Bachelor Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Traversability Estimation from RGB Images and Height Map

Jan Dočekal

Supervisor: doc. Ing. Karel Zimmermann, Ph.D.
Supervisor–specialist: Ing. Vojtěch Šalanský
Field of study: Cybernetics and Robotics
May 2020

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Dočekal Jan** Personal ID number: **478074**

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

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Traversability Estimation from RGB Images and Height Map

Bachelor’s thesis title in Czech:

Odhad traversability terénu z RGB obrázků a výškových map

Guidelines:

(1) Get acquainted with the skid-steer robot TRADR and its software developed for DARPA Subterranean Challenge under Robot Operating System (www.ros.org).
(2) Propose a method for terrain traversability estimation. Input should be the height map (generated by the existing system) and RGB images, output will be traversability map of the same spatial resolution as the height map.
(3) Implement proposed method and provide qualitative evaluation on existing BAG-files captured during previous round of the competition.
(4) Prepare dataset suitable for quantitative evaluation and report some results.

Bibliography / sources:

[1] Sebastian B.D. Traversability Estimation Techniques for Improved Navigation of Tracked Mobile Robots https://vtechworks.lib.vt.edu/handle/10919/94629
[2] R. Omar Chavez-Garcia, Jerome Guzzi, Luca M. Gambardella, Alessandro Giusti, Learning Ground Traversability from Simulations, RAL, 2018 https://arxiv.org/abs/1709.05368

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Karel Zimmermann, Ph.D., Vision for Robotics and Autonomous Systems, FEE

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

Date of bachelor’s thesis assignment: **09.01.2020** Deadline for bachelor thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

doc. Ing. Karel Zimmermann, Ph.D. Supervisor’s signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’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 bachelor’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student’s signature

Acknowledgements

I would like to express my sincere gratitude to my supervisor doc. Ing. Karel Zimmermann, Ph.D. for his guidance, motivation and continuous support during writing this thesis.

Also, I would like to thank Ing. Vojtěch Šalanský, who shared his thoughts during the research and was always helpful.

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 theses.

In Prague, 21. May 2020

Abstract

Traversability estimation is an important task for autonomous mobile robots. They should be able to decide about traversability in their surroundings to be safely navigated. In this thesis, the method of merging depth measurements as heightmaps with RGB images is proposed. Our approach consists from state-of-the-art methods for analysis of both, which are convolutional neural networks. We used self-supervised learning of convolutional neural networks on real datasets. Datasets consist from various environments such as mines, hallways, staircases and other common outdoor terrains (grass, road, pavement). Our network provides correct estimation for easier terrain such as hallways or flat terrain, and acceptable results as for challenging environments such as staircases or soft obstacles (e. g. high grass).

Keywords: computer vision, traversability, heightmap, convolutional neural networks

Supervisor: doc. Ing. Karel Zimmermann, Ph.D.
Center for Machine Perception,
Karlovo nam. 13,
Praha 2

Abstrakt

Odhad traversability je důležitá úloha pro autonomní mobilní roboty. Ti by měli být schopni rozhodnout o traversabilitě svého okolí, aby byli bezpečně naváděni. V této práci je navržena metoda spojení hloubkových měření v podobě výškových map s RGB obrázky. Náš přístup se skládá z nejmodernějších metod analýzy obou, tedy konvolučních neuronových sítí. Používáme self-supervised učení konvolučních neuronových sítí na reálných datasetech. Datasety se skládají z několika různých prostředí, jako jsou doly, chodby, schodiště a další běžné venkovní terény (tráva, cesta, chodník). Naše síť poskytuje správný odhad na jednodušších terénech, jako jsou chodby nebo rovný terén, a přijatelné výsledky pro náročný terén, jako schody nebo měkké překážky (např. vysoká tráva).

