Master Thesis

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Road Following for Hexapod Walking Robot

Bc. Martin Stejskal

Supervisor: doc. Ing. Jan Faigl, Ph.D.
Field of study: Cybernetics and Robotics
Subfield: Robotics
May 2016

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Martin S t e j s k a l

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Road Following for Hexapod Walking Robot

Guidelines:

1. Familiarize yourself with vision-based road following methods [1,2] and terrain classification approach [3].
2. Propose a combination of vision and tactile based road detection.
3. Develop a road following method based solely on proprioceptive sensing and terrain classification.
4. Develop a control strategy to learn terrain classification using vision-based road following.
5. Combine the road following methods based on exteroceptive and proprioceptive sensing.
6. Experimentally validate the combined control strategy and demonstrate robustness of the road following in cases of failures of exteroceptive sensors.

Bibliography/Sources:

[1] P. Cristoforis, M. Nitsche, T. Krajnik, and M. Mejail: Realtime monocular image-based path detection, Journal of RealTime Image Processing, pp. 1-14, 2013.
[2] T. Krajnik, J. Blazicek, J. Santos: Visual road following using intrinsic images. European Conference on Mobile Robots (ECMR) 2015.
[3] J. Mrva, J. Faigl: Feature Extraction for Terrain Classification with Crawling Robot. ITAT 2015: 179-185.

Diploma Thesis Supervisor: doc. Ing. Jan Faigl, Ph.D.

Valid until: the end of the summer semester of academic year 2016/2017

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, December 22, 2015

Acknowledgements

I would like to thank my supervisor, doc. Ing. Jan Faigl, PhD, for all the support, optimism and valuable advices during the work on the thesis.

Declaration

I declare that presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 15 2016 ....
signature

Abstract

While moving in the environment, the robot is often supposed to stay on the specified terrain to maximize its motion efficiency. A special case of such a setup is the road following problem. In this work, two different approaches for road following are utilized, i.e. the visual and tactile based approaches. Their outputs are combined to improve the overall system robustness. The proposed visual road detection allows a smooth motion on the roads. However, it is based on exteroceptive sensor, which is biased by the environmental conditions. On the other hand, the tactile terrain classification is less vulnerable to ambient conditions. It is using proprioceptive sensing, which cannot foresee surface in front of the robot, but it rather provides information about the actual or past crawled surfaces. Therefore, performance of independent visual and tactile sensing are joined together, that enables the robot to keep on the road even in case one of the sensing modality fails. The experimental results demonstrate performance of the both independent approaches and also how they mutually enhance if the visual and tactile information are used together.

Keywords: hexapod, walking robot, road following

Supervisor: doc. Ing. Jan Faigl, Ph.D.

Abstrakt

Častým požadavkem na pohyb robotu prostředím je zůstat na daném typu terénu z důvodu maximalizovat efektivitu pohybu a snížit energetické nároky. Speciálním případem tohoto požadavku je sledování cesty. V této práci jsou použity dva různé přístupy sledování cesty založené na vizuální a taktilní senzorické informaci. Jejich výstupy jsou kombinovány za účelem zvýšení robustonsti. Vizuální sledování cesty umožňuje plynulý pohyb po cestě. Nicméně jedná se o přístu založený na exteroceptivním senzoru, který je ovlivněn světelnými podmínkami prostředí. Na druhé straně taktilní klasifikace terénu je odolná vůči vlivům okolního prostředí, ale je založena na proprioceptivním měření, které je schopné poskytnout informaci pouze o aktuálním nebo minulém terénu. Proto jsou v práci oba přístupy kombinovány, což umožňuje udržet robot na cestě i v případě selhání jednoho z nich. Experimentální výsledky demonstrují výhody kombinace vizuálního a taktilního sledování cesty.

Klíčová slova: hexapod, kráčející robot, sledování cesty

Překlad názvu: Autonomní sledování cesty šestinohým kráčejícím robotem

Contents

1 Introduction 1

2 Terrain Classification 3
2.1 Related work . . . 3
2.1.1 Exteroceptive sensing based terrain classification . . . 4
2.1.2 Proprioceptive sensing based terrain classification . . . 7
2.2 Utilized terrain classification with adaptive motion gait . . . 11
2.2.1 Hexapod platform . . . 11
2.2.2 Adaptive gait . . . 11
2.2.3 Terrain classification for adaptive gait . . . 16

3 Visual Road Following 19
3.1 Related work . . . 19
3.2 Utilized vision road following methods . . . 22

4 Proposed Road Following Strategy 25
4.1 Problem definition . . . 25
4.2 Tactile road following . . . 26
4.2.1 General considerations . . . 26
4.2.2 Road following control strategy 27
4.2.3 Limits of the proposed control strategy . . . 29
4.3 Visual road following . . . 30
4.3.1 Colour based road recognition 30
4.3.2 Intrinsic image based road recognition . . . 31
4.3.3 Road following algorithm . . . 34
4.4 Fusion of visual and tactile controllers . . . 38
4.4.1 Image quality criteria . . . 38
4.4.2 Terrain classification criteria . 40
4.4.3 Rules of the controller switching . . . 40
4.4.4 System learning . . . 40

5 Experimental Results 43
5.1 Experimental setup . . . 43
5.2 Tactile road following . . . 43
5.2.1 Laboratory experiments . . . 43
5.2.2 Outdoor experiments . . . 44
5.3 Visual road following . . . 48
5.3.1 Colour based road recognition 48
5.3.2 Intrinsic image based road recognition . . . 48
5.3.3 Comparison of the colour based and intrinsic image based road recognition . . . 50
5.4 Fusion of visual and tactile controllers . . . 52

6 Conclusion 59

Bibliography 61

A Contents of attached CD 65

Figures

2.1 Hexapod platform and leg schema 12
2.2 Dynamixel AX-12A servo [29] . . 12
2.3 Position error and leg elevation during one gait cycle [27] 13
2.4 Body levelling [25] 15
2.5 Gait diagram [26] 16
2.6 Servo position error for three different terrains. 17

3.1 Log-chromaticity space 23
3.2 Projection of the surfaces in the log-chromaticity space into a grey-scale image 24

4.1 Behaviour of the robot on the road with and without considering the terrain history 26
4.2 Example of road recognition using the invariant image 35
4.3 Example of searched path with all four possible criteria. 37
4.4 Experimental results of the image quality function from the laboratory environment 39
4.5 Logic of the controller switching with necessary conditions of image quality (1) and detected terrain (2) criteria 41
4.6 Flow diagram of the system learning 41

5.1 The on-board computer Odroid U3, adopted from [15] 44
5.2 Experimental setup for the laboratory terrain classification evaluation 45
5.3 Various terrain borders included in the outdoor experiments 46
5.4 Result of the tactile road following in one of the outdoor trials. Travelled trajectory is highlighted by the red line. 47
5.5 The road is detected correctly only if the colour distance between the road terrain and off road is high enough. Otherwise false positive detection occurs. 48
5.6 Shadows in the colour based road recognition 49
5.7 Invariant image road recognition 1 ($\theta = 152°$) 50
5.8 Invariant image road recognition 2 ($\theta = 169°$) 51
5.9 Invariant image road recognition 3 ($\theta = 169°$) 51
5.10 Invariant image road recognition 4 ($\theta = 144°$) 52
5.11 Comparison of the road detection results for colour based and intrinsic image based methods. 53
5.12 Trajectory of the robot with highlighted predicted terrain classes and decision process diagram, laboratory experiment 55
5.13 Trajectory of the robot with the highlighted predicted terrain classes and decision process diagram, run 1 56
5.14 Trajectory of the robot with the highlighted predicted terrain classes and decision process diagram, run 2 57

Tables

5.1 Two–fold cross-validation confusion matrix of the sample gait cycles. The overall accuracy reached 96.2%. Class names are shortened in the table in the following way: D - Dirt, A - Asphalt, G - Grass. 47
5.2 FPS of the utilized vision based road following methods 51

A.1 Structure of the CD root 65

Chapter 1

Introduction

One of the goals in mobile robotics and artificial intelligence is to deploy results of research to challenging environments encountered in a common life. Therefore a lot of effort is put into development of robust devices and methods which allow the robot, e.g. to navigate through an outdoor terrain such as a forest or field, participate in rescue missions or patrol and monitor various areas.

In an urban environment or a slightly uneven terrain, it is possible to use wheeled vehicles. They are easy to control and can carry heavy loads. The robot's traversability through the terrain can be increased by adding tracks. These robots are nowadays used for exploration after accidents, where human life can be threatened (risk of collapse, explosion, radioactivity, etc.). A good example of employing track robots is the elimination of consequences in Fukushima where a nuclear power plant has been damaged. As reported in the popular magazine [31], many different vehicles are operating there.

Compared to the ordinary wheeled robots, crawling robots can operate in even more diverse, difficult and unstructured terrains. However, the control requirements are much more demanding due to a higher number of the degrees of freedom (DOF). The hexapod walking robot, used in this work, has 18 control DOF in constrast to 2 control DOF of a common car-like robot. Moreover, the robot's body stability has to be preserved while moving the legs, which is not a simple task, especially for bipods and tripods [6].

Predefined trajectories for each leg can be created to handle robots with high number of DOF, i.e. create a motion gait [10]. Such a predefined gaits are usually efficient only on flat terrains, because of the fixed lift height. Leg is periodically lifted to the required height, moved forward, and in the end it is laid back on the ground. However, when crawling rough terrains, the leg can stuck at an obstacle or can stand at different height than the ground terrain surface. Both cases can cause stability loss or incapability to move on.

Closing the control loop and considering the sensory input can solve the drawbacks of the regular gaits and increase traversability through rough terrains [27]. As shown in [27], the actuator information about a position error in a leg joint can be used to detect leg contact with the surface. This approach is minimalistic and requires no additional sensory equipment.

Also the actuator information can be utilized to perform terrain classification. The method presented in [4] is based on differences between expected and real leg trajectories. Besides, Discrete Fourier Transform (DFT) is used to analyse the frequency spectrum of the gait cycle. However, it considers a regular periodic gait not suitable for rough terrains with obstacles.

Therefore, this approach is not directly applicable for the adaptive gait developed in [27]. A modification of the feature extraction has been proposed in [26]. The authors shown, that differences of the expected and real leg trajectories are sufficient to correctly classify the crawled terrain. Thus, a technically blind robot is able to provide information about crawled terrain based solely on its proprioceptive sensing. At the same time, the ability of crawling challenging terrains using adaptive motion gait is preserved.

Nonetheless, it is almost impossible for a blind robot to navigate through the environment without any other sensory equipment. There is a vast number of sensors available for supporting the robot navigation, e.g. sonars, laser scanners, radars, etc. Most of them suffer from various disadvantages like weight, size, costs or precision. In a case of active sensors, such as sonars, interference may cause sensor failure or incorrect measurements.

During the last decades, vision based sensors became cheaper, small and can provide a high resolution image of the robot surroundings. With computational power available today they are ideal tool for robotic tasks. These sensors can be considered as a standard equipment, e.g. for object detection, navigation or path detection. This work is concerned with path recognition.

However, there are several issues in visual path recognition. The first one is road appearance. It is difficult to generally define the road since its colour, shape, and width may vary. The other problem is a scene illumination that may significantly change the result of road recognition, especially in outdoor environment. Also when road border is similar to the road itself (e.g. brown cobblestones and dirt), robot can be easily led off the road.

In this work, the robustness of the vision based road recognition is addressed. The goal is to propose a control strategy that will combine visual information and proprioceptive terrain classification. This combination shall remove drawbacks of the visual navigation related to illumination issues and false positive road detection, while keeping a desired behaviour.

The work is organized as follows. In Chapter 2 a brief overview of existing methods for terrain classification are introduced. Then, the utilized adaptive motion gait and terrain classification are described. Chapter 3 provides an overview of visual road recognition techniques together with detailed principles of the method used in this work. The proposed control strategy design is presented in Chapter 4 and the experimental results are reported in Chapter 5. The concluding remarks are in Chapter 6.

Chapter 2

Terrain Classification

Humans intuitively adjust their walking style according to the traversed terrain in diverse environments. This property arises from the need of maintaining stability, increasing the motion speed and energy efficiency. A walk on slippery ice, grass and rocky hill can be considered as an example. On the ice surface, humans make short slow steps and do not lift the feet too much above the surface. On grass, humans can safely run making long leaps with a medium lift height. And finally, while walking the rocky hill, the feet have to be lifted high to avoid collisions with the rocks.

Using the same walking style for all terrains can mean either stability loss (running on ice), unnecessarily slow motion (using the ice walking style on grass) or redundant energy consumption [18]. A similar behaviour can be implemented for mobile robots to improve efficiency while traversing different terrains. Therefore, it is necessary to develop on-line terrain classification method to recognize traversed terrain and switch the control to the most suitable motion strategy.

2.1 Related work

As the authors of [4] propose, the terrain classification can be described as a process of assigning terrain patches to some of the predefined classes. According to the way how terrain patches are acquired, terrain classification can be divided into two categories based on

- exteroceptive sensing,
- proprioceptive sensing.

Eventually both approaches can be combined.

For exteroceptive sensing mostly the cameras or laser range finders are used. The main advantage of these methods is the ability to predict the terrain ahead of the robot. On the other hand, it does not provide any information about the robot experience and its performance during the terrain traversing. Proprioceptive sensing allows the robot to analyse the interaction with the surface and thus, gives information about the performance, which can be utilized to change the current motion style.

In the rest of this section, selected existing approaches are briefly described. In particular, Section 2.1.1 describes various methods of exteroceptive sensing based classification. In Section 2.1.2, proprioceptive methods are discussed. The goal of this related work overview is to show common elements in the terrain classification and highlight few remarkable approaches.

2.1.1 Exteroceptive sensing based terrain classification

Although many exteroceptive sensors are available, mostly visual and range sensors are used for terrain classification, especially laser scanners and cameras. In the following text, data samples (images, laser scans) capturing more than one terrain type are referred as multi-terrain samples, whereas samples with one terrain type are referred as single-terrain samples.

In [40], two types of laser scanners are used. Both of them emit the near-infrared spectrum able to reflect certain information about material properties in addition to distance. These additional properties are returned as an intensity $\mathbf{I}_p$ of the point p. The authors propose to create a map from particular laser scans in 3 different heights and under 2 lightning conditions. The map contains terrain elevation and intensity values $\mathbf{I}_p$ representing the material properties. The method was evaluated on 12 different terrains, where some of them were flat; thus, their elevation map was the same, or at least very similar to each other. The map was then divided into a grid where particular grid cells were joined to form terrain patches of 11 $\times$ 11 cells. Over each terrain patch the feature extraction was performed to obtain a feature vector for Support Vector Machine (SVM). Four different ways of feature extraction are discussed" statistical, texton [17], Fourier Transform, and elevation map. The particular approaches of feature extraction are described in detail in following paragraphs.

In the statistical approach, the mean, variance, standard deviation and kurtosis for intensity values are computed in each patch. These values then compose the feature vector $\mathbf{x}_s$ for statistical approach:

$$\mathbf{x}_s = \left[\mu\left(\mathbf{I}_p\right), var\left(\mathbf{I}_p\right), \sigma\left(\mathbf{I}_p\right), kurtosis\left(\mathbf{I}_p\right)\right]. \tag{2.1}$$

The feature vector $\mathbf{x}_e$ for the elevation map was computed in the same manner as the vector $\mathbf{x}_s$. Only the intensity values $\mathbf{I}_p$ were replaced by cell height $\mathbf{H}_p$. Therefore the elevation map feature vector is

$$\mathbf{x}_e = \left[\mu\left(\mathbf{H}_p\right), var\left(\mathbf{H}_p\right), \sigma\left(\mathbf{H}_p\right), kurtosis\left(\mathbf{H}_p\right)\right]. \tag{2.2}$$

The texton approach relies on the texture of the surface [40]. Textons [17] describe the relation between neighbouring pixels and thus the texture itself. The map intensity values are considered as a grey-scale image. The texton feature vector $\mathbf{x}_t$ for the pixel $p_{i,j}$ is composed of the intensity differences in 3×3 neighbourhood

$$\mathbf{x}_t = \left[I_{p_{i,j}}, \left(I_{p_{i-1,j-1}} - I_{p_{i,j}}\right), \left(I_{p_{i-1,j}} - I_{p_{i,j}}\right), \cdots, \left(I_{p_{i+1,j+1}} - I_{p_{i,j}}\right)\right]. \tag{2.3}$$

The vectors $\mathbf{x}_t$ of all cells are then clustered into 12 generic classes using K-means algorithm. For each terrain patch the histogram of the generic classes contained within the patch is then calculated and used as a feature vector for SVM.

The last classification method presented in [40] is based on 2D Fast Fourier Transform (FFT) of the intensity values. Three masks, vertical, horizontal, and circular, were used in order to extract desired frequencies from the patches. A feature vector is then made from the mean values computed from the patch after performing FFT with and without the masks.

The results presented by the authors of [40] show that considering intensity values instead of the elevation map significantly improves the classification performance. With a combination of statistical and Fourier Transform approaches they reached the success rate 98.47%. Besides, the benefit of the method is possibility to classify multi-terrain samples.

Another laser based classification was proposed in [21], where an infra-red camera is used to capture reflected laser beam line. In this way, the height profile and intensity values can by acquired from each sample. Both the height profile and intensity values are considered as two separate input data and their performance is compared according to lightning conditions. A terrain profile illuminated by the laser is extracted from the greyscale image and its FFT is performed in order to create a feature vector. In contrast to texton description of the texture from [40], authors of [21] propose a statistical processing of the Grey-Level Co-occurrence Matrix (GLCM). In the GLCM, the contrast, correlation, energy, and homogeneity are computed. The feature vector for texture is then created as

$$\mathbf{x}_t = [contrast, correlation, energy, homogeneity]\,. \tag{2.4}$$

Then, the authors consider a probabilistic neural network as a classifier and terrain profile and texture were trained and tested separately [21]. Both approaches reach in most cases more than 90% success rate. However, as it is shown in [21], texture classification can be influenced by light conditions. The main drawback of the laser-camera combined approach is the possibility to classify only single terrain samples.

Classification based solely on the images from colour camera has been proposed in [2]. The feature extraction time versus class differentiability is examined and then the most efficient solution is designed. The authors propose to build a hierarchy of Bayesian classifiers and stop the classification at the moment when the class can be distinguished from the other classes with a particular confidence. Based on the error rate and computational time, the authors of [2] select the average colour, colour histogram and textons as a terrain features.

The hierarchy is build in a form of a decision tree, where at each level, the Bayesian classifier with a different feature extraction procedure is used. Classification is then considered as a problem of finding posterior probability of the class w_i while having a measurement X

$$P\left(w_i|X\right) = \frac{p(X|w_i)p(w_i)}{\sum_{j=1}^{R} p(X|w_j)p(w_j)}, \tag{2.5}$$

where probability $p(X|w_i)$ can be determined from the training set and the path in the hierarchy. If at any level classifier cannot certainly decide a proper class, the decision tree is subdivided to separate the overlapping classes from the others. The classes are passed to another classifier at the lower level of the structure, where different and more demanding features are used. This is repeated until the classification is successful (a high probability of the classified terrain or the hierarchy bottom is reached).

