CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

Faculty of Electrical Engineering

Department of Cybernetics

Master’s thesis

Terrain Classification and Traversability Assessment from Exteroceptive Data

Prágr Miloš

May 2018

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

Supervisor specialist: Ing. Petr Čížek

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Declaration

I declare that the 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 thesis.

V Praze dne 25. 5. 2018

Acknowledgement

I would like to thank my supervisors, doc. Ing. Jan Faigl, Ph.D., and Ing. Petr Čížek for their support and feedback during creation of this work and their continued support during my work in Computational Robotics laboratory.

Abstrakt

Kráčející roboty mohou být v současné době považovány za rostoucí segment autonomní mobilní robotiky se záběrem od industriální inspekce po dlouhodobé operování v nehostinném prostředí. Hlavní výhodou těchto robotů je jejich schopnost operovat v obtížných terénech, které mohou být neprostupné pro kolová a pásová vozidla. Aby roboty efektivně operovaly v takových podmínkách, musejí dokázat odhadnout nárožnost zdolání různých terénů a například zvolit takový terén, který je pro robot eneregický výhodný. Energetickou náročnost lze ohodnotit Cost of Transport (CoT) metrikou průchodnosti prostředí, která je založená na energetické spotřebě a rychlosti robotu. V této práci navrhujeme teoretický rámec pro řešení problému průchodnosti terénu, který odhaduje CoT metriku na základě dat z proprioceptivních a exteroceptivních senzorů. V práci používáme jednoduché vizuální a geometrické deskriptory postavené nad daty z RGB-D kamery. Navržené řešení je otestováno v několika scénářích, včetně scénářů se změnou úhlu pohledu a scénářů obsahujících předem neznámé terény.

Klíčová slova: kráčející robot; průchodnost terénu; Cost of Transport; CoT

Abstract

Multi-legged robots are being increasingly deployed in autonomous missions. Such missions can range from autonomous plant inspection to long-term operation in extraterrestrial environments. The main advantage of multi-legged robots is their ability to operate in hard to traverse terrains that can be impassable by wheeled or tracked vehicles. The multi-legged robots must be able to assess the terrain difficulty regarding the robot traversability capabilities based on the necessary effort to traverse the terrain. Such an assessment can be based on the Cost of Transport (CoT) that is a traversability measure computed from robot's power consumption and speed in traversing the particular terrain. In this thesis, we propose a terrain traversability regression framework to estimate CoT using information from the robot's proprioceptive and exteroceptive sensors. Namely, we utilize lightweight visual and geometric features computed from RGB-D image. We test the proposed traversability regression framework in several scenarios, including scenarios with viewpoint changes and tests using observed, but untraversed terrains.

Keywords: multi-legged robot; terrain traversability; Cost of Transport; CoT

Contents

List of Figures

List of Tables

Used Symbols

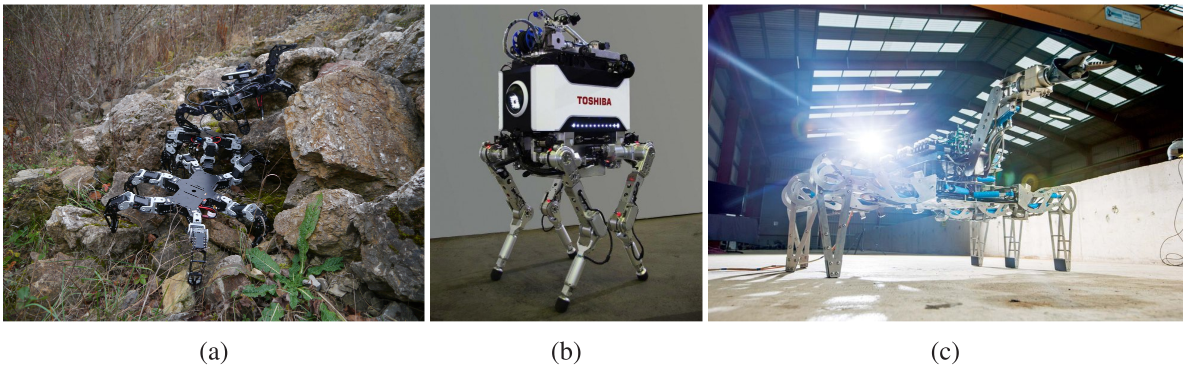
Symbol	[Unit]	Reference
c_{b}	$[-]$	Blue color channel in p_{rc} or p_{wc}
c_{g}	$[-]$	Green color channel in p_{rc} or p_{wc}
c_{r}	$[-]$	Red color channel in p_{rc} or p_{wc}
$c_{\mathrm{x}}^{\mathrm{d}}$	$[\mathrm{px}]$	x offset in K^{d}
$c_{\mathrm{y}}^{\mathrm{d}}$	$[\mathrm{px}]$	y offset in K^{d}
CoT	$[-]$	Power consumption Cost of Transport; main CoT definition in this work
$\mathrm{CoT}_{\mathrm{heat}}$	$\left[\mathrm{ms}^{-2}\right]$	Cost of Transport with heat loss
$\mathrm{CoT}_{\mathrm{mech}}$	$\left[\mathrm{kgms}^{-2}\right]$	Mechanical Cost of Transport
$\mathrm{CoT}_{\mathrm{original}}$	$\left[\mathrm{kgms}^{-2}\right]$	Original definition of Cost of Transport
$\mathrm{CoT}_{\mathrm{power}}$	$[-]$	Power consumption Cost of Transport
d	$[-]$	Danger Level
d_{img}	$[\mathrm{m}]$	Depth value in a depth image
E	$[\mathrm{J}] = \left[\mathrm{kgm}^2\mathrm{s}^{-2}\right]$	Energy expenditure, used in
$f_{\mathrm{x}}^{\mathrm{d}}$	$[\mathrm{px}]$	x pixel focal length in K^{d}
$f_{\mathrm{y}}^{\mathrm{d}}$	$[\mathrm{px}]$	y pixel focal length in K^{d}
g	$\left[\mathrm{ms}^{-2}\right]$	Gravitational acceleration
H_i	$[\mathrm{J}] = \left[\mathrm{kgm}^2\mathrm{s}^{-2}\right]$	Heat lost when moving leg i
H_{pos}		Robot position based homogeneous transformation
I_{in}	$[\mathrm{A}]$	Instantaneous current
k	$[-]$	Number of nearest points considered
K^{d}		calibration matrix of the depth sensor
l	$[\mathrm{m}]$	Stride length
L_1	$[-]$	L_1-norm absolute value distance
m	$[\mathrm{kg}]$	Robot mass
n	$[-]$	Number of robot's legs
P_{i}	$[\mathrm{W}] = \left[\mathrm{kgm}^2\mathrm{s}^{-3}\right]$	Power input
P_{in}	$[\mathrm{W}] = \left[\mathrm{kgm}^2\mathrm{s}^{-3}\right]$	Instantaneous power consumption
p_{r}	$[\mathrm{m} \times \mathrm{m} \times \mathrm{m}]$	Coordinate of reconstructed point
p_{rc}	$[\mathrm{m} \times \mathrm{m} \times \mathrm{m} \times \text{-} \times \text{-} \times \text{-}]$	Coordinate and color information of reconstructed point
p_{wc}	$[\mathrm{m} \times \mathrm{m} \times \mathrm{m} \times \text{-} \times \text{-} \times \text{-}]$	Coordinate and color information of world frame point
r	$[\mathrm{m}]$	Radius of a neighborhood of a given point
t	$[\mathrm{s}]$	Stride period
v	$\left[\mathrm{ms}^{-1}\right]$	Robot velocity
V	$[\mathrm{V}]$	Battery voltage
W_i	$[\mathrm{J}] = \left[\mathrm{kgm}^2\mathrm{s}^{-2}\right]$	Work done when moving leg i
x_{img}	$[\mathrm{px}]$	Column coordinate in a depth image
x_{r}	$[\mathrm{m}]$	x coordinate of reconstructed point p_{r} or p_{rc}
x_{w}	$[\mathrm{m}]$	x coordinate of world coordinate frame point p_{wc}
y_{img}	$[\mathrm{px}]$	Row coordinate in a a depth image
y_{r}	$[\mathrm{m}]$	y coordinate of reconstructed point p_{r} or p_{rc}
y_{w}	$[\mathrm{m}]$	y coordinate of world coordinate frame point p_{wc}
z_{r}	$[\mathrm{m}]$	z coordinate of reconstructed point p_{r} or p_{rc}
z_{w}	$[\mathrm{m}]$	z coordinate of world coordinate frame point p_{wc}
λ	$[-]$	scaling constant
λ_i	$[-]$	Eigenvalue i

Used Abbreviations

Abbreviation	Reference
CoT	Cost of Transport
Db2 wavelet	Daubechies 2-tap wavelet
DWT	Discrete wavelet transform
FOV	Field of view
F-IGMN	Fast Incremental Gaussian Mixture Network
GLCM	Gray level co-occurrence matrix
GT	Ground truth
HSV color space	Hue, saturation, value color space
IGMN	Incremental Gaussian Mixture Network model
L_1	L_1-norm absolute value distance
Lab Color Space	Lightness, green-red, blue-yellow color space
LADAR	see LIDAR
LIDAR	Light Detection and Ranging
NDVI	Normalized difference vegetation index
PCA	Principal component analysis
PLS	Partial least squares
RBF kernel	Radial basis function kernel
RGB Color Space	Red, green, blue color space
RGB-D	Red, grenn, blue + depth
SLAM	Simultaneous location and mapping
SLIC	Simple Linear Iterative Clustering
SVM	Support Vector Machine
SVR	Support Vector Regression
UAV	Unmanned aerial vehicle
YCbCr Color Space	Luma (brightness), chroma (blue/green) chroma (red/green) color colorspace

Chapter 1
Introduction

Nowadays, autonomous robots are being deployed in various missions, including long-term data collection in unknown environments [1] or inspection of areas hostile for humans [2]. Numerous types of autonomous robots exist, ranging from ground robots through marine robots to unmanned aerial vehicles (UAVs). Ground robots utilize several types of locomotion, including walking, rolling, or crawling, using components such as wheels, tracks, and legs. They also possess certain advantages over UAVs. For example, ground robots can carry higher loads than UAVs, which directly influences maximal battery/fuel capacity and therefore mission effectiveness. Besides, UAV mounted cameras cannot be used in dusty areas, as UAV propellers cause a swirl effect and are thus detrimental to visibility.



(a) (b) (c)

Figure 1: (a) Hexapod robots in rough terrain, FEL CTU; (b, c) Deployment in power plant and waste pool inspection - courtesy of [2] and [3], respectively.

However, some terrain types are almost impassable by a wheeled or tracked robot. Fortunately, such terrains can often be traversed by a legged robot with little to no difficulty. Figure 1a presents an example of hexapod crawling robots in rough terrain. The main advantage of a legged robot is its ability to choose and change its gait. By doing so, the robot is capable of traversing rough terrains, where the robot can choose more stable gait or even decides to select footholds for its legs individually [4]. The advantage is most profound for multi-legged robots, which can move individual legs while maintaining static stability by standing on at least one leg. For example, a hexapod robot is stable when using the tripod gait, i.e., when moving three of its legs while the other three are stable, see Figure 2. Moreover, a hexapod robot can further enhance its stability by using the pentapod gait, i.e., gait with five stable and one moving leg. Thus multi-legged robots appear to be the best agents for long-term autonomous operation in rough terrain.

Multi-legged robots can be deployed in various applications ranging from a single large crawler to swarms of smaller spiderbots. Walking robots are often deployed in environments that are not suitable for human workers, but are too cluttered for regular wheeled robots, e.g., in contaminated industrial areas. Figure 1b shows a four-legged robot designed for inspection of the contaminated Fukushima nuclear plant. An underwater walking hexapod robot presented in Figure 1c is being used for cleaning of nuclear waste ponds. Alternatively, walking robots are deployed in areas that are unreachable or unsuitable for the long-term presence of a human crew. A large crab-like underwater robot [5]

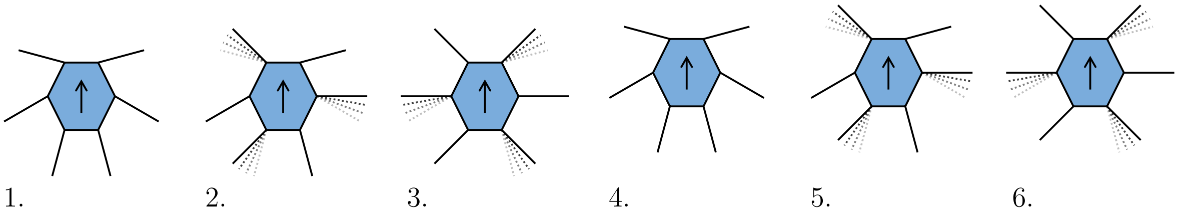
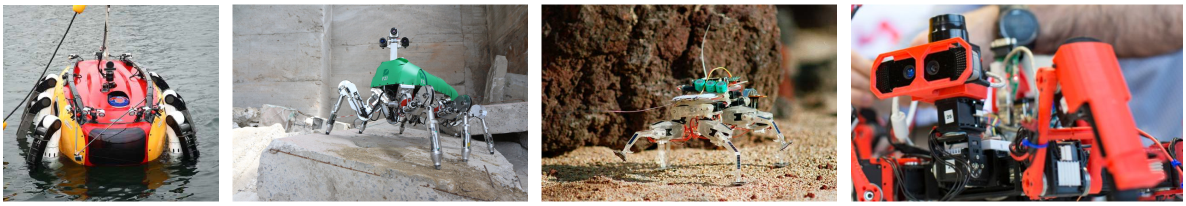


Figure 2: Tripod gait used by hexa-pedal entities.

in Figure 3a has been successfully tested. A versatile search-and-rescue hexapod robot [6] is shown in Figure 3b. Moreover, some locations in need of inspection may simply be hard to access for a full-sized human, and a small robot [7] such as the one presented in Figure 3c has to be utilized. Finally, small robots can be utilized in swarm-like configurations. Siemens is developing a swarm-like system of 3D-printing small octapod robots [8], see Figure 3d.



(a) (b) (c) (d)

Figure 3: (a) CR-200 seabed robot [5]; (b) LAURON V search-and-rescue robot [6]; (c) JPL spiderbot prototype [7]; (d) Siemens spiderbot prototype [8].

The ability to navigate effectively in the environment greatly influences the mission effectiveness for all types of ground robots. Although multi-legged robots are capable of traversing a wide range of terrains, some terrains are much more difficult to traverse than others. Therefore, multi-legged robots need to be concerned with efficient locomotion control over terrains of particular types. To do so, it is needed to differentiate between terrains that are easy and hard to traverse. This can be further utilized either for trajectory planning or gait selection, and thus a faster and safer navigation.

Various traversability metrics have been introduced in the literature to describe the ability of a given robot to traverse a particular type of the terrain. Such measures include a binary division between passable and impassable terrain [9], and more complicated proprioceptive, geometric, and appearance based measures [10]. Since multi-legged robots are capable of traversing most types of terrain, it is advantageous to measure the terrain traversability as a continuous variable, e.g., as a function of locally observed properties [11] or as an energy-based score such as the Cost of Transport (CoT) described in [12; 13], which represents a measure of the effectiveness of the robot locomotion. Note that the CoT-like approaches are inherently influenced by many factors, such as terra-mechanical properties of the terrain, construction of the robot, or by seasonal and weather condition changes in long-term missions.

If the knowledge of the terrain traversability is utilized for planning, it is necessary to have the ability to predict the metric before a given terrain is traversed by the robot, which is, e.g., not the case of the CoT defined in [12]. While some terrain traversability approaches are based on exteroceptive terrain observation, some are proprioceptive. By definition, proprioceptive data are localized to robot's position. Hence, such a terrain traversability measure cannot be known for yet untraversed terrain. Estimation based on exteroceptive data is therefore needed. However, multi-legged robots often observe terrain from viewpoints that are relatively close to the ground, thus being limited in their ability

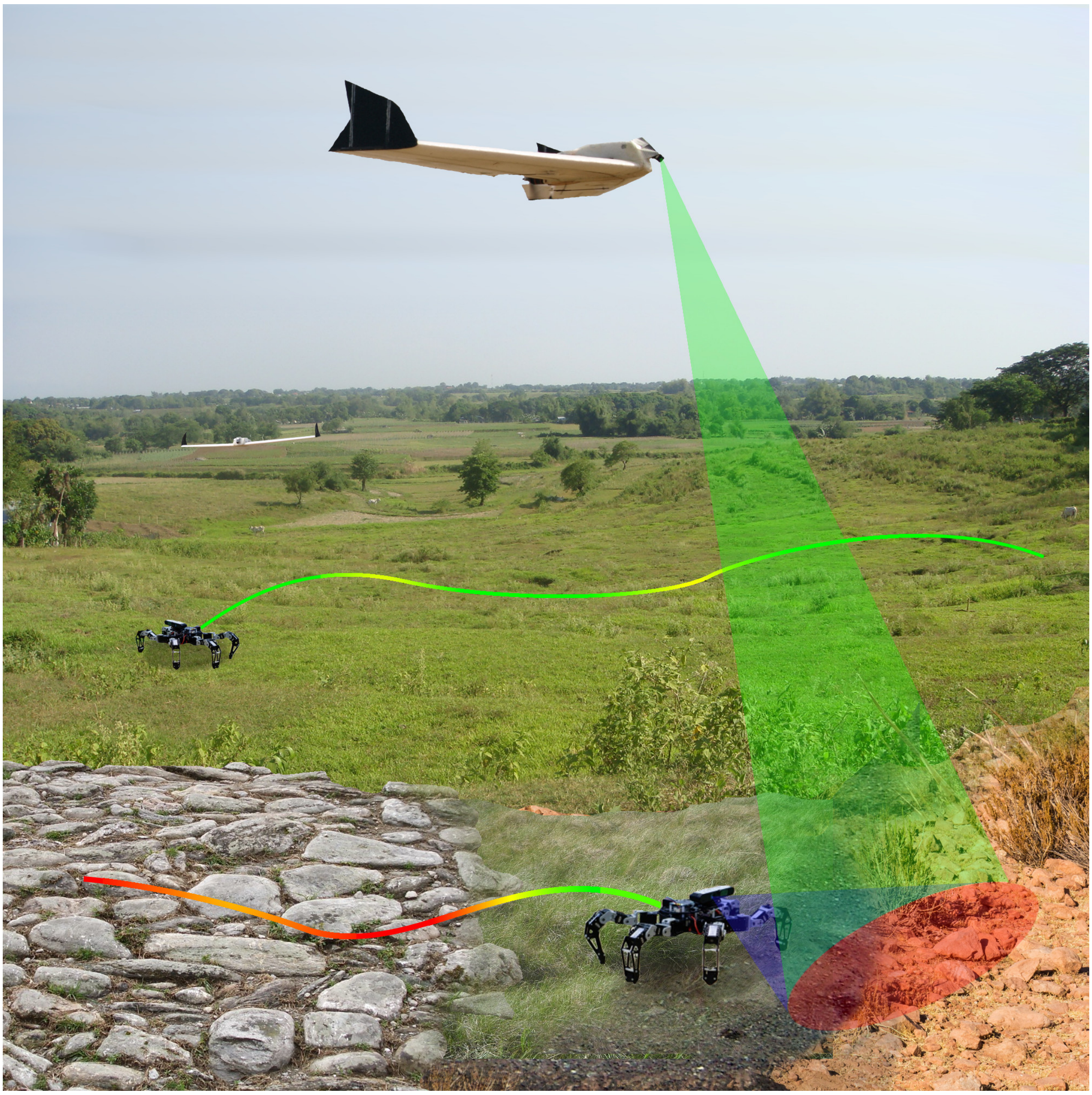


Figure 4: Deployment of crawling hexapod robots and UAVs in planning. The legged robot collects data about its traversability cost by crawling in the terrain, but it can observe only a limited area. On the other hand, the UAV can observe a larger area, thus the robot can benefit from knowledge sharing.

to observe and map large areas. Therefore it is desirable to deploy multi-legged robots in cooperation with other robot types. The authors of [14] present a system where a UAV helps a walking quadruped robot with localization and path planning. Imagine a load-bearing walking robot and a lightweight UAV that is capable of docking with the robot. The legged robot can experience how hard the terrain is to traverse but is limited in both its viewpoint and speed. The UAV cannot sample the terra-mechanical properties of the terrain but is suitable for exploration of large areas due to its raised viewpoint and high speed. Thus, this setup can benefit from advantages of both types of vehicles by inference of knowledge between the UAV and legged robot. In Figure 4, we present a visualization of our interpretation of such a setup.

In this work, we are concerning traversability cost estimation from exteroceptive data for a small hexapod multi-legged robot that is equipped with RGB-D camera. We use CoT [13] as a measure of the terra-mechanical properties of the traversed area. We select the CoT estimation from exteroceptive data since this approach allows us to model the robot's proprioceptive terrain traversal experience and to predict it from remote exteroceptive data. Moreover, we do not rely on explicit classification into a set of discrete terrain classes, which has been extensively studied, e.g. [15; 16; 17; 18]. We rely on robot learning and adaptation to new and unforeseen conditions. This setup favors continuous metrics such as CoT, as they can be gradually updated by an incremental learning algorithm. Note that continous metrics cannot be directly used for terrain classification, as visually distinct terrain classes may easily have similar CoT. Still, some of the terrain characterization approaches used for discrete terrain classification can be utilized for CoT inference. This work aims to build on terrain descriptors yielded from visual terrain classification approaches and explore their usefulness in the problem of the CoT inference.

The rest of the thesis is structured as follows,

- in Chapter 2, we provide an overview of works concerning multi-legged terrain traversal and other terrain related tasks;
- in Chapter 3, we describe the aim of this work in detail;
- Chapter 4 presents our proposed solution to the terrain traversability regression task;
- Chapter 5 reports on the performed experiments;
- finally, in Chapter 6, a conclusion is drawn and discussion of further possible developments of terrain traversability tasks is presented.

Chapter 2

Related Work

In this chapter, we provide a short overview of existing approaches and topics on multi-legged robots and efficient traversing various terrains. In Section 2.1, we discuss the state of the art terrain traversability metrics and overview of the terrain characterization features and approaches is presented in Section 2.2.

2.1 Terrain Traversability Measures

When traversing different terrains, legged robots experience various levels of difficulty, e.g., smooth flat pavement is easy to traverse, while rough woodland terrains are much harder. Robot's ability to efficiently pass over various terrains greatly influences its mission effectiveness. Hence, it is desirable for the robot to create a model of its ability to overcome different terrains, especially in missions in an a priori unknown environment.

Two types of approaches can be identified for definition of the terrain traversability property. The first type can be based on a discrete set of terrain classes, e.g., a particular terrain is traversable in binary classes passable and impassable. The second type can be based on a traversability scale. Both approaches can be combined as in [9] by creating a two-level metric. Note that since this thesis is concerned with the regression of the continuous traversability measure, the following section emphasizes existing continuous traversability measures utilized for passable terrains.

We consider two groups of traversability assessment approaches. In the first group, the traversability is computed from proprioceptive data available to the robot, e.g., using its power consumption and velocity. For the second group, one can construct terrain traversability assessment directly as a function the of captured exteroceptive data, e.g., from the ground slope or vegetation density observed in the robot's vicinity. The main advantage of the former approaches is that it is based on the experience of the robot with traversing of the terrain, i.e., it can reflect on changes in terrain traversability, and it can cope with an unknown atypical terrain. For example, a sloped surface can be much harder to pass when it is wet, yet its slope property remains the same. On the other hand, the exteroceptive approaches are easily adaptable to a wider range of robots. For example, two robots may move over a given terrain with different speed and consume various amounts of energy, but the terrain they exteroceptively perceive remains the same.

