrf	0.74	0.74	0.78	0.85	**0.87**	0.87
0.6 m/s	random	gmm	0.36	0.34	0.34	0.34	0.35	0.33
		mrf	0.49	0.45	0.50	0.50	**0.60**	0.54
	k-means	gmm	0.40	0.39	0.40	0.41	0.40	0.41
		mrf	0.54	0.53	0.63	**0.66**	0.66	0.66
average	random	gmm	0.52	0.50	0.51	0.50	0.51	0.50
		mrf	0.65	0.64	0.69	0.71	**0.77**	0.75
	k-means	gmm	0.53	0.53	0.53	0.53	0.53	0.53
		mrf	0.67	0.68	0.74	0.79	**0.80**	0.80

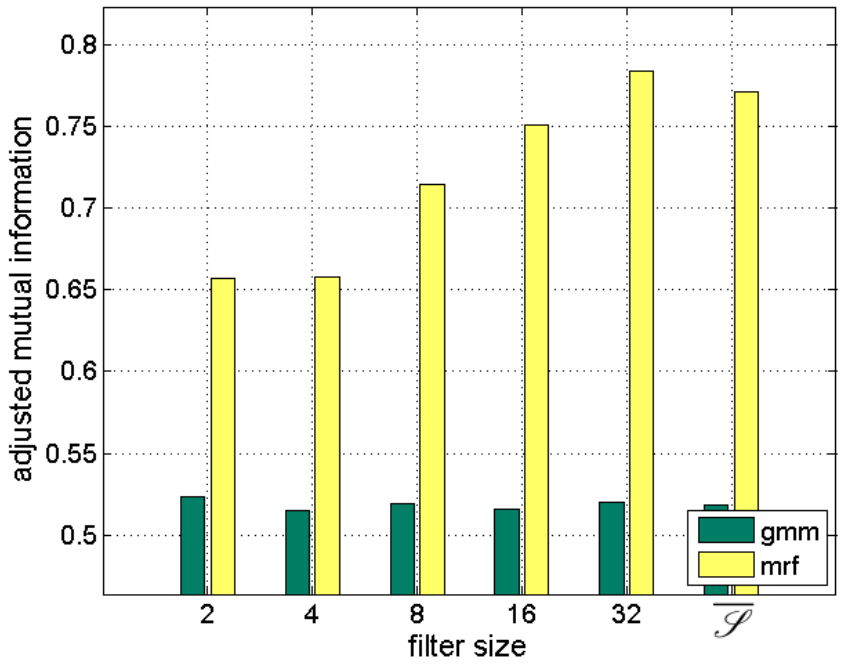


Figure A.2.: Adjusted mutual information index for the 3 classes experiments and *k*-means model initialization with respect to varying filter sizes when averaging the outcomes over the complete set of velocity profiles.

A.1.2. Results of the 5 Classes Experiments

Table A.3.: Adjusted rand index for the 5 classes experiments with respect to varying velocity profiles, mixture model initialization schemes, and filter sizes when adopting the GMM-based and the temporally coherent MRF-based clustering techniques.

init	dist	filter size					
		2	4	8	16	32	$\overline{\mathscr{S}}$
random	gmm	0.55	0.55	0.55	0.55	0.55	0.55
	0	0.02	0.02	0.01	0.01	0.01	**0.53**
	2	0.61	**0.62**	0.55	0.40	0.21	0.62
	4	0.64	0.65	0.65	0.59	0.42	**0.66**
	8	0.66	0.67	0.67	0.68	0.62	**0.69**
	16	0.66	0.67	0.69	0.71	**0.73**	0.72
	var	0.65	0.66	0.68	0.68	0.68	**0.71**
	average	0.54	0.55	0.54	0.51	0.45	**0.65**
k-means	gmm	0.54	0.55	0.54	0.55	0.54	0.54
	0	0.02	0.02	0.01	0.01	0.01	**0.54**
	2	0.60	0.60	0.55	0.40	0.21	**0.61**
	4	0.63	0.64	0.63	0.58	0.41	**0.64**
	8	0.64	0.65	**0.68**	0.67	0.61	0.66
	16	0.65	0.69	0.67	0.71	**0.72**	0.71
	var	0.65	0.65	0.66	**0.70**	0.67	0.67
	average	0.53	0.54	0.53	0.51	0.44	**0.64**

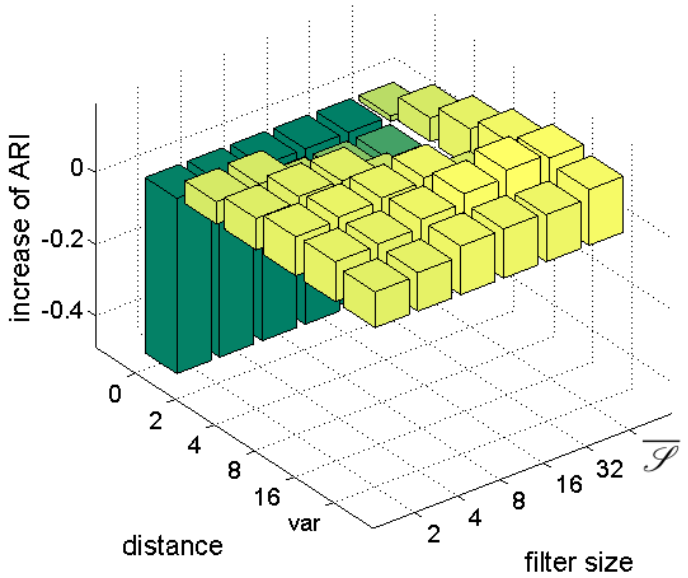


Figure A.3.: Increase in the clustering performance in terms of the adjusted rand index for the 5 classes experiments and *k*-means model initialization when averaging the outcomes over the complete set of velocity profiles and travel distances.