Klíčová slova: počítačové vidění, traversabilita, výšková mapa, konvoluční neuronové sítě

Překlad názvu: Odhad traversability terénu z RGB obrázků a výškových map

Contents

1 Introduction 1

1.1 Task 1

1.2 Related Work 3

1.3 Our Approach 5

2 Convolutional Neural Networks 7

2.1 Classification Task 7

2.2 Segmentation Task 8

3 Projection of Features 11

3.1 Image Features 12

3.2 Voxel Map 13

3.3 Camera Model 14

3.4 Ray-Tracing Method 15

4 Self-Supervised Learning 17

4.1 Labels Estimation 17

4.2 Training CNN 19

4.2.1 CNN architecture 19

4.2.2 Training 20

5 Experimental Results 23

5.1 Flat Terrains 23

5.2 Staircases 25

5.3 Prepared Experiment 26

6 Discussion and Conclusion 29

6.1 Discussion 29

6.2 Conclusion 30

6.3 Future Work 31

Bibliography 33

Figures

1.1 UGV TRADR mobile robotic platform with flippers. 2

1.2 Traversability estimation diagram. 5

2.1 Example of classification convolutional neural network VGG16.[1] 8

2.2 Example of semantic segmentation convolutional neural network.[2] 9

3.1 Image features projection to voxel map diagram. 11

3.2 ERFNet architecture with spatial resolutions according to input of shape 1024 × 512 × 3.[3] 12

3.3 Example of image from robot camera. 12

3.4 Comparison of sparse (a) and dense (b) DEM. 13

3.5 Voxel map visualized in ROS visualization tool RViz. 13

3.6 Projection of 3D object on a 2D image plane using Pinhole Camera Model.[4] 14

3.7 Tracing ray from camera origin through image plane.[5] 15

3.8 Projected image features to built voxel map. 16

4.1 Visualization of robot's position estimated on given patch of heightmap, with robot drawn as a grid skeleton. 18

4.2 Estimated labels (b) to its input heightmap (a) with marked visible area of sight of robot (red rectangle). 19

4.3 Architecture of CNN used for traversability estimation. 20

4.4 Binary Cross-Entropy loss function during training CNN. 21

5.1 Visualization of traversability in hallway. Green voxels refer to traversable, red to non-traversable. 24

5.2 Visualization of traversability on flat terrain. Green voxels refer to traversable, red to non-traversable. 24

5.3 Visualization of traversability on staircases. Green voxels refer to traversable, red to non-traversable. 25

5.4 Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on indoor dataset. 26

5.5 Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on outdoor dataset. 26

5.6 Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on combined dataset. 27

5.7 Visualization of wrong traversability on experiment with grass and rocks with the same height. CNN trained on combined dataset. 27

Chapter 1

Introduction

1.1 Task

Traversability estimation is a crucial task in autonomous mobile robotics systems, especially for explorative robots working in challenging environment. A correct traversability assessment allows for safe robot navigation. Although it is one of the most important problems, it is still an open research problem.

In general, this task is very robot-specific, because various robots can overcome different obstacles or drive through different spaces. In this thesis, the aim is at UGV TRADR robot with flippers, see Figure 1.1. Flippers are auxiliary independently controlled subtracks, which allow the robot to traverse hard terrain or obstacles. Traversability itself is mainly given by support of mobile parts of robot on terrain.

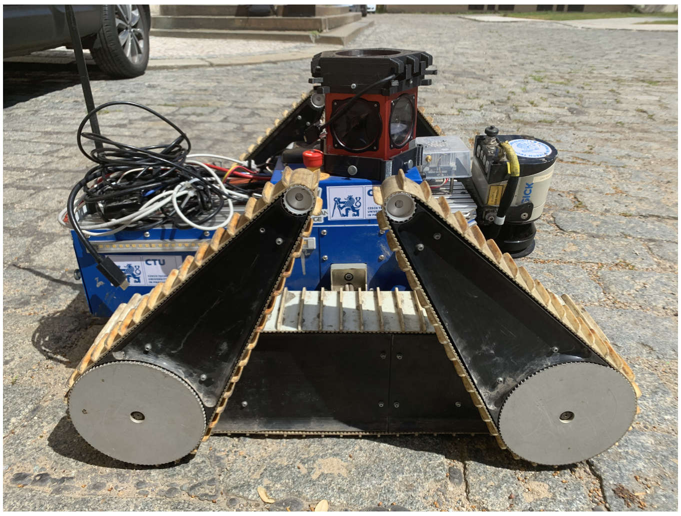


Figure 1.1: UGV TRADR mobile robotic platform with flippers.

The goal of this thesis is to estimate traversability of terrain based on exteroceptive measurements, for such UGV robot is equipped with multiple sensors. In order to do so, the LiDAR sensor and RGB cameras will be used. LiDAR is a time-of-flight depth sensor which measures depths of points in surroundings of robot using light beams. This measurement is stored as sparse point clouds of hundreds or thousands points. RGB cameras are used to obtain images of current environment in all directions around robot.