The classification of terrain types into given discrete classes can be considered as the simplest approach to the traversability assessment. Although terrain classification does not explicitly indicate that the individual classes have different traversability property, it is often implied. For example, the authors of [15] motivate their classification work by safe navigation over extraterrestrial terrain. However, as it is shown in [1], a false estimation of the traversability property may have fatal consequences to the mission itself.

A combination of discrete classes and a continuous variable is presented in [9], where the model of the traversability is denoted as *danger level*, which is perceived as exteroceptive half-continuous-half-discrete variable describing the terrain difficulty. The variable describes passable terrains with danger level values $d \in [0, 1]$, with $d = 0$ being completely smooth terrain and $d = 1$ representing a terrain that is barely traversable or unknown. Non-traversable terrain is represented by $d = \infty$. The danger level estimation is based on steep slopes, roughness, and terrain step height.

Authors of [11] use a continuous traversal cost based on position, density, and point cloud distributions of sensed obstacles. The traversal cost is constructed in such a way that very low values describe areas such as roads, while high values describe dense vegetation. The metric is designed to work on logarithmic scale in order to minimize errors in the lower portion of the traversal cost spectrum.

The *Cost of Transport (CoT)* is a continuous proprioceptive metric defined in [19] as

$$\mathrm{CoT}_{\text{original}} = \frac{P_{\text{i}}}{v}, \tag{1}$$

where P_{i} is the power input and v is speed of the robot. The metric has recently been utilized by [20; 21; 22; 13; 12] in various forms suitable for robotic and biological locomotion. The CoT metric is well defined for terrains that can be traversed, where it is a nonnegative real number, i.e., CoT $\in [0, \infty)$. For terrains that are not traversable, the robot's velocity equals 0 and the respective CoT is thus considered either undefined or infinite. Arguably, this approach is similar to the danger level approach from [9] as it assigns a continuous value to passable and infinity to impassable terrains. However, the infinity is not explicitly used as in [9].

The authors of [13] have recently reintroduced CoT for battery powered robots, defined as

$$\mathrm{CoT}_{\text{pow}} = \frac{P_{\text{in}}}{m\,g\,v}, \tag{2}$$

where P_{in} is the instantaneous power consumption, m is the weight of the robot, g is the gravitational acceleration, and v is the robot speed. The power consumption based CoT definition can be easily utilized in mobile robotics, as it is defined using instantaneous power consumption P_{in}, which can be computed as

$$P_{\text{in}} = V I_{\text{in}}, \tag{3}$$

where V is the battery voltage and I_{in} is the instantaneous current drawn from battery. The time t and speed v can be measured by the localization system and the robot body weight m is typically constant, although this may not hold for load transporting robots. As the power consumption based definition requires measuring power consumed by the robot, it is not suitable for application where such information is not available, e.g., when computing the biological cost of locomotion.

Alternatively, the mechanical CoT is defined in [21; 22] as

$$\mathrm{CoT}_{\text{mech}} = \frac{E}{l} \tag{4}$$

using the energy expenditure/work E and stride length l. However, the mechanical locomotion is often accompanied by additional energy loss via heat generation. For example, biological locomotion is accompanied by heat loss from muscle fiber. The traditional definition of CoT is expanded in [12] by accounting for heat energy loss as

$$\mathrm{CoT}_{\text{heat}} = \frac{\sum_{i=1}^{n}(W_i + H_i)}{m\,v\,t}, \tag{5}$$

where W_i and H_i are the mechanical energy consumed and heat energy lost, respectively, while moving leg i (out of n legs), m is the body mass of the robot, v its body speed, and t its stride period, i.e., the duration of one gait cycle.

The units to measure CoT in its original form (1) and (4) are given as

$$\mathrm{CoT}_{\text{original}}\left[\frac{\mathrm{W}}{\mathrm{ms}^{-1}}\right] = \mathrm{CoT}_{\text{mech}}\left[\frac{\mathrm{J}}{\mathrm{m}}\right] = \mathrm{CoT}\left[\frac{\mathrm{kgm}}{\mathrm{s}^2}\right]. \tag{6}$$

As some of the CoT definitions discount for body weight or gravitational acceleration, the respective units derived for such definitions differ. For example, the power consumption based CoT (2) in [13] is a dimensionless quantity

$$\mathrm{CoT_{pow}}\left[\frac{\mathrm{W}}{\mathrm{kgms^{-2}ms^{-1}}}\right] = \mathrm{CoT_{pow}}\left[-\right], \tag{7}$$

while [12] defines their CoT (5) as “energetic cost for moving a unit weight by a unit distance” and it is measured as

$$\mathrm{CoT_{heat}}\left[\frac{\mathrm{J}}{\mathrm{kgms^{-1}s}}\right] = \mathrm{CoT_{heat}}\left[\frac{\mathrm{m}}{\mathrm{s^2}}\right]. \tag{8}$$

2.2 Terrain Description

Terrain classes or terrain traversability property can be characterized by terrain description features. In the following section, we provide an overview of features that are utilized in various terrain related tasks, e.g., terrain classification, terrain traversability assessment, or obstacle identification in field environments. Although such features can be constructed using proprioceptive or exteroceptive data, exteroceptive features are emphasized, as this work aims to utilize them.

Some of the terrain traversability measurement approaches presented in the previous section may also be considered as representatives of standalone terrain description features. For example, the approach [9] constructs a continuous terrain traversability metric from exteroceptive data observed from the robot’s vicinity. Although the data are used only for one specific model, the local terrain properties can also be utilized as terrain descriptors in a combination with different traversability metric.

The various approaches that can be used to extract terrain characterization features from visual and geometric data can be categorized into several groups. We consider following three categorizations: whether the feature is primarily based on geometric or appearance data, whether the feature makes use of color information, and whether the feature is dependent on the current robot position and orientation. In Table 1, we classify the approaches presented in the rest of this chapter.

A feature describing the shape of the point cloud is considered to be geometric without color information. Similarly, the color channel of the point cloud segment is a geometric feature with the color information. On the other hand, the color of the image segment is an appearance-based feature with the color information. The color information enriches both appearance and geometric approaches but it also has several pitfalls. Primarily, the illumination has a considerable effect on the color, although this effect can be somewhat limited by using certain color spaces. Moreover, vegetation may change its color properties during the year, but geometric and traversability properties may remain unchanged. Besides, various man-made surfaces can be produced in a broad spectrum of colors.

An online, appearance-based approach that computes a description of the image segments or patches is position dependent, and an approach that computes geometric features for an externally localized robot from an externally supplied point cloud is not position dependent. In the former case, the feature representation itself (i.e., the image segment captured from the robot position) is dependent on the localization, while in the latter case, only the externally supplied information is location dependent. We consider a setup where a robot incrementally builds a point cloud using an onboard sensor and uses such data to localize itself and the terrain descriptor is computed based on the point cloud that remains the same even after the robot moves. Thus, the terrain description is indepedent on the robot position. However, individual point clouds are reconstructed from a particular robot positions, and thus the point cloud can be considered as position dependent because of possible view obstructions and varying point cloud density influenced by the robot viewpoint.. Approaches that are dependent on the current robot

Descriptor Type	Origin/Application	A/G	C	P
Basic Descriptor Types				
Pixel Value	[11; 16]	A	-	-
Voxel Value	[18; 17]	G	C	I
Superpixel Color	[15]	A	C	D
Wavelet	[23; 24]	A	G	-
Gabor Filters	[25; 11]	A	G	-
Terrain Height	[26; 27; 9; 28]	G	G	I
Terrain Slope	[9; 28]	G	G	I
Terrain PCA	[26; 27; 29]	G	G	I
Line of Sight Shadows	[28]	G	G	D
Vegetation Index	[30; 29; 18]	-	C	I
Object Detection Features	[31; 32; 33]	G	G	-
Overhead Local Feature Set	[11]	A	-	I
Pixel Intensity HSV		A	C	I
Pixel Intensity Gray		A	G	I
Gabor Filter		A	G	I
Grayscale Window		A	G	I
Overhead Disaster Feature	[34]	A	G	I
Image-to-Voxel Color	[17]	-	C	-
Danger Level Approach	[9]	G	G	I
Slope		G	G	I
Step Height		G	G	I
Roughness		G	G	I
General Global Feature Set	[11]	A	G	D
Obstacle Position		A	G	D
Obstacle Density		A	G	D
Obstacle Point Cloud Dist.		A	G	D
Field Terrain Feature Set	[26]	G	G	I
Terrain Height		G	G	I
Terrain Shape		G	G	I
Terrain Orientation		G	G	I
Off-Road Env. Voxel Feature Set	[18]	G	-	I
Scan Line Features		G	G	I
Point Cloud Features		G	G	I
Color Features		G	C	I
Rough Terrain Feature Set	[28]	G	G	-
Terrain Roughness		G	G	D
↳ height, slope, variance				
↳ line of sight shadows				
Step Height		G	G	I
Object Detection Features		G	G	-
Fast Point Feature Histograms	[31]	G	G	I
Viewpoint Feature Histogram	[32]	G	G	D
Ensemble Set	[33]	G	G	I

Table 1: Terrain characterization features classified, where A/G stands for Appearance/Geometric Based, C stands for Color (C - Color, G - Colorless/Grayscale), P stands for the robot position (D - Depedent, I - Independent), and the symbol '-' denotes the particular category is not defined.

position suffer from different visual properties of the terrain from different viewpoints. This problem is prevalent not only in appearance based classification but also in geometric point clouds, as for most sensors, the point cloud density is lower for longer distances. On the other hand, the position independent data may be harder to acquire, e.g., data reconstructed from multiple angles or captured from an unchanging viewpoint such as an overhead position.

The following Sections 2.2.1 and 2.2.2 present a set of various terrain description approaches organized according to the appearance-geometric categorization. Selected approaches that utilize both appearance and geometric features are presented in both sections with the particular attention given on the respective feature type.

2.2.1 Appearance Based Approaches

In this section, we discuss appearance-based approaches that are utilized in terrain classification and terrain traversability regression. These approaches range from simple color-based features to systems utilizing overhead data. We also discuss texture recognition approaches that can be utilized as terrain descriptors. Finally, we examine the recent advances in season independent features, which can potentially prove useful in terrain assessment tasks.

The approach presented in [11] uses overhead data to construct a Gaussian model to infer dense geometric LiDAR terrain features while testing the approach based on the global *general features*, i.e., features that extend well into new domains, and local *locale-specific features* that are limited to a single domain. The overhead local features are based on HSV and gray-scale pixel intensity, Gabor filter response, and gray-scale window mean and variance. As this system utilizes overhead images of the traversed terrain, it is an example of the position independent appearance approach. Such datasets can be obtained from a UAV mapping the terrain, or even from publicly available data. However, a large amount of available data has to be captured from a considerable height and the approach is suitable only for systems that do not need high detail, e.g., systems designed for car-sized vehicles. Regardless, the aerial viewpoint is advantageous, as it can capture a large area including patterns obstructed from ground viewpoints. The authors of [34] use an interesting approach to localize roads from the overhead UAV captured images of urban terrain affected by natural disasters. Their algorithm constructs a tree-like structure by adding new road representing edges based on orientation difference with known edges and brightness difference along the edge.

Although the aerial viewpoint represents a specific setup that requires either deployment of a UAV or usage of large georeferenced database, the appearance features used in such systems do not necessarily differ from those used in systems utilizing only near to ground viewpoints. In [15], the authors consider a simple two-dimensional feature based on superpixel means of color dimensions of the Lab color space, where the superpixels are extracted with the Simple Linear Iterative Clustering (SLIC). The visual feature is used as one component of co-training scheme for discrete classification of the simulated extraterrestrial terrain. The other component is a vibration based feature and both components use support vector machines (SVM).

Similarly to [15], the approach proposed in [16] uses a combination of vibration, vision, and additionally also a laser scan based classifier. Unlike SVM co-training in [15], Bayes model with decision rules is used to combine the classifiers. The vision-based classifier uses Gaussian Mixture Models in the RGB space (with equalized Y dimension in YCbCr) to model different terrain types.

An alternative technique that can be utilized for terrain description is texture recognition. Naturally, we classify texture characterization approaches as appearance based. A considerable amount of work has been done to classify textures using various frequency-based approaches. Authors of [35] compare several approaches to texture classification of steel samples with varying roughness and report that wavelet-based features outperform both gray level co-occurrence matrix (GLCM) and PCA/PLS based approaches.

The wavelet texture representation is further developed in [24], where are compared various setups for wavelet features on several natural textures, including textures that can occur in terrain datasets. The authors conclude that texture characteristics are encoded mainly in high and middle-frequency regions. The best wavelet energy signature is a four-dimensional feature extracted from two level discrete wavelet transform (DWT), that outperformes other more complicated setups for all wavelet bases. Moreover, they have found that wavelet base choice affects texture discrimination, but the best performing base has not been selected.

Other works commenting on wavelet transforms in the context of texture classification or segmentation include [23], which pioneered wavelet transform in texture recognition, DWT-based texture segmentation algorithm proposed in [36], and various scale and rotation invariant wavelet extensions reported in [37; 38; 39].

Although works concerning wavelet-based texture analysis such as [36] often consider wavelets to be superior, other frequency-based approaches have also been developed. 2D Gabor filters are linear filters combining Gaussian and cosine functions. An analysis of number of Gabor filter based features is reported in [25], where it is concluded that grating cell operators from [40] perform the best. A more recent approach for texture classification using Steerable Pyramid based Laws' Masks is proposed in [41].

Virtually all appearance-based approaches can suffer from seasonal changes that cause changes in vegetation structure and color. While the geometric based approaches also suffer from this problem, the problem is more prevalent for appearance-based methods, especially if color is used. This problem is addressed in [42], where various image feature extractors are evaluated with cross-seasonal performance in mind. Some attempts to transform images from one season to another also exist, e.g., [43]. However, it should be noted that an application to the terrain classification would be more complicated than just transforming the appearance, as the terrain traversability property may also differ across seasons.

2.2.2 Geometry Based Approaches

In this section, we provide an overview of geometric features utilized in terrain classification and terrain traversability assessment. Geometric features range from simple features describing color properties of a point cloud section or voxels to complicated features sets based on data captured using multi-sensor arrays. We shortly discuss general properties of geometric features and present examples of the terrain characterization features, including some features that were not originally designed as such.

Geometric features have several advantages over appearance features. Most notably, geometric features describe shape property of the terrain and they are not affected by lighting changes. However, this holds only if the features are captured by certain sensor types. If the geometric features are reconstructed from an RGB-D camera or have some special property, e.g., added color information, illumination changes can still have some effect. Other than greater invariance to illumination changes, the geometric features are also less likely to be influenced by weather and seasonal changes.

Geometric features are often utilized as a basis for evaluation of other predictions, serving as input data for fixed traversability assessment models. The approach presented in [11] uses position, density, and point cloud distributions of the sensed obstacles as global features, i.e., features that extend to other domains, and utilizes them to evaluate predictions from local features. Authors of [9] propose to construct a danger level metric based on locally observed slope, roughness, and maximal height difference.

The approach [17] uses the combination of a geometric and an appearance-based approaches. A simple average voxel color in the HSV space is utilized to classify particular terrain types. However, unlike in most of the previous cases, the SVM classifier is not trained on the same type of data, i.e., voxels, but rather on raw images with hand-labeled pixels. This approach is combined with an elevation map to plan motion of the robot.

A number of geometric features in terrain related tasks is specifically designed to cope with vegetation. For example, the authors of [26] use a set of thirteen features to describe terrain, vegetation, and obstacles in an agricultural environment. The features are computed from a local neighborhood of the point of the interest and features are divided into four height features based on the features' z coordinate, four shape features based on the PCA, three orientation features based on the normal vectors of the local plane, a distance feature, and a reflectance feature. The PCA features are based on various statistics computed from the 3-by-3 covariance matrix eigenvalues $\lambda_1, \lambda_2, \lambda_3$. The PCA features are specifically capable of distinguishing vegetation from a plain terrain, using the intuition that for planar structures $\lambda_2, \lambda_3 \gg \lambda_1$ and for linear structures $\lambda_3 \gg \lambda_1, \lambda_2$. The classifier is trained by the SVM from the labeled data. The variable size of the local neighborhood is considered to compensate for growing sparsity of the point cloud at longer distances from the robot position, where the relation is captured by a linear equation. This approach requires the z-coordinate of the point cloud to be orthogonal to the surface to properly compute the height of the features. This is done by fitting a global ground plane to the dataset. Although this seems to be sufficient in a flat field environment, a more structured environment can be problematic. Note a similar approach that is proposed in [27].

The approach presented in [18] uses a set of various LIDAR based features to estimate the height of the support ground surface with possible vegetation occlusion. The used features are based on scan-line, point cloud, and color information. Scan-line features are computed for constructed scan line segments. Used statistics are the number of returns, the distance between the first and last points in the segment, and the cumulative distance between the neighboring points in the segment divided by the total distance between the first and last points in the segment. Point cloud features represent a large set of features based on various previous approaches. The used color features are the number of points with the properly exposed color tags, average intensity normalized values for red, blue and green channels, the normalized difference vegetation index (NDVI) as in [29; 30], the green channel relative to other channels, and the saturation and value channels from the HSV color space. The random forest classifier [44] is used with a hand-labeled dataset.

Another feature set has been developed for rough terrains in [28], where data are recovered from a stereo camera to describe the area of interest in front of the robot, which traverses a test track comprised of a rocky and a brick-filled area. The features are based on the center line average, slope, average local variance, line of sight shadows, maximum step height, and even run length. Similarly to [26], this presumes the existence of the elevation, i.e., the approach requires the ability to fit a reasonable plane for the obtained point cloud data. The motion efficiency is then computed as the power consumption CoT, i.e., it is based on the power consumption and robot speed.

Similarly to features used in texture recognition, numerous features developed for object detection and description can be potentially utilized in terrain related tasks. The Point Cloud Library[1] provides an implementation of a considerable amount of such features. Some of such approaches are [45; 31; 32; 46]. The main advantage of the commonly used geometric approaches is available implementation. Moreover, works comparing at least some of these approaches do exist. On the other hand, a comparison reported in [47] concerns general object description, and the actual performance for a particular

[1] http://pointclouds.org/

terrain may differ significantly.

A considerable portion of geometric feature approaches is based on a statistic using the normals of the points in the question. The point normals are most commonly computed by fitting a plane to some local neighborhood and computing its normal. However, unlike in image-based approaches, where a window centered over a pixel always covers the same amount of points (if border regions are omitted) two-point cloud regions may differ in density. Therefore, if we compute normals in two regions that differ in density, two situations may arise. If the region under investigation is defined by a fixed radius, the normal is based on regions that differ in the number of points. If the normal is computed from a fixed number of the closest neighbors, it is based on regions that differ in radius.

Authors of [45] propose a point cloud feature based on the normal vector comparison. All point pairs in its neighborhood are considered to compute the feature for a point of interest, and four statistics based on the angle differences between the points' normals and the distance vector between them are computed and binned into a histogram. In [31], an improvement of this approach is proposed: only pairs with the point of interest are considered, and only three statistics are computed. The histogram is enriched by distance weighted histograms of the points in the neighborhood to reintroduce the information lost by the limited point selection. A close-to-realtime online incremental implementation based on scanlines is possible. Further extension of this principle is provided in [32] and [46] by combining the previous approaches with statistics based on the viewpoint and point cloud centroid relation. However, the viewpoint approaches are primarily designed for object description in robotic manipulation and would probably not generalize well to terrain descriptors.

Another approach is presented in [48], where an interest point is represented by the minimal and maximal curvature estimates for its neighborhood. These estimates are computed from the normal vectors and distances between the points. The authors of [33] present an ensemble approach that uses a number of shape functions histograms (10 histograms) to capture the shape property. The histograms are matched using the L1 distance [49].

Based on reported literature survey, we have selected a set of the terrain characterization features and traversability metric for the terrain traversability assessment scenario. The individual features, the used metric, and our learning framework are detailed in Chapter 4.

Chapter 3

Problem Statement

This work searches for suitable terrain descriptors that are invariant to certain types of viewpoint changes in the task of terrain traversability regression. Specifically, we are interested in viewpoint changes between near-to-ground robot viewpoints and aerial UAV viewpoints. We explore several combinations of existing terrain descriptors and learning models. The individual combinations are used for the regression of the terrain traversability property in one- or two-viewpoint setups. A simple chart describing the flow of data in this work is presented in Figure 5. The power consumption based CoT metric (2) is used as the terrain traversability measure.

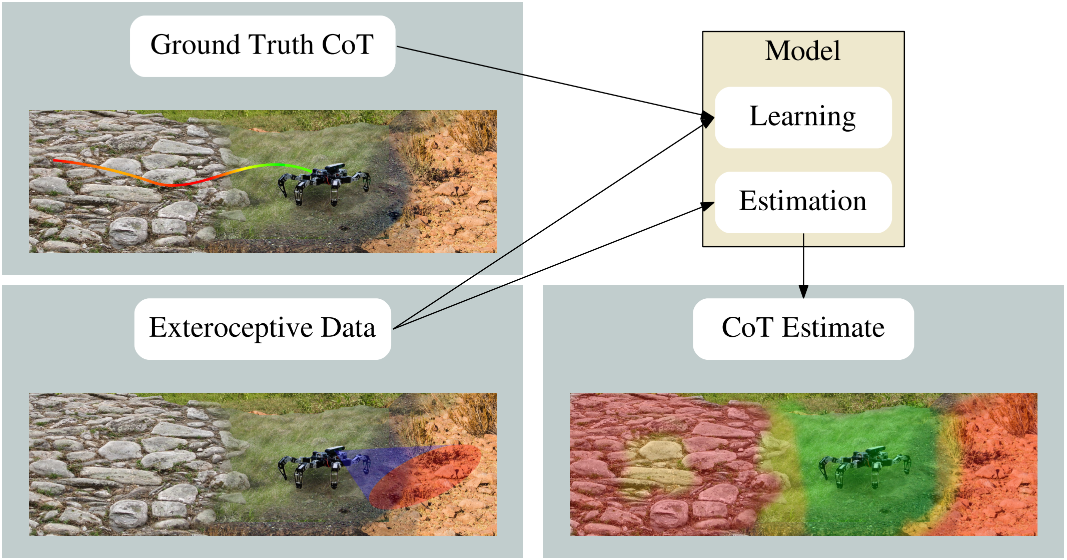


Figure 5: Input and Output Data in Terrain Treversability Regression Task.

The considered scenario consists of a hexapod walking robot and a UAV. The robot walks along its trajectory on the ground and observes the terrain using both proprioceptive and exteroceptive sensors. Specifically, it captures the cost to traverse the terrain and RGB-D representation of its surroundings. The UAV senses the terrain only exteroceptively, capturing an additional RGB-D dataset. When viewpoint invariant terrain features are utilized, this setup allows for evaluation of the terrain traversability property in much larger areas than those visited solely by the hexapod robot, as the UAV is not constrained by the ground robot trajectory and its raised viewpoint allows it to cover a larger area. In the scope of this work, two alternative dataset types allow us to test for viewpoint invariance of the terrain characterization features.

Besides, particular scenarios for a single viewpoint are also utilized, but they do not test viewpoint invariance of the terrain descriptors. Still, they provide additional opportunity to evaluate the quality of the various terrain descriptors and learning approaches.

We consider both incremental and nonincremental approaches to terrain traversability estimation, because both types have specific merits. If the robot is deployed in an unknown environment, it has to be able to learn the terrain property incrementally. On the other hand, nonincremental approaches can provide more informed results, if a large enough dataset is available. In the next section, we introduce the concept of problem life stages to further describe and compare the incremental and non-incremental models.

3.1 Problem Life Stages

The problem of terrain traversability regression is divided into two life stages: the learning stage and the inference stage. The model learns the terrain traversability property in the learning stage and responds to queries in the inference stage. The learning stage accepts pairs of exteroceptive terrain descriptors and proprioceptive ground truth CoT. The inference stage accepts the terrain descriptors and returns the respective CoT estimate. This is not to say that the individual stages are executed in a pre-set order. For the class of incremental learning approaches, it is possible to use and update the model in any given order, making the two life stages concurrent.