Table A.4.: Adjusted mutual information index for the 5 classes experiments with respect to varying velocity profiles, mixture model initialization schemes, and filter sizes when adopting the GMM-based and the temporally coherent MRF-based clustering techniques.

init	dist	filter size					
		2	4	8	16	32	$\overline{\mathscr{S}}$
random	gmm	0.61	0.61	0.61	0.60	0.61	0.61
	0	0.04	0.04	0.03	0.03	0.04	**0.60**
	2	0.67	**0.67**	0.61	0.47	0.28	0.67
	4	0.69	0.71	0.70	0.65	0.49	**0.71**
	8	0.71	0.72	0.73	0.74	0.68	**0.74**
	16	0.71	0.72	0.74	0.76	**0.78**	0.77
	var	0.70	0.71	0.73	0.74	0.73	**0.75**
	average	0.59	0.60	0.59	0.56	0.50	**0.71**
k-means	gmm	0.60	0.61	0.60	0.61	0.61	0.60
	0	0.04	0.04	0.03	0.03	0.04	**0.61**
	2	0.66	**0.66**	0.62	0.47	0.29	0.66
	4	0.68	0.70	0.69	0.64	0.49	**0.70**
	8	0.70	0.71	**0.73**	0.73	0.67	0.72
	16	0.71	0.74	0.73	0.76	**0.77**	0.76
	var	0.70	0.70	0.72	**0.75**	0.72	0.73
	average	0.58	0.59	0.58	0.56	0.50	**0.70**

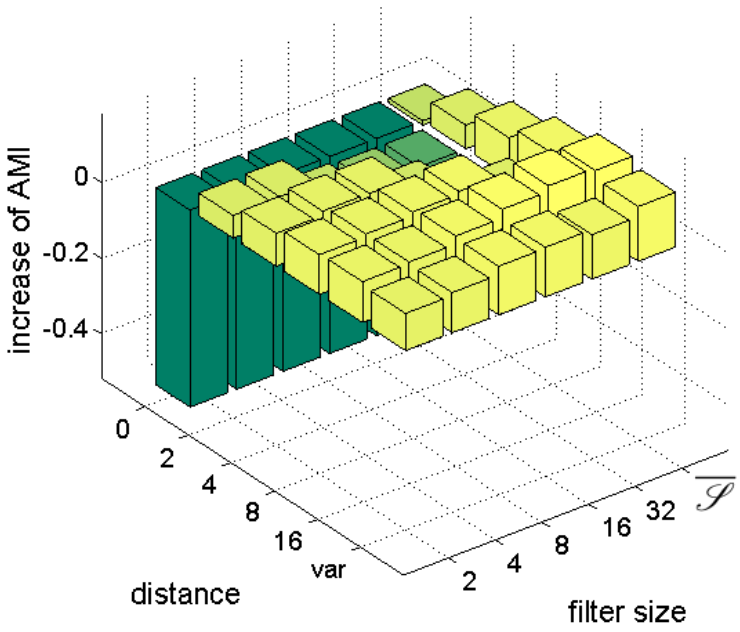


Figure A.4.: Increase in the clustering performance in terms of the adjusted mutual information index for the 5 classes experiments and *k*-means model initialization with respect to varying filter sizes when averaging the outcomes over the complete set of velocity profiles and travel distances.

Bibliography

[AC05] O. Arandjelovic and R. Cipolla. Incremental learning of temporally-coherent gaussian mixture models. In W.F. Clocksin, A.W. Fitzgibbon, and P.H.S. Torr, editors, *The 16th British Machine Vision Conference (BMVC '05)*, pages 25–32, Oxford, UK, September 2005. British Machine Vision Association.

[AD52] T.W. Anderson and D.A. Darling. Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952.

[Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Transaction on Automatic Control*, 19(6):716–723, 1974.

[AMH+06] A. Angelova, L. Matthies, D.M. Helmick, G. Sibley, and P. Perona. Learning to predict slip for ground robots. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2006 (ICRA '06)*, pages 3324–3331, Orlando, FL, USA, 2006.

[AMHP07a] A. Angelova, L. Matthies, D. M. Helmick, and P. Perona. Fast terrain classification using variable-length representation for autonomous navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pages 1–8, Minneapolis, MN, USA, 2007.

[AMHP07b] A. Angelova, L. Matthies, D.M. Helmick, and P. Perona. Learning slip behavior using automatic mechanical supervision. In *Proceedings of the IEEE International Conference on Robotics and Automation 2007 (ICRA '07)*, pages 1741–1748, Roma, Italy, April 2007. IEEE.

[AP05] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17:203–215, February 2005.

[AR81] N. Ahuja and A. Rosenfeld. Mosaic models for textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(1):1–11, 1981.