A classification based on colour might be remarkably influenced by illumination conditions of the environment. More robust classification can be accomplished by using terrain landmarks as proposed in [11], where for each terrain a set of Speeded Up Robust Features (SURF) [3] are gathered to generate a bag of visual words (BOVW). To create such a vocabulary a set of single-terrain images is acquired and SURF descriptors are extracted. The descriptors are clustered by the K-means algorithm and cluster centres are added to the vocabulary.

Moreover, the authors of [11] discuss the problem of single-terrain and multi-terrain images. In the single-terrain case, only one terrain type for the whole image is expected as an output. Therefore, each SURF descriptor $\mathbf{d}_j$ found in the image is assigned to a vocabulary word $\mathbf{v}_i$ so that the Euclidean distance is minimized

$$\arg\min_i ||\mathbf{v}_i - \mathbf{d}_j||. \tag{2.6}$$

The histogram of assigned words $\mathbf{v}_i$ is then computed and treated as a feature vector for the SVM classifier. However, if the image contains more terrains, the aforementioned procedure is not applicable. Therefore, pixels on the regular grid are selected from the image and a circular neighbourhood containing specified number of features is established. The SVM classifier then predicts the terrain in the bounded area and all pixels within are labelled. Note, pixels can be involved in more than one area; thus, all labels for the pixels are stored and at the end of the classification, a voting procedure is performed for each pixel. The class with the highest number of votes is determined as the final pixel class.

This terrain classification was afterwards utilized to switch the gait patterns for the LittleDog robot. A planning algorithm was supposed to select one of three predefined gaits:

- fast gait with low ground clearance,
- slow gait with high ground clearance,
- hybrid gait with medium clearance.

However, experimental results of [11] are not promising since the authors encountered problems with camera focusing during the online evaluation.

A similar approach is presented in [44]. In addition to the SURF features, also SIFT features are tested and the performance of both is compared. Beside the terrain classification, the authors also address a selection the gait according to the energy consumption of the hexapod walking robot. Each learned terrain is examined by a neural network and the most effective gait with the lowest energy consumption is selected. Employing this knowledge it is shown how the robot changes gaits as it crawls different terrains.

2.1.2 Proprioceptive sensing based terrain classification

Contrary to the exteroceptive sensing, proprioceptive sensing provides information about robot own experience with the environment. The majority of the existing methods is based on measuring vibrations, force or torque. As it can be concluded from the following state of the art, the vibrations are utilized mainly for wheeled robots, whereas walking robots take advantage of the force and torque. In this section, both approaches are discussed without differentiating between legged and wheeled robots unless stated otherwise.

A feasibility of the vibration based terrain classification is presented in [41]. Data from accelerometer were collected while the experimental platform (wheeled cart) was moving on six different terrains. Raw data were then processed and feature vector were generated by four different methods as follows.

The first feature vector is composed from the power spectral density (PSD) using Welch's estimate. The second vector is composed by 128-points produced by the FFT. The authors present their own feature vector which is computationally less demanding than PSD and FFT and can be described in the following way. Let $\mathbf{v}$ be the vector of raw data. Then, μ is its mean value, n is the number of sign changes in $\mathbf{v}$ and t is the number of occurrences where data pass the mean value. Autocorrelation r_1 for lag $k = 1$ is calculated. The minimum min and maximum max values are found and the norm of the raw data $\|\mathbf{v}\|$ is computed. Combining all these values, the feature vector is generated in the form

$$\mathbf{x}_t = [n, t, \sigma_v, r_1, max, \|\mathbf{v}\|, min, \mu] \,. \tag{2.7}$$

The last type of feature vector is created by a simple concatenation of PSD and the authors' feature vector.

The SVM with RBF kernel was used for the classification. Performance of all four feature types was compared with respect to the needed computational time. It is shown that the best classification rate is achieved with the proposed combined feature vector presented in Eq. 2.7.

In [42], the authors extend their previous work based solely on vibration classification [41] by visual classification. A description of the image is provided by integral invariants, which are computed for each of the HSV channels and the invariants are then treated as a feature vector. Feature vectors of all three channels are merged together for classification with SVM. The vibration classification slightly differs from the method presented in [41].

Raw data from all three sensor axes are divided into time segments of 1 s and features are extracted using 128-point FFT. This results in three vectors with 64 features each being concatenated into one 192-element feature vector.

Both visual and vibration classification methods can be applied independently. However, their success rate is lower. Therefore predictions are combined and a stronger classifier is created as follows. At first, vibration data have to be assigned to the correct visual data. Therefore a local coordinate frame is created for the image. With the odometry information, the robot is able to estimate its translation and assign the correct image to the current vibrations. Prediction integration is accomplished by summing the probabilities of each class from visual and vibration based classifications and the class with the highest probability is selected as the final terrain class.

An estimation of the terrain solely based on vibrations may fail since vibrations are highly dependent on the particular vehicle speed and weight. This issue is addressed in [8], where the authors propose to interpolate several basic datasets, instead of learning a large number of datasets gathered for different speeds and loads. The tracked values for classification are frequency response of acceleration, roll and pitch angular velocities. The feature vector is constructed as

$$\mathbf{x} = \left[|\ddot{y}(j\omega)|, |\omega_{roll}(j\omega)|, |\omega_{pitch}(j\omega)| \right]. \tag{2.8}$$

For learning the terrain, only few point clouds consisting particular feature vectors with known speed and load are collected as follows. Let γ be a set of conditions for which the point cloud is valid, i.e. a combination of known speed and weight, and γ_k is a set of conditions for a learned point cloud. Each learned point cloud is represented by a matrix $\mathbf{X}_{\gamma_k}$. Then, it can be decomposed by the Singular Value Decomposition (SVD) into

$$\mathbf{X}_{\gamma_k} = \mathbf{U}_{\gamma_k} \mathbf{\Sigma}_{\gamma_k} \mathbf{V}_{\gamma_k}^T \tag{2.9}$$

and mean value $\overline{x}\gamma_k$ can be computed from the cloud points. For two learned point clouds with conditions γ_k and γ_{k+1} a point cloud with a condition γ satisfying $\gamma_k < \gamma < \gamma_{k+1}$ can be interpolated. The estimated mean value $\hat{\overline{x}}_\gamma$ of the interpolated point cloud is found using the cubic Catmull-Rom splines. The singular value matrix $\mathbf{\Sigma}\gamma$ is estimated using a linear interpolation between $\mathbf{\Sigma}_{\gamma_k}$ and $\mathbf{\Sigma}_{\gamma_{k+1}}$. Matrix logarithm and exponential are utilized to interpolate the matrices $\mathbf{U}_\gamma$ and $\mathbf{V}_\gamma$ as

$$\mathbf{U}_\gamma = e^{t\mathbf{Q}_{U,k}} \mathbf{U}_{\gamma_k} \tag{2.10}$$

$$\mathbf{Q}_{U,k} = \log\left(\mathbf{U}_{\gamma_{k+1}} \mathbf{U}_{\gamma_k}^T \right). \tag{2.11}$$

Another remarkable approach of dealing with a speed dependency is presented in [7]. However, it is applicable only for wheeled robots. The authors suggest to model each wheel as a single input – single output (SISO) system. At first, it is shown (using control theory), how the vibration output, i.e. vibrations measured on the robot body, depends on the speed. Four wheel models are then combined into one multiple input – multiple output (MIMO) system representing the robot. The transfer function is denoted $G(s)$, output $Z(s)$ and the input $X(s)$

$$G(s) = \begin{bmatrix} G_1(s) & G_1(s) & G_1(s) & G_1(s) \\ G_2(s) & G_2(s) & -G_2(s) & -G_2(s) \\ G_3(s) & -G_3(s) & -G_3(s) & G_3(s) \end{bmatrix}, \tag{2.12}$$

$$Z(s) = \begin{bmatrix} \dot{Y}(s) \\ \omega_{pitch}(s) \\ \omega_{roll}(s) \end{bmatrix}, \quad X(s) = \begin{bmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \\ X_4(s) \end{bmatrix} \tag{2.13}$$

$$Z(s) = G(s)X(s), \tag{2.14}$$

where the corresponding time-domain variable $\dot{y}(t)$ is the vertical speed of the robot body and $\omega_{pitch}(t)$, $\omega_{roll}(t)$ are pitch and roll angular velocities. $X_i(s)$ represents terrain responses for particular wheel i. The transfer functions $G_i(s)$ define the relations between the wheels and robot body in relation to $\dot{y}(t)$, $\omega_{pitch}(t)$, and $\omega_{roll}(t)$. With independent system input for each wheel, the multi-terrain classification can be performed as follows.

First, the system from Eq. 2.14 is inverted, since values of the vector $Z(s)$ are easily measurable on the robot body and it is much more complicated to obtain values in $X(s)$; so, the system equations are solved for $X(s)$

$$X(s) = G^{-1}(s)Z(s). \tag{2.15}$$

For such a system, multiple solutions exist. Thus, the solution is constrained and simplified to estimate a virtual input for the left and right side of the robot as

$$\hat{\mathbf{X}}(s) = \begin{bmatrix} \hat{X}_1(s) \\ \hat{X}_2(s) \end{bmatrix}. \tag{2.16}$$

These virtual inputs are then processed by FFT and learned by probabilistic neural network. Even though the result shows feasibility of the method, it was tested only in simulation.

Vibration based classification is applicable for both wheeled and legged robots. Beside, the vibrations, force, torque and motor current in the legs are useful for terrain classification for legged robots. In [16], force and motor current are measured. The classification is shown on the leg detached from the body. Note, the leg is divided into coxa, femur, and tibia. In tibia, three force sensitive resistors (FSR) are placed around the leg perimeter and the current is measured in the active joint. The authors aim to predict terrain shape and surface independently in the following way.

For the terrain shape classification, the leg oscillates with the amplitude 1° and frequency 0.5 Hz. The scratching motion is executed to capture the terrain friction and thus, allow the surface type classification. In both cases, the feature vector is generated from the vector containing data for a given terrain (three FSR and one current measurements). The values of interest are the standard deviation $\sigma_{\mathbf{s}}$, minimum and maximum values and the mean $\mu_{\mathbf{s}}$ in the time domain. Then, the FFT is computed and in frequency domain, the dominant frequencies larger than 5 Hz and standard deviation of the frequency spectrum σ_{fs} are determined. The feature vector is finally composed as

$$\mathbf{x}_t = [\sigma_{\mathbf{s}}, min(\mathbf{s}), max(\mathbf{s}), \mu_{\mathbf{s}}, FFT(\mathbf{s}).\sigma_{fs}] \,. \tag{2.17}$$

The feature vectors are then trained by the AdaBoost algorithm with strategy one-versus-all.

Instead of placing the sensors in the middle of the leg, 6-DOF force-torque sensor is placed at the tip of the leg in [39]. The sensor provides three-axis information about the applied force and torque. The author exploits data from all 6 axes separately and one combination of force and torque in the z-axis to classify the terrain. The feature vector $\mathbf{x}_t$ is generated by the data variance, skewness, kurtosis and fifth moment:

$$\mathbf{x}_t = \left[\sigma^2, skewness, kurtosis, fifth\ moment\right] . \tag{2.18}$$

The authors of [39] present three different types of discriminant analysis for the classification - linear (LDA), quadratic (QDA) and function with Mahalanobis distances. The best reported performance is achieved by using a combination of z-axis force and torque with QDA.

All of the aforementioned approaches need additional sensory equipment to measure vibrations, forces etc. The minimalistic way of terrain classification is proposed in [4] where intelligent servo drives were used. The servos communicate via half-duplex serial line and can provide information about its state, such as the current speed, motor load, current and goal positions and more. The authors propose to exploit the differences between goal and actual position read from the servos of two front legs. Raw data with the sample rate around 20 Hz from the last three whole gait cycles are saved and interpolated using cubic Hermite spline and resampled again with the rate 100 Hz. The gait cycle is divided into 16 subwindows. In each subwindow, the minimum, maximum, mean, median, and standard deviation of the error signal are calculated giving 480 features for all six servos of the front legs.

Beside the gait phase domain, frequency domain features are also generated in [4]. The required assumption for frequency domain features is the periodicity of the gait cycle which is fulfilled for standard gait patterns. Then, the FFT can be computed for position error data of the whole gait cycle. The first 25 frequency bins are directly used as features, which gives 150 features for 6 servos. Additional features are obtained in a form of centroid, standard deviation, skewness and kurtosis of the spectrum. Moreover, the spectrum energy is calculated, which is then utilized in constructing the feature vector. The whole feature extraction results in the 660-dimensional feature vector.

Before training the classifier, all features are scaled to have mean of 0 and standard deviation of 1 in order to suppress domination of the features with large numeric ranges. The scaling factors are saved and used in the same way during the classification itself. SVM is used as the classifier.

2.2 Utilized terrain classification with adaptive motion gait

Many methods of the terrain classification have been published in literature and some of them are highlighted in the previous section. However, the goal of this work is to propose the minimalistic approach and utilize only servo drive feedback. Moreover, the hexapod walking robot is considered to traverse rough terrains with obstacles. The standard gaits with constant leg trajectories provide reliable locomotion only on flat terrains. A lower traversability is usually caused by the fixed leg trajectory endpoints that force the leg to be placed in every gait cycle to exactly the same position, and therefore, the robot is not able to adapt to the current terrain. Since the outdoor environment is uneven, gait adaptation is a crucial part of the gait pattern generation.

Therefore, the adaptive motion gait presented in [27] accompanied by the terrain classification method [26] are used in the proposed solution. In the following parts of this section, the hexapod walking robot, adaptive gait, and terrain classification used in this work are described.

2.2.1 Hexapod platform

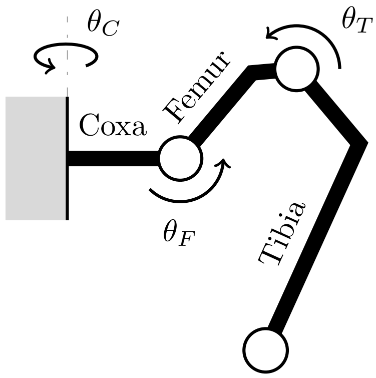
The hardware platform is based on the commercial hexapod kit PhantomX Hexapod Mark II marketed by Trossen Robotics [30]. The basic robot structure is shown in Fig. 2.1a. The robot has six identical legs, each composed of three servo drives dividing the leg into coxa, femur, and tibia as shown in Fig 2.1b. Thus, the whole robot has 18 controllable DOF. The legs are mounted to the body symmetrically with respect to the body axis dividing it to the left and right side.

The joints are actuated by the Dynamixel AX-12A servo drives [29] depicted in Fig. 2.2. These are intelligent servos with build-in 10-bit digital controller. Their operating radius is 300° with the angular resolution approximately 0.29°. Unlike the common servos, where the only input signal is PWM, the Dynamixel AX-12A servos communicate via a half-duplex serial line. Each device has its unique ID and a simple protocol with a variable number of data bytes is used. The packets are secured with a checksum to provide data consistency and allow an error detection.

Due to the build-in intelligence, each servo provides several information about its state and allows to set various parameters for its operation such as speed, maximal torque, and goal position. Also safety flags like overheat, overload, and more are implemented to prevent any damage to the servos. Detailed description of the servos is available online at [29].

2.2.2 Adaptive gait

Adaptive gait for the hexapod with the Dynamixel AX-12A servos was proposed in [25] and [27]. It was designed with two fundamental requirements.



(a): Hexapod platform [30] **(b):** The hexapod leg [27]

Figure 2.1: Hexapod platform and leg schema. Leg consists of coxa, femur, and tibia.

Figure 2.2: Dynamixel AX-12A servo [29]

The first one is to detect the leg contact with the surface without any additional sensors except the data from servo itself. The second one is to preserve the robot stability and uniformly distribute the load among all legs. Although in [27] the gait is proposed in a pentapod variant, the current version supports tripod gait allowing the robot to move much faster.

The task of the adaptive gait design can be divided into two subtasks – surface contact detection and body levelling. When detecting the leg's contact with the surface, the most natural way is to measure the force applied to the leg. This force results in the torque in the servos. However, the servos do not provide information about the actual torque. In [27], the servo position error is considered to be linearly dependent on the torque, and therefore, it may be used as an estimate of the torque.

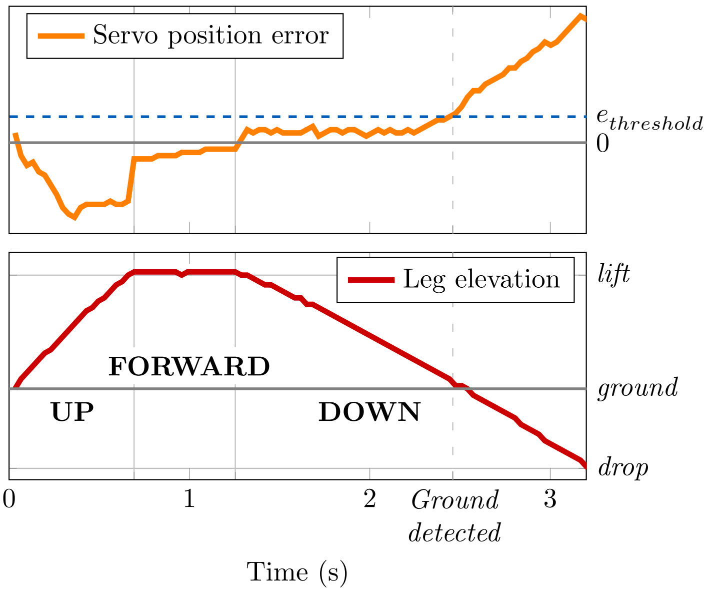


Figure 2.3: Position error and leg elevation during one gait cycle [27]

During the gait cycle, the torque affects mainly the tibia and femur joint actuators. Regarding the robot's construction (Fig. 2.1a) the most of the momentum is reflected by the femur joint and thus, it can be used as a sufficient approximation of the torque applied to the leg. Taking into account this consideration, the surface detection is a simple task of thresholding the position error from the femur joint actuator.

An example of the position error trend with respect to the leg elevation is shown in Fig. 2.3. As it can be seen after the leg hits the ground, the error grows. If the leg motion is not stopped in such a case, it may happen that the leg will be overloaded or the robot can lose the support of other legs. Thus, the threshold $e_{threshold}$ has to be set properly to stop the leg motion.