When a non-incremental approach is used, the two life stages are easily differentiated as they are separated in time. First, the model is prepared from the descriptor-CoT pairs in the learning stage. It is done only once at a given instant which separates the learning phase and the inference phase. Further on, the model is in the inference stage and cannot be expanded with new data. It is only used for the terrain evaluation, i.e., it returns CoT estimate for descriptor queries. The individual stages quite often correspond to different types of datasets. For example, in our two-viewpoint setup, the individual life stages correspond to the individual viewpoints, i.e., the model is created using one viewpoint and tested using the other.

When an incremental approach is used, the two life stages are executed concurrently. Each datapoint can be used both to estimate the CoT and to learn the model if the ground truth CoT is known. As a result, there is not a difference between the ways how the data captured from different viewpoints are added to the model. However, for a single datapoint, a given sequence of actions still exists. First, if CoT estimation is required, the model is queried for CoT using the datapoint. This query is a part of the inference stage. Only after this query, the datapoint is used to update the model. This update is a part of the learning stage. This specific sequence is necessary to avoid spurious results, i.e., in this case of querying a descriptor immediately after it has been added to the model.

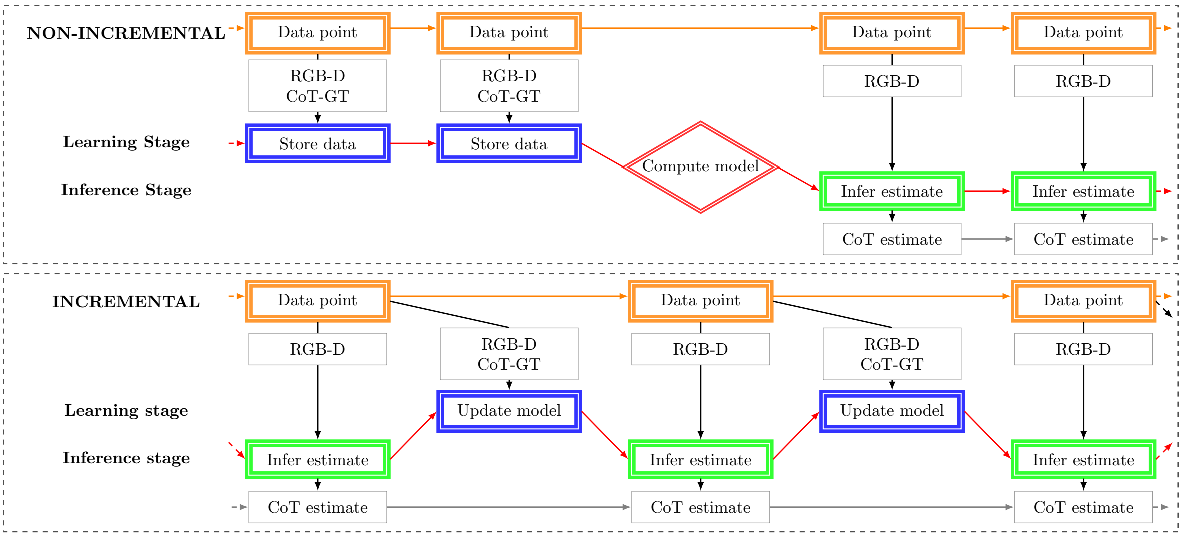


Figure 6: Life stages of incremental and nonincremental model.

In this work, we aim to create a terrain CoT model which utilizes the two life stages problem formulation. A comparison of the life stages in different setups is presented in Figure 6.

3.2 Quality Metrics

We utilize both quantitative and qualitative analysis of our CoT estimation. Our quantitative metric is based on the absolute error of the terrain traversability inference, which is computed as the absolute difference between the ground truth traversability, i.e., the traversability computed from the real data collected by the robot, and traversability inferred by the model. The mean and variance of the inference error are considered, with the mean being the primary metric.

In the qualitative analysis, we first check how the model reacts on areas with different traversability. This check should provide a result similar to the result of the quantitative analysis. Afterward, we investigate how the model behaves near areas with high CoT variances which is usually exhibited by spikes in the CoT estimate. Typically, such areas are results of the robot being temporarily stuck in a particularly tricky area, meaning that spikes are a direct results of the particular terrain properties. Still, we consider such spikes to be undesirable, as we aim to model the terrain property on a scale larger than a single robot step and our robot moves using a specific gait. Finally, we investigate border areas between different terrain types.

Chapter 4
Proposed Solution

The primary aim of this work is to search for suitable terrain characterization features invariant to certain types of viewpoint changes. These features are used in the inference of CoT perceived by the hexapod crawling robot in several experimental setups. In this chapter, a framework for terrain feature extraction and CoT learning and estimation is introduced. The core components of the framework are the following individual building blocks, i.e., the RGB-D datasets, utilized terrain features, the learning procedure and the CoT metric, and the strategies used for obtaining, storing, and exploiting aforementioned building blocks. The individual building blocks and strategies are described in the following sections.

4.1 Framework Building Blocks

In this section, we describe the individual building blocks used in the CoT inference framework. We also provide information about the made implementation of the specific blocks, mainly reporting on the utilized third party toolboxes.

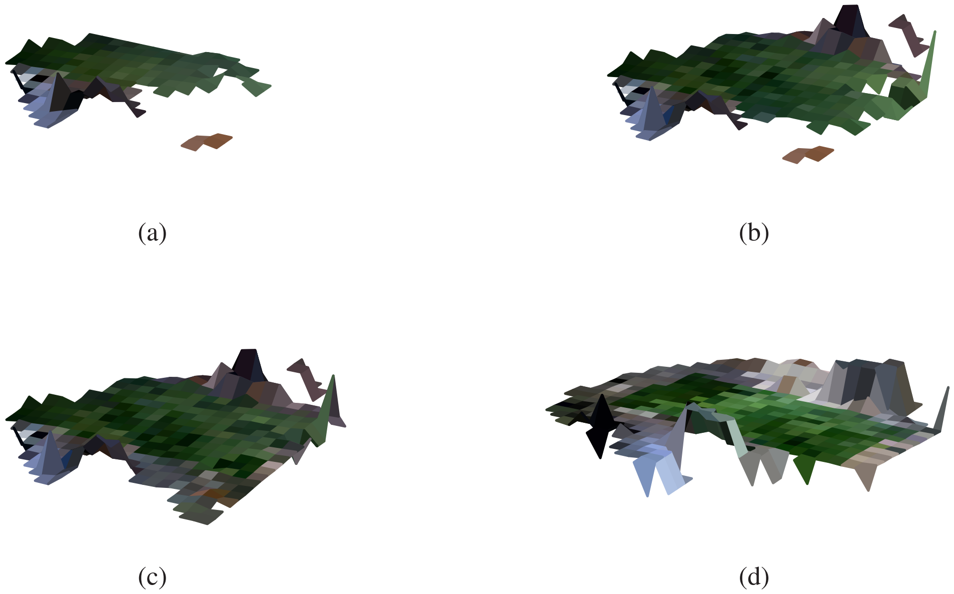
4.1.1 Exteroceptive Data Representaion

Regarding the form of our exteroceptive datasets, the utilized data differ both in viewpoint (ground, aerial) and type (image, point cloud). Near-to-ground viewpoint captures a *walk scan*, i.e., data representing the robot's field of view when traversing the terrain. These ground datasets additionally carry proprioceptive information regarding the robot's traversal of the terrain. The aerial viewpoint captures an *environment scan* and represents an overview of the traversed terrain. In each of our datasets, we store the exteroceptive data in two forms: RGB-depth image pairs and point clouds with color information.

Each image carries a time tag and a georeference information representing the robot's position. We use georeference information in the point cloud reconstruction. This holds for both ground and aerial datasets, as the same sensor is used to capture both types. The localization of individual image pairs has been obtained using the ORB simultaneous localization and mapping (SLAM) [50]. In such a way, the localized image represents an unbiased believe of the robot about the structure of its surroundings. This has shown to be the most precise solution available for our indoor experiments, where a precise GPS-based localization is not available.

The walk scan point clouds are computed incrementally along the robot's trajectory to provide the robot with up to date information about the environment. Hence, at any given time, the reconstructed point cloud represents only areas already observed by the robot. The terrain assessment framework is passed point cloud "snapshots" of the environment, i.e., the robot is provided with the ability to incrementally learn the terrains. An environment scan, on the other hand, represents the terrain visible from an aerial location, e.g., a UAV passing over the terrain. Therefore it represents the terrain as a whole and we construct only one environment scan point cloud that is used at any time. A visualization of typical walk and environment scans is shown in Figure 7.

Point clouds are reconstructed in the following two steps. First, each RGB-depth image pair is used to



(a)

(b)

(c)

(d)

Figure 7: Visualization of (a,b,c) three stages of the incremental walk scan construction; and the (d) environment scan.

reconstruct a single-view point cloud. Simple colorless point coordinates denoted $p_{\mathrm{r}} = [x_{\mathrm{r}}, y_{\mathrm{r}}, z_{\mathrm{r}}]$ are obtained using

$$\begin{array}{rcl} z_{\mathrm{r}} & = & d_{\mathrm{img}} \\ x_{\mathrm{r}} & = & z_{\mathrm{r}}\frac{x_{\mathrm{img}} - c_{\mathrm{x}}^{\mathrm{d}}}{f_{\mathrm{x}}^{\mathrm{d}}} \\ y_{\mathrm{r}} & = & z_{\mathrm{r}}\frac{y_{\mathrm{img}} - c_{\mathrm{y}}^{\mathrm{d}}}{f_{\mathrm{y}}^{\mathrm{d}}} \end{array}, \tag{9}$$

where d_{img} is the depth from the undistorted depth image, x_{img} and y_{img} are the column and row coordinates in the depth image, and the depth sensor calibration matrix K^{d} is

$$K^{\mathrm{d}} = \begin{bmatrix} f_{\mathrm{x}}^{\mathrm{d}} & 0 & c_{\mathrm{x}}^{\mathrm{d}} \\ 0 & f_{\mathrm{y}}^{\mathrm{d}} & c_{\mathrm{y}}^{\mathrm{d}} \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Afterwards, the points are paired with color information from the undistorted RGB image, making the reconstructed RGB point denoted $p_{\mathrm{rc}} = [x_{\mathrm{r}}, y_{\mathrm{r}}, z_{\mathrm{r}}, c_{\mathrm{r}}, c_{\mathrm{g}}, c_{\mathrm{b}}]$. Only points sufficiently close to the sensor (2.0 m for walk scans and 1.5 m for environment scan) are considered in further computation. The world coordinate points denoted $p_{\mathrm{wc}} = [x_{\mathrm{w}}, y_{\mathrm{w}}, z_{\mathrm{w}}, c_{\mathrm{r}}, c_{\mathrm{g}}, c_{\mathrm{b}}]$ are then obtained using,

$$\lambda \begin{bmatrix} x_{\mathrm{w}} \\ y_{\mathrm{w}} \\ z_{\mathrm{w}} \\ 1 \end{bmatrix} = H_{\mathrm{pos}} \begin{bmatrix} x_{\mathrm{r}} \\ y_{\mathrm{r}} \\ z_{\mathrm{r}} \\ 1 \end{bmatrix}, \tag{11}$$

where $H_{\mathrm{pos}} \in M^{4\times 4}$ is the homogeneous transformation obtained from the robot's known position, orientation, and $\lambda \neq 0$.

Then a global point cloud is reconstructed by merging reconstructions of the individual images. In the merging process, we sequentially add new information to the global merged point cloud. This is done in a two-step process. First, for each point to be added, i.e., a point p_{wc} reconstructed from the last image, we query the merged point cloud (represented in a search tree), and add the point only if its local area is not dense enough, i.e., there is less than 5 points in the 10 mm neighborhood. Second, we filter outliers by keeping only new points that have at least 3 points in their 10 mm neighborhood. Note that for each added image, each of the two search trees is constructed only once. First, the first tree for filtering dense areas is constructed from the global merged point cloud. Then, all the new points are tested. The global point cloud and the new points that passed the first test are used to construct the second tree, and the new points are tested again. Therefore, this design actually allows density higher than that specified in the first test, i.e.. higher than 5 points in 10 mm neighborhood, as the design only limits where new points are added in the bulk. The bulk of the new points is only checked for the minimal, not the maximal density. Moreover, the total size of the point cloud is limited. When the limit is exceeded, the point cloud is randomly subsampled, i.e., a third of its points is randomly selected and removed until the point cloud does not exceed the limit. The size of the point cloud is limited to $2,000,000$ points in the creation process, and the maximal size of the point cloud passed the learning framework is $500,000$ points.

4.1.2 Terrain Characterization Features

Based on the literature survey and preliminary results, we considered the following features for benchmarking. Both geometric and appearance data, including frequency-based features that are not originally designed for terrain description have been utilized. Note that appearance-based features are based on time-referenced images captured from an onboard sensor and cannot be extracted for datasets captured from the aerial viewpoint, as the aerial images cannot be referenced with respect to the robot's trajectory. Therefore, they are tested only in a subset of the performed experimental runs. The following features are considered:

Image segmentation features are appearance based image-segment color-channel features. They are similar to segmentation features utilized in the approach [15] that segments images using the SLIC algorithm and uses patch means of the ab channels of the Lab color space as the features. First, we segment the image by the SLIC algorithm. Contrary to [15], which segments on grayscale images for the sake of computation efficiency, we segment on the Lab color space. A SLIC implementation available in the scikit-image package [51] is utilized. The channel means of the individual segments serve as feature descriptors, i.e., each image point is described by the channel mean of its respective segment. We construct three features based on two color spaces: the RGB channel means (3 dim), Lab channel means (3 dim), and ab channel means on the Lab space (2 dim).

Image wavelet features are appearance image frequency features based on Db2-F2 feature presented in [24]. The feature is computed in the following manner. First, the DWT response is computed by applying the Daubechies 2-tap DWT to a grayscale image. Then the cumulative sum of the DWT response for each image point is computed. Finally, the energy signature feature at a given point is the cumulative sum of the responses in its respective local image area, i.e., the 72×72 pixel neighborhood with the DWT response normalized by the area size. Note that this neighborhood is extracted from the raw image, as image transformations, e.g., resizing the image, appear to have a negative impact on the frequency-based feature discrimination. We use the F2 (based on the classification listed in [24]) feature, i.e., our feature is three dimensional, representing the level 1 horizontal, vertical, and diagonal DWT response. The Python PyWavelet toolbox [52] is utilized for the DWT computation.

Point cloud color features are geometric color-channel features computed on reconstructed point clouds. Every point cloud color feature is based on sampled point of interest. Two types of features are used: the channel values of the sampled point (denoted as P in the reported results) and channel means of the points in the spherical neighborhood with the 0.2 m radius spherical neighborhood (denoted as M in the reported results). The RGB and Lab color spaces are used with both types, making four point cloud color features in the total.

Point cloud geometric features are a set of geometric features based on terrain classification features presented in [26]. We have used 11 out of 13 therein presented features, combining them in four sets. Namely, we use the shape feature set, height feature set, orientation feature set, and all of them combined in a full feature set, leaving out the reflectance and distance features which are not suitable for our experimental hardware and setup (see Sections 5.1 and 5.2 for details). As the features are not invariant to coordinate system rotation, it is necessary to base the coordinate frame on a known ground plane. If the ground plane is not known, it is estimated by fitting a single plane to the whole point cloud. The normals needed for the orientation feature computation are estimated by fitting a plane to the 5 nearest neighbors of the sampled point. Feature statistics are computed for a spherical region with 0.3 m radius around the sampled point.

4.1.3 Learning Algorithms

We have considered four learning approaches on top of the selected terrain features. Two of them support online incremental learning. The following approaches have been evaluated:

Support Vector Regression (SVR) [53] is a maximum-margin regression algorithm. We use a scikit-learn [54] Python implementation of the SVR with the radial basis function (RBF) kernel. SVM and assorted support vector variations have been commonly used in terrain classification tasks as it is reported in [15; 26; 17].

Regression trees are a precomputed decision structures that utilize recursive partitioning to infer a continuous variable. We used a scikit-learn Python implementation. Previously, a random forest classifier has been used in the terrain classification task in [18].

Incremental Gaussian Mixture Network Model (IGMN) presented in [55; 56] is an online incremental learning approach thats creates and updates Gaussian mixture model based on streamed data points. The IGMN supports a full prediction of a data point based on an incomplete input of any kind. The Fast-IGMN is an IGMN improvement presented in [56], which improves the IGMN time complexity to $\mathcal{O}(NKD^2)$, where N is the number of data points, K is the number of components, and D is the data point dimensionality. We have used our implementation of the Fast-IGMN. Gaussian mixture models has been used for terrain classification task in [16].

Hoeffding tree or Very Fast Decision Tree Learner (VFDT) [57] is an online incremental decision tree learning algorithm that utilizes the Hoeffding bound. Its output is asymptotically identical to that of a conventional learner. We have used a slightly modified VFDT implementation based on [58]. However, unlike the other utilized regression approaches, Hoeffding tree is used with a discrete number of classes and nine evenly spaced intervals and one interval for any larger value are used for the results reported in this thesis.

4.1.4 Cost of Transport

This work utilizes the power consumption CoT (2) as the terrain traversability metric. CoT is computed based on the robot's power readings, its weight, and its localization which is utilized for robot speed estimation. As there is a considerable difference between the robot's stride period and the frequency of the power consumption readings, CoT has a periodic nature. Therefore CoT is smoothed using a median filter roughly sized to the typical stride period to mitigate this problem

4.2 Framework Strategies

This section explains how the above-selected data representation, terrain features, learning algorithms, and cost of transport (i.e., the forming building blocks) are combined in an inference learning framework that estimates CoT. There are two main parts of the framework: Data Aquisition and CoT Modelling. See Figure 8 for a scheme of the complete CoT inference framework. In the Data Aquisition framework, the individual images, point clouds, and CoT ground truth data are utilized to create the terrain descriptors. The task of the CoT Modelling framework is to create the CoT-feature descriptor pairs, i.e., dealing with the time delay between the acquisition of the terrain visual descriptors and its CoT characterization, and using them in the CoT regression model itself.

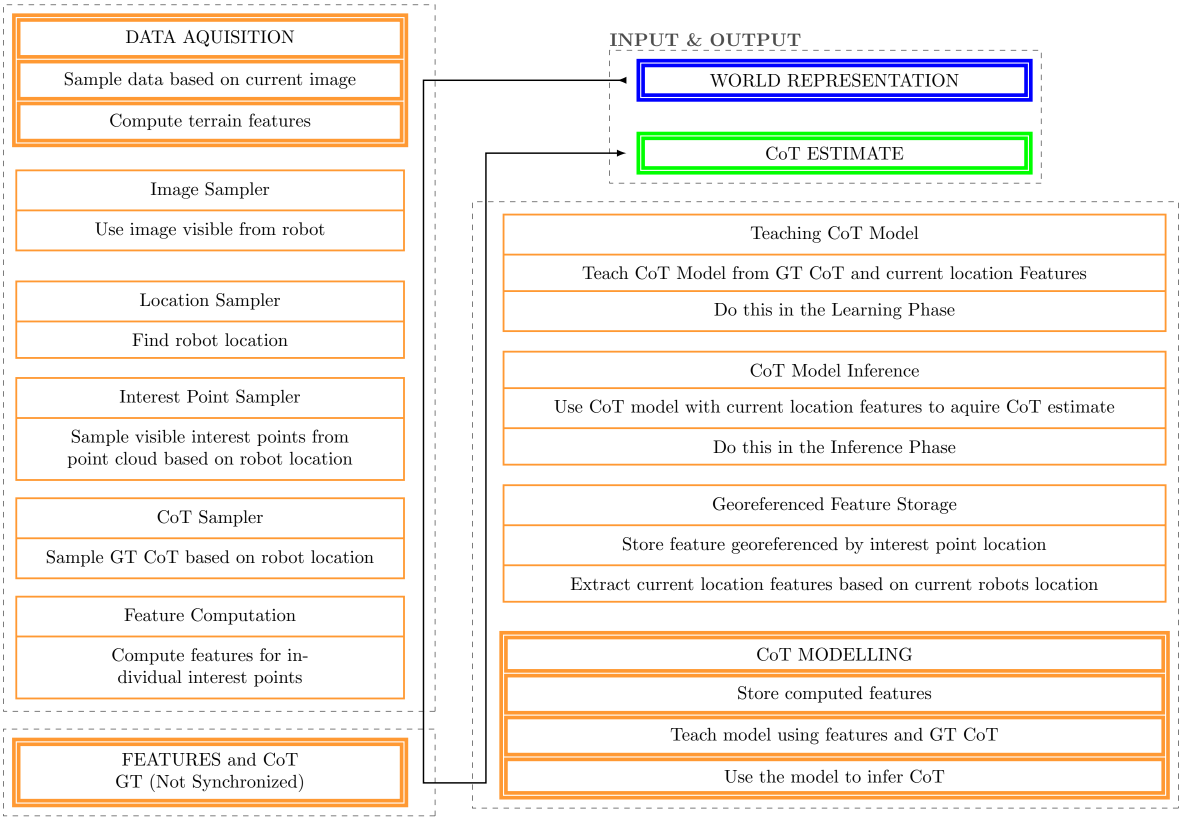

DATA AQUISITION
Sample data based on current image
Compute terrain features
Image Sampler
Use image visible from robot
Location Sampler
Find robot location
Interest Point Sampler
Sample visible interest points from point cloud based on robot location
CoT Sampler
Sample GT CoT based on robot location
Feature Computation
Compute features for individual interest points
FEATURES and CoT GT (Not Synchronized)
INPUT & OUTPUT
WORLD REPRESENTATION
CoT ESTIMATE
Teaching CoT Model
Teach CoT Model from GT CoT and current location Features
Do this in the Learning Phase
CoT Model Inference
Use CoT model with current location features to aquire CoT estimate
Do this in the Inference Phase
Georeferenced Feature Storage
Store feature georeferenced by interest point location
Extract current location features based on current robots location
CoT MODELLING
Store computed features
Teach model using features and GT CoT
Use the model to infer CoT

Figure 8: Scheme of the CoT inference framework.

4.2.1 Data Acquisition

The data acquisition framework operates on individual datasets consisting of georeferenced RGB-D images, i.e., the color image data RGB with depth, from the robot and georeferenced RGB point clouds of the operational environment. The task of the data acquisition framework is to sample points in the robot's field of view (FOV), compute the terrain descriptors at these points, and pass these further on to the CoT Modelling framework. This is repeated for the given number of iterations, i.e., either for each of the available RGB-D images, or for a subset of the images.

The point cloud and robot's location is utilized to compute the terrain descriptors. First, the point cloud section located in the robot's FOV is isolated. Although the point cloud may contain points that are located out of the robot's FOV, the FOV is selected as it represents an area that is located in front of the robot, i.e., an area that the robot is likely to traverse. Then, visible points of interest are sampled semi-randomly from the FOV, with each new point being discounted by its distance from the points already sampled in this iteration. Usually, 30 points are sampled in each iteration. Afterward, if a walk scan point cloud is used, an image captured by the RGB sensor is sampled based on the time tag. The sampled points, the image, and the point cloud are then used to compute the individual features.

If such data are available and required, the robot's location is then used to sample CoT information from the known trajectory. The individual sampled points, their respective feature descriptors, the robot's location, and the CoT ground truth are then passed to the CoT modeling framework.