[Asa99] A. Asano. Texture analysis using morphological pattern spectrum and optimization of structuring elements. In *10th International Conference on Image Analysis and Processing (ICIAP' 99)*, pages 209–214, Los Alamitos, CA, USA, 1999. IEEE Computer Society.

[Bar01] E.I. Barakova. An integration principle for multimodal sensor data based on temporal coherence of self-organized patterns. In *Proceedings of the 6th International Work-Conference on Artificial and Natural Neural Networks: Bio-inspired Applications of Connectionism-Part II (IWANN '01)*, pages 55–63, London, UK, 2001. Springer-Verlag.

[Bar09] A. Barbu. Learning real-time MRF inference for image denoising. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pages 1574–1581, 2009.

[BB89] R.C. Bolles and A.F. Bobick. Exploiting temporal coherence in scene analysis for autonomous navigation. In *IEEE International Conference on Robotics and Automation (ICRA '89)*, volume 2, pages 990–996, Scottsdale, AZ, May 1989.

[BBMG09] A.R. Bates, A.S. Bijral, J. Mulligan, and G. Grudic. Traversable path identification in unstructured terrains: a Markov random walk approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pages 4394–4401, Piscataway, NJ, USA, 2009. IEEE Press.

[BCG90] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73, January 1990.

[BCG00] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:719–725, July 2000.

[BCG03] C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4):561–575, January 2003.

[Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.

[Bes86] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.

[BG70] D. Blackwell and M. A. Girschik. *Theory of Games and Statistical Decisions*. New York: Wiley, 1970.

[BI05] C. A. Brooks and K. Iagnemma. Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics*, 21(6):1185 – 1191, 2005.

[BI07] C. A. Brooks and K. Iagnemma. Self-supervised terrain classification for planetary rovers. In *Proceedings of the NASA Science Technology Conference*, pages 1–8, 2007.

[BI09] C.A. Brooks and K. Iagnemma. Visual detection of novel terrain via two-class classification. In *Proceedings of the ACM Symposium on Applied Computing (SAC '09)*, pages 1145–1150, New York, NY, USA, 2009. ACM.

[BID05] C. A. Brooks, K. Iagnemma, and S. Dubowsky. Vibration-based terrain analysis for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pages 3426–3431, Barcelona, Spain, 2005.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BJ79] D. Birch and G.H. Jacobs. Spatial contrast sensitivity in albino and pigmented rats. *Vision Research*, 19(8):933–937, 1979.

[BL11] N. Brodu and D. Lague. 3D terrestrial LiDAR data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. July 2011.

[Bon35] C.E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. *Studi in Onore del Professore Salvatore Ortu Carboni*, pages 13–60, 1935.

[Bou10] C. Bouveyron. Adaptive mixture discriminant analysis for supervised learning with unobserved classes. June 2010.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, August 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45:5–32, October 2001.

[BSR05] S. Blackmore, B. Stout, and B. Runov. Robotic agriculture - the future of agricultural mechanisation? In *Proceedings of the 5th European Conference on Precision Agriculture*, pages 621–628, 2005.

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[BV94] C.W. Baum and V.V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6):1994–2007, November 1994.

[BZM07] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *IEEE 11th International Conference on Computer Vision (ICCV '07)*, pages 1–8, October 2007.

[CC08] E. Coyle and E. Collins. A comparison of classifier performance for vibration-based terrain classification. In *26th Army Science Conference*, pages 1–7, Orlando, FL, USA, December 2008.

[CCJD+08] E. Coyle, E. G. Collins Jr, E. DuPont, D. Ding, H. Wang, R. A. Cooper, and G. Grindle. Vibration-based terrain classification for electric powered wheelchairs. In *Proceedings of the IASTED Conference on Telehealth and Assistive Technologies*, pages 139–144, Baltimore, USA, 2008. ACTA Press.

[CCL10] E. Coyle, E.G. Collins, and Liang Lu. Updating control modes based on terrain classification. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pages 4417–4423, Anchorage, AK, May 2010.

[CD07] P. Cunningham and S.J. Delany. k-nearest neighbour classifiers. Technical report, UCD School of Computer Science and Informatics, 2007.

[CKL96] L.W. Chan, I. King, and K.S. Leung. Levenberg-marquardt learning and regularization. *Progress in Neural Information Processing*, pages 139–144, 1996.

[CKW94] T. M. Chin, W.C. Karl, and A.S. Willsky. Probabilistic and sequential computation of optical flow using temporal coherence. *IEEE Transactions on Image Processing*, 3(6):773–788, 1994.

[CL05] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines*. Department of Computer Science and Information Engineering, National Taiwan University, Taipeh 106, Taiwan, 2005.

[Com94] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, April 1994.

[CS02] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, March 2002.

[CTL06] C. Constantinopulus, M.K. Titsias, and A. Likas. Bayesian feature and model selection for Gaussian mixture models. *IEEE Transanctions on Pattern Analysis and Machine Intelligence*, 28:1013–1018, 2006.

[Das00] S. Dasgupta. Experiments with random projection. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI '00)*, pages 143–151, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[DB04] J.G. Dy and C.E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, December 2004.

[DFR04] D. DeCarlo, A. Finkelstein, and S. Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering (NPAR '04)*, pages 15–145, New York, NY, USA, 2004. ACM.