The value of $e_{threshold}$ depends on the robot weight and the easiest way to estimate the proper $e_{threshold}$ is to stand the robot on the ground so that all the legs support the robot equally. Then, each femur joint servo has a position error e_i implying the approximation equation

$$e_{threshold} \approx \frac{1}{6} \sum_i e_i. \tag{2.19}$$

After the leg detects the ground and new foothold position is established, the body is moved and levelled. The body levelling is intentionally separated from the leg motion. Firstly, the body has to rotate to reflect the leg position. For this purpose, a plane, which minimizes the square distance from each leg endpoint, is found. The equation of the plane is

$$z = ax + by + c, \tag{2.20}$$

where the parameters a, b and c can be computed using the linear regression from the feet positions [27].

The coordinate system of the robot is placed in the center of the body. It forms a right-handed system, where the z-axis points upwards and is perpendicular to the body, the x-axis points forward, and the y-axis points to the left. When levelling the body, it is rotated in a way that the $x-y$ plane is parallel to the one found in Eq. 2.20.

The second step of body motion is translation. A new body center is considered only in x and y coordinates, since the body height h in the direction of the z-axis is predefined constant and $x-y$ plane is parallel to the found plane. The body position is calculated as an average position of the legs. The whole body transformation to the new coordinate system is composed of a rotation represented by the matrix $\mathbf{R}$ and a translation represented by the vector $\overrightarrow{t}$ and can be written as

$$\begin{bmatrix} x'_B \\ y'_B \\ z'_B \\ 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R} & & \mathbf{R}\overrightarrow{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix}. \tag{2.21}$$

In fact, if the robot shall move forward, the body has to follow the legs, and therefore, the legs, after they are place to the new foothold positions, have to move backwards with respect to the body. Transformed leg positions can be expressed as

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix} = \begin{bmatrix} & \mathbf{R}^T & & -\overrightarrow{t} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}. \tag{2.22}$$

In order to create the rotation matrix $\mathbf{R}$, the orthogonal base is created from the plane parameters estimated by Eq. 2.20. The vector $\overrightarrow{b}_x$ is defined directly from the plane, since it has to preserve the forward motion. Also $\overrightarrow{b}_z$ perpendicular to the plane can be created directly. Finally $\overrightarrow{b}_y$ is constructed to be linearly independent and orthogonal to $\overrightarrow{b}_x$ and $\overrightarrow{b}_z$. All three vectors have the form

$$\overrightarrow{b}_x = \begin{bmatrix} 1 \\ 0 \\ a \end{bmatrix}, \quad \overrightarrow{b}_y = \begin{bmatrix} -ab \\ a^2+1 \\ b \end{bmatrix}, \quad \overrightarrow{b}_z = \begin{bmatrix} -a \\ -b \\ 1 \end{bmatrix}. \tag{2.23}$$

The rotation matrix is then constructed from the basis vectors and normalized:

$$\mathbf{R} = \begin{bmatrix} 1 & -ab & -a \\ 0 & a^2+1 & -b \\ a & b & 1 \end{bmatrix} \begin{bmatrix} \| \overrightarrow{b}_x \| & 0 & 0 \\ 0 & \| \overrightarrow{b}_y \| & 0 \\ 0 & 0 & \| \overrightarrow{b}_z \| \end{bmatrix}^{-1}. \tag{2.24}$$

Rewriting Eq. 2.22 with knowledge of Eq. 2.24, the transformed leg position can be expressed as

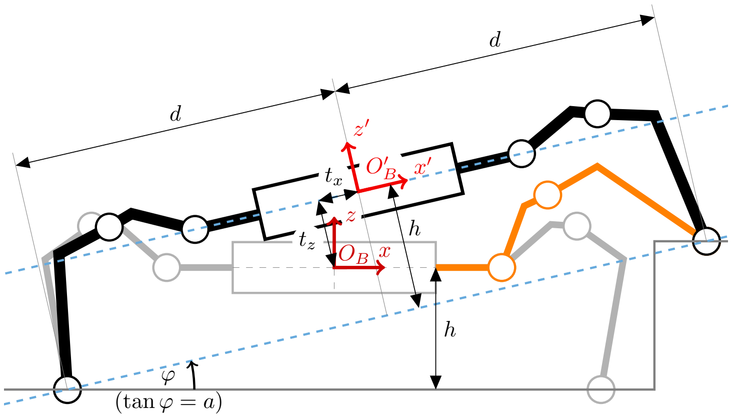


Figure 2.4: Body levelling [25]

$$x_i' = \frac{\overrightarrow{b}_x \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}}{\| \overrightarrow{b}_x \|} - t_x, \tag{2.25}$$

$$y_i' = \frac{\overrightarrow{b}_y \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}}{\| \overrightarrow{b}_y \|} - t_y, \tag{2.26}$$

$$z_i' = \frac{\overrightarrow{b}_z \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}}{\| \overrightarrow{b}_z \|} - t_z. \tag{2.27}$$

From Eq. 2.25 for all the legs, the final body translation $\overrightarrow{t}$ can be computed. The x and y components of the translation follow the idea of computing the new body position as a center of the foot positions. However, the z-axis has to compensate the change of the heigh above the ground to reach the predefined height h:

$$\overrightarrow{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \frac{\sum x_i + a \sum z_i}{6\|\overrightarrow{b}_x\|} \\ \frac{-ab\sum x_i + (a^2+1)\sum y_i + b\sum z_i}{6\|\overrightarrow{b}_y\|} \\ \frac{c}{6\|\overrightarrow{b}_z\|} - h \end{bmatrix}. \tag{2.28}$$

The simplified model of the body levelling is shown in Fig. 2.4. The grey robot outline depicts the robot in the default position. Then, the right leg (orange) is lifted and a new foothold is found on a terrain obstacle. When all the lifted legs are laid, the plane is estimated by the linear regression (bottom light blue dashed line) and shifted to the height h (the top light blue dashed line). From the plane parameters, the rotation matrix is computed and finally, the translation is determined. Note that the rotation is performed prior the translation. The new robot coordinate system is denoted as (O_B', x', y', z').

Fig. 2.5 shows the whole gait cycle with the separated ground detection and body levelling. After a new cycle is started, legs to move are selected. In

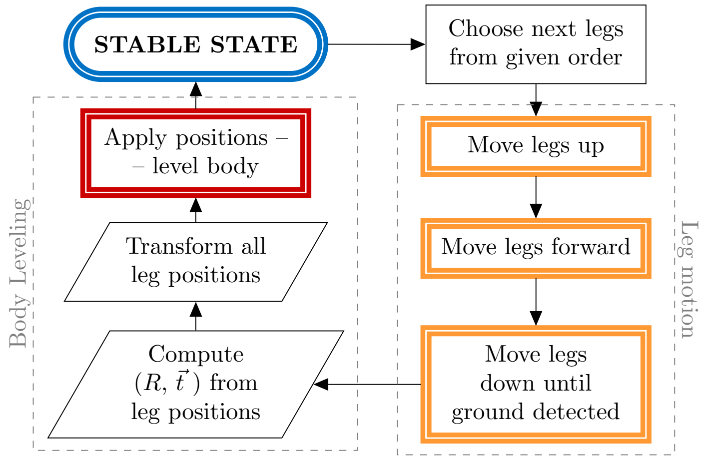


Figure 2.5: Gait diagram [26]

a case of the tripod scheme, alternating triples of legs are selected at once. Then, the legs are moved up and forward. Afterwards, the legs are laid down until all of them reach the ground. From such a new foothold positions the rotation and translation are computed. The legs positions are transformed and body levelling is applied in order to move the body forward.

The adaptive gait [27] allows the robot to traverse various rough terrains including stairs. The approach is minimalistic and does not need any additional sensors. The only bottleneck is the communication, because all the servos are connected to the same serial line. Therefore, this configuration allows to communicate only with one servo at the time. This drawback slows the overall movement of the robot.

2.2.3 Terrain classification for adaptive gait

The principles of the terrain classification based on the servo drive feedback only [4] have been discussed in Section 2.1.2. It is worth mentioning, that the approach [4] relies on the periodicity of the gait to build features in the frequency domain by the FFT. However, this is not directly applicable for the adaptive gait, because the time when all the legs are laid is a priori unknown. Therefore, the ideas of [4] were modified in [26] to allow the terrain classification. The terrain classification method [26] works as follows.

The position error in all servo drives of the two front legs are measured with a sample rate approximately 20 Hz. Then, the gathered data are interpolated and resampled at the frequency of 100 Hz and windowed according to the following gait phases. The leg motion in the adaptive tripod gait is divided into 4 phases:

- body levelling,
- leg lifting,
- moving the leg forward,

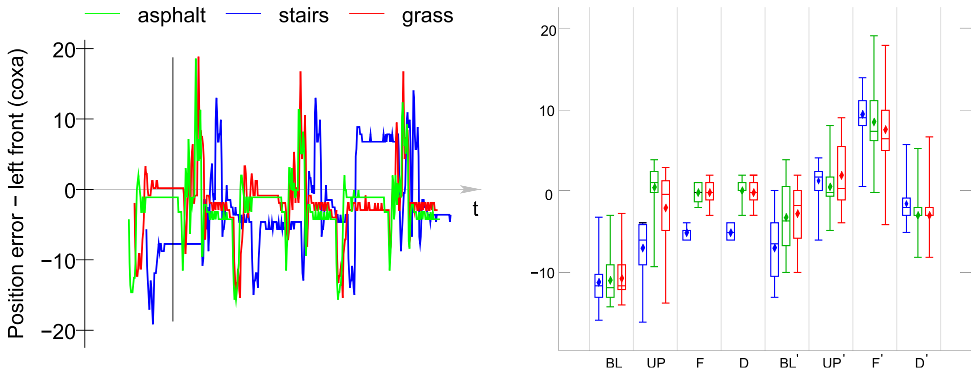


Figure 2.6: Servo position error (the left chart) of the servo drive for three different terrains and its statistical values (the right chart) in the particular subwindows of the whole tripod gait cycle. The subwindows are denoted as follows: BL - body levelling, UP - leg lifting, F - moving the legs forward, D - laying the legs down. The first triple of the legs is moving during the phases denoted without the prime superscript, whereas the second triple of the legs is moving during the phases denoted with the prime superscript

- laying the leg down.

These phases are repeated for both triples of the legs and thus, 8 phases are considered in total for the whole gait cycle. For each phase, the basic statistics of the servo position error are computed, i.e. the minimum, maximum, mean, median, and standard deviation. This gives 5 features (the statistical values) in the 8 subwindows for each of the 3 servo drives of the front legs resulting in 240-dimensional feature vector per one complete gait cycle. An example of the resampled position error data in one servo drive and statistical values in the particular gait phases are shown in Fig. 2.6. The feature vectors are then trained by the SVM, which performs the classification.

The authors of [26] report a high success rate of the terrain classification of the various terrains. That is why this approach has been considered in this work for the terrain classification.

Chapter 3

Visual Road Following

One of the most desired feature for mobile robots is autonomy. It is necessary to obtain a complex view of the robot surrounding to achieve a high level of robot autonomy. Many sensors is available and the most suitable sensor seems to be a digital camera. It is affordable, small, and easy to use. Various features and regions can be extracted from a camera image. In the context of this thesis, the main task for image processing is a robust road detection. However, due to a wide range of road surfaces and shapes, it is difficult to define a road generally. Therefore, it is a very complex task to develop road detection method for all types of road. Moreover, environmental conditions (like light, season changes, etc.) in which the autonomous systems operate make the task even harder. Many approaches for vision based road following have been proposed in literature. Each of them relies on different road features. In the following section, selected techniques are highlighted and summary of the theoretical foundations of the approaches employed in the work of this thesis is provided in Section 3.2.

3.1 Related work

A survey of vision based road detection is presented in [23]. The authors divide the techniques into three main categories:

1. activity driven;
2. feature driven;
3. model driven.

Activity driven approaches are aimed mainly for the lane detection in traffic with a static camera. Even though these methods are not well suitable for mobile robots, two examples are briefly discussed to provide complex overview of the existing road detection methods. Feature driven methods rely on a feature extraction from a particular image. The features are classified and assigned to one of the learnt surfaces. Finally, the model based methods try to assign some defined road model to the image.

A simple activity driven lane detection is proposed in [37]. Motion of the vehicles is detected from the difference of two consecutive frames. The difference image is then filtered to remove light conditions effect and suppress the false positive detection. The filtered image is then accumulated in the activity map. Also a decay process for the activity map is proposed in order to ensure adaptability.

More robust approach of lane detection is presented in [22]. Firstly, a background model is created from the static parts of the camera image. Moving objects are recognized by subtraction of the background model from a new image. The object (vehicles) are then tracked with Kalman filer allowing to keep all positions of the object. Finally, the positions are clustered by K-means or RANSAC algorithms to estimate the centres of lanes.

According to the division of road detection methods from [23] texture and colour based methods can be considered as a feature based detection. In [45], the authors propose to differentiate textures according to statistical values. For each pixel a covariance matrix of the window 5×5 pixels is computed. The texture anisotropy strength is calculated from the covariance matrix and used together with the pixel position as a three dimensional feature vector.

Following feature segmentation is divided into two phases. First, unsupervised segmentation with Mahalanobis distance is used to obtain the probabilities of particular textures. After the statistical model is created, a Bayessian classifier compares each pixel with the model and assigns it to a road or non-road regions.

Another way of texture based road segmentation is presented in [5], where a texture descriptor is composed of the actual pixel colour and intensity of the surrounding pixels. 16 texton classes are found by K-means clustering. All the feature vectors in the image are represented by these classes in a way that the Euclidean distance of the classes from the feature vectors is minimized.

After the classes are established, their histograms are computed in 32×32 windows. The histograms are then again clustered by K-means in order to find representative histogram profiles. The texture from a new image is classified by the histogram comparison with the histogram profiles using the Euclidean distance.

Described texture classification is combined with stereo-vision to detect the road in [5]. The image is segmented according to the texture and stereo-vision marks the segment as a road based on their flatness and width.

Beside a texture, a colour segmentation can be considered as a feature based road recognition method. A robust approach of the colour segmentation is proposed in [9]. At first, the horizon is found in the image considering one assumption - horizon appears approximately as a straight line. The Sobel operator is applied and resulting grey-scale image is thresholded by Otsu method. This gives a binary image which is eroded to filter out noise. Then, the image is divided into ten horizontal sub-images in two phases: the first phase divides the image from the border and the second phase starts division with offset equal to the half of the sub-image height. The number of pixels marked as a horizon is counted in each sub-image and the sub-image with the highest amount is chosen as a horizon. The whole image is ten cropped and only the part under the horizon is further processed for the road detection.

After the image is cropped, segmentation is performed. Two methods are proposed - graph-based and quick shift segmentations. A quick shift is based on the mean shift filter and is explained in [38]. The graph-based segmentation creates a fully connected undirected graph with pixels as nodes. Each edge is evaluated by a weight computed from colour difference, intensity, and position of the connected pixels. The weight represents the dissimilarity of the pixels. The high weight indicates pixels belonging to different segments, whereas the low weight indicates pixels from the same segment. The image is segmented according to these weights. Each segment is further represented in the RGB space by a Gaussian with the given mean and covariance matrix. The RGB mean and covariance are then converted to the HSV space.

The road is marked with a rectangular region of interest (ROI) at the bottom of the image. Within the ROI, similar segments are merged together. When merging two segments, their Gaussians are also merged. Thus, the road area is described by a mixture of Gaussians. The similarity is established from Mahalanobis distance for two Gaussian distributions. Once no other segments can be merged, other segments out of ROI are classified as road or non-road comparing them to the mixture of Gaussians from ROI.

However, ROI can also contain outliers that do not belong to the road. Therefore, only segments with area larger than some specified threshold are considered as the road. The results presented in [9] show that the method is very robust and work in various environments and can handle even slightly unstructured roads.

For a well structured roads a shape model can be defined and Hough transformation can be used to find defined shapes in the image. This approach is utilized in [43]. The authors assume that the road boundary can be parametrized as a parabolic curve. Canny edge detection is applied to find edges in the image. Multiresolution Hough transformation then searches the parabolic model in the binary edge image. Described approach works well for the roads with clear boundaries or drawn lanes.

On the other hand, mobile robots often have to operate in unstructured environments where most of the described methods fail. The road can be outlined only by a tracks, for example. The method that can deal with such cases is vanishing point detection. The authors of [24] utilize two dimensional Gabor filter to find the dominant orientation at each pixel. The image is convolved with a bank of predefined filters with various orientations.

The Gabor energy is computed for all filtered images. For each pixel, the orientation with the highest energy is found and marked as a dominant orientation. However, sufficiently precise results can be achieved only by convolving a large number of filters. This time demanding process is overcome in [24] where only four filters with basic angles are used. The final dominant orientation is calculated as a linear combination of the two orientations with the highest energy. Coefficients of the linear combination are determined by the energy. After that, the vanishing point is selected by voting. Each pixel has a weighted vote for every pixel above itself in the dominant direction. The weight is calculated from the orientation and pixel distance. The pixel with the most votes is chosen as the resulting vanishing point.

3.2 Utilized vision road following methods

Challenging part of the image processing for road recognition are shadows. Due to shadows, additional edges can be observed in the image and also colours of the same surface may differ in different parts of the image. This is the reason, why some of the road following algorithms may fail under presence of shadows. Since in this work, the robot is expected to crawl in outdoor environments, a shadow removal method has been considered as the crucial part of the proposed solution, and therefore, it has been implemented and evaluated. The employed approach is described in this section. Also for a clear description of the approach, theoretical background of the shadow removal is introduced first.

One of the most utilized method for shadow removal is presented in [12]. The approach assumes that the camera sensor has indefinitely narrow band and the bands do not overlap. Moreover, all surfaces are considered Lambertian and the illumination is provided by a black body. None of these assumptions are completely satisfied, but [12] shows feasibility of the proposed method. The main idea of method is as follows.