4.2.2 CoT Modelling

The CoT Modelling framework accepts data sampled by the Data Acquisition framework and uses them to either enrich its CoT regression model or to infer the CoT estimate. Primarily, the framework accepts sampled points described by terrain features and CoT ground truth data, which are either used in learning or paired with CoT estimates and passed to be evaluated by the error metric. The Data Acquisition framework provides the CoT ground truth related to the robot's position and descriptors of the terrain visible from that position, i.e., the descriptors and CoT are not on the same location, since the robot knows the CoT ground truth only after successfully traversing the terrain and estimating the speed it traversed the particular part of the terrain from the georeferenced data. We introduce feature storage that maintains a georeferenced feature dictionary to deal with the RGB-D-CoT pair acquisition delay. The dictionary is queried when the robot reaches any arbitrary location, and all features located in close proximity, i.e., in a spherical region with radius 0.2 m, are passed to the learning framework together with CoT for the current location. The size of the feature storage is limited, and it is randomly pruned when its capacity overflows. Hence, a persistent feature map of the environment is not created, and the robot rather incrementally learns the forthcoming terrain.

The features from the storage are paired with the CoT ground truth respective to their location and used to expand the model or to query the model for the CoT estimate. When they are used for both, the estimation is executed first in order to avoid spurious results. Note that the model expansion corresponds to the learning stage and the CoT estimation corresponds to the inference stage of the CoT regression problem presented in Section 3.1. The incremental models can be simply passed any data at any time, i.e., at any point, it is able to accept new feature-CoT pair and learn or respond with the CoT estimate on the feature query. When learning, the model updates online as the data are presented. The non-incremental models need a more strict distinction between the learning and inference stages. In the learning stage, the model only accepts new feature-CoT pairs and stores them. After the learning stage is finished, the model uses the stored data to finalize itself, i.e., to precompute the part of the model that is capable of the CoT estimation. This computed model is then used in the inference stage, where it is only capable of the CoT estimation.

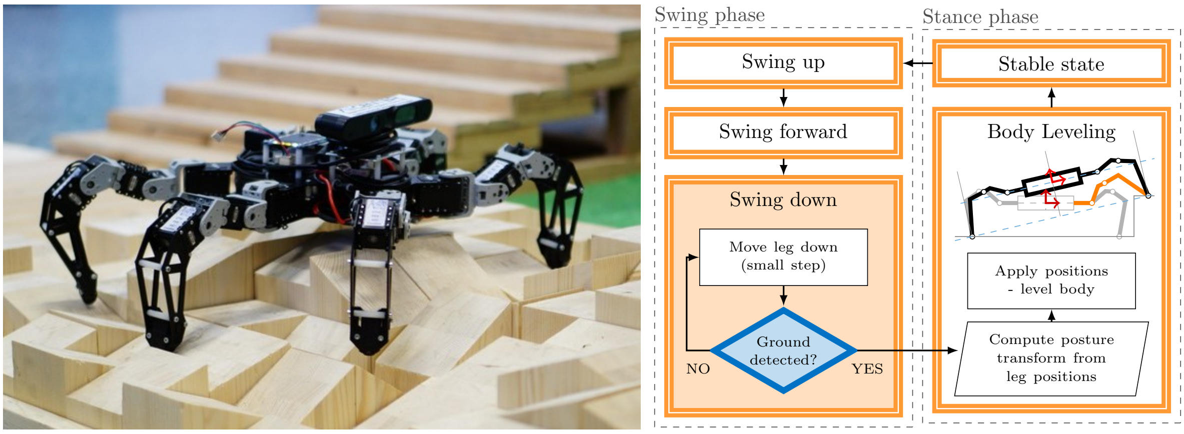
Chapter 5
Experiments

The solution for the terrain traversability regression proposed in Chapter 4 has been evaluated using several test scenarios. We utilize an indoor test track consisting of six different terrains of varying roughness, foothold stability, and appearance. The test track has been traversed by our hexapod crawling robot that senses the terrain both proprioceptively, i.e., measuring its power consumption and motion progress, and exteroceptively, using the captured RGB-D representation of the terrain.

In this chapter, we report on the achieved results in the terrain traversability regression experiments as follows. In Section 5.1, we describe our experimental platform. In Section 5.2, we report on the specifics of our individual experimental setups. Specifically, we describe our indoor test track and the way it is traversed by the hexapod crawling robot. Finally, we discuss the results of our experiments regarding quantitative and qualitative analysis in Section 5.3.

5.1 Experimental Platform

The used robot is an electrically actuated low-cost hexapod crawling robot depicted in Figure 9a. It features six legs, each with three joints (formed by Dynamixel AX12A servomotors) attached to the trunk. The trunk hosts the electronics and sensory equipment. In the default configuration, the robot dimensions are approx. $45 \times 40 \times 25$ cm.



(a) (b)

Figure 9: (a) The used hexapod crawling robot and (b) a scheme of its adaptive motion gait, courtesy of [59].

The adaptive motion gait [59] is utilized for locomotion over the rough terrains. It uses the estimation of the ground-reaction forces based on position data provided by the joint actuators, which allows the robot to negotiate individual footsteps and traverse irregular terrain. The adaptive motion gait is independent of exteroceptive sensors. The basic premise of the adaptive motion gait is to distribute the weight of the robot evenly among its legs, which increases the robot's stability.

The adaptive motion gait operates as follows. During the locomotion, individual legs alternate in swing phase, reaching new footholds, and stance phase, supporting the body. In [59], the swing phase is further divided into 3 individual parts with the ground contact detection only in the swing down phase of the motion as it is visualized in Figure 9b. When the swinging leg touches the ground, the ground reaction force causes an increase in the load applied to the individual leg joints. This allows the hexapod robot to identify the ground level using a position error threshold. After the foothold position is acquired, the robot's body is moved into an equilibrium position, i.e., a stable position from which the robot can easily move in any direction. This part is called body leveling and it is a part of the stance phase. As the equlibrium position is always given by the new footholds, it makes the whole robot thrust forward.

The RGB-D ASUS Xtion Pro Live camera has been utilized for the terrain perception and hall-effect current sensor for estimation of the robot instantaneous power consumption. On the robot the camera is mounted on its frontal part approx. 17 cm above the ground level. The camera provides the data with 30 Hz, and the power consumption data are provided with 62 Hz. The RGB-D data are further processed by the Robot Operating System (ROS) [60] nodes to extract the individual RGB and Depth images and the trajectory using the ORB-SLAM [50].

5.2 Experimental Setups

The experimental data have been obtained on our indoor laboratory test-track. The laboratory test-track is used to create several artificial terrains. The individual terrain types differ in roughness, color, and foothold visibility. Specifically, each of our test-track terrains is a three-meter length path over a different surface. The terrains with the increasing presumed difficulty of traversing are (with short identifiers in brackets): PVC flooring *(flat)*, turf-like carpet *(grass)*, and semi-transparent soft black fabric *(black)* represent different flat terrains. In rough terrain scenarios we use turf-like carpet covered wooden blocks *(grass rough)*, black fabric covered wooden blocks *(black rough)*, and bare wooden blocks *(blocks)*. The wooden blocks are 10 × 10 cm large with variable height and top slope. The three rough terrain setups are shown in Figure 10. Note that the different flat and rough terrains have been covered by the same artificial turf and black fabric. This selection of the terrain types allowed us to simulate a rough and flat surfaces with various levels of occlusion. The artificial turf occluded the wooden bricks fully, including the shape of the covered terrain. The black fabric occluded the terrain only partially, with the bricks being visible at certain locations, and sticking to the shape of the terrain somewhat better.

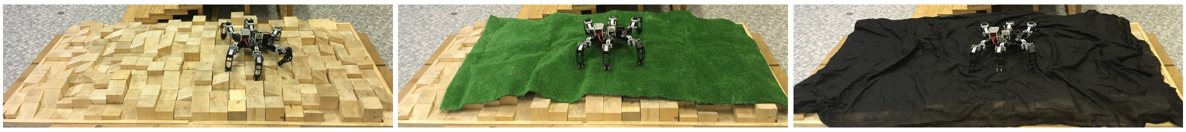


Figure 10: Rough terrains in laboratory trials.

We guide the robot remotely over the terrains during each of our experiments, while we collect visual and power consumption data. The incremental localization technique [50] is then utilized to process the visual data to obtain the robot pose that is utilized to estimate the robot velocity, and thus calculation of CoT according to (2). Besides, the ground-based data, simulated aerial scans have been collected for the individual terrain types using an elevated camera to provide a simulation of a UAV overfly over the whole test track. Note that the same camera was used in both ground and aerial scans. Generally, at least two ground walks and one environment scan have been captured for each of the terrains.

In each of the experiments, we have merged the trials over the individual terrains into a single dataset. Typically, for each terrain, it is repeated two times and data from the one trial is utilized for learning and the second trial for testing. After merging, the proposed framework in a given setup is used according to the description presented in Section 4.2. In the non-increnemental setups, the algorithm learns on the first half of the terrain sections, whereas the CoT estimate for the following sections is inferred from the rest. In the incremental setups, the full dataset is utilized for both learning and inference.

We test four main experimental setups: *walk-to-flight test*, *walk-to-walk test*, *flight-to-flight test*, and *exploration test*. The individual tests are designed to evaluate the performance of the system when concerning a certain situation.

In the *walk-to-flight setup*, the model learns on data from robot walk through the environment, but CoT is inferred from the data recovered from the environment scans. This setup simulates the situation when a robot "samples" an area by learning properties of the traversed terrain on a smaller patch of the land and passes this information to the UAV that explores and evaluates the whole area. For this setup, a feature has to be invariant to viewpoint changes, i.e., the setup is designed to test for feature viewpoint invariance. In the other setups, the learning and testing is performed using different robot walks, i.e., different walk scans and trajectories are used in learinng and testing to avoid spurious results. This is not the case in the walk-to-flight setup. The goal of the walk-to-flight setup is not to test the general performance of the inference system, but rather to test the invariance of the features to viewpoint changes. Note that while the point clouds utilized in the inference part of this setup are based on the environment scans, the respective trajectory and ground truth CoT are based on the robot walks, as no such information exists for the environment scans. The same goes for all environment scan setups.

The *walk-to-walk* and *flight-to-flight setups* are both designed to evaluate performance of the system when simply traversing various terrains. The two setups differ in the utilized data. The walk setup uses the walk scan point clouds, while the flight setup uses the environment scans. However, both setups use the same walk-scan based trajectories and traversability ground truths. Different trajectories and scans are used in learning and testing to avoid spurious results. Note that unlike the environment scan utilizing setups, the walk setup can be used to test the performance of the appearance-based features, as the location-specific images are available.

Finally, the *unknown terrain exploration setup* is designed to evaluate the system's performance when encountering unknown terrain. The scenario is specific in that different terrains are used for learning and testing. Specifically, the flat, grass, black, and cubes terrains are used for learning, and grass rough, black rough, and flat are used for testing. Otherwise, the setup is similar to the walk or flight setups.

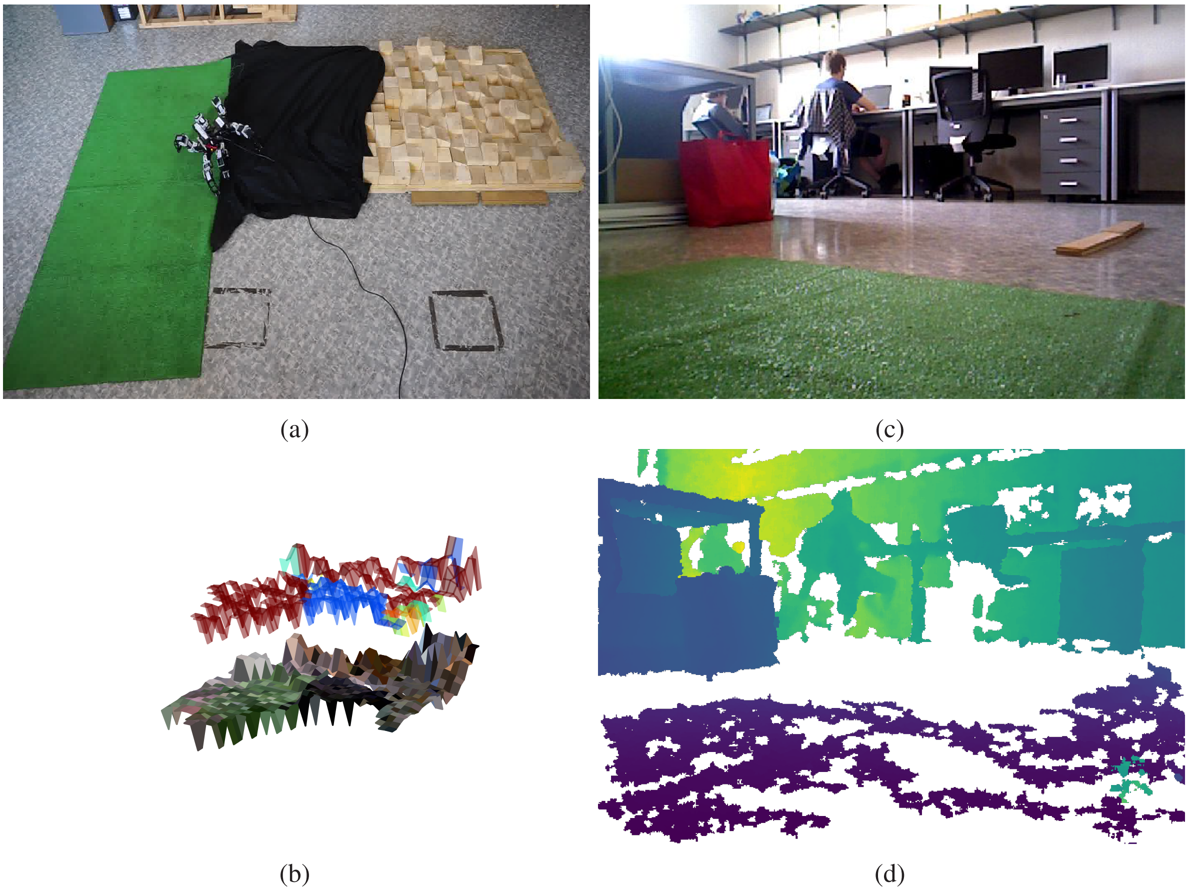
5.3 Experiment Results

This section discusses the achieved results and it is divided based on the setups presented in the previous section, i.e., *walk-to-flight*, *walk-to-walk*, *flight-to-flight*, and *unknown terrain exploration setup*. For each of these setups, we compare the selected feature sets and learning approaches using quantitative and qualitative analysis according to the description presented in Section 3. Note that not all possible combinations of the selected features sets and approaches are explored. We combine geometry based features with color/appearance describing features in such a way that the feature combinations are lightweight and easy to compute, a typical combination feature is comprised of 6–7 double precision floating point numbers.

5.3.1 Walk-to-Flight Viewpoint Invariance Testing

In Table 2, the quantitative measures, i.e, the mean absolute error between the estimated and ground-truth CoT, and its variance, are reported. In Figures 12 and 19, we present the model responses in the individual walk-to-flight setups. The Hoeffding trees and SVR learning algorithms have not performed well in the preliminary analysis of the walk-to-flight scenario, thus shifting the focus of the evaluation on the IGMN and Regression trees.

As this setup utilizes the environment scan data, it was not possible to use image-based features. Therefore we test standalone point cloud color and geometric features, and combinations thereof. In Figure 11, we present the representations of the terrain traversed by our hexapod robot in a walk-to-flight scenario with multiple terrains.



(a)

(c)

(b)

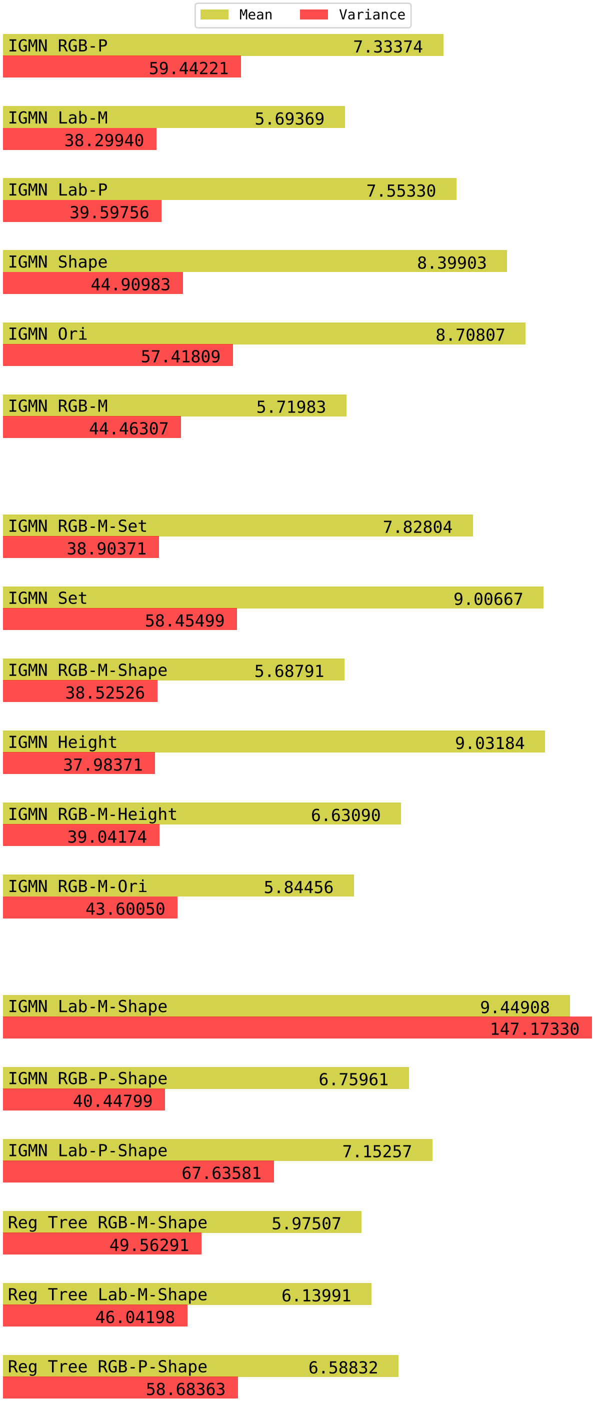
(d)

Figure 11: (a) External overview, (b) CoT and Terrain Visualization (Walk scan, Lab-M-Shape Feature), (c) robots's RGB and (d) depth view in a walk-to-flight-like scenario with multiple indoor terrains.

Only a limited difference between the CoT values for the flat, grass and grass rough terrains has been observed. However, this is an expected behavior. We are interested in the CoT estimation rather than the terrain classification. The cost similarity for the flat and grass terrains is not surprising, as the grass track is basically the flat track covered by artificial turf carpet. The grass rough cost is somewhat more surprising, as covering the cubes by the artificial turf leads to the existence of areas where turf is not lying directly on the cubes, and thus it is not directly supported by the firm ground. It appears that the artificial turf is firm enough to support the robot and prevent its legs from sinking down to the cube level.

Table 2: Walk-to-flight scenario statistics

Learning	Feature	Mean	Var
Hoeffding	RGB-M Shape	7.460	69.869
IGMN	Height	9.032	37.984
IGMN	Lab-M	5.694	38.299
IGMN	Lab-M Shape	9.449	147.173
IGMN	Lab-P	7.533	39.598
IGMN	Lab-P Shape	7.153	67.636
IGMN	Ori	8.708	57.418
IGMN	RGB-M	5.720	44.463
IGMN	RGB-M Height	6.631	39.042
IGMN	RGB-M Ori	5.845	43.600
IGMN	RGB-M Set	7.828	38.904
IGMN	RGB-M Shape	5.688	38.525
IGMN	RGB-P	7.334	59.442
IGMN	RGB-P Shape	6.760	40.448
IGMN	Set	9.007	58.455
IGMN	Shape	8.399	44.910
Reg Tree	Height	7.303	58.589
Reg Tree	Lab-M	6.968	66.616
Reg Tree	Lab-M Shape	6.140	46.042
Reg Tree	Lab-P	6.417	37.860
Reg Tree	Lab-P Shape	7.075	70.896
Reg Tree	Ori	8.135	60.325
Reg Tree	RGB-M	6.352	48.358
Reg Tree	RGB-M Height	6.352	48.358
Reg Tree	RGB-M Ori	5.833	47.513
Reg Tree	RGB-M Set	6.864	63.257
Reg Tree	RGB-M Shape	5.975	49.563
Reg Tree	RGB-P	6.227	41.483
Reg Tree	RGB-P Shape	6.588	58.684
Reg Tree	Set	7.945	59.022
Reg Tree	Shape	6.938	77.151
SVR	RGB-M Shape	6.757	76.273



Further, we observed several other traits of our terrain datasets. The CoT value over the wooden blocks is less uniform. The low-cost areas are comparable to flat or grass datasets but it also contains high-cost peaks. This is not surprising, as some areas form elevated plateaus, which can be traversed relatively easily. Other areas are rugged with a considerable height difference between the individual flat areas.

Finally, the black, and black rough datasets are the most costly with the high-cost peaks, which makes the black the most costly flat setup. The black rough dataset is the most costly terrain in total, with the highest CoT peaks and the highest non-peak areas. Presumably, the robot is unable to find proper grip on the fabric covered terrain. Moreover, unlike the artificial turf, the black fabric is not firm enough to support the robot.

The RGB-M-Shape feature, i.e., the combination of the RGB-M point cloud color feature and the point cloud geometric shape feature, and the standalone point cloud Lab-M color feature, perform best quantitatively. Since there are four distinguishable colors on the terrains, a good performance (i.e., low absolute error mean and variance) of the standalone Lab-M feature is not surprising. Although there are two similarly colored pairs with a different terrain shape, the pairs had surprisingly similar CoT, allowing the standalone color-based features to distinguish CoT quite well. The standalone color features learned by the IGMN are presented in Figure 12a.

From the qualitative analysis in Figures 12b and 12c, we conclude that the sole geometric features perform badly with both the incremental and non-incremental models. The color and shape feature combination is able, in our opinion, to better cope with the CoT peaks and valleys. Therefore, it provides the best results. Moreover, the combination does not exhibit overfitting as the model does not exhibit any particularly strong reaction on the CoT spikes. We present various color-geometric combination features in Figures 12d and 12e.