[DGM02] T. Downs, K.E. Gates, and A. Masters. Exact simplification of support vector solutions. *Journal of Machine Learning Research*, 2:293–297, 2002.

[Die02] T.G. Dietterich. Machine learning for sequential data: A review. In T. Caelli, A. Amin, R.P.W. Duin, M.S. Kamel, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science, pages 227–246. Springer, 2002.

[DKSM95] I.L. Davis, A. Kelly, A. Stentz, and L. Matthies. Terrain typing for real robots. In *Proceedings of the Intelligent Vehicles Symposium*, pages 400–405, 1995.

[DMCJC08] E. M. DuPont, C. A. Moore, E. G. Collins Jr., and E. Coyle. Frequency response method for terrain classification in autonomous ground vehicles. *Autonomous Robots*, 24(4):337–347, 2008.

[DMR08] E. M. DuPont, C.A. Moore, and R. G. Roberts. Terrain classification for mobile robots traveling at various speeds: An eigenspace manifold approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pages 3284–3289, Pasadena, CA, USA, 2008.

[DMRCJ05] E. M. DuPont, C.A. Moore, R. G. Roberts, and E. G. Collins Jr. Terrain classification using probabilistic neural networks. In *Proceedings of the Florida Conference on Recent Advances in Robotics*, Gainesville, Florida, USA, May 5-6 2005.

[DMS+05] E. M. DuPont, C. A. Moore, M. Selekwa, E. G. Collins Jr., and R. G. Roberts. Online terrain classification for mobile robots. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, pages 1643–1648, Orlando, Florida, USA, November 5-11 2005.

[DRM06a] E. M. DuPont, R. G. Roberts, and C. A. Moore. The identification of terrains for mobile robots using eigenspace and neural network methods. In *Proceedings of the Florida Conference on Recent Advances in Robotics (FCRAR '06)*, pages 25–26, Miami, FL, USA, 2006.

[DRM06b] E. M. DuPont, R. G. Roberts, and C. A. Moore. Speed independent terrain classification. In *Proceedings of the 37th Southeastern Symposium on System Theory (SSST '06)*, pages 240–244, Cookeville, TN, USA, March 5-7 2006.

[DVG07] 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.

[DVH04] C. Dima, N. Vandapel, and M. Hebert. Classifier fusion for outdoor obstacle detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pages 665 – 671, New Orleans, LA, USA, 2004.

[EG91] B. Eisenberg and B.K. Ghosh. The sequential probability ratio test. *Handbook of Sequential Analysis*, pages 47–66, 1991.

[EHS+07] A. Erkan, R. Hadsell, P. Sermanet, J. Ben, U. Muller, and Y. LeCun. Adaptive long range vision in unstructured terrain. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pages 2421–2426, San Diego, CA, USA, 2007.

[EKL+02] A. Ellery, C. Kolb, H. Lammer, J. Parnell, H. Edwards, L. Richter, M. Patel, J. Romstedt, D. Dickensheets, A. Steele, and C. Cockell. Astrobiological instrumentation for Mars - the only way is down. *International Journal of Astrobiology*, 1(4):365–380, 2002.

[Ell08] A. Ellery. Space robotics: Robotic rovers for planetary exploration. *International Journal of Advanced Robotic Systems*, 1(4):303–307, 2008.

[FC97] J. Fernandez and A. Casals. Autonomous navigation in ill-structured outdoor environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97)*, volume 1, pages 395–400, Grenoble, France, 1997.

[FH07] 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.

[FJL03] M.A.T. Figueiredo, A.K. Jain, and M.H.C. Law. A feature selection wrapper for mixtures. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA '03)*, pages 229–237, 2003.

[FR98] C. Fraley and A.E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[FS95] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[GD08] P. Giguere and G. Dudek. Clustering sensor data for terrain identification using a windowless algorithm. In *Proceedings of Robotics Science and System (RSS '08)*, pages 25–32, Zürich, Switzerland, June 2008.

[GD09] P. Giguere and G. Dudek. Surface identification using simple contact dynamics for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, ICRA'09, pages 3045–3050, Piscataway, NJ, USA, 2009. IEEE Press.

[GDP+09] P. Giguere, G. Dudek, C. Prahacs, N. Plamondon, and K. Turgeon. Unsupervised learning of terrain appearance for automated coral reef exploration. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 268–275, Washington, DC, USA, 2009. IEEE Computer Society.

[GFB10] T. Grundmann, M. Fiegert, and W. Burgard. Probabilistic rule set joint state update as approximation to the full joint state estimation applied to multi object scene analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pages 2153–0858, Taipei, Taiwan, October 2010.

[GG87] S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[GJW82] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen. The complexity of the generalized Lloyd-Max problem. *IEEE Transactions on Information Theory*, 28(2):255–255, 1982.

[GO10] A. Ghorbani and I.-V. Onut. Y-Means: An autonomous clustering algorithm. In M. Graña Romay, E. Corchado, and M. Garcia Sebastian, editors, *Hybrid Artificial Intelligence Systems*, volume 6076 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin, Heidelberg, 2010.

[GRB00] G. Giacinto, F. Roli, and L. Bruzzone. Combination of neural and statistical algorithms for supervised classification of remote-sensing images. *Pattern Recognition Letters*, 21:385–397, May 2000.