Let $\mathbf{R} = (R_r, R_g, R_b)$ be the colour intensity of a RGB pixel sensed by a camera. The value of the colour, with the previous assumptions, can be calculated as

$$R_k = \sigma \int E(\lambda) S(\lambda) Q_k(\lambda) \mathrm{d}\lambda, \quad k = r, g, b, \tag{3.1}$$

where σ is Lambertian shading, $E(\lambda)$ is the illumination spectral power distribution, $S(\lambda)$ is the surface reflectance, $Q_k(\lambda)$ is the sensor sensitivity function for each colour, and λ is a wavelength. Considering indefinitely narrow band camera, Q_k becomes Dirac impulse and Eq. 3.1 can be simplified to

$$R_k = \sigma E(\lambda_k) S(\lambda_k) q_k. \tag{3.2}$$

It is possible to approximate the light by the Wien's approximation for the Planck's law with

$$E(\lambda, T) \simeq I k_1 \lambda^{-5} e^{-\frac{k_2}{T\lambda}}, \tag{3.3}$$

where k_1 and k_2 are constants and T is a temperature characterizing the light colour. I defines the overall illumination intensity. Substitution of Eq. 3.3 into Eq. 3.2 gives

$$R_k = \sigma I k_1 \lambda_k^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k. \tag{3.4}$$

After reduction of the three dimensional RGB space to the two dimensional log-chromaticity, intensity and shading information are removed:

$$c_{r,g} = \frac{R_{r,g}}{R_b}, \quad s_k = k_1 \lambda_k^{-5} S(\lambda_k) q_k, \quad e_k = -\frac{k_2}{\lambda_k}, \tag{3.5}$$

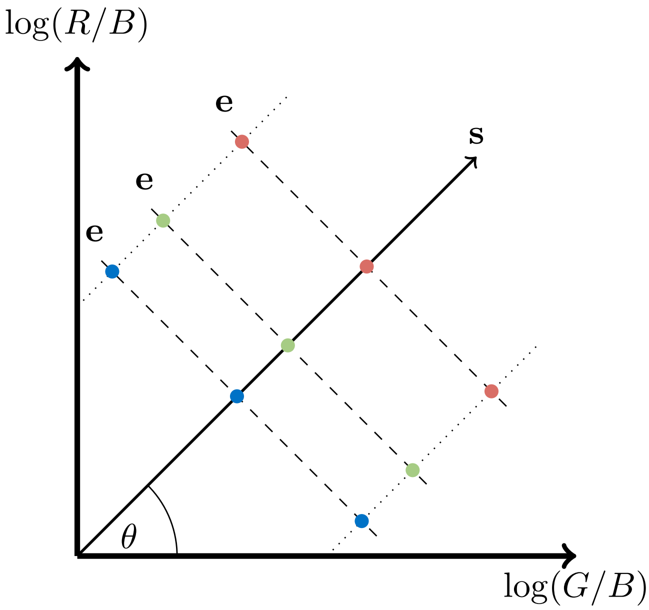


Figure 3.1: Log-chromaticity space. Three surface types are highlighted by different colours. Vector **e** denote the illumination change. As the light changes, surface colours are projected onto line. Vector **s** is the illumination invariant vector.

$$\rho_{r,g} = \log c_{r,g} = \log\left(\frac{s_{r,g}}{s_b}\right) + \frac{(e_{r,g} - e_b)}{T} = \mathbf{s} + \frac{1}{T}\mathbf{e}. \tag{3.6}$$

The two vector **s** is independent of illuminant, while vector **e** is independent of surface. A graphical representation of Eq. 3.6 in the log-chromaticity space is shown in Fig. 3.1. With a known correct angle θ same surface colours are projected onto straight lines as the light varies. θ depends on a camera sensor and can be obtained from the sensor sensitivity. However, sensitivity is usually unknown. Nevertheless, the angle θ can be estimated by the calibration method presented in [13], which is described later in this chapter.

Eq. 3.5 and 3.6 a division by a blue channel is considered for a conversion to the chromaticity space. The selection of the dividing channel may not be clear. Therefore, in real implementation, the geometric mean is used (denoted by the M subscript)

$$R_M = \sqrt[3]{R_r R_g R_b}, \quad s_M = \sqrt[3]{s_r s_g s_b} \tag{3.7}$$

which changes Eq.3.6 to

$$\rho_{r,g,b} = \log c_{r,g,b} = \log\left(\frac{s_{r,g,b}}{s_M}\right) + \frac{(e_{r,g,b} - e_M)}{T}. \tag{3.8}$$

As the result, all three components of the chromaticity are obtained. Since the goal is to transform the RGB space into the two dimensional log-chromaticity

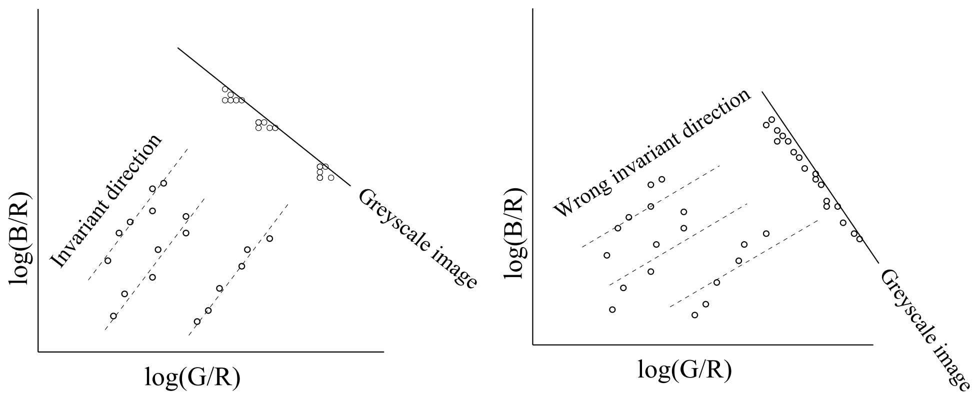


Figure 3.2: Projection of the surfaces in the log-chromaticity space into a grey-scale image (adopted from [13]). For the correct angle, peak appear in the pdf. Incorrect angle spreads the pdf and increase the entropy.

space, a projection, which transforms the three dimensional chromaticities to a plane, is then performed as

$$\chi = \mathbf{U}\rho. \tag{3.9}$$

From χ a grey-scale invariant image is formed as

$$\mathcal{I} = \chi_1 \cos\theta + \chi_2 \sin\theta. \tag{3.10}$$

The idea of the θ estimation proposed in [13] is following: a vector perpendicular to the vector $\mathbf{e}$ can be found and all the projected surfaces in the log-chromaticity space can be projected onto this line in a direction of the vector $\mathbf{e}$. The perpendicular line represents a grey-scale invariant image. If the angle is determined correctly, the probability density function (pdf) of the image contains several peaks and thus, the entropy is low. Otherwise, the pdf will be spread and the entropy will be higher. An illustration of the idea is shown in Fig. 3.2.

Thus, an invariant image can be computed for angles from the interval $\langle 0°, 180° \rangle$ and the angle with the lowest Shannon entropy is selected as the best value of θ.

Reliability of the estimation approach is discussed in [1]. The authors show cases of the self-calibration instability and propose a new calibration method. First, the Chebyshev's theorem is used to remove outliers from the image. Secondly, the entropy from several images is analysed and θ is estimated from a set of samples. However, for the purpose of this work, the calibration method [13] showed to be sufficient, and it is utilized in the proposed road following with the hexapod crawling robot.

The road detection method utilized in this work is based on the approach [19]. A greyscale invariant image $\mathcal{I}$ is computed following Eq. 3.7 to 3.10. The image $\mathcal{I}$ is then filtered and thresholded. The thresholded binary image contains separated road and surrounding pixels. Details about the emplyed filtering and thresholding together with the particular implementation are described in Section 4.3.2.

Chapter 4

Proposed Road Following Strategy

4.1 Problem definition

When visual road following is used to keep the robot on the road, few drawbacks have to be considered in order to make the control robust and do not allow the robot to leave the road. In the outdoor environments, any image processing method has to deal with changing light conditions. The influence of the scene illumination may have crucial impacts on the quality of the processed output. The key issue in the road detection is how to define a road. There are several approaches in literature and couple of the most representative are discussed in Chapter 3.

The problem addressed in this work is considered in the context of a hexapod walking robot crawling structured roads with different terrain structure than the surroundings areas, e.g. an asphalt road with a grass or dirt around. For a road defined in this manner, colour or border detection is in most cases sufficient. Advantage of such road detection methods is that they are computationally efficient and can be deployed on the on-board, low powerfull computational resources of the hexapod walking robot. However, a simple road detection is usually unable to adapt to new conditions and can be easily affected by environmental changes. For example when a colour of the surrounding terrain is similar to the road colour or overexposure of a camera sensor occurs, it can lead to the false positive detection. The robot is then led off the road. Because the goal of this work is to keep the robot on the road, a false positive detection is considered to be much more costly than false negative.

Therefore, the proprioceptive sensing is employed to eliminate false positive detections of the vision-based road following method. The idea is to use servo drive feedback to determine the currently crawled terrain. A desired behaviour of the control strategy is shown in Fig. 4.1. Based on this motivation, one of the main problems addressed in this thesis is to propose control strategy, that will steer the robot in a way of the red dashed trajectory shown in Fig. 4.1. To achieve it, a history of the crawled terrains have to be stored. However, it is not expected the robot will stay entirely on the road all the time, since it needs to perceive sensor information of crawling off the road.

The proposed solution shall combine complementary visual based and

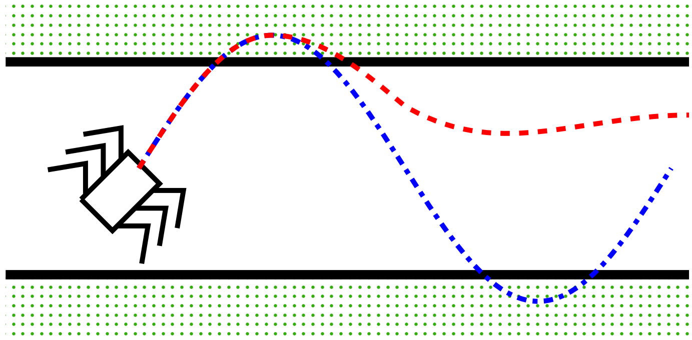


Figure 4.1: Behaviour of the robot on the road with and without considering the terrain history. The blue dash-dotted trajectory depicts steering where only the current terrain is used to control the robot. The red dashed trajectory shows desired motion where the robot is able to estimate the real road direction based on the history of the terrain types.

tactile based road following methods in a way that increases the overall system robustness and keeps the robot on the road even in cases of false positive visual detections and in conditions where the visual detection is unreliable, i.e. overexposure or darkness. The terrain classification steering uses no additional sensors; thus, the robot is practically blind and can react to current or past events only. Since the visual road following is able to foresee the road ahead the robot and produce smooth control, it shall be prioritized over the terrain classification. The task can be divided into three phases:

- tactile road following,
- visual road following,
- fusion of the produced control actions.

The proposed tactile road following is described in Section 4.2, the adopted visual methods are discussed in Section 4.3 and the proposed fusion of control actions from the tactile and visual controllers is designed in Section 4.4.

4.2 Tactile road following

4.2.1 General considerations

The construction of the robot platform, as discussed in Section 2.2.1, divides the body into left and right sides. A road following method based on the terrain classification requires the ability to recognize different terrains on both sides of the robot. Prior the road following approach presented below, we firstly consider a separate classification for left and right sides in which one SVM model has been created for each side. However, our early experimental results have not been promising even on easily distinguishable terrains.

After further investigation it was revealed that during walking, one side of the legs significantly affects servo trajectories on the other one. The influence then causes failure of the terrain classification. Thus, more general framework has been defined, in which four basic types of the terrain can be distinguished with respect to the robot:

1. *On road* - terrains of the defined road, the robot has all legs on the desired road surface.
2. *Off road* - all other terrains than *on road*, the robot has all legs off the road surface.
3. *Left off road* - the left border of the road, left legs are off the road.
4. *Right off road* - the right border of the road, right legs are off the road.

The introduced terrain types define a set of generic terrain classes $\mathcal{T}_\mathcal{C}$ (for brevity, the names are shortened):

$$\mathcal{T}_\mathcal{C} = \{\texttt{On}, \texttt{Off}, \texttt{Off}_\texttt{left}, \texttt{Off}_\texttt{right}\}. \tag{4.1}$$

Notice, multiple terrains can be represented by a single class from $\mathcal{T}_\mathcal{C}$, e.g. grass and dirt can both represent `Off` class.

Because separate prediction for both sides failed, a single classification model has been created in a way described in [26]. Terrain surfaces (grass, asphalt, dirt, etc.) and their borders were trained to follow the idea of $\mathcal{T}_\mathcal{C}$ set. A disadvantage of this approach is in increased complexity of the training phase, which requires training of the borders between the `On` and `Off` classes (the borders assigned to $\texttt{Off}_\texttt{left}$ and $\texttt{Off}_\texttt{right}$, respectively).

It is worth noticing that for training the classifier data from a straight walk are currently used. Rotation speed changes the trajectories of the servo drives, which are used directly as a features for the SVM. This fact is further taken into account during the design of a control law. Here, we would also like to highlight that the proposed approach has been accepted as the ICRA 2016 conference paper [35].

4.2.2 Road following control strategy

As depicted in Fig. 4.1, the challenge of the tactile road detection without any other sensory equipment is to estimate the road direction. A simple reactive controller, which steers the robot using the single last measurement, may exhibit oscillations (blue dash-dotted line in Fig. 4.1), since it only pushes the robot away from the off road. The goal is to design a controller that will achieve a trajectory shown by the red dashed line in Fig. 4.1. Such a controller design problem can be described as follows.

The robot is considered to walk forward with a constant velocity. Therefore, the problem of controller design is to compute the angular velocity steering the robot away from the off road. Note that for the tactile terrain classification, a robot has to crawl the terrain for some time to collect new data. Due to this constrain, it is not expected the robot will stay entirely on the road, but it is rather assumed the robot can partially leave the road. The controller's task is then to return the robot back on the road and approximately follow the road direction. For this purpose a history of the last N crawled terrains samples T_i is utilized, where

$$T_i \in \mathcal{T_C} \tag{4.2}$$

holds. Each sample contributes to the final angular velocity $\dot{\theta}$, but the older the sample is, the weaker its contribution should be. Moreover, for each sample, all preceding history has to be taken into account, i.e. the same terrain types from $\mathcal{T_C}$ for a different T_i may have various contribution if the previous samples are different. Thus, the control low is proposed as

$$\dot{\theta} = k_p \sum_{i=1}^{N} \frac{1}{\eta i} s_i w_i, \tag{4.3}$$

where $\dot{\theta}$ is the angular velocity directly applicable to the gait controller, k_p stands for a proportional control gain, η is an ageing factor, and s_i is a sign of the steering weight w_i. The terrain sample T_1 is the most recent and T_N is the oldest one.

The ageing factor η decreases the influence of the particular sample to the final control action as i is increasing. The weight w_i and its sign s_i reflect the history according to the actual and previously detected terrains T_i. At the end of each gait cycle, the terrain is classified from data gathered in the last 3 gait cycles. A new terrain sample $T_i \in \mathcal{T_C}$ is predicted and added at the beginning of the terrain history queue. For a new T_i, a weight w_i is set from one of three possible values:

$$w_i(T_i) = \begin{cases} W_{off} & \text{if } T_i \text{ is } \texttt{Off}, \\ W_b & \text{if } T_i \text{ is } \texttt{Off}_{\texttt{left}} \text{ or } \texttt{Off}_{\texttt{right}}, \\ w_{on}(i) & \text{if } T_i \text{ is } \texttt{On}. \end{cases} \tag{4.4}$$

W_{off} and W_b are predefined constant values standing for the off road and border weights. When the robot hits the terrain border, usually a slight correction by the weight W_b is enough to return the robot back to the road. On the other hand, after going completely off the road, more aggressive action, represented by W_{off}, is required. Thus, the values are considered to satisfy inequality

$$0 \leq W_b \leq W_{off}. \tag{4.5}$$

The weight $w_{on}(i)$ is, however, a function of the previous terrains T_i. Therefore, once a robot is on the road for a few last gait cycles, it should keep its straight heading. But if the robot was previously off the road, it should keep the turning radius to return the robot to the expected road direction. Similarly a slight change is required in the case the robot was crawling the border. For these reasons, the weight function $w_{on}(i)$ is computed as

$$w_{on}(i) = \begin{cases} W_{off} & \text{if } n_{off} \geq N_{off}, \\ W_b & \text{if } n_b \geq N_b \text{ and } n_{off} < N_{off}, \\ 0 & \text{otherwise} - \textit{keep the direction}. \end{cases} \tag{4.6}$$

The number of samples T_i in the history queue classified as `Off` is denoted n_{off}, the number of $\texttt{Off}_{\texttt{left}}$ or $\texttt{Off}_{\texttt{right}}$ is denoted n_b. The constant values N_{off} and N_b determine the number of T_i samples for which the history is considered mostly off road or mostly as the border terrain, respectively.

The second variable to reflect the terrain history is a sign s_i of the weight w_i. The sign has two possible values

$$s_i \in \{-1, 1\}. \tag{4.7}$$

The positive sign denotes the counter-clockwise rotation, whereas the negative sign stands for the clockwise rotation. The value of s_i assigned to the weight w_i is determined from the actual terrain class T_i and the previous terrain T_{i-1} in following way:

$$s_i = \begin{cases} -1 & \text{if } T_i = \texttt{Off}_{\texttt{left}} - \textit{left side border}, \\ 1 & \text{if } T_i = \texttt{Off}_{\texttt{right}} - \textit{right side border}, \\ -s_{i-1} & \text{if } T_i = \texttt{On} \text{ and } T_{i-1} \neq \texttt{On}, \\ s_{i-1} & \text{otherwise}. \end{cases} \tag{4.8}$$

The first two conditions in Eq. 4.8 simply force the robot to crawl away from the road boundaries. The third and fourth conditions capture the transition from the off road or the road border to the on road and set the sign opposite to the previous one. In combination with Eq. 4.6, the current terrain class T_i contributes conrary to the final action. This allows the robot to compensate the whole manoeuvre and partially return to the original road direction. After w_i and its corresponding s_i are established, the control action is computed using Eq. 4.3.

4.2.3 Limits of the proposed control strategy

The proposed strategy and control law can handle most of the common situations encountered during crawling the robot on the road. However, it is necessary to mention the following important assumption – the control strategy determines the direction (represented by s_i) from the border terrain. Thus, if the direct transition from `On` to `Off` occurs, there is no information about the road direction and s_i cannot be set correctly. Using tactile information only, this situation is impossible to handle.

The SVM model is trained from straight walks data only. Since it uses leg trajectories to create the feature vector, different trajectories can cause a confusion in the classification. It follows that for classification only the straight or slightly rotated steps can be used. With respect to the size of the robot and size of the outdoor roads it is acceptable to alter one straight step (without applied control action) with turning steps if the rotation is too large. In a case the rotation does not exceed the defined threshold, the terrain prediction is carried out for all steps. This approach allows to react to the terrain other than on road as quickly as possible, because on the road, the action is equal to zero.

4.3 Visual road following

In contrast to the tactile road following, which can recognize the terrain after it is crawled, the vision based approach can recognize the scene ahead of the robot. The goal is therefore to steer the robot to the region with detected road. This allows a smooth motion through the environment.

A general task in the visual road following is to divide the image pixels or pixel areas to two groups of road and non-road pixels. From the road pixels, the forward and angular robot velocities are then computed. In this work, two different methods for road pixel recognition are employed. The first method is a simple approach based on the colour distance of the pixels from the previously learned colours, that is presented in Section 4.3.1. Section 4.3.2 describes the second approach based on shadow removal and road search in the intrinsic image. Both methods use the same algorithm to compute the forward and angular speed, which is discussed in Section 4.3.3.