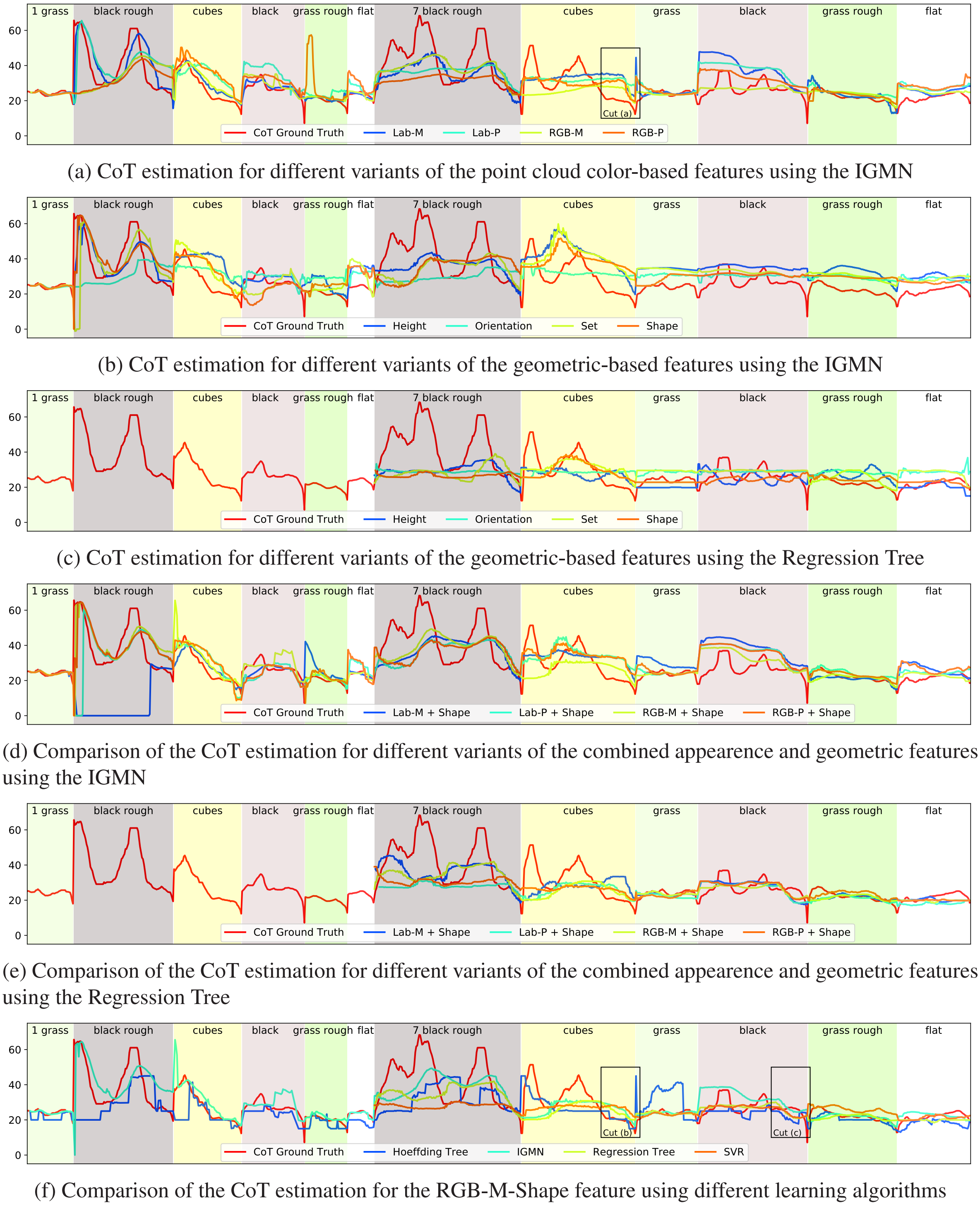
Results of a comparison of different models favor the IGMN setup in both quantitative and qualitative measures. IGMN's incremental learning property allows it to quickly adapt to the CoT changes. In Figure 12f, we present all four models learning the RGB-M-Shape feature. The incremental approaches do not overfit the current terrain and do remember previously visited areas. See Figure 13a, where it is possible to induce this from the existence of inferred CoT jumps when a new terrain is visited. Moreover, in this particular setup, the incremental model does not only learn from the walk-scan, but also from the second aerial environment scan.

Another property of our datasets presented in Figures 13b and 13c is that at the far end of each of the examined terrain, a small flat area is often traversed by the robot. This area can be interpreted as a terrain type border. We conclude that the most of the well-performing setups are capable of reacting on this terrain type border and estimate a lower CoT value in that region.

5.3.2 Walk-to-Walk

The results of the walk-to-walk scenario, i.e., a scenario when the framework learns from a walk scan of a particular trajectory and is tested using a walk scan of another trajectory, are similar to the results of the walk-to-flight setup, although some interesting differences exist. Table 3 presents an integral portion of the quantitative results of the walk-to-walk and Table 6 presents the full exhaustive test results of the setup. In Figure 14 and Figure 20, the framework responses in individual runs are presented.

An additional trajectory covering the same set of terrains has been utilized in this and following scenarios. However, although the new trajectory covers the same set of terrains, it is still slightly different, and some difference in the captured CoT ground truth has been observed. Most notably, the robot has



(a) CoT estimation for different variants of the point cloud color-based features using the IGMN

(b) CoT estimation for different variants of the geometric-based features using the IGMN

(c) CoT estimation for different variants of the geometric-based features using the Regression Tree

(d) Comparison of the CoT estimation for different variants of the combined appearence and geometric features using the IGMN

(e) Comparison of the CoT estimation for different variants of the combined appearence and geometric features using the Regression Tree

(f) Comparison of the CoT estimation for the RGB-M-Shape feature using different learning algorithms

Figure 12: Learned and predicted values of CoT for a different combination of features and learning algorithms in the walk-to-flight setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.

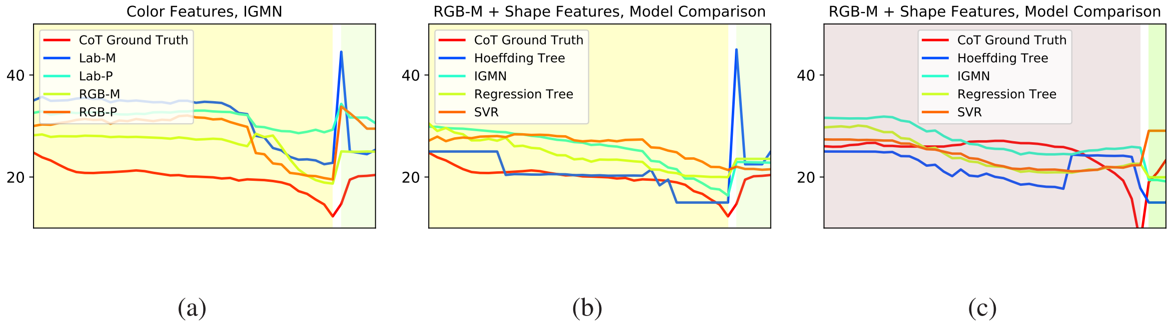


Figure 13: (a) Jump in inferred CoT when reaching a new area, the IGMN for Color Feautures, (b) and (c) Terrain change to flat CoT drops, RGB-M-Shape features.

been able to traverse the black covered flat ground smoothly, thus making the overall CoT in the area lower. Similarly, in the cubes terrain, trajectory partially leads over a plateau on top of the wooden bricks which makes CoT lower.

The IGMN learned combination of the point cloud Lab-M and Orientation features is the quantitatively best-performing feature in the walk-to-walk setup. This is somewhat similar to the results observed in the walk-to-flight setup, where another combination feature, the RGB-M-Shape feature has performed the best. Other features notable for their performance in the walk-to-walk setup are the sole Lab-M feature, which performs quite well under multiple learning approaches despite being rather limited, and the sole orientation feature, which performs well combined with color features, but terribly when it is a lone feature.

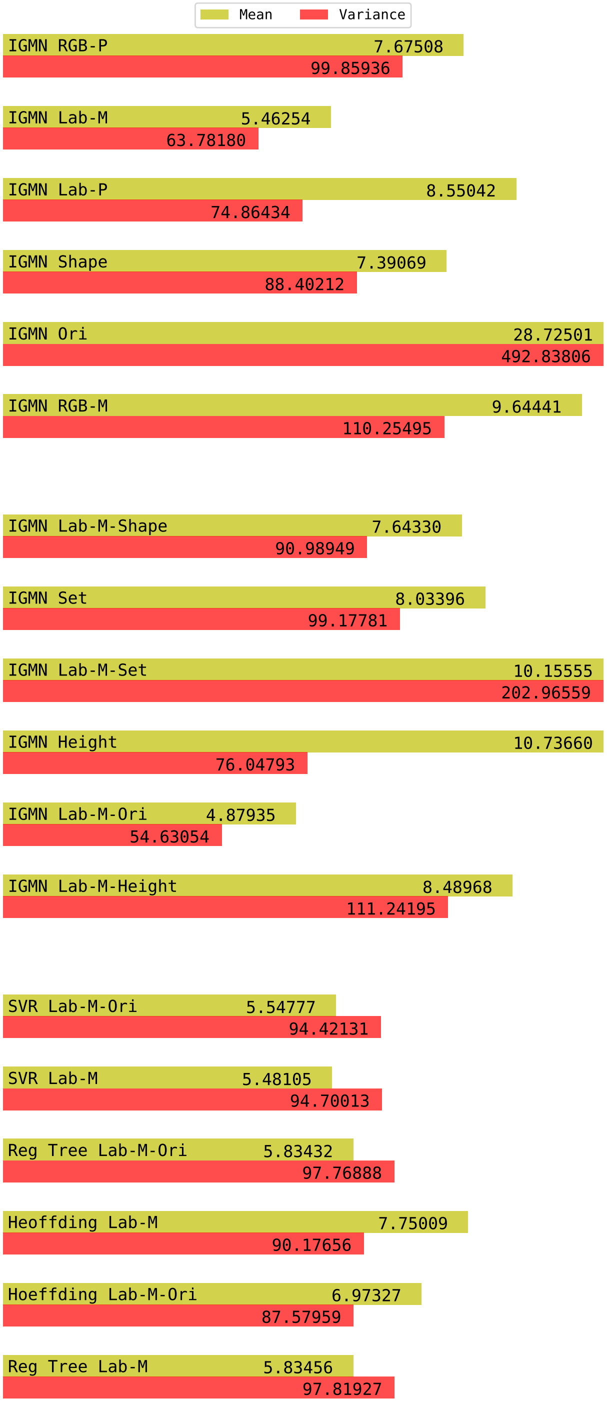
When compared with the other IGMN learned color-geometric combination feature, the Lab-M-Orientation feature is the best performing features, but does not differ drastically. The Lab-M feature correctly identifies areas of relatively low CoT, including the black fabric covered flat terrain and bare cubes. Other Lab-M-Geometric feature combinations, such as those presented in Figure 14a, have problems identifying some areas. The Lab-M-Shape combination feature drastically overestimates CoT for the black flat terrain. The Lab-M-Set and Lab-M-Height features additionally overestimate some turf covered and flat areas. On the other hand, the Lab-M-Orientation feature appears to have only a limited reaction on terrain changes. In Figure 15, a detail of the CoT estimation around the border area is presented. Other features, such as the Lab-M-Set and Lab-M-Shape, react on low-CoT areas that are often present at the ends of the individual terrain trajectories by estimating the relatively low CoT of the flat area, while the Lab-M-Orientation feature usually assigns the CoT values of the more expensive terrain.

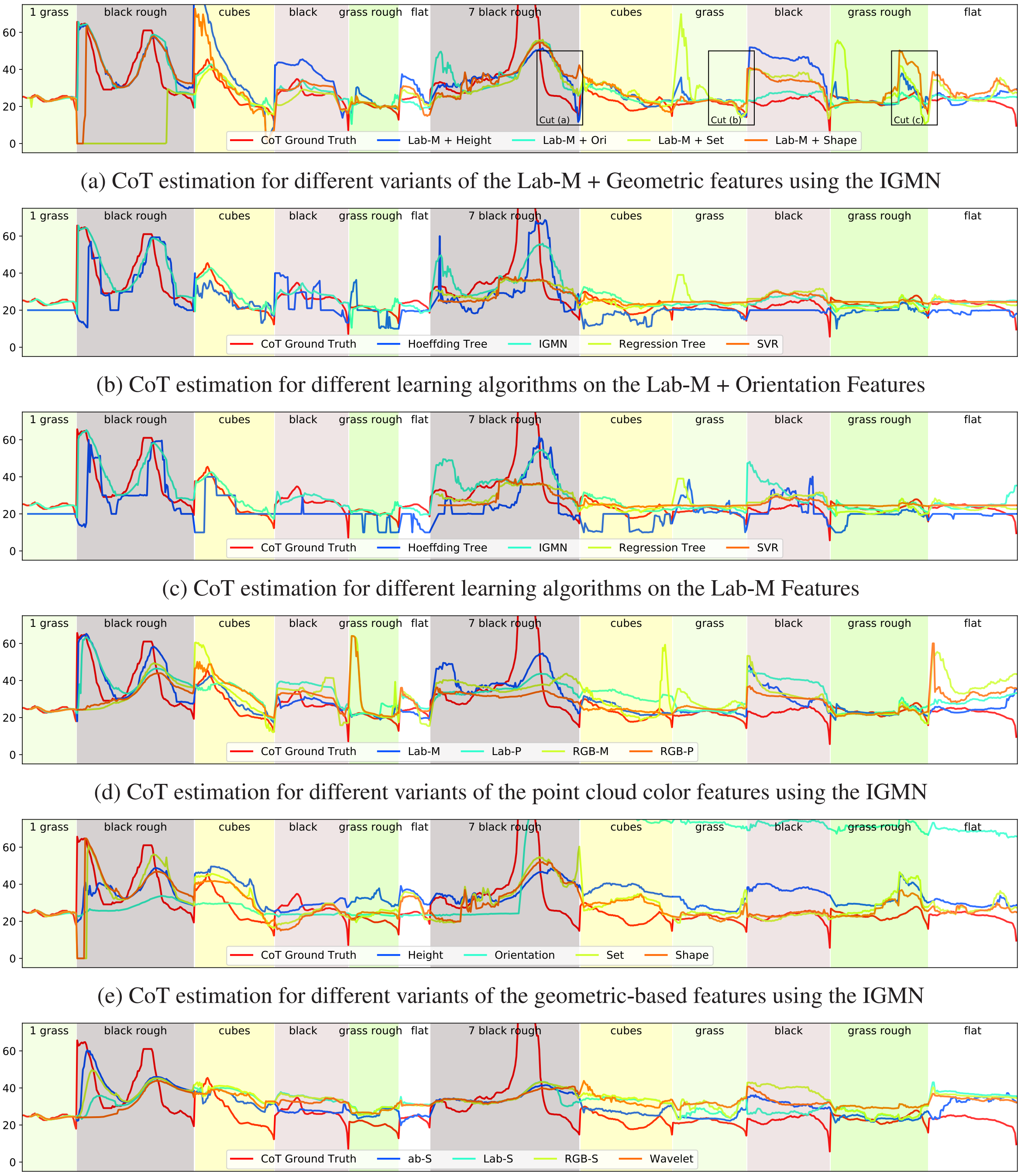
The Lab-M-Orientation feature performs well with most of the learning algorithms. In Figure 14b, we present the Lab-M-Orientation feature used under all four learning approaches. All the models evaluate the individual terrains correctly, recognizing high- and low-cost areas. The IGMN and the Hoeffding tree react on the the large high-CoT area of black rough terrain somewhat better than the regression tree and SVR. This is not surprising given the IGMN and Hoeffding tree are the two incremental approaches.

The second best performing feature is the Lab-M feature. This simple point cloud color feature performs well when used with the IGMN, SVR, and regression tree. In Figure 14c, the feature is presented with all four of our models. The feature behaves as one would expect a color based feature to behave, i.e., it overestimates the black flat area, as it is hard to distinguish it from the black rough based only on color information. This behavior is especially prevalent for incremental learning, as its estimation is influenced by the expensive black rough area on the second trajectory. Moreover, such a behavior

Table 3: Walk-to-walk scenario statistics

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.750	90.177
Hoeffding	Lab-M Ori	6.973	87.580
IGMN	ab-S	7.527	73.985
IGMN	Height	10.737	76.048
IGMN	Lab-M	5.463	63.782
IGMN	Lab-M Height	8.490	111.242
IGMN	Lab-M Ori	4.879	54.631
IGMN	Lab-M Set	10.156	202.966
IGMN	Lab-M Shape	7.643	90.989
IGMN	Lab-P	8.550	74.864
IGMN	Lab-S	9.265	77.643
IGMN	Ori	28.725	492.838
IGMN	RGB-M	9.644	110.255
IGMN	RGB-M Ori	7.550	93.046
IGMN	RGB-P	7.675	99.859
IGMN	RGB-P Ori	6.888	86.462
IGMN	RGB-S	10.644	73.111
IGMN	Set	8.034	99.178
IGMN	Shape	7.391	88.402
IGMN	Wave	10.201	75.290
Reg Tree	Height	8.287	108.883
Reg Tree	Lab-M	5.835	97.819
Reg Tree	Lab-M Height	6.852	87.882
Reg Tree	Lab-M Ori	5.834	97.769
Reg Tree	Lab-M Set	6.438	91.429
Reg Tree	Lab-M Shape	5.925	81.897
Reg Tree	Lab-P	7.106	106.065
Reg Tree	Lab-S	8.552	98.232
Reg Tree	Ori	9.455	95.993
Reg Tree	RGB-M	5.508	90.968
Reg Tree	RGB-P	7.643	107.963
Reg Tree	RGB-S	10.679	91.996
Reg Tree	Set	6.591	79.512
Reg Tree	Shape	6.407	94.094
SVR	Height	10.889	155.765
SVR	Lab-M	5.481	94.700
SVR	Lab-M Ori	5.548	94.421
SVR	Lab-M Shape	7.455	120.396
SVR	Lab-P	6.155	120.056
SVR	Ori	9.062	103.553
SVR	RGB-M	6.602	119.747
SVR	RGB-P	7.007	109.752
SVR	Set	6.986	133.726
SVR	Shape	6.245	95.858





(a) CoT estimation for different variants of the Lab-M + Geometric features using the IGMN

(b) CoT estimation for different learning algorithms on the Lab-M + Orientation Features

(c) CoT estimation for different learning algorithms on the Lab-M Features

(d) CoT estimation for different variants of the point cloud color features using the IGMN

(e) CoT estimation for different variants of the geometric-based features using the IGMN

(f) CoT estimation for different variants of the image-based features using the IGMN

Figure 14: Learned and predicted values of CoT for a different combination of features and learning algorithms in the walk-to-walk setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.

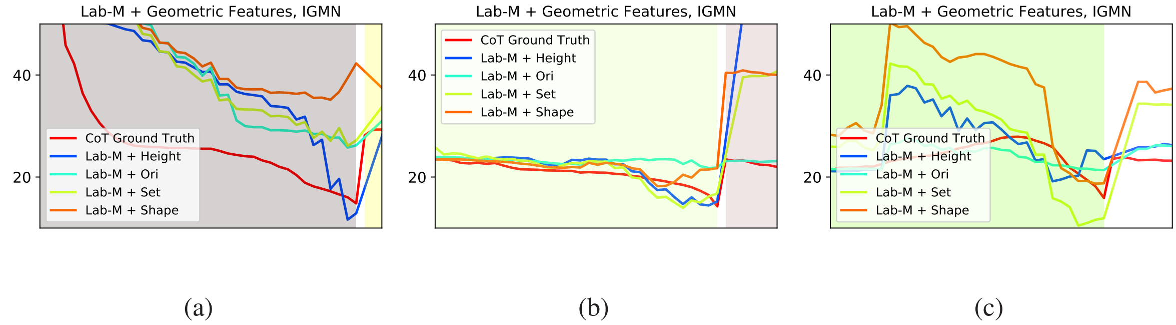


Figure 15: Walk-to-walk terrain change details.

is not specific for the Lab-M feature, but rather is a property of the utilized color-based features in general. In Figure 14d, we present the point cloud features under the IGMN learning approach.

The behavior of the sole orientation feature under the IGMN learning is rather peculiar. In general, geometric features, such as those presented in Figure 14e, are capable of discriminating between flat and rough terrains. However, the sole Orientation feature, which is a part of the best performing Lab-M-orientation feature, overestimates CoT for most areas and assigns values normally found only in the most extreme black rough terrain. Although the exact reason for this behavior is not known, we presume that this is caused by the incremental learning in the IGMN approach. More specifically, the model probably receives a "shock" from the CoT spikes on the black rough terrain.

Finally, in the walk-to-walk setup, we also test our image features, i.e., the Wavelet feature and the Image segmentation features. The results are presented in Figure 14f. The behavior of the image features is similar to the point cloud color features, although the image features also considerably overestimate CoT for flat areas. Moreover, the image features do not perform particularly well for the terrain border areas.

5.3.3 Flight-to-Flight

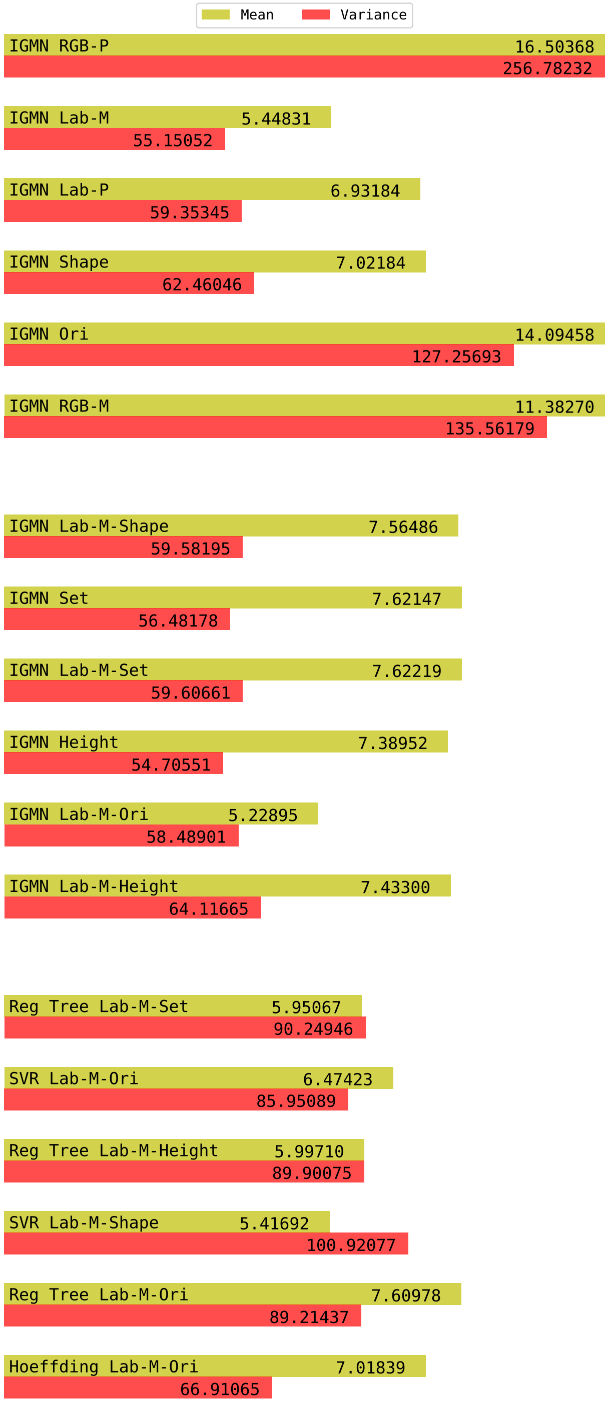
The flight-to-flight scenario is a scenario where the robot both learns and estimates using one environment scan for each terrain, albeit different trajectories and associated CoT time series are used in the learning and inference phases. Tables 4 and 7 present the selection from and the full statistics of the flight-to-flight scenario testing. In Figures 16 and 21 the individual runs are shown.