[HA85] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[HALP11] M. Häselich, M. Arends, D. Lang, and D. Paulus. Terrain classification with markov random fields on fused camera and 3d laser range data. In *Proceedings of the 5th European Conference on Mobile Robotics (ECMR '11)*, pages 1–6, 2011.

[Har09] A. Harwood. First look: 2010 land rover LR4. http://www.motortrend.com/roadtests/suvs/112_0904_2010_land_rover_lr4_first_look/index.html, April 2009. (08/04/2011).

[HAZ+06] J. Hipp, E. Arabzadeh, E. Zorzin, J. Conradt, C. Kayser, M.E. Diamond, and P. König. Texture signals in whisker vibrations. *Journal of Neurophysiology*, 95(3):1792–1799, March 2006.

[HCHE05] E.R. Hruschka, T.F. Covoes, E.R. Jr. Hruschka, and N.F.F. Ebecken. Feature selection for clustering problems: a hybrid algorithm that iterates between k-means and a Bayesian filter. In *Fifth International Conference on Hybrid Intelligent Systems (HIS' 05)*, pages 405–410, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[HE03] G. Hamerly and C. Elkan. Learning the k in k-means. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS)*, pages 281–288, Vancouver, BC, Canada, 2003.

[HE04] G. Hamerly and C. Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems (NIPS '03)*, pages 1–8. MIT Press, 2004.

[HN03] X. He and P. Niyogi. Locality preserving projections. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 (NIPS '03)*, pages 1–8, Cambridge, MA, 2003. MIT Press.

[HRV11] M. Hubert, P.J. Rousseeuw, and T. Verdonck. A deterministic algorithm for the MCD. Technical Report TR-10-01, Department of Mathematics, Katholieke Universiteit Leuven, 2011.

[HSD73] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

[ID02] 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*, volume 4715, pages 256–266, Orlando, FL, USA, 2002.

[Jaz70] A.H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in Science and Engineering. Academic Press, New York, NY, 1970.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, September 1999.

[JWL+06] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics*

and the 44th annual meeting of the Association for Computational Linguistics, pages 209–216, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[JZ96] D. Jung and A. Zelinsky. Whisker based mobile robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)*, volume 2, pages 497–504, Osaka , Japan, November 1996.

[Kaf03] K. Kafadar. Testing for normality. *Journal of the American Statistical Association*, 98:765–765, 2003.

[KFDB07] P. Komma, J. Fischer, F. Duffner, and D. Bartz. Lossless volume data compression schemes. In *Simulation and Visualization (SimVis '07)*, pages 169–182, Magdeburg, Germany, March 2007.

[KH95] C. Kervrann and F. Heitz. A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics. *IEEE Transactions on Image Processing*, 4:856–862, 1995.

[KKBZ11] Y. Khan, P. Komma, K. Bohlmann, and A. Zell. Grid-based visual terrain classification for outdoor robots using local features. In *IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS '11)*, pages 16–22, Paris, France, April 2011.

[KKZ94] I. Katsavounidis, C.C. Jay Kuo, and Z. Zhang. A new initialization technique for generalized Lloyd iteration. *IEEE Signal Processing Letters*, 1(10):144–146, October 1994.

[KKZ11] Y. Khan, P. Komma, and A. Zell. High resolution visual terrain classification for outdoor robots. In *IEEE ICCV Workshop on Challenges and Opportunities in Robot Perception*, Barcelona, Spain, November 2011. To appear.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international Joint Conference on Artificial Intelligence, Volume 2*, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[KPS09] A. Krebs, C. Pradalier, and R. Siegwart. Comparison of boosting based terrain classification using proprioceptive and exteroceptive data. In O. Khatib, V. Kumar, and G. Pappas, editors, *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 93–102. Springer Berlin, Heidelberg, 2009.

[Krz88] W. J. Krzanowski. *Principles of multivariate analysis: a user's perspective*. Oxford University Press, New York, NY, USA, 1988.

[KS96] D. Koller and M. Sahami. Toward optimal feature selection. Technical Report 1996-77, Stanford InfoLab, February 1996.

[KS07] E. Kokiopoulou and Y. Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2143–2156, 2007.

[KSG04] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, June 2004.

[KSI03] C. Kruengkrai, V. Sornlertlamvanich, and H. Isahara. Refining a divisive partitioning algorithm for unsupervised clustering. In *Design and application of hybrid intelligent systems*, pages 535–542, 2003.

[KSO+06] D. Kim, S. Sun, S. Oh, J. Rehg, and A. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pages 518–525, Orlando, FL, USA, 2006.

[Kuh55] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[KWZ09a] P. Komma, C. Weiss, and A. Zell. Adaptive Bayesian filtering for vibration-based terrain classification. In *IEEE International Conference on Robotics and Automation (ICRA '09)*, pages 3307–3313, Kobe, Japan, May 2009.

[KWZ09b] P. Komma, C. Weiss, and A. Zell. Improved vibration based terrain classification using temporal coherence. In *40th International Symposium on Robotics (ISR '09)*, pages 359–364, Barcelona, Spain, March 2009.

[KZ09a] 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 '09)*, MIT, Cambridge, Massachusetts, USA, May 2009.

[KZ09b] P. Komma and A. Zell. Towards real-time and memory efficient predictions of valve states in diesel engines. In *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems (CIVVS '09)*, pages 8–15, Nashville, TN, USA, March 2009.