4.3.1 Colour based road recognition

The colour similarity can serve as a simple feature for differentiating the road pixels from the rest of the image. This method has been mostly implemented according to [20] and works as follows. After the image is captured by the camera, the default RGB colours are converted to the HSV model. The HSV model partially eliminates the influence of the various illumination of the same colour. The pixels are then evaluated independently. For each pixel a colour distance from know road colour is computed as

$$\Delta E^* = \sqrt{(H_e - H_l)^2 + \frac{(S_e - S_l)^2}{4} + \frac{(V_e - V_l)^2}{16}}, \tag{4.9}$$

where H, S, V are particular components of the pixel colour, the subscript e denotes the current pixel and the subscript l denotes the learnt pixel or pixels, respectively. If the distance is lower than the predefined threshold, the pixel is marked as a road pixel. However, it can be computationally demanding to compute Eq. 4.9 for each pixel in the image. Moreover, it is required to learn more than one reference pixel and thus, the computation would be repeated for every learnt pixel.

Repeated calculation of Eq. 4.9 during the online road recognition has been avoided by the following approach. Assuming 24-bit colour depth, the RGB space can be represented by a three dimensional array with size $256 \times 256 \times 256$. This space can be reduced, since absolute accuracy is not necessary. Therefore, during the road recognition the whole RGB space is four times reduced to array $64 \times 64 \times 64$, where each cell substitutes neighbouring colour in the neighbourhood $4 \times 4 \times 4$. This smaller array is then used in the decision process as a memory of the classification results indexed by the particular colour of the examined pixel. Logical 1 is stored in the array cell in a case the colour represented by the cell belongs to the road, logical 0 is stored otherwise. This approach simplifies the online road recognition to checking the values in the offline precomputed array.

When learning new colour, the distance from the learnt colour is computed for each RGB value according to Eq. 4.9. Logical 1 is added to the array if the evaluated colour has lower distance to the learnt pixel than the given threshold. Cells are not reset to logical 0 since it is required to learn more colours at once and storing logical 0 can overwrite results from previous learning.

Road colour learning can be accomplished in two different ways: manually and autonomously. In the manual mode, the camera image is displayed and user can teach the colour by selecting the road pixel with a mouse click. The system is also able to learn the colour autonomously and periodically refresh the knowledge. In this case, it is assumed that the road is in the middle of the lower part of the image. Thus, this area can be marked as a source of the road pixels. Every N steps, where N should be selected according to the robot forward velocity, the algorithm resets the evaluation array to zeros and learns the colours from the pixel within the marked area.

The colour based road recognition is simple and with described array checking it is also very computationally efficient algorithm. Nonetheless, it is not robust to light changes and works only in environments, where the road and surrounding terrain have sufficiently different colour. Therefore, additional vision-based algorithm has been developed.

4.3.2 Intrinsic image based road recognition

The main disadvantage of the simple colour based approach from Section 4.3.1 is incapability do deal with varying light and shadows on the road. As discussed in Section 3.2, assumptions for shadow removal were proposed in [12] together with a complete method of removing the shadows from colour images. However, based on initial evaluation, the process [12] is too computationally demanding to obtain a fully shadowless colour image in real-time.

On the other hand, computation of the 2D log-chromaticity, as an intermediate step in the full shadow removal [12], is quite fast and can be done in real-time using on-board computational resource of the utilized hexapod walking robot. The authors of [19] suggest to search the road in the 2D log-chromaticity since it already does not contain shadows. With the theoretical background described in Section 3.2 the implementation of the 2D log-chromaticity image is straightforward and the algorithm is listed in Alg. 4.1.

Algorithm 4.1: Computation of the 2D log-chromaticity image.

```
function computeLogChromaticity($\mathcal{I}$, $\theta$)
    for x = 0 to $\mathcal{I}$.width do
        for y = 0 to $\mathcal{I}$.height do
            $R_{r,g,b}$ = $\mathcal{I}$[x,y] scale to $\langle 0, 1 \rangle$
            $R_M$ = $\sqrt[3]{R_r \cdot R_g \cdot R_b}$
            $\rho_{r,g,b}$ = $\log(R_{r,g,b}/R_M)$
            $\chi$ = $\mathbf{U} \cdot \rho$

            $\mathcal{I}_i$[x,y] = $(cos\theta \quad sin\theta) \cdot \chi$
        end for
    end for

    $\mathcal{I}_i$ = $\mathcal{I}_i$ scale to $\langle 0, 255 \rangle$

    return $\mathcal{I}_i$
end function
```

The image is walked through pixel by pixel. Colours of each RGB channel are normalized to the interval $\langle 0, 1 \rangle$ and saved in the variables R_r, R_g, and R_b. The geometric mean R_M of all normalized channels is calculated. The log-chromaticity $\rho_{r,g,b}$ is then obtained and projected into a plane with the matrix $\mathbf{U}$, which is an orthogonal matrix with the dimensions 2×3. After the projection, $\rho_{r,g,b}$ is represented in the plane by χ. In [13], the suggested value of $\mathbf{U}$ is

$$\mathbf{U} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{bmatrix}. \tag{4.10}$$

The greyscale intrinsic image $\mathcal{I}_i$ is finally formed from χ as

$$\mathcal{I}_i = \chi_1 \cos\theta + \chi_2 \sin\theta. \tag{4.11}$$

The angle θ is a parameter of the camera, which can be determined from the image entropy [13]. The estimation is implemented as an iterative process, where an intrinsic image is calculated for all angles from 0° to 180° and the lowest entropy is found with the corresponding angle θ. Alg. 4.2 shows how θ estimation proceeds.

Algorithm 4.2: Estimation of the angle θ.

```
function estimateAngle($\mathcal{I}$)
    size = $\mathcal{I}$.width · $\mathcal{I}$.height
    $I_{max}$ = 0.95 · 255
    $I_{min}$ = 0.05 · 255
    $H_{min}$ = inf
    $\theta$ = -1

    for $\theta_i$ = 0 to 180 do
        $\mathcal{I}_i$ = computeLogChromaticity($\mathcal{I}$, $\theta_i$)
```

```
        I_in = ∀i : i ∈ I_i ∧ I_min < i < I_max
        inliers = |I_in|
        outliers = size - inliers

        μ = mean(I_in)
        σ = std_dev(I_in)

        bin_width = 3.49 · σ · inliers^(-1/3)
        H_i = ShannonEntropy(I_in, bin_width)

        if H_i < H_min then
            H_min = H_i
            θ = θ_i
        end if
    end for

    return θ
end function
```

The algorithm iterates through the integer values of the angle θ_i from $0°$ to $180°$. In each iteration, an intrinsic image $\mathcal{I}_i$ is computed. 5% of the brightest and darkest pixels from $\mathcal{I}_i$ are discarded and the numbers of inlier and outlier pixels are determined. In Alg 4.2, the set of all inlier pixels is denoted $\mathcal{I}_{in}$. From inliers only the mean μ and standard deviation σ are calculated.

After the outliers are removed from the image, a histogram of the image is created and its bin width is established from the Scott's rule that defines the bin width h as

$$h = 3.5\,\sigma\,n^{-1/3}, \tag{4.12}$$

where σ is the standard deviation of the processed data and n is the number of samples. With the known bin width h the histogram and consequently a sample probability $p_t(i)$ can be computed. The Shannon entropy H of the image $\mathcal{I}_{in}$ is then calculated as

$$H = -\sum_t p_t(i)\log_2 p_t(i), \quad i \in \mathcal{I}_{in}. \tag{4.13}$$

During the road detection, an image passes several stages of the processing. At first, the Gaussian blur is used to remove sharp edges and noise. After the initial filtering, the intrinsic image $\mathcal{I}_i$ is computed and its histogram is equalized. The image is then thresholded; so, the road is separated from the surrounding. Then, the opening morphology is applied to filtered out the noise in a form of the small granularity. Opening transformation is composed of an erosion followed by a dilatation that removes small objects from the image. At the end, the contours are found and redrawn to smooth the road edges. The result of the whole processing is a binary image.

The whole image processing is summarized in Alg. 4.3. The parameter $\mathcal{I}$ denotes the input colour image, σ_{GB} is the standard deviation of the

Gaussian blur, θ stands for the best angle for the intrinsic image computation, τ is a threshold for separation of the road and non-road pixels. Due to the independent pixel-wise processing, the algorithm execution can be easily parallelized. Results of particular algorithm phases are shown in Fig.4.2.

Algorithm 4.3: Image processing from the source to the binary image.

```
function processImage(𝓘, σ_GB, θ, τ)
    𝓘 = gaussianBlur(𝓘, σ_GB)
    𝓘_i = computeLogChromaticity(𝓘, θ)
    𝓘_i = equalizeHistogram(𝓘_i)
    𝓘_bin = threshold(𝓘_i, τ)
    𝓘_bin = morphologyOpen(𝓘_bin)
    c = findContours(𝓘_bin)
    drawContours(𝓘_bin, c)

    return 𝓘_bin
end function
```

4.3.3 Road following algorithm

The output of the road recognition described in Section 4.3.1 and 4.3.2 is a binary image in which road is represented by 1 and 0 denotes the surroundings. A method based on [20] has been developed to find a path in the binary image and compute the steering action. Additionally to [20], a road border search and off road search have been implemented. These features are required for the semi-autonomous learning procedure described in Section 4.4.4.

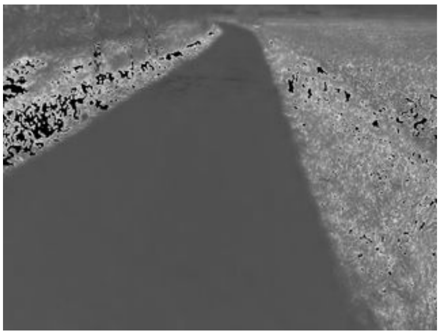
A basic form of the algorithm (road search only) is listed in Alg. 4.4. The input parameters are as follows: $\mathcal{I}$ - binary image (product of the road recognition), ϵ_{max} - the maximal noise in the row, δ_{max} - the maximal number of consecutive image rows without a path, w_{min} - the minimal path width. The image is considered to have the origin in the top left corner.

The image is processed from the bottom to the half in order to reduce the required computation time. Each row is searched from the middle to the sides - first to the right, then to the left. If a pixel is marked as a road, the search continues. Otherwise, the column noise counter ϵ is increased. If the counter exceeds the threshold ϵ_{max}, the search on the current side is stopped and the actual cursor position is saved as the road border position.

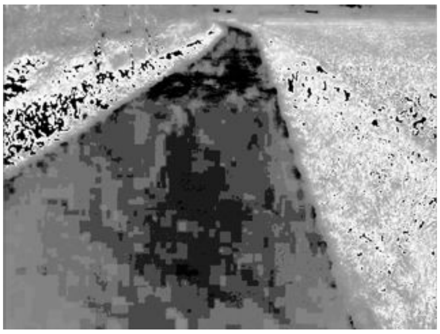
After the borders in the row are set, the path center and width are calculated. The width has to be larger than the defined limit w_{min}, otherwise the path is considered too narrow to traverse. If it is not, the row noise counter δ is increased. Similarly to the noise in the row ϵ, the threshold δ_{max} is defined for the consequent rows with the narrow or none path found. The processing is terminated when the noise δ exceeds the threshold δ_{max}.

The steering action, i.e. the angular velocity ω and the forward velocity v, are calculated from the path center in the last row and the path length (the number of rows with the detected path), respectively.

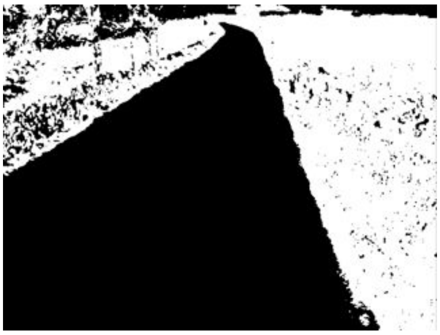
(a): Input image



(b): Invariant image



(c): Equalized histogram



(d): Thresholded image

(e): Filtered and inverted image

(f): Detected path

Figure 4.2: Example of road recognition using the invariant image. From input image (4.2a) an invariant image (4.2b) is computed first. Then, its histogram is equalized (4.2c) and the image is thresholded (4.2d). Morphological opening is applied to filter the binary image (4.2e). At the end, the image is inverted; so, logical 1 represents a road and its surroundings is denoted by logical 0. The found path is shown in Fig. 4.2f

Algorithm 4.4: Searching of the path in binary image.

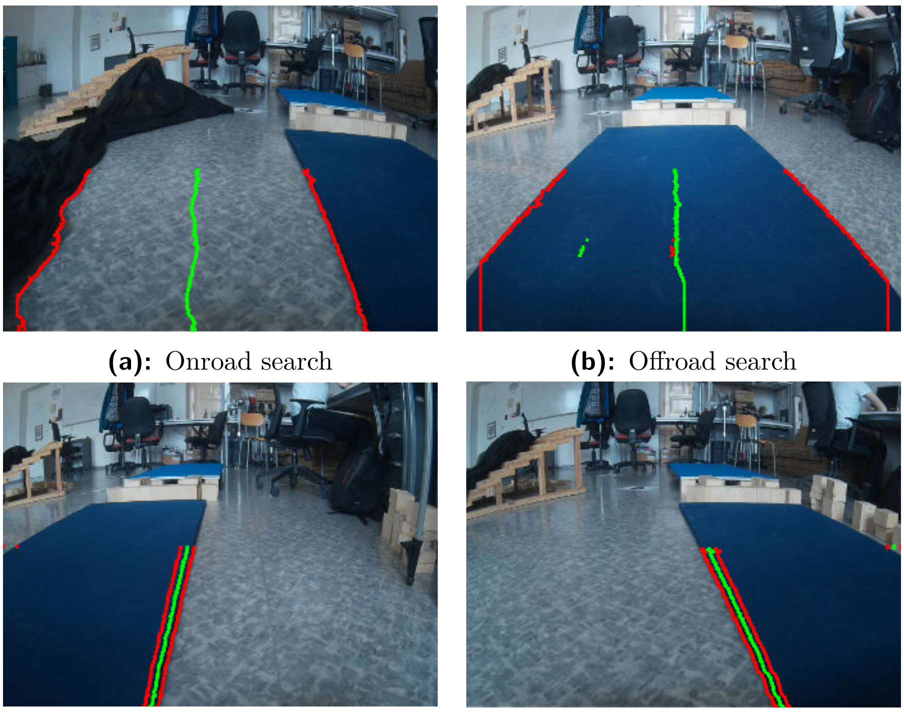
```
function FindPath(𝓘, ε_max, δ_max, w_min)
    δ = 0
    path_length = 𝓘.height / 2

    for y = 𝓘.height to 𝓘.height / 2 do
        ε = 0
```

```
        right_border = 𝓘.width - 1
        path_center = 𝓘.width / 2

        for x = path_center to 𝓘.width do
            if 𝓘[x,y] == 0 then
                ϵ = ϵ + 1
                if ϵ ≥ ϵ_max then
                    right_border = x - ϵ_max
                    break
                end if
            else
                ϵ = 0
            end if
        end for

        ϵ = 0
        left_border = 𝓘.width - 1

        for x = path_center to 0 do
            if 𝓘[x,y] == 0 then
                ϵ = ϵ + 1
                if ϵ ≥ ϵ_max then
                    left_border = x + ϵ_max
                    break
                end if
            else
                ϵ = 0
            end if
        end for

        path_width = right_border - left_border
        if path_width < w_min then
            δ = δ + 1
            if δ ≥ δ_max then
                path_length = I.height / 2 - y
                break
            else
                δ = 0
            end if
        end if

        path_center = (right_border + left_border) / 2
    end for

    ω = c_ω · path_center        // angular speed
    v = c_v · path_length               // forward speed
```

```
53
54      return ω, v
55  end function
```

Alg. 4.4 has been parametrized to allow additional off road and border search as follows. The off road search can be achieved by modification of the conditions on lines *11* and *26*, which in Alg. 4.4 check the presence of the non-road pixels (logical 0). If the conditions are modified to check the road pixel (logical 1), the off road search can be performed, since the noise counter is increased in a case the road pixel is encountered. However, the border search requires the change in direction of the row search. Each row is originally searched from the center to sides. Switching the search direction from sides to the center results in the border search. Thus, the for-loop conditions on lines *10* and *25* have to be adjusted depending on the side, from which the border is searched.

The results of all search modes of the proposed road detection are depicted in Fig. 4.3. For the laboratory evaluation, the colour based road detection has been utilized. The grey floor was marked as a road and highly contrast materials (blue board and black cloth) have been used to bound the path.



(a): Onroad search **(b):** Offroad search

(c): Left border search **(d):** Right border search

Figure 4.3: Example of searched path with all four possible criteria.

4.4 Fusion of visual and tactile controllers

Fusion of the visual and tactile controllers should result in increased robustness of the proposed road following system. The goal is to compute one final action from the outputs of both controllers. Actions computed by each controller are independent, and therefore, one possibility is to combine both actions into a single command, however, the output of the controllers may differ significantly and the resulting action may be incorrect.

Therefore, the propose solution is based on switch strategy to select the most suitable control action based on evaluation of the image quality and terrain classification reliability. It selects exactly one controller which has a complete control over the robot's motion. The controllers are switched according to the defined criteria. The vision based technique is prioritized because it provides a smooth crawling on the road for appropriate conditions. There are two typical cases, where visual system shall be disabled and tactile following shall take over the steering:

1. inappropriate light conditions,
2. other terrain that `On` is detected.

These two cases has been covered by two evaluation criteria proposed in the following sections. In Section 4.4.1, the scene light conditions are checked directly from the image. Then, the history of the crawled terrain is examined and its reliability is analysed in Section 4.4.2.

4.4.1 Image quality criteria

The scene illumination can be evaluated using the image brightness and contrast. The image brightness is computed as follows. The input image is converted from the RGB to the HSV colour model. The average brightness $\overline{B}$ is computed from the HSV model, namely from the V channel as

$$\overline{B} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} V_{i,j}, \tag{4.14}$$

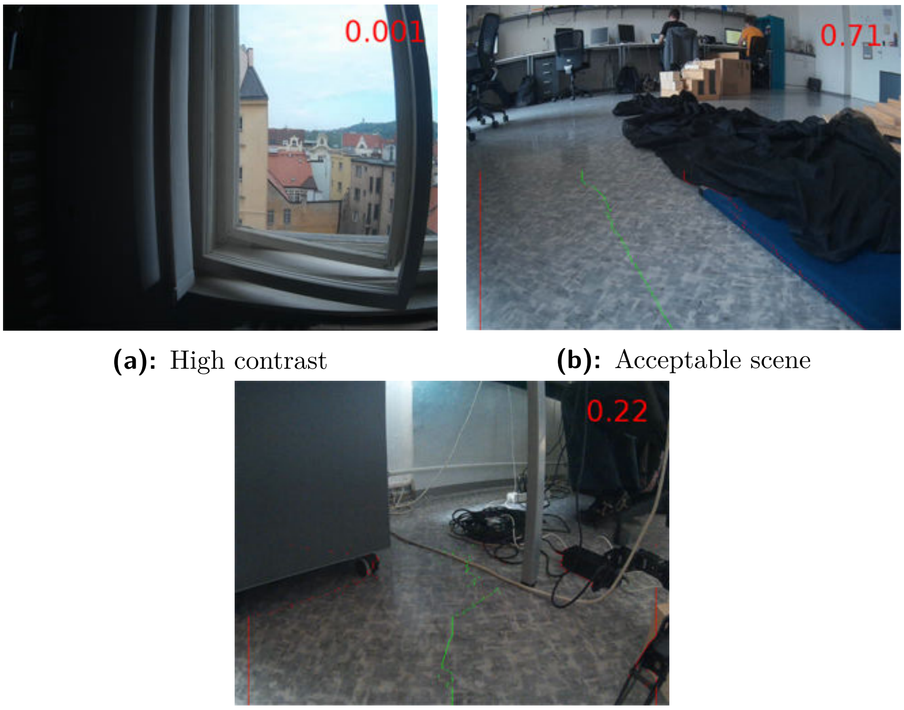
where M is the image height, N is the image width, and $V_{i,j}$ is the value of the V channel at the pixel position (i, j).