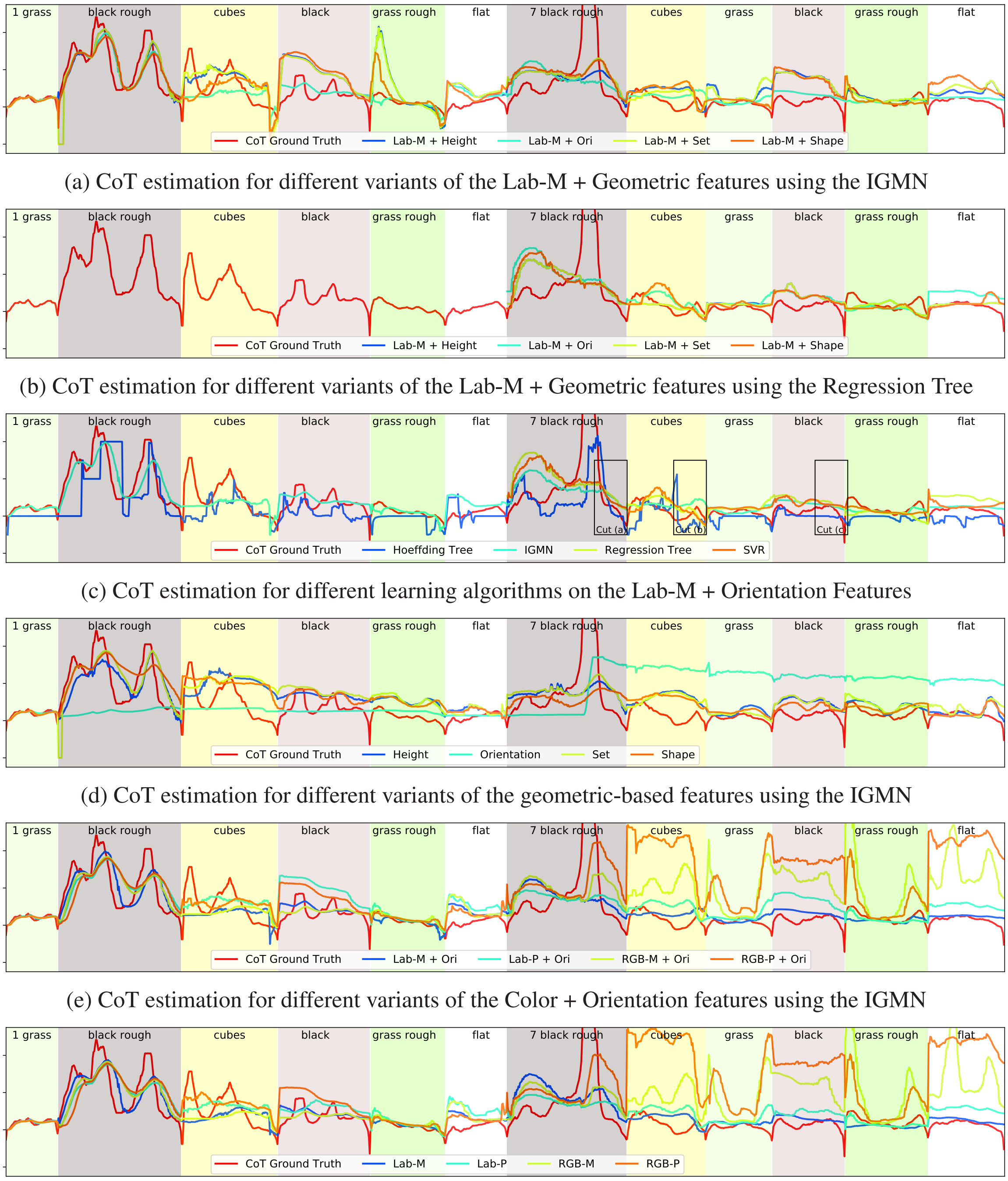
Similarly to the walk-to-walk setup, the IGMN learned Lab-M-Orientation feature is the best performer in the flight-to-flight setup. The Lab-M based features dominated the setup in general, i.e., the sole Lab-M, Lab-M-shape, and Lab-M-set features performed well under some of the learning approaches.

The IGMN Lab-M-Orientation feature behaves quite predictably, as its behavior is similar to the walk-to-walk setup. It is able to discriminate between high-cost and low-cost terrains. The black flat and cube terrains are slightly overestimated, although much less than for the other color-geometric feature combinations, such as those presented in Figure 16a. On the other hand, some of the feature combinations perform decently when paired with some of the nonincremental approaches. Specifically, the Lab-M-Shape features perform well under the SVR, while the Lab-M-Set and Lab-M-Height features do better under the regression tree. The performance of various Lab-M-Geometric features under the regression tree learning approach is presented in Figure 16b. Note that the Lab-M-Orientation feature performs decently under all of the approaches, as it is presented in Figure 16c. Although the Lab-M-Orientation feature is the best performer regarding quantitative analysis and appears to discriminate

Table 4: Flight-to-flight scenario statistics

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.189	76.481
Hoeffding	Lab-M Ori	7.018	66.911
IGMN	Height	7.390	54.706
IGMN	Lab-M	5.448	55.151
IGMN	Lab-M Height	7.433	64.117
IGMN	Lab-M Ori	5.229	58.489
IGMN	Lab-M Set	7.622	59.607
IGMN	Lab-M Shape	7.565	59.582
IGMN	Lab-P	6.932	59.353
IGMN	Lab-P Ori	7.768	63.331
IGMN	Lab-P Shape	7.677	58.118
IGMN	Ori	14.095	127.257
IGMN	RGB-M	11.383	135.562
IGMN	RGB-M Ori	11.622	137.297
IGMN	RGB-M Shape	7.264	61.659
IGMN	RGB-P	16.504	256.782
IGMN	RGB-P Ori	15.641	233.964
IGMN	RGB-P Shape	7.348	57.839
IGMN	Set	7.621	56.482
IGMN	Shape	7.022	62.460
Reg Tree	Lab-M	7.610	89.214
Reg Tree	Lab-M Height	5.997	89.901
Reg Tree	Lab-M Ori	7.610	89.214
Reg Tree	Lab-M Set	5.951	90.249
Reg Tree	Lab-M Shape	6.732	95.003
Reg Tree	Lab-P	7.195	85.103
Reg Tree	RGB-M	6.022	92.347
Reg Tree	RGB-P	7.068	88.858
SVR	Height	7.786	131.404
SVR	Lab-M	6.393	86.490
SVR	Lab-M Ori	6.474	85.951
SVR	Lab-M Shape	5.417	100.921
SVR	Lab-P	5.865	98.150
SVR	Lab-P Ori	6.986	88.912
SVR	Lab-P Shape	6.189	93.494
SVR	Ori	7.930	94.635
SVR	RGB-M	6.198	97.596
SVR	RGB-M Ori	6.369	100.920
SVR	RGB-M Shape	6.360	96.901
SVR	RGB-P	6.917	94.211
SVR	RGB-P Ori	6.892	94.106
SVR	RGB-P Shape	6.274	96.316
SVR	Set	7.938	110.359
SVR	Shape	6.110	104.596





(a) CoT estimation for different variants of the Lab-M + Geometric features using the IGMN

(b) CoT estimation for different variants of the Lab-M + Geometric features using the Regression Tree

(c) CoT estimation for different learning algorithms on the Lab-M + Orientation Features

(d) CoT estimation for different variants of the geometric-based features using the IGMN

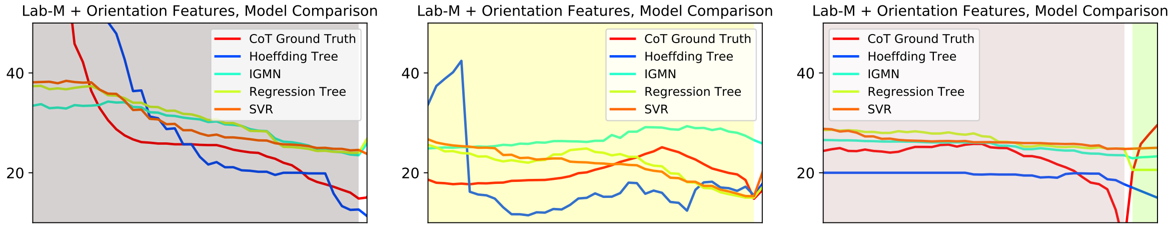
(e) CoT estimation for different variants of the Color + Orientation features using the IGMN

(f) CoT estimation for different variants of the point cloud color features using the IGMN

Figure 16: Learned and predicted CoT values for a different combination of the features and learning algorithms in the flight-to-flight setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.

terrains quite well, it suffers from the same problems as we have observed in the walk-to-walk setup, i.e., it exhibits only a limited reaction to partial terrain changes. Figure 17 presents a detail of the behavior for the border areas.

Again, the behavior of the sole geometric features does not differ drastically between the walk-to-walk and flight-to-flight scenarios. The sole geometric features, such those under the IGMN model which are presented in Figure 16d, perform decently with the exception of the IGMN learned orientation feature, which again drastically overestimates CoT for all areas. However, unlike in the walk-to-walk scenario, this behavior influences the composite features, when certain color based features are utilized. In Figure 16e, we present the Color-Geometric combination features under the IGMN. The likely reason why this behavior is propagated to the combination features is that the behavior is not prevalent only in the sole geometric features but also in the some of the sole point cloud color features. Specifically, the RGB based features appear to suffer from this problem, as can be observed in Figure 16f.



(a) (b) (c)

Figure 17: Flight-to-flight terrain change details.

5.3.4 Unknown Terrain Exploration

In the unknown terrain exploration scenario, the framework learns from four basic terrains (cubes, flat, grass flat, and black flat) and is tested on the combined terrains (black rough and grass rough), and flat terrain, which is added to test framework's ability to retain the information learned several terrains back. The setup uses both walk and environment scans with either walk- or environment-scan point cloud, i.e., the point clouds are used in the same manner as in the walk-to-walk or flight-to-flight scenario, respectively. The overall quantitative statistics are presented in Tables 5, 8, and 9. In Figures 18, 22, and 23, we present the model responses in individual the unknown terrain exploration setups.

Unlike in the simple walk-to-walk and flight-to-flight setups, the best performing feature is the IGMN learned RGB-M-Shape feature under the flight-to-flight-exploration setup. However, the Lab-M-Geometric features that have proven successful in other setups also perform decently. Namely, under the flight-to-flight-exploration setup, the sole Lab-M, Lab-M-Orientation, Lab-M-Set, and also the Lab-M-Height performed decently. Under the walk-to-walk-exploration scenario, the performance is notably worse with the sole Lab-M feature being the best. All the mentioned features were learned using the IGMN approach.

The performance of the RGB-M-Shape feature is surprising. In the walk-to-walk and flight-to-flight setups, the Lab based features performed better, presumably because some lighting changes may have been present during capturing the dataset. Presumably, in the exploration setup, somewhat lower variance of the terrains and trajectories allows the RGB based features to perform quite decently. A similar phenomenon can be observed in the walk-to-flight scenario, where each terrain is represented

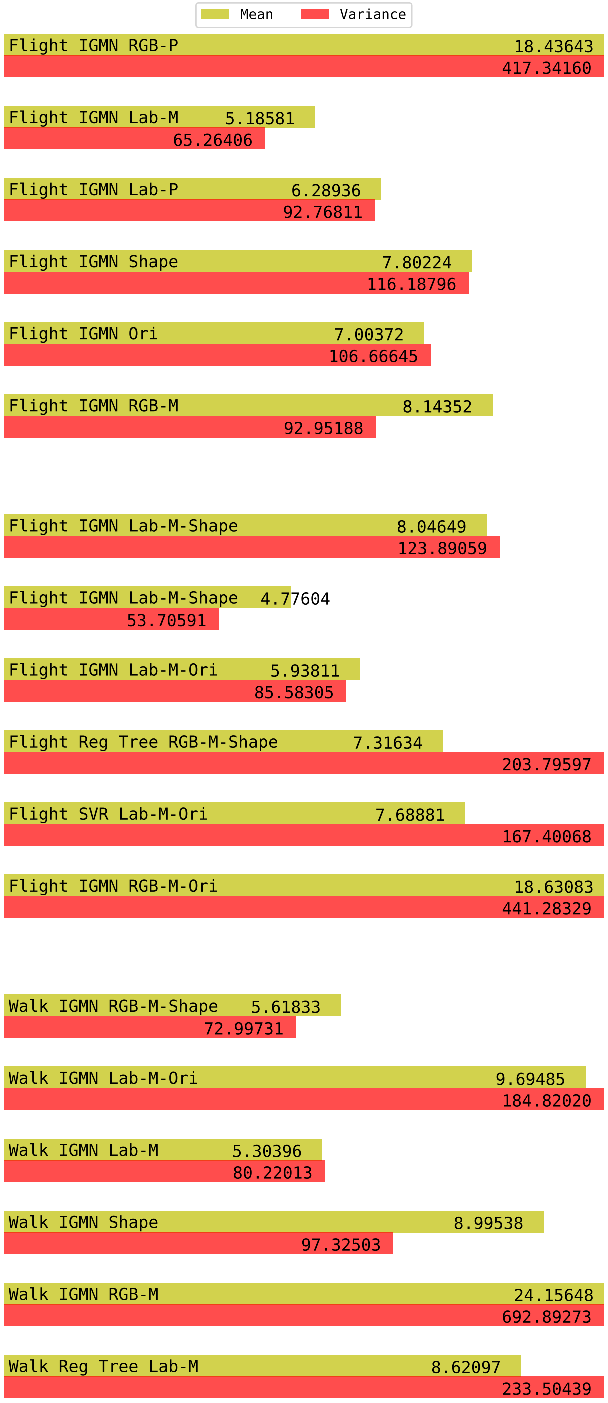
Table 5: Exploration scenario statistics

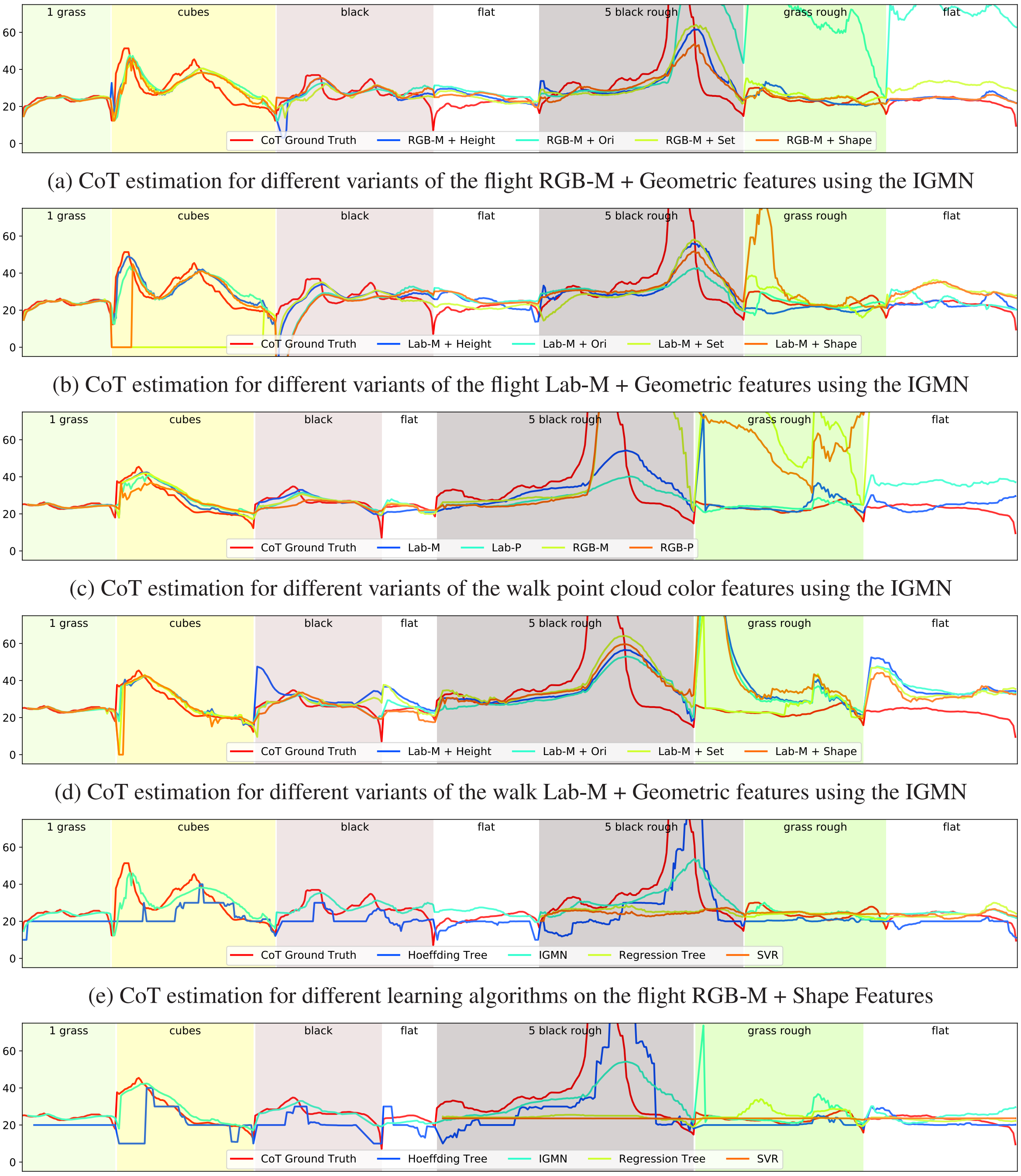
Flight-to-flight

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.416	67.174
IGMN	Height	6.084	80.776
IGMN	Lab-M	5.186	65.264
IGMN	Lab-M Height	5.413	56.304
IGMN	Lab-M Ori	5.938	85.583
IGMN	Lab-M Set	9.821	149.407
IGMN	Lab-M Shape	8.046	123.891
IGMN	Lab-P	6.289	92.768
IGMN	Ori	7.004	106.666
IGMN	RGB-M	8.144	92.952
IGMN	RGB-M Ori	18.631	441.283
IGMN	RGB-M Shape	4.776	53.706
IGMN	RGB-P	18.436	417.342
IGMN	Set	5.763	56.175
IGMN	Shape	7.802	116.188
Reg Tree	Lab-M	7.690	167.393
Reg Tree	RGB-M Shape	7.316	203.796
SVR	Lab-M	7.616	205.117
SVR	Lab-M Shape	7.414	201.844
SVR	Lab-P	7.606	190.827

Walk-to-walk

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.692	105.646
Hoeffding	Lab-M Ori	7.800	93.814
IGMN	Height	10.156	92.699
IGMN	Lab-M	5.304	80.220
IGMN	Lab-M Height	10.294	181.473
IGMN	Lab-M Ori	9.695	184.820
IGMN	Lab-M Set	6.840	87.844
IGMN	Lab-M Shape	9.736	219.595
IGMN	Lab-P	7.475	106.035
IGMN	Ori	28.547	721.531
IGMN	RGB-M	24.156	692.893
IGMN	RGB-M Shape	5.618	72.997
IGMN	Set	12.044	136.896
IGMN	Shape	8.995	97.325
Reg Tree	Lab-M	8.621	233.504
Reg Tree	Lab-M Ori	8.621	233.517
Reg Tree	Ori	8.670	252.610
Reg Tree	RGB-M	8.815	233.840
Reg Tree	RGB-M Shape	8.933	228.813
SVR	Lab-M	8.557	252.507





(a) CoT estimation for different variants of the flight RGB-M + Geometric features using the IGMN

(b) CoT estimation for different variants of the flight Lab-M + Geometric features using the IGMN

(c) CoT estimation for different variants of the walk point cloud color features using the IGMN

(d) CoT estimation for different variants of the walk Lab-M + Geometric features using the IGMN

(e) CoT estimation for different learning algorithms on the flight RGB-M + Shape Features

(f) CoT estimation for different learning algorithms on the walk Lab-M color features

Figure 18: Learned and predicted values of CoT for a different combination of the features and learning algorithms in the exploration setup. The first four terrains (parts) represent the learning phase. The following three parts represent the inference phase.

by one specific trajectory. Figure 18a presents the various RGB-M-Geometric features. Note that the RGB-M-Orientation feature suffers from the same problem as some of the RGB-M and Orientation based features in the previous setups, i.e., after encountering a high-CoT area, it does estimate all further terrains as a high CoT. The Lab-M-Geometric features presented in Figure 18b, despite of a worse quantitative performance, perform quite decently and do not suffer from this problem. The Lab-M-Shape feature produces a high-CoT spike in the grass rough terrain area, but it is not of the same magnitude as the aforementioned problem.

Although the sole Lab-M feature perfomes the best in the walk-to-walk-exploration scenario quantitative analysis, a detailed examination of Figure 18c shows otherwise. Most of the color based features exhibited a considerable inaccuracy for several terrains. The Lab-M performs best quantitatively, albeit the Lab-P feature appears to be better, as it does not produce any considerable peak. The Lab-M-Geometric features, presented in Figure 18d, do not perform particularly well in the walk-to-walk-exploration setup. All variants produce high CoT spikes in areas where there is a low ground truth CoT.

The performance of the individual features somewhat differs between the walk-to-walk and flight-to-flight exploration setups, and the previous setups. However, the performance of the learning algorithms appears to be straightforward. Figure 18e and Figure 18f present model comparisons for the flight-to-flight-exploration RGB-M-Shape feature and walk-to-walk-exploration Lab-M feature, respectively. The non-incemental methods have not encountered many high-CoT areas in the learning stage, and thus are not capable of the proper reaction on such areas in the inference phase. On the other hand, the incremental approaches, are suitable for exploration of unknown terrains, as they can add new information on the go.

Chapter 6

Conclusion

In this work, we present a framework for regression of exteroceptively perceived terrain traversability assessment for a hexapod walking robot. The framework clusters terrain data based on terrain traversability assessment from visual and geometric features. It also supports incremental learning and can be used with datasets captured from multiple viewpoints. Based on the presented overview of the state of the art in the field of terrain traversability and terrain description, we have selected a traversability metric and a set of terrain description features. Namely, we have utilized the power consumption based Cost of Transport and various lightweight and easy to compute color and geometry based terrain description features. We compare these lightweight terrain description features and comment on their impact on the precision of the traversability assessment estimation.

Our hexapod walking robot captures an RGB-D dataset, which is further utilized to construct point cloud representation of the traversed terrain. The robot also collects proprioceptive data which is utilized in the CoT computation. The dataset consists of hexapod robot locomotion data on different terrains in an indoor laboratory scenario. Moreover, we simulate a UAV overflight over our test track and create addition point clouds captured from aerial viewpoints. These point clouds, combined with the RGB images from the robot, are further utilized by our framework to learn the terrain description features. Moreover, we also investigate the possibility of using the framework for traversability assessment estimation in observed, yet untraversed terrain.

We compare the selected combination of the approaches on several scenarios designed to test specific properties of the individual learning and feature combinations. In walk and flight tests, we simply test the performance of the individual combinations when traversing arbitrary terrains using the point clouds captured from the robot and aerial viewpoints, respectively. In the walk-to-flight test, we test the viewpoint invariance of the individual features, where the framework learns from the ground viewpoint data, and it is tested on the aerial viewpoint. Finally, in the exploration tests, we test the ability of the framework to deal with new terrains and the robot is tested on terrains that were not used in the learning process.

Our testing has revealed several interesting results. We conclude that the combined point cloud color-geometric features are the best performers. The individual color or geometric features may have performed decently in some setups, but appear to be much more susceptible to a reaction on high-CoT peaks and other overfitting problems. Additionally, frequency-based image wavelet features and image segmentation features were tested but did not perform particularly well.

Among the point cloud color-geometric combinations, we consider the combinations of the Lab color channel means feature with either the eigenvalue-based shape feature or normal-based orientation feature to be the best performers. Some variation of the Lab feature is among the best performing features in all of our setups, either alone or combined with one of the geometric features. Although in some of the setups it is outperformed by RGB based features, its overall performance is convincing. Moreover, the RGB based features perform well only in setups where there is limited possibilty of the ilumination change for individual terrains. The shape and orientation based features appear to be comparable in performance. The orientation based combination features perform better in the simple walk and flight scenarios, while the shape feature is better in the walk-to-flight and exploration setups and appears to be less prone to overfitting in general. The shape feature is four dimensional, while the orientation feature is only three dimensional, but requires a fixed coordinate frame orientation.

We observe that the terrain traversability estimation benefits greatly from incremental learning approaches. In all of our setups, the best performing feature has been learned using the IGMN approach. The benefit of the incremental learning is best observable in the exploration setup, where the non-incremental approaches are not capable of coping with new terrains, while the incremental methods simply add them on-the-go. Similarly, the multi-viewpoint setups benefits from incremental learning, because the possible differences between the individual representations can be mitigated on-the-go.

Our work is open to various extensions. For example, features more specific to terrain types that can be encountered could be developed. Moreover, for practical applications, it would be beneficial to extend the system with assigning terrain types with direction based traversability metric. On a more technical note, our work could be further utilized by creating a multi-robot system, where the robot traversing the terrain is accompanied by a UAV that explores the area around the robot and allows a terrain evaluation in areas that are occluded to the ground robot exteroceptive sensors.

References

[1] D. Brown and G. Webster, “Now a stationary research platform, nasa’s mars rover spirit starts a new chapter in red planet scientific studies,” *NASA Press Release*, 2010.