[KZ10a] P. Komma and A. Zell. Clustering vibration data using a temporally coherent expectation maximization approach. In *In 7th Symposium on Intelligent Autonomous Vehicles (IAV '10)*, pages 1–6, Lecce, Italy, September 2010.

[KZ10b] P. Komma and A. Zell. Markov random field-based clustering of vibration data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pages 1902–1908, Taipei, Taiwan, October 2010. Best Paper Award Nominee.

[LDV05] A.C. Larson, G.K. Demir, and R.M. Voyles. Terrain classification using weakly-structured vehicle/terrain interaction. *Autonomous Robots*, 19:41–52, July 2005.

[Leo06] J. Leong. Number of colors distinguishable by the human eye. http://-hypertextbook.com/facts/2006/JenniferLeong.shtml, 2006. (08/04/2011).

[Ley79] P. Leyhousen. *Cat Behavior*. Garland STMP Press, New York, 1979.

[LFJ02] M. Law, M. Figueiredo, and A. Jain. Feature saliency in unsupervised learning. Technical report, Department of Computer Science and Engineering, Michigan State University, 2002.

[Lil67] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318):399–402, June 1967.

[LMP01] J.D. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[LOCD09] L. Lu, C. Ordonez, E. Collins, and E.M. DuPont. Terrain surface classification for autonomous ground vehicles using a 2D laser stripe-based structured light sensor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pages 2174–2181, Piscataway, NJ, USA, 2009. IEEE Press.

[LP96] F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:722–733, July 1996.

[LSLZ09] H. Liu, J. Sun, L. Liu, and H. Zhang. Feature selection with dynamic mutual information. *Pattern Recognition*, 42:1330–1339, July 2009.

[LTJ04] M. H. Law, A. Topchy, and A. K. Jain. *Clustering with soft and group constraints*, volume 3138 of *LNCS*. Springer, January 2004.

[LVV03] A. Likas, N. Vlassis, and J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, February 2003.

[Mas51] F.J. Massey. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[MBC+03] L. Matthies, C. Bergh, A. Castano, J. Macedo, and R. Manduchi. Obstacle detection in foliage with ladar and radar. In P. Dario and R. Chatila, editors, *11th International Symposium on Robotics Research (ISRR '03)*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 291–300, Siena, Italy, October 2003. Springer.

[MBW08] O. Mattausch, J. Bittner, and M. Wimmer. Chc++: Coherent hierarchical culling revisited. *Computer Graphics Forum*, 27(2):221–230, 2008.

[MC91] B.S. Manjunath and R. Chellappa. Unsupervised texture segmentation using Markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:478–482, May 1991.

[MCW09] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pages 737–744, New York, NY, USA, 2009. ACM.

[MFM04] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:530–549, May 2004.

[Mil91] R.G. Miller. *Simultaneous Statistical Inference*. Springer-Verlag, McGraw Hill, New York, 1991.

[Min86] M. Minoux. *Mathematical Programming: Theory and Algorithms*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York, 1986.

[MMM00] J. Macedo, L. Matthies, and R. Manduchi. Ladar-based discrimination of grass from obstacles for autonomous navigation. In *Proceedings of the International Symposium on Experimental Robotics (ISER '00)*, pages 111–120, Hawaii, USA, 2000.

[MTMG03] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1):91–118, July 2003.

[Mun57] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[MZ92] 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.

[MZ02] R.A. Maronna and R.H. Zamar. Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 44(4):307–317, November 2002.

[Nab02] I. T. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer, 2002.

[NC07] B. Nelson and I. Cohen. Revisiting probabilistic models for clustering with pairwise constraints. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pages 673–680, New York, NY, USA, 2007. ACM.

[Nil02] M. Nilsson. Hierarchical clustering using non-greedy principal direction divisive partitioning. *Journal of Information Retrieval*, 5:311–321, October 2002.

[OBWK06] L. Ojeda, G. Borenstein, G. Witus, and R. Karlsen. Terrain characterization and classification with a mobile robot. *Journal of Field Robotics*, 23(2):103–122, 2006.

[OCLF10] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.

[OD92] P.P. Ohanian and R.C. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833, 1992.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[Pea88] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[Pla98] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[Pla00] J. Platt. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. In A.J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[PM00] D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[PPM+07] M.J. Pearson, A.G. Pipe, C. Melhuish, B. Mitchinson, and T.J. Prescott. Whiskerbot: A robotic active touch system modeled on the rat whisker sensory system. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 15:223–240, September 2007.

[PUH03] C. Pantofaru, R. Unnikrishnan, and M. Hebert. Toward generating labeled maps from color and range data for robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, volume 2, pages 1314–1321, October 2003.

[QCD05] A. Quattoni, M. Collins, and T. Darrel. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems 17 (NIPS '04)*, 2005.

[Rab90] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in speech recognition*, chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[Raf95] A.E. Raftery. Bayesian model selection in social research. *Sociological Methodology*, 25:111–163, 1995.

[Ras02] C. Rasmussen. Combining laser range, color, and texture cues for autonomous road following. In *IEEE International Conference on Robotics and Automation (ICRA '02)*, volume 4, pages 4320–4325, 2002.

[RD99] P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, August 1999.