For the contrast evaluation, the RMS contrast is computed. To do so, the input image has to be converted to greyscale. Then, average pixel intensity $\overline{I}$ is computed from the particular intensities $I_{i,j}$ as

$$\overline{I} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} I_{i,j}. \tag{4.15}$$

With the known average intensity $\overline{I}$, the RMS contrast C_{RMS} can be calculated as

$$C_{RMS} = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\overline{I} - I_{i,j}\right)^2}. \tag{4.16}$$



(a): High contrast **(b):** Acceptable scene

(c): Dark area

Figure 4.4: Experimental results of the image quality function from the laboratory environment. High contrast in Fig. 4.4a caused the quality to drop to very low value of 0.001. The quality is lower (0.22) as well if the image is too dark (Fig. 4.4c). An acceptable image with the quality 0.71 is shown in Fig. 4.4b.

For both, the brightness and the contrast, separate Gaussians are created. The mean values of brightness μ_B and contrast μ_C, respectively, with corresponding variances σ_B^2 and σ_C^2 are determined experimentally from a set of images. For various light conditions, the set is collected with a human supervision. Visual road recognition algorithm is executed and the supervisor evaluates the quality of the recognized road. If the road is recognized correctly, the image is added into the set.

After the parameters are established, the Gaussians are scaled to the interval $\langle 0, 1\rangle$. When the image quality is evaluated, the values of the brightness (q_B) and contrast (q_C) Gaussians are obtained. The resulting image quality q is computed as the multiplication of both values

$$q = q_B \cdot q_C, \quad q \in \langle 0, 1\rangle. \tag{4.17}$$

Experimental results of the proposed image quality criteria are shown in Fig. 4.4.

A threshold with a specified hysteresis is used to switch the active controller from the visual to the tactile and back. The hysteresis reduces influence of the image noise in situations, where the image quality is near to the specified threshold value.

4.4.2 Terrain classification criteria

The second switching criteria covers false positive detection, i.e. the scenario where the visual system detects road on the different terrain than `On`. This may happen, for example, if the road surrounding has similar colour to the road. The robot then crawls off the road and the terrain classification recognizes the road border or the off road. Then, it is a task for the tactile road following method presented in Section 4.2 to lead the robot back on road.

In the tactile road following, a history of the crawled terrains T_i is utilized. For each terrain its contribution in a form of the weight w_i and the sign s_i is determined according to the previously detected terrains. From all the contributions, the variance σ^2_{TC} is calculated ($E(x)$ denotes the mean value)

$$\sigma^2_{TC} = \frac{1}{N} \sum_{i=1}^{N} \left(w_i s_i - E\left(w_i s_i\right)\right)^2 . \tag{4.18}$$

Also the number of terrain changes n_{CH} in the history, i.e. $T_i \neq T_{i-1}$, is counted.

The values σ^2_{TC} and n_{CH} are then used as a tactile road following reliability indicator. For both indicators, threshold values are set and tactile following is considered reliable only if both values are lower than the thresholds. n_{CH} eliminates frequent altering between terrain types. The variance σ^2_{TC} filters out misclassifications with a radical impact on the output action, e.g. $\texttt{Off}_{\texttt{left}}$ between $\texttt{Off}_{\texttt{right}}$ terrains.

Besides, the action computed by the tactile road following is checked. If it is large enough and conditions of the variance and the number of terrain changes are fulfilled, tactile following can be activated.

4.4.3 Rules of the controller switching

The switching logic is depicted in Fig. 4.5. It can be represented by a simple state machine with two states. The current state can be changed only if the criteria presented in Section 4.4.1 and 4.4.2 pass the defined conditions. The system starts in the visual road following mode. If at least one of the criteria requests to disable visual following, steering is passed to the tactile following. The control is handed back to the visual following in a case the image quality is high enough and the robot crawls the `On` terrain.

4.4.4 System learning

A raw terrain data from servo drives has to be gathered first for all expected terrain types of $\mathcal{T}_C$ to train the SVM for terrain classification. Therefore, a learning method has been designed to simplify the deployment in a new environment. It offers two possible ways of learning - manual and semi-autonomous. In the manual mode, the user controls the robot with a standard USB joystick. In the semi-autonomous mode, the visual road following leads

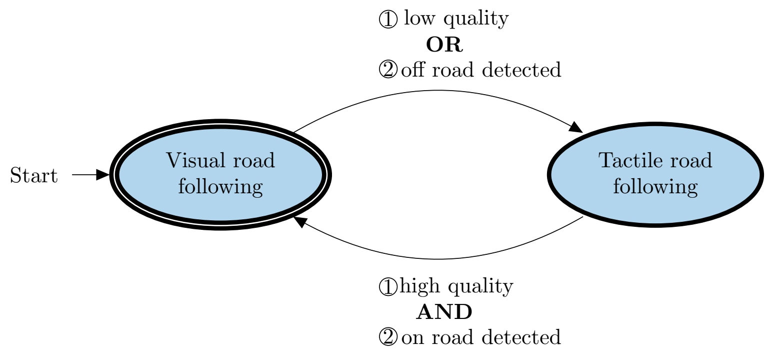


Figure 4.5: Logic of the controller switching with necessary conditions of image quality (1) and detected terrain (2) criteria

the robot using algorithms for the border and off road detection described in Section 4.3.3.

The learning procedure is the same for both modes of the learning. It is based on a simple state machine. The robot crawls the specified number of steps on the given terrain while controlled by the selected mode. A user is then asked if the data collection for the terrain should be repeated or it can continue with a new terrain. When a new terrain is selected, the robot has to be manually placed on the new terrain. Gathering can then continue. After data for all terrains are collected, the SVM model is trained. The flow diagram of the learning process is shown in Fig. 4.6.

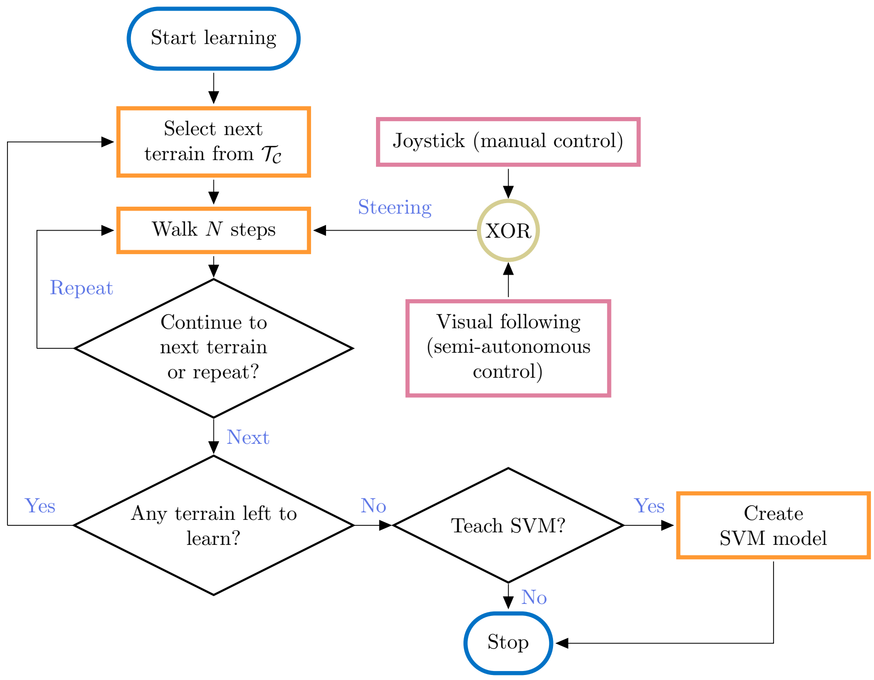


Figure 4.6: Flow diagram of the system learning

Chapter 5
Experimental Results

5.1 Experimental setup

The proposed method has been experimentally verified in a series of indoor and outdoor experiments to provide representative and realistic validations of the developed solution. Moreover, the method has been further evaluated in the competition Robotem rovně 2016 held annually by [28].

The experimental setup has been prepared in a way allowing offline or laboratory evaluation of the proposed methods. First, the particular components of the system have been tested separately. After the functionality has been verified, outdoor experiments of the whole system with fully on-line processing have been carried out. In Section 5.2 and 5.3, results of the tactile and visual road following are presented including drawbacks of each method. Section 5.4 then shows how these drawbacks are compensated when the proposed combination is employed.

All the algorithms have been implemented in C/C++ in Robot Operating System (ROS) [14], version Hydro. Implementation in ROS allows to split the software into smaller functional units and simplify the parametrization while launching the algorithms. On-board processing has been provided by the ARM-based computer Odroid U3 [15] with 2GB of RAM and 1.7GHz Quad-Core processor, which is shown in Fig. 5.1. Camera and communication with servo drives were connected via USB.

Even though the system is able to run onboard, during the experiments the robot has been controlled via USB cable from a laptop computer, to allow simple data logging, synchronization, and system monitoring. The interaction with the robot (start/stop, learning) has been provided by the standard PC joystick.

5.2 Tactile road following

5.2.1 Laboratory experiments

The feasibility of the control strategy proposed in Section 4.2.2 has been verified on the terrains easy to recognize. Simple surface distinguishability

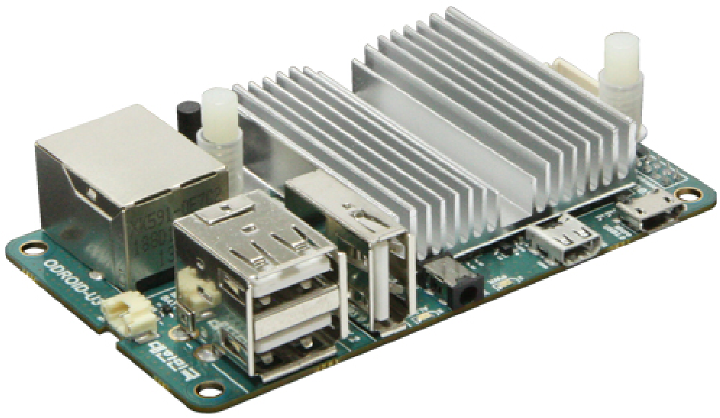


Figure 5.1: The on-board computer Odroid U3, adopted from [15]

reduces the impact of the misclassified samples to the decision process and control action calculation. A high contrast in the surface softness proved to be the good sign for a reliable terrain recognition. These conditions were assured in the laboratory by the wooden board marked as `On` terrain and soft pillows representing the `Off` terrain class. The considered setup is shown in Fig. 5.2.

Prior the evaluation itself, the parameters from Eq. 4.3, 4.4 and 4.6 have been also tuned in the laboratory setup. The used values of the parameters of the control law have been set as follows

- History length $N = 10$,
- Ageing factor $\eta = 2$,
- Proportional control gain $k_p = 0.5$,
- Offroad weight $W_{off} = 1$,
- Border weight $W_b = 0.5$,
- Offroad count $N_{off} = 5$,
- Border count $N_b = 4$.

5.2.2 Outdoor experiments

The outdoor experiments have been performed in urban park at Karlovo náměstí[1]. The asphalt paths are surrounded by a grass or a dirt. According to $\mathcal{T}_\mathcal{C}$, the `Off` class has represented a grass and a dirt, whereas an asphalt has been marked as the `On` class. Examples of the terrain borders considered during the experiments are depicted in Fig. 5.3 with their assignment of surface to $\mathcal{T}_\mathcal{C}$ classes.

[1] GPS coordinates of the park are 50.0767292N, 14.4202022E

Figure 5.2: Experimental setup for the laboratory evaluation. The wooden board and soft pillows were chosen because of their easy distinguishability. During the experiment, the surfaces have been assigned to $\mathcal{T}_C$ in the following way: wood - `On`, wood-pillows - $\texttt{Off}_{\texttt{right}}$, pillows-wood - $\texttt{Off}_{\texttt{left}}$, pillows - `Off`.

The classification model fidelity has been validated first. For the training data gathering, the robot has been controlled manually by the joystick. The model has been then created offline. The accuracy has been evaluated by two-fold the cross-validation, as in [26]. The results are summarized in Tab. 5.1.

The overall accuracy achieved during the validation is 96.2%. However, almost 86% of the misclassified samples have been related to the dirt surface. Regarding the feature vector described in Section 2.2.3, where position errors of the legs are assumed, the terrain classification is sensitive to deviations in leg trajectories. Since the dirt or the borders between a dirt and an asphalt, respectively, were flat as well as an asphalt, such a misclassification has been expected. The other false predictions arise from the similarity of the different road borders. Thus, they belong to the same $\mathcal{T}_C$ class and are acceptable for the road following.

During the experiments, the robot has been supposed to deal with several types of the borders. The most reliable borders to detect have been the ones with the large difference in the surface. A significant transition from an asphalt to a dirt can be seen in Fig. 5.3a. The stones rise the off road to a higher level and the robot can easily sense the deviation in the leg position error used in the classification feature vectors.

Another reliably predicted borders have been between an asphalt and a grass (Fig. 5.3b and 5.3c). Grass is in contrast to asphalt soft and thus, the situation is close to the conditions prepared artificially in the laboratory. Moreover, there is usually a small difference in the elevation.

The last dirt surface proved to be the most difficult to classify. Basically two types of the asphalt-dirt transitions have been included. The first one

(a): Border stone

(b): Flat border with grass

(c): Steep border with grass

(d): Strict border with dirt

(e): Blurred border of the dirt

Figure 5.3: Various terrain borders included in the outdoor experiments. According to the $\mathcal{T}_\mathcal{C}$, surfaces were assigned as follows: asphalt - `On`, asphalt-grass - $\texttt{Off}_{\texttt{right}}$, grass-asphalt - $\texttt{Off}_{\texttt{left}}$, asphalt-dirt - $\texttt{Off}_{\texttt{right}}$, dirt-asphalt - $\texttt{Off}_{\texttt{left}}$, grass and dirt - `Off`

had a strict border with a small elevation difference (Fig. 5.3d), the second one has been flat and blurred (Fig. 5.3e). Because of the lower accuracy on the dirt terrains, it has taken the robot a longer time to return back to the road.

As discussed in Section 4.2.3, the algorithm needs information about the side where border is detected. Therefore, the robot approached additionally to the border under various angles to evaluate the limits of the control strategy. Regarding the length of the step and the fact that the terrain is predicted from the last three gait cycles, a direct transition from `On` to `Off` has been

Table 5.1: Two–fold cross-validation confusion matrix of the sample gait cycles. The overall accuracy reached 96.2%. Class names are shortened in the table in the following way: D - Dirt, A - Asphalt, G - Grass.

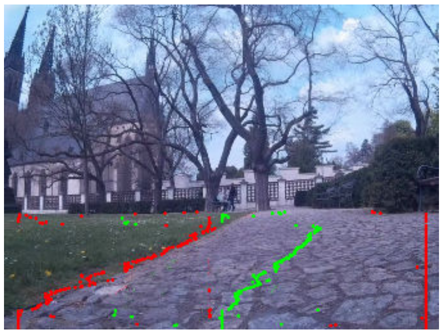
Class	G	D – A	D	A – D	A	G – A	A – G
G	99	0	0	0	0	0	0
D – A	0	116	7	3	0	1	0
D	0	9	62	0	0	0	0
A – D	0	4	1	82	0	0	1
A	0	0	0	0	112	0	0
G – A	0	0	0	0	0	117	0
A – G	0	0	0	2	0	0	122

Figure 5.4: Result of the tactile road following in one of the outdoor trials. Travelled trajectory is highlighted by the red line.

reported for angles larger than approximately 45°. For larger values, it is impossible to return to the road in the desired way described in Section 4.2.

Described trials have been documented in the video published in [36]. Tracked trajectory from one of the trials is highlighted by the red line in Fig. 5.4. The robot has been set to approach the border on the left side. The $\texttt{Off}_{\texttt{left}}$ class has been detected and the control strategy pushed the robot back on the asphalt road. When `On` has been detected again, the maneuvre has been compensated by the counter action and the robot aligned to the road direction.

(a): High constrast between road and grass ensures correct road recognition



(b): Low colour distance causes false positive detection at the half of the image

Figure 5.5: The road is detected correctly only if the colour distance between the road terrain and off road is high enough. Otherwise false positive detection occurs.

5.3 Visual road following

5.3.1 Colour based road recognition

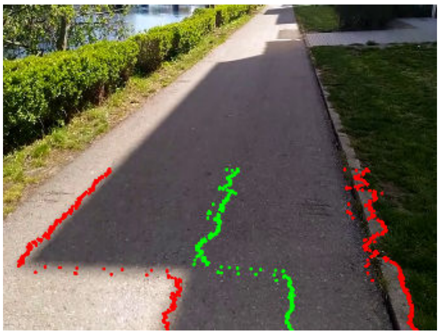
As discussed in Section 4.3.1, the colour based recognition is simple, but not robust enough for outdoor environments. It works well in cases, where the road and surrounding terrain are contrastive and the road boundaries are sharp. Such a case is shown in Fig. 5.5a. However, if the image quality is lower or road and off road are similar, the surrounding terrain may be mistaken for the the road. This is the worst case scenario, since it leads the robot out of the road. An example of the false positive detection is in Fig.5.5b, where the road is reported through whole image width because of the flowers on the grass in the half of the image.

Another weakness of the simple color-based vision road following approach are shadows. In the RGB space, shadows may have absolutely different colour from the surface directly illuminated by the sun. The HSV space reduces partially this effect, but in most cases it is not sufficient. Therefore, the shadows cause false negative detections, that lead to blurred (Fig. 5.6a) or narrowed (Fig. 5.6b) path. There is a possibility, to teach system both colours, in shadow and at the sun. Nonetheless, this may lead to overfitting and in contrary to the initial issue, false positives are then reported. The slight overfitting can be seen in Fig. 5.6c. Therefore, intrinsic image has been considered in such situations.