[2] J. Falconer, “Toshiba unveils four-legged nuclear plant inspection robot,” *Innovation Toronto*, 2012, [Accessed April 10, 2018]. [Online]. Available: http://www.innovationtoronto.com/2012/11/toshiba-unveils-four-legged-nuclear-plant-inspection-robot/

[3] A. Satariano and M. Lloyd, “One job the robots can have: Cleaning nuclear waste,” *Bloomberg*, 2017, [Accessed April 10, 2018]. [Online]. Available: https://www.bloomberg.com/news/features/2017-02-16/one-job-the-robots-can-have-cleaning-nuclear-waste

[4] D. Belter and P. Skrzypczyński, “Rough terrain mapping and classification for foothold selection in a walking robot,” *Journal of Field Robotics*, vol. 28, no. 4, pp. 497–528, 2011.

[5] B. H. Jun, H. Shim, B. Kim, J. Y. Park, H. Baek, S. Yoo, and P. M. Lee, “Development of seabed walking robot CR200,” in *OCEANS MTS/IEEE Bergen: The Challenges of the Northern Dimension*, 2013.

[6] A. Roennau, G. Heppner, M. Nowicki, and R. Dillmann, “LAURON V : A Versatile Six - Legged Walking Robot with Advanced Maneuverability,” in *AIM*, 2014, pp. 82–87.

[7] J. P. L. NASA, “Creepy crawlers may unravel web of mysteries planetary,” *JPL News Release*, 2002, [Accessed April 11, 2018]. [Online]. Available: https://www.jpl.nasa.gov/releases/2002/232.cfm

[8] M. Murphy, “Siemens is building a swarm of robot spiders to 3d-print objects together,” *Quartz*, 2016, [Accessed April 11, 2018]. [Online]. Available: https://qz.com/672708/siemens-is-building-a-swarm-of-robot-spiders-to-3d-print-objects-together/

[9] A. Stelzer, H. Hirschmüller, and M. Görner, “Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain,” *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 381–402, 2012.

[10] P. Papadakis, “Terrain traversability analysis methods for unmanned ground vehicles: A survey,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013.

[11] S. Boris, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, “Improving Robot Navigation Through Self-Supervised Online Learning,” *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1059–1075, 2006.

[12] J. Nishii, “An analytical estimation of the energy cost for legged locomotion,” *Journal of theoretical biology*, vol. 238, no. 3, pp. 636–645, 2006.

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

[14] P. Fankhauser, M. Bloesch, P. Krüsi, R. Diethelm, M. Wermelinger, T. Schneider, M. Dymczyk, M. Hutter, and R. Siegwart, "Collaborative navigation for flying and walking robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2859–2866.

[15] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, "Autonomous Terrain Classification with Co- and Self-Training Approach," *Robotics and Automation Letters*, pp. 1–6, 2016.

[16] W. Mou and A. Kleiner, "Online Learning Terrain Classification for Adaptive Velocity Control," in *SSRR*, 2010, pp. 1–7.

[17] S. Bartoszyk, P. Kasprzak, and D. Belter, "Terrain-aware motion planning for a walking robot," in *RoMoCo*, 2017, pp. 29–34.

[18] "Scene Understanding for a High-mobility Walking Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1144–1151.

[19] V. A. Tucker, "The Energetic Cost of Moving About: walking and running are extremely inefficient forms of locomotion. Much greater efficiency is achieved by birds, fish and bicyclists," *American Scientist*, vol. 63, no. 4, pp. 413–419, 1975.

[20] Y. Mulgaonkar, B. Araki, J. S. Koh, L. Guerrero-Bonilla, D. M. Aukes, A. Makineni, M. T. Tolley, D. Rus, R. J. Wood, and V. Kumar, "The flying monkey: A mesoscale robot that can run, fly, and grasp," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4672–4679.

[21] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1140–1154, 2016.

[22] R. M. Alexander, "Models and the scaling of energy costs for locomotion," *Journal of Experimental Biology*, vol. 208, no. 9, pp. 1645–1652, 2005.

[23] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, vol. 4, no. 11, pp. 1549–1560, 1995.

[24] X. Li and Z. Tian, "Wavelet Energy Signature: Comparison and Analysis," in *International Conference on Neural Information Processing*, 2006, pp. 474–480.

[25] S. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1160–1167, 2002.

[26] M. Kragh, R. N. Jørgensen, and H. Pedersen, "Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data," in *ICVS*, 2015, pp. 188–197.

[27] C. Wellington and A. Stentz, "Online Adaptive Rough-Terrain Navigation in Vegetation," in *IEEE International Conference on Robotics and Automation (ICRA)*, no. April, 2004, pp. 96–101.

[28] T. Homberger, M. Bjelonic, N. Kottege, and P. V. K. Borges, "Terrain-dependant Control of Hexapod Robots using Vision," in *International Symposium on Experimental Robotics*, 2016, pp. 92–102.

[29] D. M. Bradley, R. Unnikrishnan, and J. Bagnell, "Vegetation detection for driving in complex environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 503–508.

[30] C. Ünsalam, K. Boyer, and Anonymous, "Linearized vegetation indices based on a formal statistical framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1575–1585, 2004.

[31] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3212–3217.

[32] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2155–2162.

[33] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 2987–2992.

[34] B. Soleimani, M. H. Z. Ashtiani, B. H. Soleimani, and H. Moradi, "A disaster invariant feature for localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1096–1101.

[35] M. H. Bharati, J. J. Liu, and J. F. MacGregor, "Image texture analysis: Methods and comparisons," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, no. 1, pp. 57–71, 2004.

[36] S. Arivazhagan and L. Ganesan, "Texture segmentation using wavelet transform," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3197–3203, 2003.

[37] E. H. S. Lo, M. Pickering, M. Frater, and J. Arnold, "Scale and rotation invariant texture features from the dual-tree complex wavelet transform," *Image Processing*, pp. 227–230, 2004.

[38] R. Manthalkar, P. K. Biswas, and B. N. Chatterji, "Rotation and scale invariant texture features using discrete wavelet packet transform," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2455–2462, 2003.

[39] K. Muneeswaran, L. Ganesan, S. Arumugam, and K. R. Soundar, "Texture classification with combined rotation and scale invariant wavelet features," *Pattern Recognition*, vol. 38, no. 10, pp. 1495–1506, 2005.

[40] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells." *Biological Cybernetics*, vol. 76, no. 2, pp. 83–96, 1997.

[41] S. Dash and U. R. Jena, "Texture classification using Steerable Pyramid based Laws' Masks," *Journal of Electrical Systems and Information Technology*, pp. 1–13, 2016.

[42] T. Krajník, P. Cristóforis, M. Nitsche, K. Kusumam, and T. Duckett, "Image Features and Seasons Revisited," in *ECMR*, 2015, pp. 1–7.

[43] P. Neubert, S. Niko, and P. Protzel, "Appearance Change Prediction for Long-Term Navigation Across Seasons," in *ECMR*, 2013, pp. 198–203.

[44] Tin Kam Ho, "Random decision forests," in *International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282.

[45] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Learning informative point classes for the acquisition of object model maps," in *International Conference on Control, Automation, Robotics and Vision*, 2008, pp. 643–650.

[46] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues,” in *IEEE International Conference on Computer Vision*, 2011, pp. 585–592.

[47] G. Arbeiter, S. Fuchs, R. Bormann, J. Fischer, and A. Verl, “Evaluation of 3D feature descriptors for classification of surface geometries in point clouds,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1644–1650.

[48] Z. C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, “General 3D modelling of novel objects from a single view,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 3700–3705.

[49] S.-H. Cha and S. N. Srihari, “On measuring the distance between histograms,” *Pattern Recognition*, vol. 35, no. 6, pp. 1355–1370, 2002.

[50] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[51] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, 2014.

[52] L. G. Wasilewski, F. R. Gommers, W. O’Leary, and H. A. K Nahrstaedt, “Pywavelets - wavelet transforms in python,” 2006, [Accessed April 18, 2018]. [Online]. Available: https://github.com/PyWavelets/pywt

[53] C.-C. Chang and C.-J. Lin, “LIBSVM : A Library for Support Vector Machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–39, 2013.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[55] M. R. Heinen, P. M. Engel, and R. C. Pinto, “IGMN : An Incremental Gaussian Mixture Network that Learns Instantaneously from Data Flows,” in *CSBC XXXI*, 2011.

[56] R. C. Pinto and P. M. Engel, “A Fast Incremental Gaussian Mixture Model,” *PLoS ONE*, vol. 10, no. 10, pp. e0 139 931+, 2015.

[57] P. Domingos and G. Hulten, “Mining high-speed data streams,” in *Sixth ACM SIGKDD International Conference*, 2000, pp. 71–80.

[58] V. da Silva and A. T. Winck, “Video popularity prediction in data streams based on context-independent features,” in *SAC*, 2017, pp. 95–100.

[59] J. Mrva and J. Faigl, “Tactile sensing with servo drives feedback only for blind hexapod walking robot,” in *RoMoCo*, 2015, pp. 240–245.

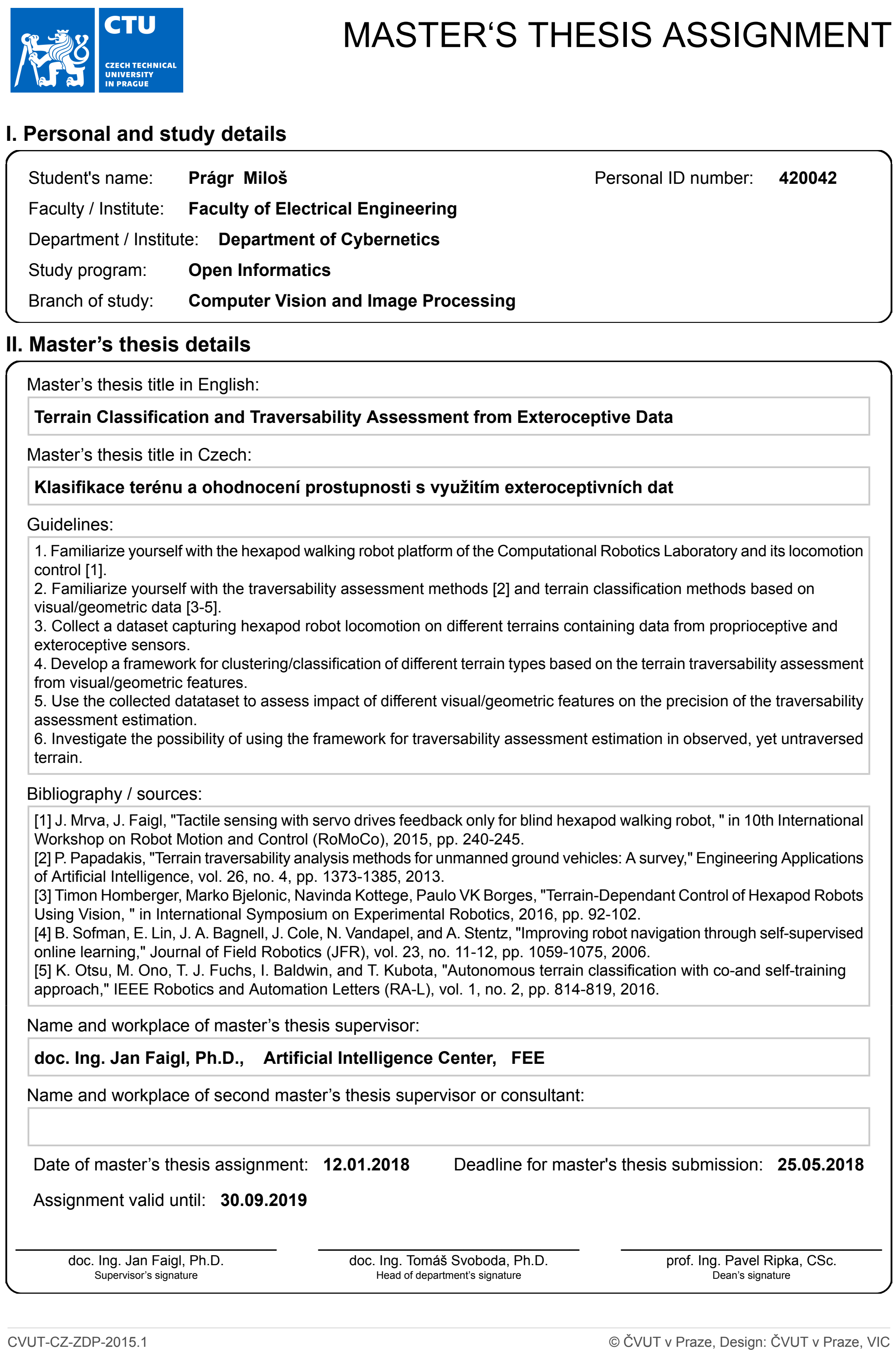
[60] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, 2009, p. 5.

Appendix A
List of Attachments

A.1 Contents of the Attached CD

- thesis.pdf
- visualiation.mp4

Appendix B
Thesis Assignment



MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Prágr Miloš** Personal ID number: **420042**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer Vision and Image Processing**

II. Master’s thesis details

Master’s thesis title in English:

Terrain Classification and Traversability Assessment from Exteroceptive Data

Master’s thesis title in Czech:

Klasifikace terénu a ohodnocení prostupnosti s využitím exteroceptivních dat

Guidelines:

1. Familiarize yourself with the hexapod walking robot platform of the Computational Robotics Laboratory and its locomotion control [1].
2. Familiarize yourself with the traversability assessment methods [2] and terrain classification methods based on visual/geometric data [3-5].
3. Collect a dataset capturing hexapod robot locomotion on different terrains containing data from proprioceptive and exteroceptive sensors.
4. Develop a framework for clustering/classification of different terrain types based on the terrain traversability assessment from visual/geometric features.
5. Use the collected datataset to assess impact of different visual/geometric features on the precision of the traversability assessment estimation.
6. Investigate the possibility of using the framework for traversability assessment estimation in observed, yet untraversed terrain.

Bibliography / sources:

[1] J. Mrva, J. Faigl, "Tactile sensing with servo drives feedback only for blind hexapod walking robot, " in 10th International Workshop on Robot Motion and Control (RoMoCo), 2015, pp. 240-245.

[2] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," Engineering Applications of Artificial Intelligence, vol. 26, no. 4, pp. 1373-1385, 2013.

[3] Timon Homberger, Marko Bjelonic, Navinda Kottege, Paulo VK Borges, "Terrain-Dependant Control of Hexapod Robots Using Vision, " in International Symposium on Experimental Robotics, 2016, pp. 92-102.

[4] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," Journal of Field Robotics (JFR), vol. 23, no. 11-12, pp. 1059-1075, 2006.

[5] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, "Autonomous terrain classification with co-and self-training approach," IEEE Robotics and Automation Letters (RA-L), vol. 1, no. 2, pp. 814-819, 2016.

Name and workplace of master’s thesis supervisor:

doc. Ing. Jan Faigl, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **12.01.2018** Deadline for master's thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

doc. Ing. Jan Faigl, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

CVUT-CZ-ZDP-2015.1

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student's signature

Appendix C
Detailed Results

Table 6: Full walk-to-walk scenario statistics

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.750	90.177
Hoeffding	Lab-M Ori	6.973	87.580
Hoeffding	Lab-M Shape	7.246	87.628
Hoeffding	Lab-P	7.415	117.329
Hoeffding	RGB-M	7.396	98.870
IGMN	ab-S	7.527	73.985
IGMN	ab-S Ori	8.091	72.056
IGMN	ab-S Shape	6.720	73.752
IGMN	Height	10.737	76.048
IGMN	Lab-M	5.463	63.782
IGMN	Lab-M Height	8.490	111.242
IGMN	Lab-M Ori	4.879	54.631
IGMN	Lab-M Set	10.156	202.966
IGMN	Lab-M Shape	7.643	90.989
IGMN	Lab-P	8.550	74.864
IGMN	Lab-P Ori	8.435	72.631
IGMN	Lab-P Shape	7.533	77.848
IGMN	Lab-S	9.265	77.643
IGMN	Lab-S Ori	10.042	75.499
IGMN	Lab-S Shape	7.693	65.152
IGMN	Ori	28.725	492.838
IGMN	RGB-M	9.644	110.255
IGMN	RGB-M Ori	7.550	93.046
IGMN	RGB-M Shape	6.175	64.173
IGMN	RGB-P	7.675	99.859
IGMN	RGB-P Ori	6.888	86.462
IGMN	RGB-P Shape	7.409	66.755
IGMN	RGB-S	10.644	73.111
IGMN	RGB-S Ori	10.550	72.426
IGMN	RGB-S Shape	8.772	119.959
IGMN	Set	8.034	99.178
IGMN	Shape	7.391	88.402
IGMN	Wave	10.201	75.290
IGMN	Wave Ori	9.675	75.966
IGMN	Wave Shape	7.040	64.586
Reg Tree	ab-S	8.227	98.684
Reg Tree	ab-S Ori	8.131	98.065
Reg Tree	ab-S Shape	6.202	91.656
Reg Tree	Height	8.287	108.883
Reg Tree	Lab-M	5.835	97.819
Reg Tree	Lab-M Height	6.852	87.882
Reg Tree	Lab-M Ori	5.834	97.769
Reg Tree	Lab-M Set	6.438	91.429
Reg Tree	Lab-M Shape	5.925	81.897
Reg Tree	Lab-P	7.106	106.065
Reg Tree	Lab-P Ori	7.113	106.377
Reg Tree	Lab-P Shape	6.700	92.712
Reg Tree	Lab-S	8.552	98.232
Reg Tree	Lab-S Ori	8.507	98.566
Reg Tree	Lab-S Shape	5.957	88.278
Reg Tree	Ori	9.455	95.993
Reg Tree	RGB-M	5.508	90.968
Reg Tree	RGB-M Ori	5.522	91.137
Reg Tree	RGB-M Shape	5.507	75.188
Reg Tree	RGB-P	7.643	107.963
Reg Tree	RGB-P Ori	7.892	105.488
Reg Tree	RGB-P Shape	6.507	92.986
Reg Tree	RGB-S	10.679	91.996
Reg Tree	RGB-S Ori	10.256	89.574
Reg Tree	RGB-S Shape	6.075	89.669
Reg Tree	Set	6.591	79.512
Reg Tree	Shape	6.407	94.094
Reg Tree	Wave	8.728	107.873
Reg Tree	Wave Ori	8.812	106.819
Reg Tree	Wave Shape	6.174	93.282
SVR	ab-S	7.830	97.605
SVR	ab-SOri	7.625	97.490
SVR	ab-SShape	5.803	83.509
SVR	Height	10.889	155.765
SVR	Lab-M	5.481	94.700
SVR	Lab-M Ori	5.548	94.421
SVR	Lab-M Shape	7.455	120.396
SVR	Lab-P	6.155	120.056
SVR	Lab-POri	7.389	114.499
SVR	Lab-PShape	6.105	84.536
SVR	Lab-S	8.219	96.409
SVR	Lab-SOri	7.990	92.611
SVR	Lab-SShape	5.597	83.539
SVR	Ori	9.062	103.553
SVR	RGB-M	6.602	119.747
SVR	RGB-M Ori	5.656	105.255
SVR	RGB-M Shape	6.093	69.685
SVR	RGB-P	7.007	109.752
SVR	RGB-P Ori	7.286	109.204
SVR	RGB-P Shape	5.850	84.981
SVR	RGB-S	9.294	103.206
SVR	RGB-S Ori	9.012	91.019
SVR	RGB-S Shape	5.835	95.590
SVR	Set	6.986	133.726
SVR	Shape	6.245	95.858
SVR	Wave	9.001	111.738
SVR	Wave Ori	8.242	108.734
SVR	Wave Shape	5.965	94.845

Table 7: Full flight-to-flight scenario statistics

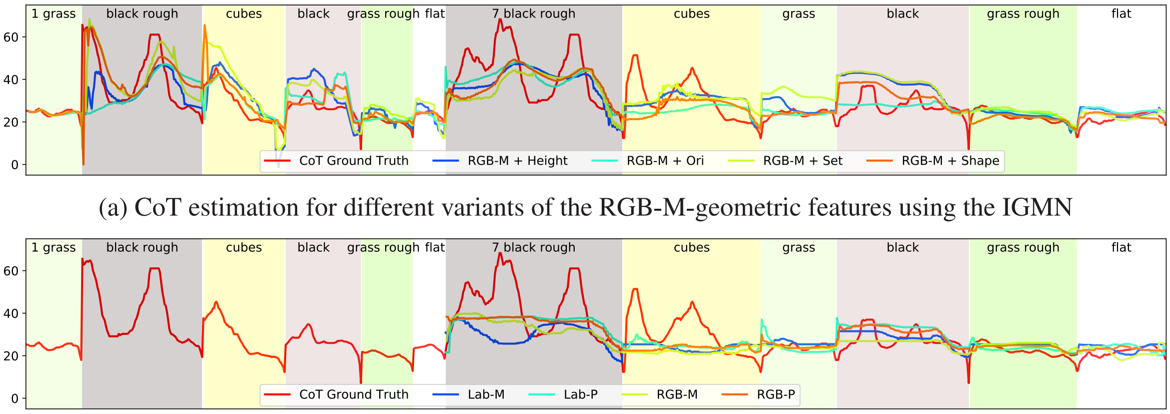
Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.189	76.481
Hoeffding	Lab-M Ori	7.018	66.911
Hoeffding	Lab-M Shape	7.142	78.206
Hoeffding	Lab-P	6.744	70.794
Hoeffding	RGB-M	7.055	67.859
IGMN	Height	7.390	54.706
IGMN	Lab-M	5.448	55.151
IGMN	Lab-M Height	7.433	64.117
IGMN	Lab-M Ori	5.229	58.489
IGMN	Lab-M Set	7.622	59.607
IGMN	Lab-M Shape	7.565	59.582
IGMN	Lab-P	6.932	59.353
IGMN	Lab-P Ori	7.768	63.331
IGMN	Lab-P Shape	7.677	58.118
IGMN	Ori	14.095	127.257
IGMN	RGB-M	11.383	135.562
IGMN	RGB-M Ori	11.622	137.297
IGMN	RGB-M Shape	7.264	61.659
IGMN	RGB-P	16.504	256.782
IGMN	RGB-P Ori	15.641	233.964
IGMN	RGB-P Shape	7.348	57.839
IGMN	Set	7.621	56.482
IGMN	Shape	7.022	62.460
Reg Tree	Height	7.515	111.048
Reg Tree	Lab-M	7.610	89.214
Reg Tree	Lab-M Height	5.997	89.901
Reg Tree	Lab-M Ori	7.610	89.214
Reg Tree	Lab-M Set	5.951	90.249
Reg Tree	Lab-M Shape	6.732	95.003
Reg Tree	Lab-P	7.195	85.103
Reg Tree	Lab-P Ori	7.189	85.142
Reg Tree	Lab-P Shape	6.324	90.253
Reg Tree	Ori	8.286	92.904
Reg Tree	RGB-M	6.022	92.347
Reg Tree	RGB-M Ori	6.029	92.179
Reg Tree	RGB-M Shape	6.316	88.825
Reg Tree	RGB-P	7.068	88.858
Reg Tree	RGB-P Ori	7.065	89.594
Reg Tree	RGB-P Shape	6.763	99.664
Reg Tree	Set	7.449	98.419
Reg Tree	Shape	6.585	102.228
SVR	Height	7.786	131.404
SVR	Lab-M	6.393	86.490
SVR	Lab-M Ori	6.474	85.951
SVR	Lab-M Shape	5.417	100.921
SVR	Lab-P	5.865	98.150
SVR	Lab-P Ori	6.986	88.912
SVR	Lab-P Shape	6.189	93.494
SVR	Ori	7.930	94.635
SVR	RGB-M	6.198	97.596
SVR	RGB-M Ori	6.369	100.920
SVR	RGB-M Shape	6.360	96.901
SVR	RGB-P	6.917	94.211
SVR	RGB-P Ori	6.892	94.106
SVR	RGB-P Shape	6.274	96.316
SVR	Set	7.938	110.359
SVR	Shape	6.110	104.596