[RDF96] N. Roy, G. Dudek, and P. Freedman. Surface sensing and classification for efficient mobile robot navigation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '96)*, volume 2, pages 1224–1228, Minneapolis, MN, USA, 1996.

[RF03] D. Ramanan and D. A. Forsyth. Using temporal coherence to build models of animals. *IEEE International Conference on Computer Vision*, 1:338, 2003.

[Rip96] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, January 1996.

[RL87] P.J. Rousseeuw and A.M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[Rou85] P.J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications*, B:283–297, 1985.

[RSPX01] L. Renqiang, W. Shuguo, L. Pinkuan, and W. Xiaohui. Curvature optical fiber whiskers for mobile robot guidance. *High Technology Letters*, 7(3):79–83, September 2001.

[Rus84] R.A. Russell. Closing the sensor-robot-computer control loop. *Robotics Age*, 6(4):15–20, 1984.

[Sar06] R.K. Sarvadevabhatla. Predicting terrain traversability from visual and accelerometric feature correlation. Technical Report 98105, Department of Computer Science, University of Washington, Seattle, WA, 2006.

[SBCL02] S.D. Spiegelhalter, N.G. Best, B.P. Carlin, and A.V.D. Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

[SBHHW04] 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 (NIPS '04)*. MIT Press, Cambridge, MA, 2004.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[Sch04] G. Schaefer. Robust dichromatic colour constancy. In *International Conference on Image Analysis and Recognition (ICIAR '04)*, volume 3212 of *Springer Lecture Notes on Computer Science*, pages 257–264, Porto, Portugal, September 2004. Springer.

[SD07] T. Su and J. Dy. In search of deterministic methods for initializing K-means and Gaussian mixture clustering. *Intelligent Data Analysis*, 11:319–338, December 2007.

[SDK+11] S. Scherer, D. Dumbe, P. Komma, A. Masselli, and A. Zell. Robust real-time number sign detection on a mobile outdoor robot. In *Proceedings of the 6th European Conference on Mobile Robots*, pages 1–7, Örebro, Sweden, September 2011.

[SG03] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, March 2003.

[SMTn03] N. Søndberg-Madsen, C. Thomsen, and J.M. Pe na. Unsupervised feature subset selection. In *Proceedings of the Workshop on Probabilistic Graphical Models for Classification*, pages 71–82, 2003.

[Spe90] D.F. Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, 1990.

[SR95] S.B. Serpico and F. Roli. Classification of multisensor remote-sensing images by structured neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 33:562–578, 1995.

[SR04] G.R. Scholz and C.D. Rahn. Profile sensing with an actuated whisker. *IEEE Transactions on Robotics*, 20(1):124–127, 2004.

[SR09] R. Steele and A.E. Raftery. Performance of Bayesian model selection criteria for Gaussian mixture models. Technical Report 559, University of Washington, Department of Statistics, September 2009.

[SS04] D. W. Scott and S. R. Sain. *Multi-Dimensional Density Estimation*, pages 229–263. Elsevier, Amsterdam, 2004.

[SSM11] D. Scherzer, M. Schwärzler, and O. Mattausch. Fast soft shadows with temporal coherence. In Wolfgang Engel, editor, *GPU Pro 2*. A.K. Peters, February 2011.

[SYM10] D. Scherzer, L. Yang, and O. Mattausch. Exploiting temporal coherence in real-time rendering. In *ACM SIGGRAPH Asia 2010 Courses (SA '10)*, pages 24:1–24:26, New York, NY, USA, 2010. ACM.

[Tar89] A. G. Tartakovskii. Sequential testing of many simple hypotheses with independent observations. *Problemy Peredachi Informatsii*, 24(4):299–309, 1989.

[TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. MIT Press, 2005.

[TC09] D.R. Thompson and N.A. Cabrol. Fast onboard texture analysis for planetary exploration. 2009.

[THKS88] C. Thorpe, M.H. Hebert, T. Kanade, and S.A. Shafer. Vision and navigation for the Carnegie-Mellon navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence - Special Issue on Industrial Machine Vision and Computer Vision Technology*, 10:362–372, May 1988.

[TJ98] M. Tuceryan and A.K. Jain. *Texture Analysis*, chapter 2.1, pages 207–248. The Handbook of Pattern Recognition and Computer Vision (2nd Edition). World Scientific Publishing Co., 1998.

[TT08] S. Tasoulis and D. Tasoulis. Improving principal direction divisive clustering. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08), Workshop on Data Mining using Matrices and Tensors*, Las Vegas, NV, USA, 2008.

[TV10] P.J. Rousseeuw T. Verdonck, M. Hubert. DetMCD in a calibration framework. In *19th International Conference on Computational Statistics (COMPSTAT '10)*, Statistics Section, Leuven Statistics Research Centre. Springer, Paris, August 2010.

[TWH01] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the Gap statistic. *Journal Of The Royal Statistical Society Series B*, 63(2):411–423, 2001.

[Van05] D. Vanderwerp. What does terrain response do? Testing the LR3's terrain response system. http://www.caranddriver.com/features/05q1/what_does_terrain_-response_do_-feature, February 2005. (08/04/2011).

[VEB09] N.X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pages 1073–1080, New York, NY, USA, 2009. ACM.