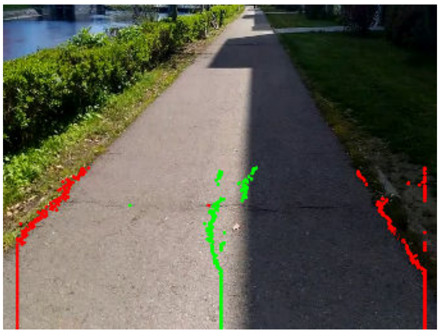
5.3.2 Intrinsic image based road recognition

The intrinsic image recognition in the form presented in Section 4.3.2 requires thresholding of the equalized image. This seems as a fundamental disadvantage of the method. It implies an assumption on a road and surrounding log-

(a): Blurred path due to the tree shadow



(b): Consistent road within the shadow



(c): Overfitted road colour

Figure 5.6: Shadows in the colour based road recognition

chromaticity. Because the threshold is by default a constant, log-chromaticity (mainly of the road) is supposed to vary only a little. Regarding this drawback, approach presented in [1] is more suitable for environments with the wide range of road surfaces, since it technically implements adaptive segmentation. However, for the purposes of this work, the method from Section 4.3.2 has been found sufficient.

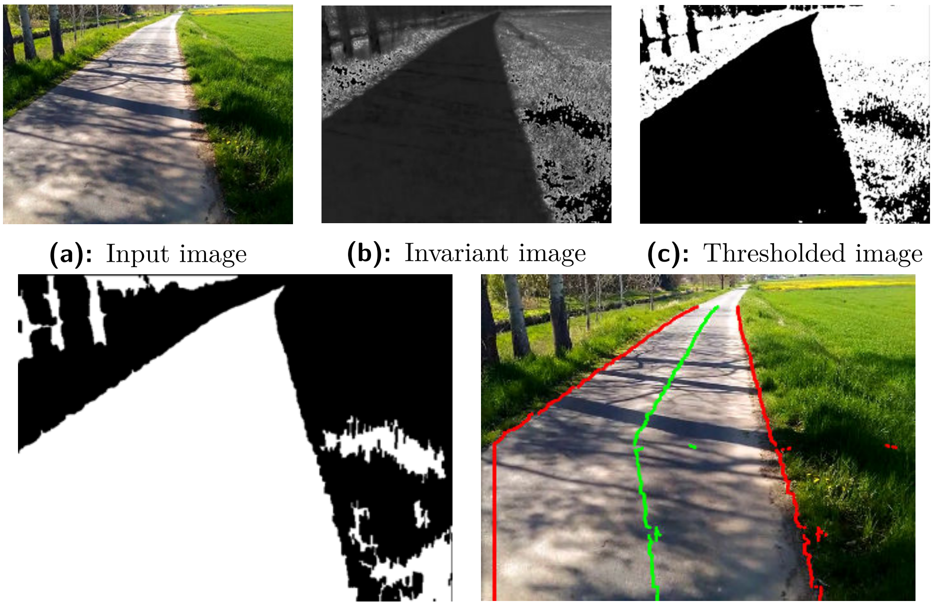
Before each experiment itself, the angle θ has been estimated and the threshold τ set. For the rest of the experiment, these values remained the same. During evaluation, three datasets have been used. The first one has been dataset from [19]. Here, θ has been varying in wide ranges, as the video contains different environmental conditions. It was necessary to tune θ before entering the new area.

The other two datasets have been gathered by the smart-phone camera. The range of the estimated θ for the phone camera varied in the interval $\langle 144°, 169° \rangle$. According to the estimation method, which relies only on the image content, the range has been narrow contrary to [1] allowing good results of the invariant image computation.

Figures 5.7 to 5.10 display the results of invariant image computation followed by the road recognition. Majority of the shadows have been removed successfully. However, with an inaccurate θ produced by the self calibration, strong shadows remain visible in some cases, e.g. see Fig 5.8b. Nevertheless, due to the properly set threshold τ shadows have been neglected in the thresholding phase.

Fig. 5.7 and 5.10 show straight road without crossroad. The recognized path is also straight with the center marked by the green pixels. Fig. 5.8 and 5.9 contain crossroad. Therefore, the path center is not continuous and in the area of the crossroads, it deflects from the straight direction.

The invariant image implementation has been considered successful. Despite the fact that θ has to be tuned before the area with different conditions can be traversed, there has not been a situation encountered where the algorithm recognized the path incorrectly.



(a): Input image **(b):** Invariant image **(c):** Thresholded image

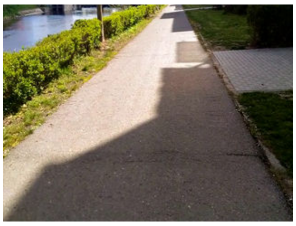
(d): Morphology, inversion **(e):** Recognized path

Figure 5.7: Invariant image road recognition 1 ($\theta = 152°$)

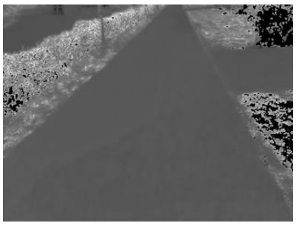
5.3.3 Comparison of the colour based and intrinsic image based road recognition

The performance of the both visual and tactile based road detection algorithms have been compared on dataset with images of the size 640 $\times$ 480 pixels. The evaluation has been performed on-board on the Odroid U3 computer [15] presented in Section 5.1. The number of frames per second has been measured to evaluate the computational requirements of the algorithms.

The results are summarized in Tab. 5.2 and examples of the detected path are shown in Fig. 5.11. As it can be seen, the colour based approach is slightly faster; however, the detected path in Fig. 5.11b is blurred due to the shadow and thus, the computed steering action may be incorrect. On the contrary, the intrinsic image based approach detected the road correctly in Fig. 5.11d at the cost of about 5 FPS slower performance. Since the robot motion is slow, the frame rate 15 FPS of the intrinsic image based method is sufficient and therefore, it is used by default.



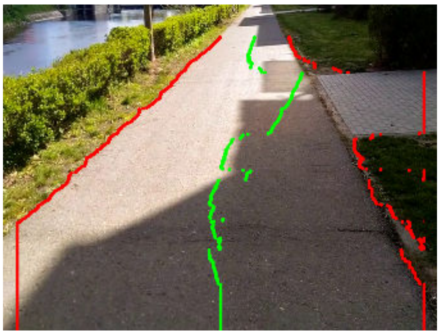
(a): Input image



(b): Invariant image

(c): Thresholded image

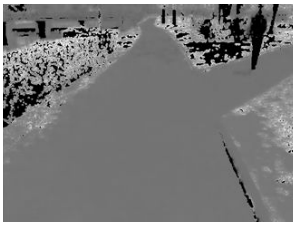
(d): Morphology, inversion



(e): Recognized path

Figure 5.8: Invariant image road recognition 2 ($\theta = 169°$)

(a): Input image



(b): Invariant image

(c): Thresholded image

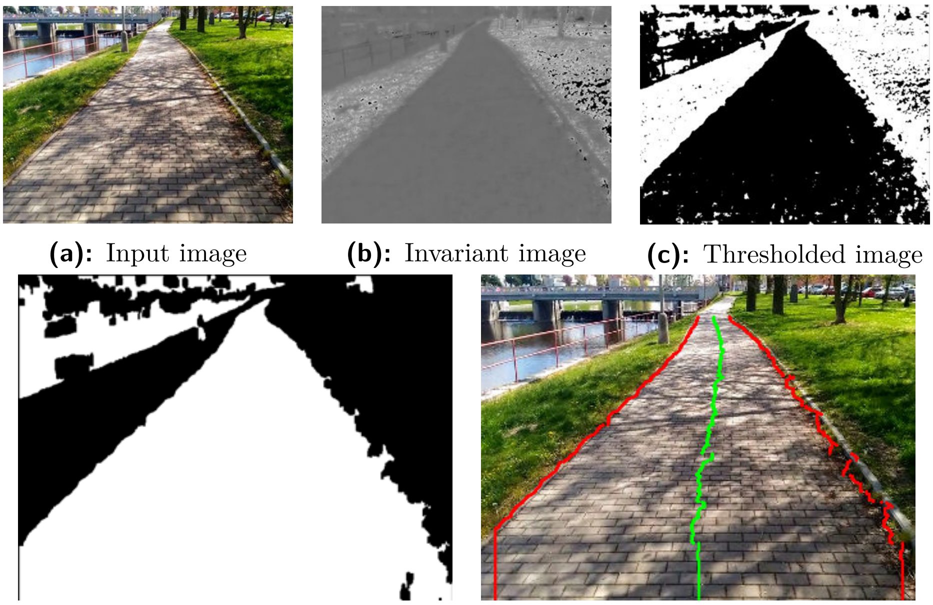
(d): Morphology, inversion

(e): Recognized path

Figure 5.9: Invariant image road recognition 3 ($\theta = 169°$)

Table 5.2: FPS of the utilized vision based road following methods

Method	FPS
Colour based	20
Intrinsic image based	15



(a): Input image **(b):** Invariant image **(c):** Thresholded image

(d): Morphology, inversion **(e):** Recognized path

Figure 5.10: Invariant image road recognition 4 ($\theta = 144°$)

5.4 Fusion of visual and tactile controllers

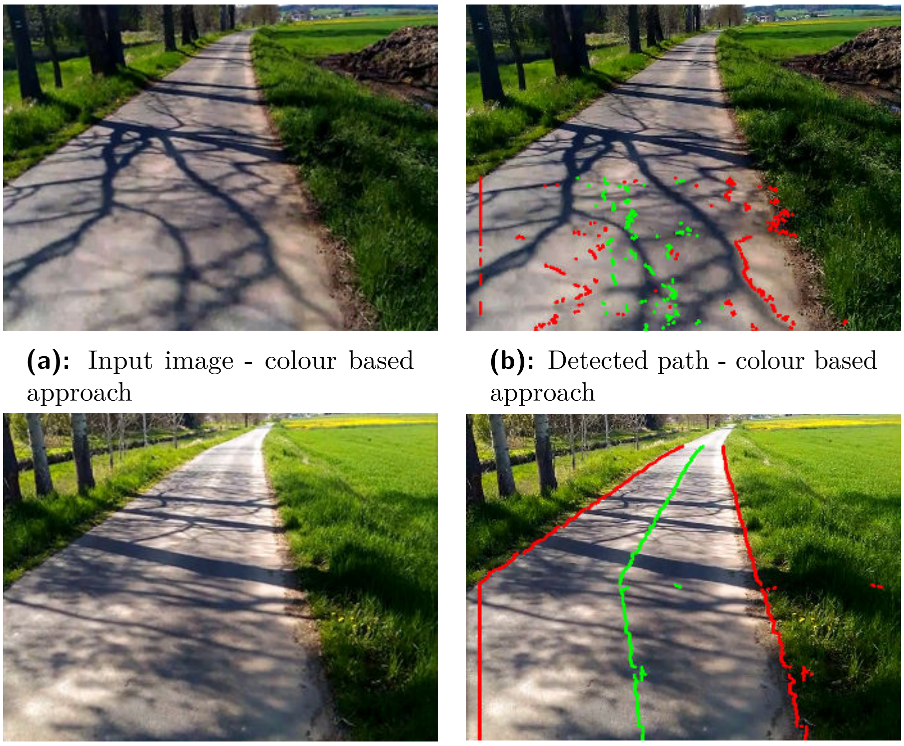
The switching strategies have been firstly evaluated in the laboratory. An artificial road corridor has been created. The laboratory floor has been considered as the `On` terrain, while the partially soft boards from Fig. 4.3 formed the road borders and the `Off` terrain. The first scenario simulated the case where the image quality is high enough, but false positive detection occurs. The tactile following has been expected to take over the steering and lead the robot out back on the road. Using the algorithm from Section 4.3.3, the robot has been intentionally led to the road border. After it entered the `Off` terrain and walked several steps, the tactile following has been activated.

The proposed control strategy successfully steered the robot from the `Off` terrain and when the first `On` terrain has been detected, it has been switched back to the visual road following method. Trajectory travelled during the experiment, predicted terrain and steering action are depicted in Fig. 5.12. After the robot left $\texttt{Off}_{\texttt{right}}$ terrain, two misclassified samples occurred. However, the switching criteria filtered them out and the visual following kept the control. The video of this experiment is available at [32].

After the prior laboratory evaluation, the main experiments have been performed outdoors. As a testing environment, an urban park with asphalt and stone roads surrounded by grass has been chosen[2]. From several trials, two have been selected to be presented in the following paragraphs.

The first run is shown in Fig. 5.13, the video of the whole experiment is available at [33]. The predicted terrain is highlighted in the robot's trajectory

[2] Vyšehrad park, 50.0657458N, 14.4174986E



(a): Input image - colour based approach

(b): Detected path - colour based approach

(c): Input image - intrinsic image based approach

(d): Detected path - intrinsic image based approach

Figure 5.11: Comparison of the road detection results for colour based and intrinsic image based methods.

in Fig. 5.13a. The control strategy currently used, the image quality, steering action applied to the motion controller and predicted terrain are charted in Fig. 5.13b.

The robot started on the left side of the road. At the beginning, it has been controlled by the visual following approximately straight. Before it gathered enough data to reliably predict the terrain, the `On` terrain has been considered as an initial terrain class. At the time of roughly 40 s, the first terrain prediction has been reported; however, it has been misclassified. But since the following step, the predictions have been correctly reported as the `On` terrain.

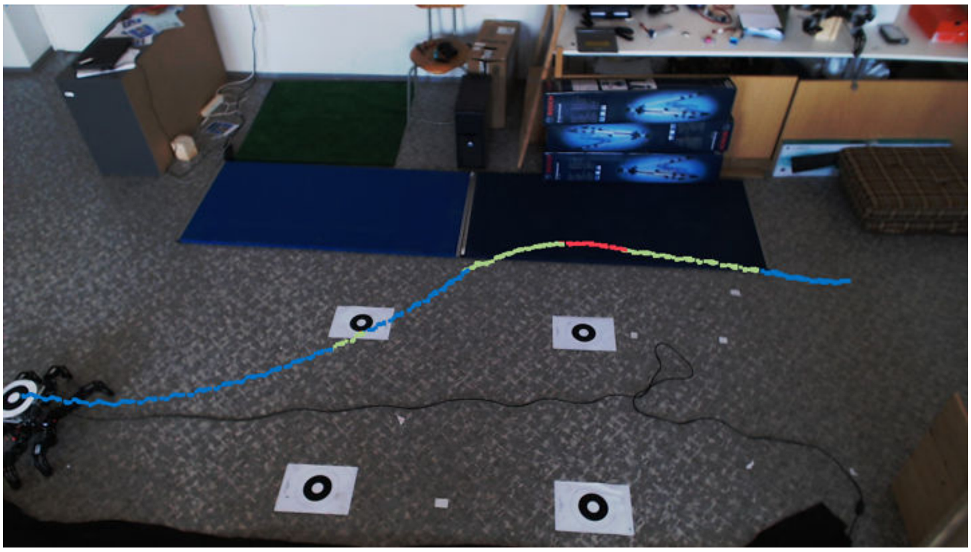
At the time 100 s, the camera has been covered by the black lid. Thus, the image quality dropped to 0 and tactile following has been activated immediately. Then, the robot has been turned towards the grass by hand to provide a disturbance and lead the robot towards the border of the road. In few steps, the road border has been encountered. The steering action has been calculated so it leads the robot from $\mathtt{Off}_{\mathtt{left}}$. Note the altering straight and turning steps in steering action in Fig. 5.13b in order to collect data for the terrain classification. Because the applied action overflows the threshold, altering is necessary to correctly predict terrains.

When the robot has been heading back to the road, the camera lid has been removed (approximately at the time 235 s). It can be seen in Fig. 5.13b that visual following has been not activated at the moment the image quality raised, but rather until the `On` terrain has been reported. The lower image quality in the second visual control part has been caused by the dark bush in front of the robot.

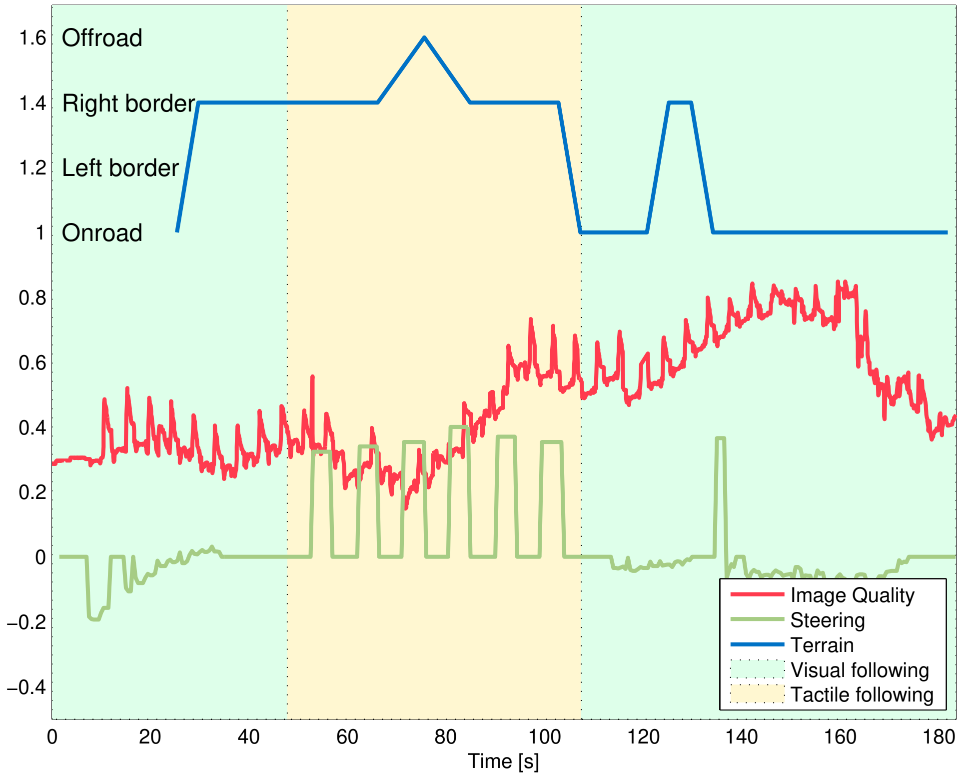
Before the experiment has been terminated, one misclassified terrain sample occurred. Due to the terrain classification criteria proposed in Section 4.4.2, it has been filtered and had no influence on the control.

The second trial followed the scenario of the first one. The tracked position is in Fig. 5.14a, the internal system information in Fig. 5.14b, and recorded experiment is available at [34]. At the beginning, the visual control applied large actions, therefore, again, altering between straight and turning steps has been necessary.

The $\texttt{Off}_{\texttt{left}}$ terrain has been detected earlier before the robot actually traversed it. It has been caused by the small declination at the road border, which has been due to the terrain classification approach mistaken with the real border. At the time 110 s, the `On` terrain has been predicted. Since the image quality has been still low, tactile following kept the control. Finally at the time 120 s, the visual following has been activated. However, the robot has been not able to return completely to the road direction, because the bush and the bench formed unfortunate shape of the road boundary. Since the main camera is low, it is impossible for the robot to get the wide perspective of the road and handle such cases. This issue can be solved by rising the camera higher above the ground.