Table 8: Full walk-to-walk-exploration scenario statistics

Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.692	105.646
Hoeffding	Lab-M Ori	7.800	93.814
Hoeffding	Lab-M Shape	10.364	193.357
Hoeffding	Lab-P	8.073	117.782
Hoeffding	RGB-M	9.430	146.286
IGMN	ab-S	6.586	131.555
IGMN	ab-S Ori	9.743	107.790
IGMN	ab-S Shape	8.330	98.607
IGMN	Height	10.156	92.699
IGMN	Lab-M	5.304	80.220
IGMN	Lab-M Height	10.294	181.473
IGMN	Lab-M Ori	9.695	184.820
IGMN	Lab-M Set	6.840	87.844
IGMN	Lab-M Shape	9.736	219.595
IGMN	Lab-P	7.475	106.035
IGMN	Lab-P Ori	7.185	93.333
IGMN	Lab-P Shape	11.140	132.079
IGMN	Lab-S	8.269	108.692
IGMN	Lab-S Ori	8.595	90.184
IGMN	Lab-S Shape	10.483	142.965
IGMN	Ori	28.547	721.531
IGMN	RGB-M	24.156	692.893
IGMN	RGB-M Ori	22.922	659.637
IGMN	RGB-M Shape	5.618	72.997
IGMN	RGB-P	23.965	642.121
IGMN	RGB-P Ori	24.195	664.523
IGMN	RGB-P Shape	7.606	83.073
IGMN	RGB-S	9.063	108.171
IGMN	RGB-S Ori	8.765	105.558
IGMN	RGB-S Shape	11.097	138.984
IGMN	Set	12.044	136.896
IGMN	Shape	8.995	97.325
IGMN	wv	8.002	130.635
IGMN	wv Ori	7.473	134.422
IGMN	wv Shape	8.726	92.329
Reg Tree	ab-S	9.042	254.305
Reg Tree	ab-S Ori	8.926	251.878
Reg Tree	ab-S Shape	9.077	238.272
Reg Tree	Height	10.982	225.083
Reg Tree	Lab-M	8.621	233.504
Reg Tree	Lab-M Height	8.348	215.078
Reg Tree	Lab-M Ori	8.621	233.517
Reg Tree	Lab-M Set	8.422	214.863
Reg Tree	Lab-M Shape	7.735	213.843
Reg Tree	Lab-P	9.138	220.355
Reg Tree	Lab-P Ori	8.117	229.471
Reg Tree	Lab-P Shape	8.933	234.272
Reg Tree	Lab-S	8.576	223.699
Reg Tree	Lab-S Ori	8.480	223.813
Reg Tree	Lab-S Shape	8.252	236.942
Reg Tree	Ori	8.670	252.610
Reg Tree	RGB-M	8.815	233.840
Reg Tree	RGB-M Ori	8.810	233.681
Reg Tree	RGB-M Shape	8.933	228.813
Reg Tree	RGB-P	8.139	229.757
Reg Tree	RGB-P Ori	8.104	231.208
Reg Tree	RGB-P Shape	9.223	233.997
Reg Tree	RGB-S	8.435	235.220
Reg Tree	RGB-S Ori	8.357	235.896
Reg Tree	RGB-S Shape	8.597	246.280
Reg Tree	Set	10.790	226.379
Reg Tree	Shape	8.177	244.333
Reg Tree	wv	8.555	252.282
Reg Tree	wv Ori	8.563	253.574
Reg Tree	wv Shape	8.630	240.265
SVR	Lab-M	8.557	252.507
SVR	Lab-M Ori	8.511	251.005
SVR	Lab-M Shape	8.499	248.929
SVR	Lab-P	8.570	248.463
SVR	RGB-M	7.913	231.387

Table 9: Full flight-to-flight-exploration scenario statistics

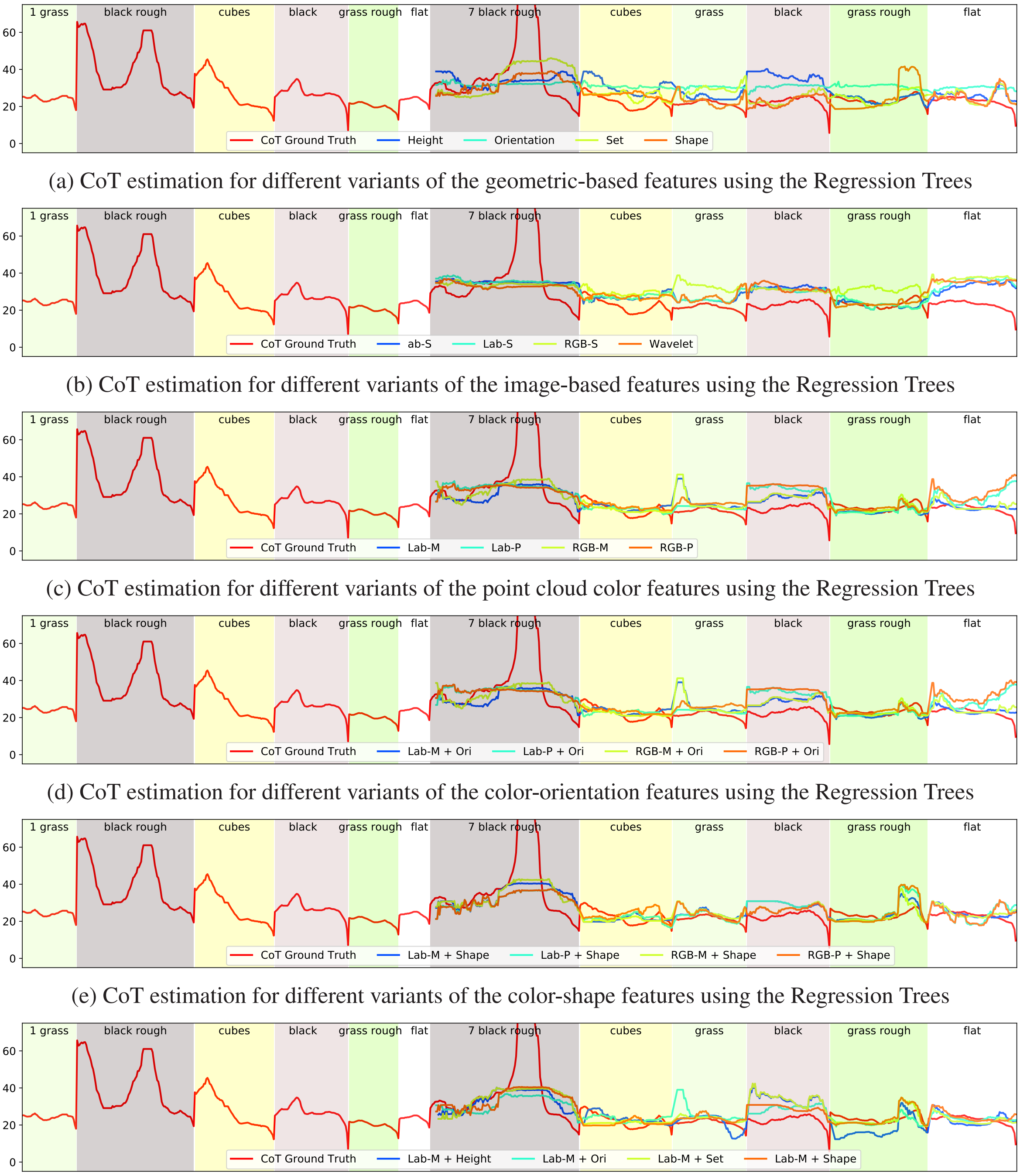
Learning	Feature	Mean	Var
Hoeffding	Lab-M	7.416	67.174
Hoeffding	Lab-M Ori	8.146	91.096
Hoeffding	Lab-M Shape	7.230	81.491
Hoeffding	Lab-P	7.170	85.642
Hoeffding	RGB-M	7.443	73.546
Hoeffding	RGB-M Shape	7.031	61.895
IGMN	Height	6.084	80.776
IGMN	Lab-M	5.186	65.264
IGMN	Lab-M Height	5.413	56.304
IGMN	Lab-M Ori	5.938	85.583
IGMN	Lab-M Set	9.821	149.407
IGMN	Lab-M Shape	8.046	123.891
IGMN	Lab-P	6.289	92.768
IGMN	Lab-P Ori	6.280	88.832
IGMN	Lab-P Shape	6.290	67.883
IGMN	Ori	7.004	106.666
IGMN	RGB-M	8.144	92.952
IGMN	RGB-M Height	5.267	51.814
IGMN	RGB-M Ori	18.631	441.283
IGMN	RGB-M Shape	4.776	53.706
IGMN	RGB-P	18.436	417.342
IGMN	RGB-P Ori	17.265	366.808
IGMN	RGB-P Shape	8.043	92.712
IGMN	Set	5.763	56.175
IGMN	Shape	7.802	116.188
Reg Tree	Height	7.801	167.335
Reg Tree	Lab-M	7.690	167.393
Reg Tree	Lab-M Height	7.456	164.962
Reg Tree	Lab-M Ori	7.689	167.401
Reg Tree	Lab-M Set	7.289	167.390
Reg Tree	Lab-M Shape	7.977	216.385
Reg Tree	Lab-P	7.284	186.605
Reg Tree	Lab-P Ori	7.415	191.274
Reg Tree	Lab-P Shape	7.753	214.104
Reg Tree	Ori	7.556	198.452
Reg Tree	RGB-M	7.790	212.173
Reg Tree	RGB-M Ori	7.790	212.173
Reg Tree	RGB-M Shape	7.316	203.796
Reg Tree	RGB-P	7.346	195.571
Reg Tree	RGB-P Ori	7.324	196.993
Reg Tree	RGB-P Shape	8.217	213.407
Reg Tree	Set	8.101	185.670
Reg Tree	Shape	7.831	220.484
SVR	Lab-M	7.616	205.117
SVR	Lab-M Ori	7.615	203.203
SVR	Lab-M Shape	7.414	201.844
SVR	Lab-P	7.606	190.827
SVR	RGB-M	7.536	202.858
SBR	RGB-M Shape	7.625	208.615



(a) CoT estimation for different variants of the RGB-M-geometric features using the IGMN

(b) CoT estimation for different variants of the color features using the Regression trees

Figure 19: Learned and predicted values of CoT for different combinations of the features and learning algorithms in the walk-to-flight setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.



(a) CoT estimation for different variants of the geometric-based features using the Regression Trees

(b) CoT estimation for different variants of the image-based features using the Regression Trees

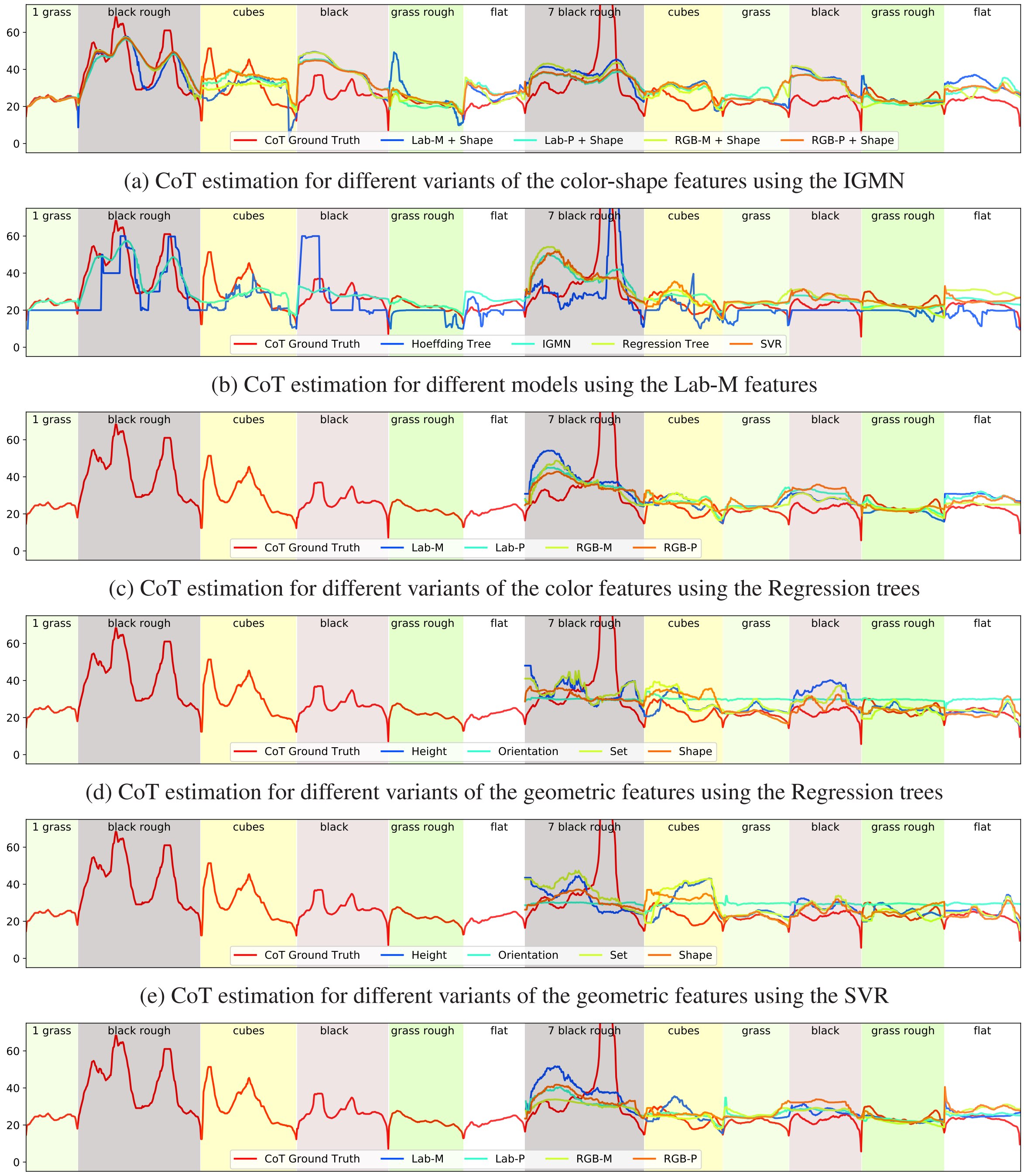
(c) CoT estimation for different variants of the point cloud color features using the Regression Trees

(d) CoT estimation for different variants of the color-orientation features using the Regression Trees

(e) CoT estimation for different variants of the color-shape features using the Regression Trees

(f) CoT estimation for different variants of the Lab-M-geometric features using the Regression Trees

Figure 20: Learned and predicted values of CoT for different combinations of the features and learning algorithms in the walk-to-walk setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.



(a) CoT estimation for different variants of the color-shape features using the IGMN

(b) CoT estimation for different models using the Lab-M features

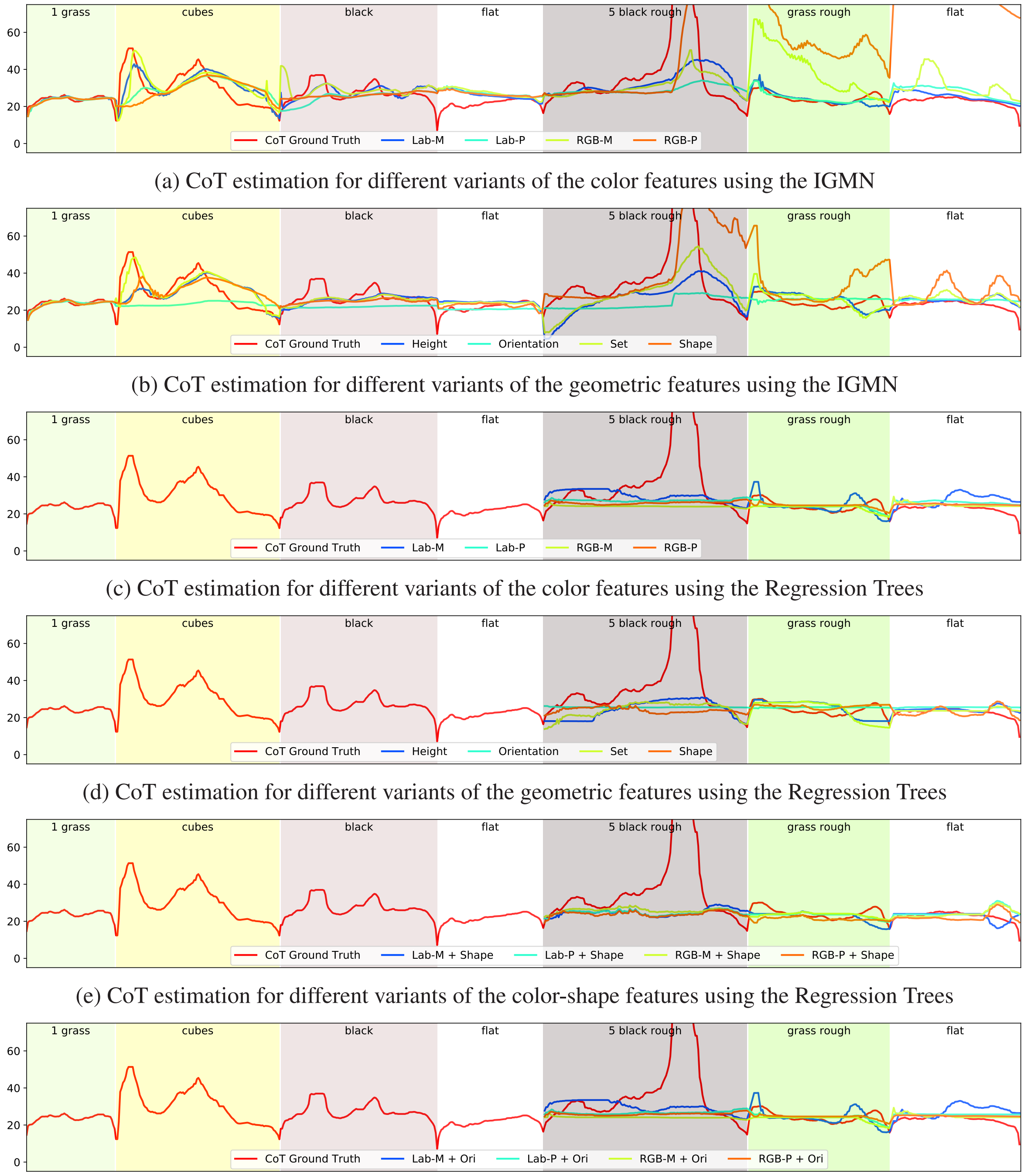
(c) CoT estimation for different variants of the color features using the Regression trees

(d) CoT estimation for different variants of the geometric features using the Regression trees

(e) CoT estimation for different variants of the geometric features using the SVR

(f) CoT estimation for different variants of the color features using the SVR

Figure 21: Learned and predicted values of CoT for different combinations of the features and learning algorithms in the flight-to-flight setup. The first six terrains (parts) represent the learning phase. The following six parts represent the inference phase.



(a) CoT estimation for different variants of the color features using the IGMN

(b) CoT estimation for different variants of the geometric features using the IGMN

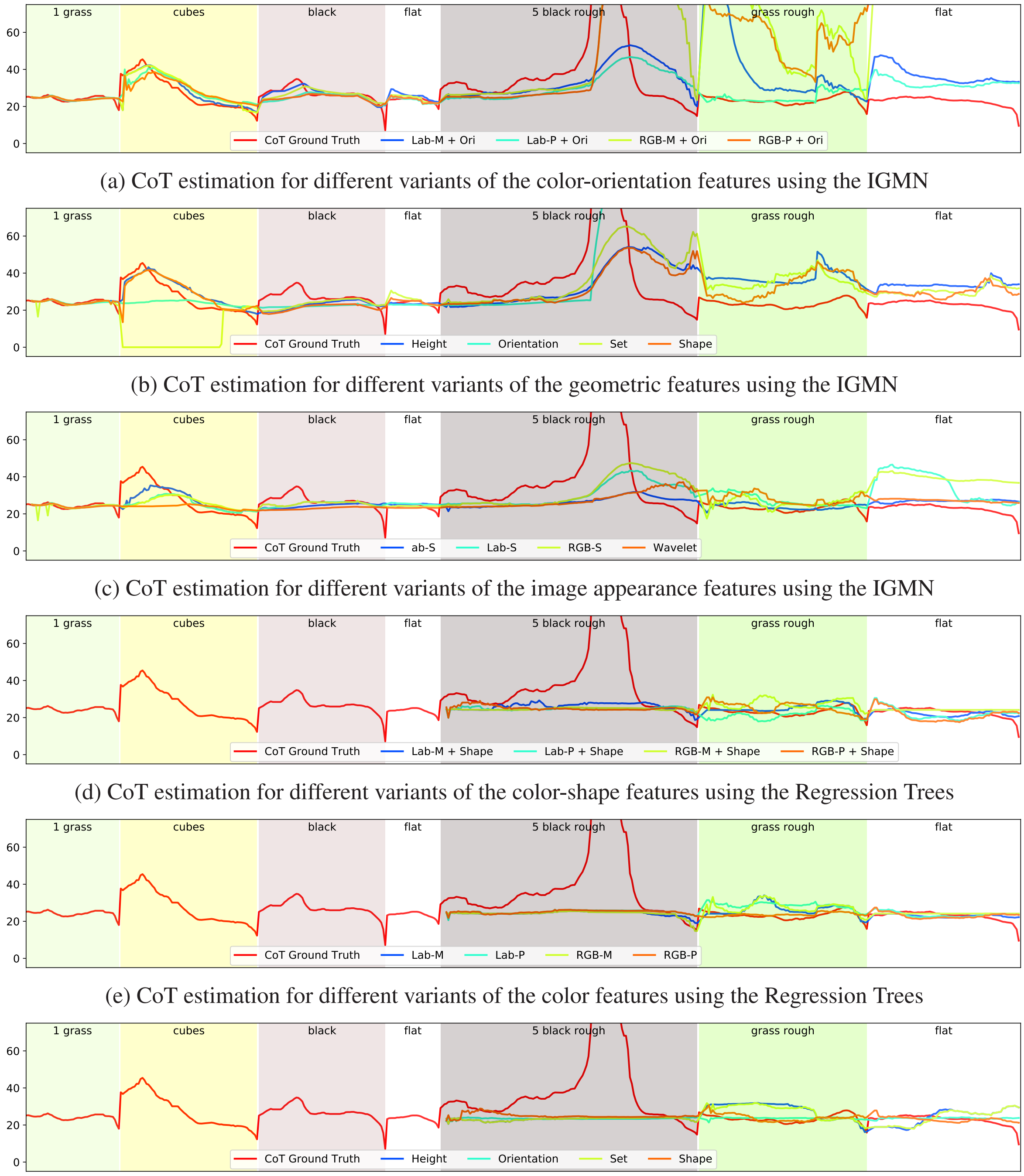
(c) CoT estimation for different variants of the color features using the Regression Trees

(d) CoT estimation for different variants of the geometric features using the Regression Trees

(e) CoT estimation for different variants of the color-shape features using the Regression Trees

(f) CoT estimation for different variants of the color-orientation features using the Regression Trees

Figure 22: Learned and predicted values of CoT for different combinations of the features and learning algorithms in the flight-to-flight-exploration setup. The first four terrains (parts) represent the learning phase. The following three parts represent the inference phase.



(a) CoT estimation for different variants of the color-orientation features using the IGMN

(b) CoT estimation for different variants of the geometric features using the IGMN

(c) CoT estimation for different variants of the image appearance features using the IGMN

(d) CoT estimation for different variants of the color-shape features using the Regression Trees

(e) CoT estimation for different variants of the color features using the Regression Trees

(f) CoT estimation for different variants of the geometric features using the Regression Trees

Figure 23: Learned and predicted values of CoT for different combinations of the features and learning algorithms in the walk-to-walk-exploration setup. The first four terrains (parts) represent the learning phase. The following three parts represent the inference phase.