[Veg05] D.P. Vega. Spatial supervised learning using recurrent sliding window classifiers. School of Electrical Engineering and Computer Science, Oregon State University, June 2005.

[VHKH04] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3-d ladar data. In *IEEE International Conference on Robotics and Automation*, volume 5, pages 5117–5122, April 2004.

[Vin12] S.B. Vincent. The function of the vibrissae in the behavior of the white rat. *Behavior Monographs*, 1(5):1–81, 1912.

[VT07] J. Verbeek and B. Triggs. Scene segmentation with conditional random fields learned from partially labeled images. In *Proceedings of Neural Information Processing Systems (NIPS '07)*, 2007.

[VTL08] P. Vernaza, B. Taskar, and D.D. Lee. Online, self-supervised terrain classification via discriminatively trained submodular Markov random fields. In *IEEE International Conference on Robotics and Automation (ICRA '08)*, pages 2750–2757. IEEE, 2008.

[Wal50] W.G. Walter. An electromechanical animal. *Dialectica*, 4:42–49, 1950.

[WB91] I.H. Witten and T.C. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.

[WCS05] C. Wellington, A. Courville, and A.T. Stentz. Interacting Markov random fields for simultaneous terrain modeling and obstacle detection. In *Proceedings of Robotics Science and Systems (RSS '05)*, pages 1–8, June 2005.

[WCS06] C. Wellington, A. Courville, and A.T. Stentz. A generative model of terrain for autonomous navigation in vegetation. *International Journal of Robotics Research*, 25:1287–1304, December 2006.

[WFSZ07] C. Weiss, N. Fechner, M. Stark, and A. Zell. Comparison of different approaches to vibration-based terrain classification. In *Proceedings of the 3rd European Conference on Mobile Robots (ECMR '07)*, pages 7–12, Freiburg, Germany, 2007.

[WFZ06] 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 '06)*, pages 4429–4434, Beijing, China, October 2006.

[Wil89] S.S. Wilson. Vector morphology and iconic neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1636–1644, 1989.

[Wil94] 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.

[WKSB09] K.M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems (IROS '09)*, pages 1217–1222, Piscataway, NJ, USA, 2009. IEEE Press.

[WLW04] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

[WM95] D. H. Wolpert and W. G. Macready. No free lunch theorems for search. Technical Report No. SFI-TR-95-02-010, Santa Fe Institute, Santa Fe, NM, 1995.

[WS08] D.F. Wolf and G.S. Sukhatme. Semantic mapping using mobile robots. *IEEE Transactions on Robotics*, 24(2):245–258, April 2008.

[WSFB05] D.F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard. Autonomous terrain mapping and classification using hidden markov models. In *IEEE International Conference on Robotics and Automation (ICRA '05)*, pages 2038–2043, Barcelona, Spain, April 2005.

[WSG+09] H. Wang, B. Salatin, G.G. Grindle, D. Ding, and R.A. Cooper. Real-time model-based electric powered wheelchair control. *Medical Engineering and Physics*, 31(10):1244–1254, December 2009.

[WSZ07] C. Weiss, M. Stark, and A. Zell. SVMs for vibration-based terrain classification. In *Proceedings of Autonome Mobile Systeme (AMS '07)*, pages 1–7, Kaiserslautern, Germany, 2007.

[WTZ08] C. Weiss, H. Tamimi, and A. Zell. A combination of vision and vibration-based terrain classification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pages 2204–2209, Nice, France, September 2008.

[WZ08] C. Weiss and A. Zell. Novelty detection and online learning for vibration-based terrain classification. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS '08)*, pages 16–25, Baden-Baden, Germany, 2008.

[WZTL10] M. Wang, J. Zhou, J. Tu, and C. Liu. Long-range terrain perception based on conditional random fields. *International Journal of Advanced Robotic Systems*, 7(1):55–66, 2010.

[XK01] E.P. Xing and R.M. Karp. CLIFF: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. In *Proceedings for the 9th International Conference on Intelligent Systems for Molecular Biology (ISMB '01)*, volume 17, pages 306–315, Copenhagen, Denmark, 2001. Supplement 1.

[YCL10] C.-C. Yen, L.-C. Chen, and S.-D. Lin. Unsupervised feature selection: Minimize information redundancy of features. In *International Conference on Technologies and Applications of Artificial Intelligence*, pages 247–254, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[YH09] B. Yin and G. Hamerly. Hierarchical stability-based model selection for clustering algorithms. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA '09)*, pages 217–222, Washington, DC, USA, 2009. IEEE Computer Society.

[Yia93] P.N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms (SODA '93)*, pages 311–321, Philadelphia, PA, USA, 1993.

[Yua00] Y.-X. Yuan. A review of trust region algorithms for optimization. In *Fourth International Congress on Industrial and Applied Mathematics (ICIAM '99)*, pages 271–282. Oxford University Press, Oxford, July 2000.

[Zac71] S. Zacks. *The Theory of Statistical Inference.* New York: Wiley, 1971.

[ZG03] D. Zeimpekis and E. Gallopoulos. PDDP(l): Towards a flexible principal direction divisive partitioning clustering algorithm. In D. Boley, I. Dhillon, J. Ghosh, and J. Kogan, editors, *IEEE ICDM Workshop on Clustering Large Data Sets*, pages 26–35, Melbourne, Florida, USA, November 2003.