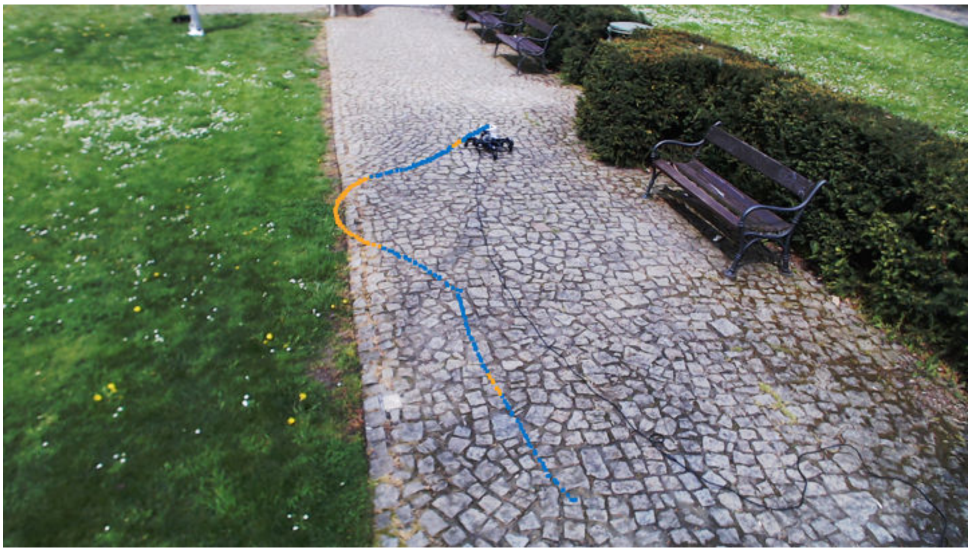


(a): Tracked position. Blue markers denote `On` terrain, the green denote $\texttt{Off}_{\texttt{right}}$ and `Off` is red.

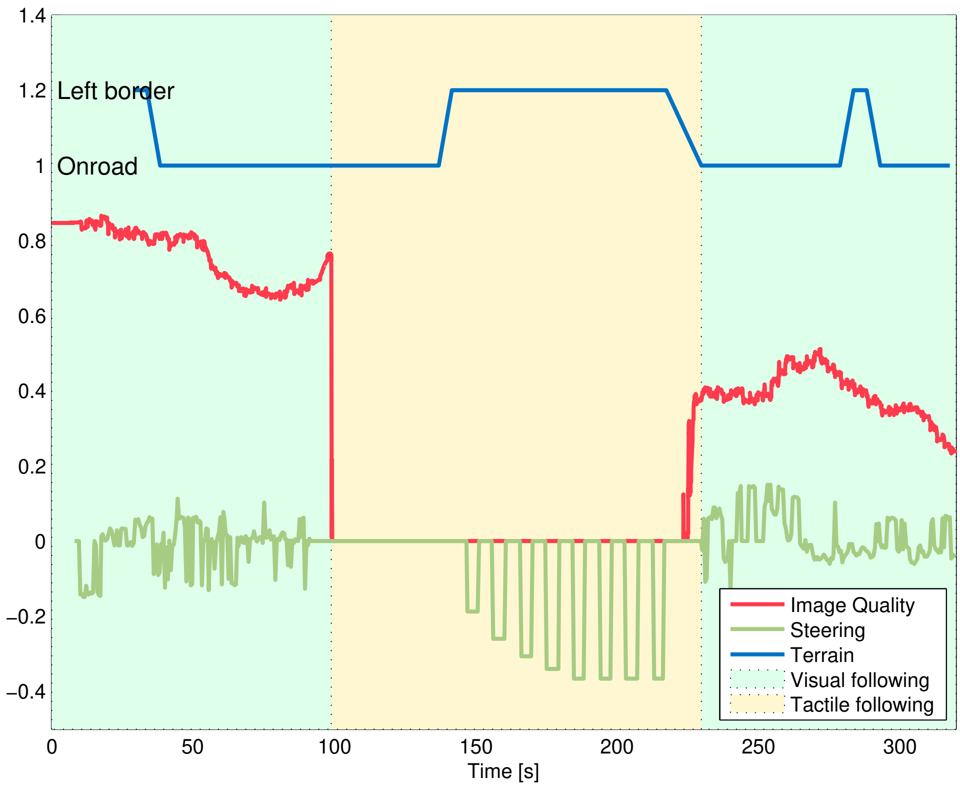


(b): Predicted terrain, image quality, and applied steering action. The colour of the background fill determines the current control strategy - visual (green) or tactile (yellow).

Figure 5.12: Trajectory of the robot with highlighted predicted terrain classes and decision process diagram, laboratory experiment



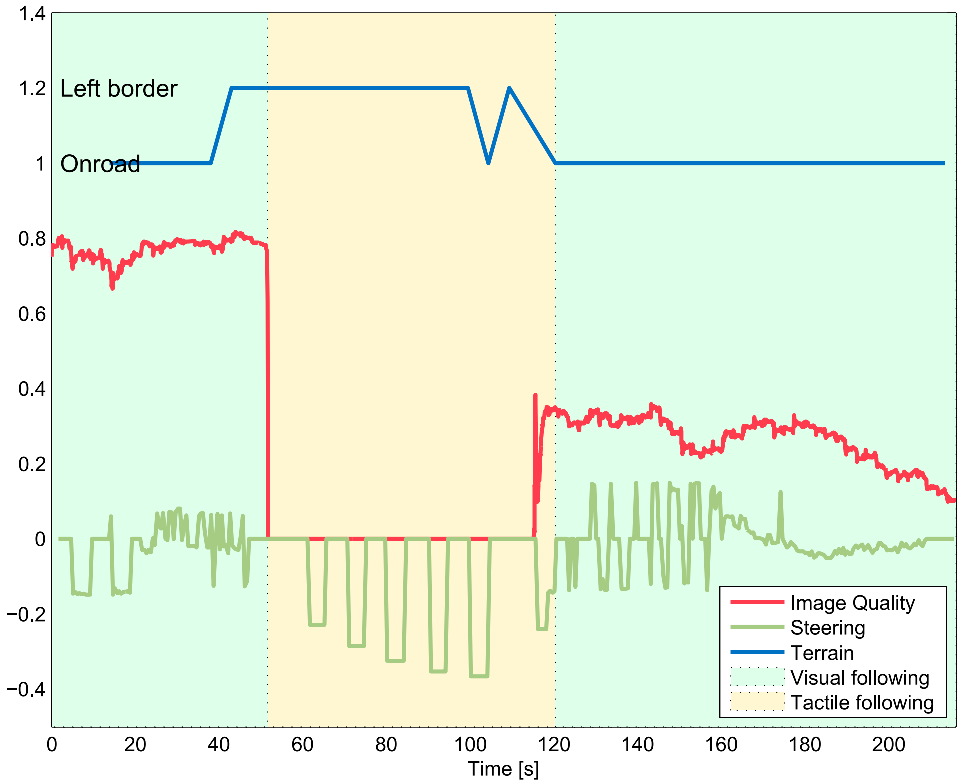
(a): Tracked position. Blue markers denote `On` terrain, while the orange denote $\texttt{Off}_{\texttt{left}}$



(b): Predicted terrain, image quality, and applied steering action. The colour of the background fill determines the current control strategy - visual (green) or tactile (yellow).

Figure 5.13: Trajectory of the robot with the highlighted predicted terrain classes and decision process diagram, run 1

(a): Tracked position. Blue markers denote `On` terrain, while the orange denote `Off`$_{\mathtt{left}}$



(b): Predicted terrain, image quality, and applied steering action. The colour of the background fill determines the current control strategy - visual (green) or tactile (yellow).

Figure 5.14: Trajectory of the robot with the highlighted predicted terrain classes and decision process diagram, run 2

Chapter 6
Conclusion

The mobile robots are desired to reach the highest level of autonomy possible. One of the most common tasks is the road following. In this work, a robust road following method is proposed to address the problem of terrain classification in road following. The approach is based on the visual road recognition supported by the proprioceptive information. The main requirements on the solution are to keep the robot on the road even in the case the visual following fails, e.g. due to light conditions or false positive road detection. Both situations can cause the robot off the road. These drawbacks are compensated by the terrain classification based on the tactile information from the robot actuators.

The work is concerned with two fundamental problems: visual road following and tactile terrain classification. Both problems are solved separately first. Then, they have been combined in the proposed switching strategy to achieve the desired behaviour.

The terrain classification is discussed generally for various types of mobile robots. Then, the specific solution for the hexapod platform is presented. It exploits the proprioceptive information provided by the intelligent servo drives. From the actual and the required position a feature vector is created at several phases of the gait cycle. The vectors are then trained by the SVM classifier.

Based on the terrain classification, the tactile road following has been proposed. For this problem, the robot is considered technically blind. The tactile strategy uses history of the classified terrain classes to compute the steering action. However, the terrain can be classified after it is crawled. Thus, the robot is allowed to crawl partially off the road to gather necessary data to make a decision. Note that the tactile road following method has been published at ICRA 2016 [35].

The second addressed problem is concerned with the visual road following. Several approaches are discussed and two of them are chosen to be employed in this work. The simple colour based method recognizes road pixels using the colour distance of the current pixel and pixels in the set of learnt colours. Such a set is defined by a human supervisor prior to the robot traversing. However, this approach suffers from many drawbacks. Therefore, more robust method has been considered. It is based on the intrinsic image computation.

The input colour image is projected into the log-chromaticity space, where the invariant image is computed. The invariant image does not contain shadows, which are the main source of issues in the simple colour based recognition. Then, using the thresholding, the path is found in the invariant image.

Finally, the tactile and visual controller are combined to create a robust road following strategy. The robot is controlled by exactly one controller at the time and the controllers are switched whenever it is necessary. Two criteria are proposed for switching between the proposed vision and tactile based methods of road following. The first criterion evaluates the image quality in the sense of brightness and contrast. The second criterion measures the reliability of the terrain classification.

The proposed approach has been experimentally evaluated in the laboratory and outdoor environments. Both experiments support feasibility of the proposed solution. The main considered contribution of the work is in the novel fusion of the visual and tactile road following. Processing of both, visual and tactile information, allows to compensate drawbacks of the particular approaches and thus, increase the overall system robustness.

Bibliography

[1] J. M. Á Alvarez and A. M. Lopez, *Road detection based on illuminant invariance*, IEEE Transactions on Intelligent Transportation Systems **12** (2011), no. 1, 184–193.

[2] A. Angelova, L. Matthies, D. Helmick, and P. Perona, *Fast terrain classification using variable-length representation for autonomous navigation*, IEEE Conference on Computer Visionand Pattern Recognition (CVPR), 2007, pp. 1–8.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, *Surf: Speeded up robust features*, In Proceedings: Computer Vision – ECCV 2006: 9th European Conference on Computer Vision (2006), 404–417.

[4] G. Best, P. Moghadam, N. Kottege, and L. Kleeman, *Terrain classification using a hexapod robot*, In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA), 2013.

[5] M. Rufus Blas, M. Agrawal, A. Sundaresan, and K. Konolige, *Fast color/texture segmentation for outdoor robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 4078–4085.

[6] A. Chemori and A. Loria, *Control of a planar underactuated biped on a complete walking cycle*, IEEE Transactions on Automatic Control **49** (2004), no. 5, 838–843.

[7] E. G. Collins and E. J. Coyle, *Vibration-based terrain classification using surface profile input frequency responses*, IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 3276–3283.

[8] E. Coyle, E. G. Collins Jr., and R. G. Roberts, *Speed independent terrain classification using singular value decomposition interpolation*, IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 4014–4019.

[9] P. Cristoforis, M. Nitsche, T. Krajnik, and M. Mejail, *Realtime monocular image-based path detection*, Journal of Realtime Image Processing (2013), 1–14.

[10] G. Dudek and M. Jenkin, *Computational principles of mobile robotics, 2nd edition*, Cambridge University Press, 2010.

[11] P. Filitchkin and K. Byl, *Feature-based terrain classification for littledog*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 1387–1392.

[12] G. D. Finlayson, S. D. Hordley, Cheng Lu, and M. S. Drew, *On the removal of shadows from images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 1, 59–68.

[13] G. D. Finlayson, Cheng Lu, and M. S. Drew, *Entropy minimization for shadow removal*, International Journal of Computer Vision **85** (2009), 35–57.

[14] Open Source Robotics Foundation, *ROS Homepage*, 2016, [Online], `http://www.ros.org`, Last visited 2016-05-13.

[15] Hardkernel, *Odroid U3 Product Page*, 2016, [Online], `http://www.hardkernel.com/main/products/prdt_info.php?g_code=g138745696275`, Last visited 2016-05-13.

[16] M. A. Hoepflinger, C. D. Remy, M. Hutter, L. Spinello, and R. Siegwart, *Haptic terrain classification for legged robots*, IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 2828–2833.

[17] B. Julesz and J. R. Bergen, *Textons, the fundamental elements in preattentive vision and perception of textures*, The Bell System Technical Journal **62** (1983), 1619–1645.

[18] N. Kottege, C. Parkinson, P. Moghadam, A. Elfes, and S. P. N. Singh, *Energetics-informed hexapod gait transitions across terrains*, In proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 5140–5147.

[19] T. Krajnik, J. Blazicek, and J. Santos, *Visual road following using intrinsic images*, European Conference on Mobile Robots (ECMR), 2015, pp. 1–6.

[20] T. Krajník, H. Szücsová, M. Svědirohová, J. Chudoba, and V. Vonásek, *Jak to vidí roboti v*, Robot revue: Magazín ze světa robotiky (2010), 32–35.

[21] L. Lu, C. Ordonez, E. G. Collins, and E. M. DuPont, *Terrain surface classification for autonomous ground vehicles using a 2d laser stripe-based structured light sensor*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009, pp. 2174–2181.

[22] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, *Detection and classification of highway lanes using vehicle motion trajectories*, IEEE Transactions on Intelligent Transportation Systems **7** (2006), no. 2, 188–200.

[23] V. H. Mistry and R. Makwana, *Surevey: Vision based road detection techniques*, International Journal of Computer Science and Information Technologies **5** (2014), no. 3, 4741–4747.

[24] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, *Fast vanishing-point detection in unstructured environments*, vol. 21, June 2012, pp. 425–430.

[25] J. Mrva, *Tactile sensing with servo drives feedback only for blind hexapod walking robot*, Master's thesis, ČVUT v Praze, Fakulta elektrotechnická, 2014.

[26] J. Mrva and J. Faigl, *Feature extraction for terrain classification with crawling robots*, Informačné Technológie - Aplikácie a Teória (ITAT), 2015, pp. 179–185.

[27] ———, *Tactile sensing with servo drives feedback only for blind hexapod walking robot*, In: Proceedings of the 10th International Workshop on Robot Motion and Control (RoMoCo), 2015, pp. 240–245.

[28] Radioklub Písek, *Radioklib OK1KPI*, 2016, [Online], `http://www.kufr.cz`, Last visited 2016-05-13.

[29] Trossen Robotics, *Dynamixel AX-12A Robot Actuator*, [Online], `http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx`, Last visited 2016-05-13.

[30] ———, *PhantomX AX Hexapod Mark II Kit*, [Online], `http://www.trossenrobotics.com/hex-mk2`, Last visited 2016-05-13.

[31] IEEE Spectrum, *Dismantling Fukushima: The World's Toughest Demolition Project*, 2014, [Online], `http://spectrum.ieee.org/energy/nuclear/dismantling-fukushima-the-worlds-toughest-demolition-project`, Last visited 2016-05-13.

[32] M. Stejskal, *Road following for the hexapod walking robot - office*, 2016, [Online], `https://youtu.be/BMuhzOHI_uk`, Last visited 2016-05-23.

[33] ———, *Road following for the hexapod walking robot - vysehrad run 1*, 2016, [Online], `https://youtu.be/qnBukObE5J4`, Last visited 2016-05-23.

[34] ———, *Road following for the hexapod walking robot - vysehrad run 2*, 2016, [Online], `https://youtu.be/hYb5UcNVWfY`, Last visited 2016-05-23.

[35] M. Stejskal, J. Mrva, and J. Faigl, *Road following with blind crawling robot*, In proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 3612–3617.

[36] ———, *Road following with blind crawling robot - demonstration video*, 2016, [Online], `https://youtu.be/3LGbtGN4zDI`, Last visited 2016-05-22.

[37] B. D. Stewart, I. Reading, M. S. Thomson, T. D. Binnie, K. W. Dickinson, and C. L. Wan, *Adaptive lane finding in road traffic image analysis*, Seventh International Conference on Road Traffic Monitoring and Control, 1994, pp. 133–136.

[38] A. Vedaldi and S. Soatto, *In proceedingd: Computer vision – eccv 2008: 10th european conference on computer vision*, ch. Quick Shift and Kernel Methods for Mode Seeking, pp. 705–718, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[39] K. Walas, *Tactile sensing for ground classification*, Journal of Automation, Mobile Robotics and Inteligent Systems **7** (2013), 18–23.

[40] K. Walas and M. Nowicki, *Terrain classification using laser range finder*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014, pp. 5003–5009.

[41] C. Weiss, H. Frohlich, and A. Zell, *Vibration-based terrain classification using support vector machines*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2006, pp. 4429–4434.

[42] C. Weiss, H. Tamimi, and A. Zell, *A combination of vision- and vibration-based terrain classification*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2008, pp. 2204–2209.

[43] B. Yu and A. K. Jain, *Lane boundary detection using a multiresolution hough transform*, In Proceedings: International Conference on Image Processing, 1997, pp. 748–751 vol.2.

[44] S. Zenker, E. E. Aksoy, D. Goldschmidt, F. Wörgötter, and P. Manoonpong, *Visual terrain classification for selecting energy efficient gaits of a hexapod robot*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2013, pp. 577–584.

[45] Jinyou Zhang and H. H. Nagel, *Texture-based segmentation of road images*, In Proceedings: Intelligent Vehicles Symposium, 1994, pp. 260–265.

Appendix A
Contents of attached CD

Attached CD contains electronic version of this work and all source codes necessary to run presented approach. The codes are in form of ROS nodes, thus to run them it is sufficient to copy them to arbitrary ROS workspace and compile. Moreover, video datasets used for visual road recognition evaluation are available. Also examples of the raw training data for terrain classification and trained SVM model are included. To demonstrate method feasibility CD contains videos recorded during the experiments.

Table A.1: Structure of the CD root

Folder, file	Description
`stejsma6_dp.pdf`	Electronic version of the work
`datasets`	Contains subdirectories with datasets for visual road recognition and terrain classification
`datasets/video`	Video datasets for visual road recognition
`datasets/tc`	Examples of training data and trained SVM model for terrain classification from office and outdoor experiments
`video`	Demo videos from experiments
`src`	Source codes of the ROS nodes