For our purpose, LiDAR measurements are crucial and RGB images provide extra terrain information. LiDAR by itself is insufficient because of various occurances which can cause wrong measuring. LiDAR measurement can provide missing spaces, where we get no information or wrong depth, such as water or other bright spots. It is caused by incorrect light beam reflection on these surfaces. Another issue are soft terrains, for example grass or snow, where LiDAR would measure very hard to traverse shape, but robot can easily overcome these terrains. That is why we will also use RGB cameras to identify and fix such problematic surfaces.

On the other hand, we should be rather pessimistic than optimistic in terms of traversability, because we do not want our robot to be damaged. How hard and challenging obstacles can robot overcome? Will our approach reach desired precision and turn out useful?

1.2 Related Work

Terrain traversability estimation is a problem necessary to be solved for autonomous mobile robotics platforms. Therefore, there are more approaches to this problem, which usually depend on sensors available on given robot.

There are two main sensors, which obtain information containing terrain features. Depth sensors and cameras will be both used in this thesis.

As for depth sensors two approaches are relevant.

Geometrical analysis In [JGH08] the 2.5D grid-based heightmap is obtained first. Then the terrain traversability is estimated based on features such as height difference, slope and roughness. These features are analysed only on a small patch of heightmap. Height difference is just simple subtraction of two adjacent grid cell's height. Slope is computed using normal estimation of local plane which allows the computation of the slope. Last feature, roughness, is obtained as residues from local plane represented with normal vector.

Similar approach is used in [BVS+13]. Unlike previous article, they construct 3D heightmap. Also, fast normal computation is applied for better time requirements. Thanks to the 3D map the robot height can be taken into consideration to decide whether it can fit into low places. On the other hand, the roughness is not taken into consideration for traversability estimation.

Convolutional neural networks The heightmap features analysis is based on convolutional neural network (CNN) in [CGGGG18]. Used data are represented as the heightmap images, which can be classified by CNN, whether it is traversable class or non-traversable class. Feature-based approach is also provided, in which the Histogram of Gradients (HOG) , that corresponds to slope, is computed. Resulting features are classified by means of a Random Forest classifier with 10 trees. But as expected, state-of-the-art approach with CNN outperforms this feature-based approach. In this thesis, RGB image information will be added to the heightmap and the traversability estimation will be provided by CNN.

With cameras we obtain RGB images, which are analysed with state-of-the-art approaches.

Deep learning methods Application of convolutional neural networks for RGB images is provided in [SRL$^+$19]. The focus is on semantic image segmentation along with roughness estimation using CNN. For image segmentation 3 different architectures of networks, SegNet, ENet and ERFNet are used and compared. In comparison, the best results has ERFNet architecture which is considered as state-of-the-art. For roughness estimation bottom feature maps (feature maps close to original input image) are used as they correspond to basic appearance features. These feature maps are upsampled with simpler decoder than in terrain segmentation part.

Similar approach is presented in [WDR$^+$19]. ERFNet with added skip connections is used for terrain segmentation. Also, the terrain is analysed using force-torque sensors on legs (they work with legged robot ANYmal) using continuous wavelet transformation using Morse wavelet and principal component analysis (PCA) to get "ground reaction score".

In this thesis, the ERFNet to obtain image features will be used as well.

Appearance-based classification The appearance-based classification is used in [MBS11] and [MB12]. In [MBS11] the image classifiers based on color histograms and discrete cosine transformation (DCT) are used. These classifiers are learnt and updated with laser scans, from which ground plane is estimated and features are assigned to it. The goal is to use as little laser scans as possible, thus it is used only if the appearance of surroundings change rapidly, e. g. change of the floor colour.

Another method is in [MB12], where the same type of classifiers, except the laser scans learning, is used. The classifiers learn using iterative ground plane estimation with floor homography. The initial estimation of floor homography is estimated from odometry and subsequently optimized with nonlinear optimization of correspondence between image pairs taken while robot is moving. Both [MB12] and [MBS11] aim for humanoid robot in indoor environment. For this reason, both studies focus only on the floor, otherwise a much more complex approach would be needed. In conclusion this method is not useful for the type of robotic platform used in this thesis.

1.3 Our Approach

In this thesis, a new method for fusing LiDAR and RGB cameras measurements to estimate traversability of robots surroundings is introduced. The aim is to combine state-of-the-art methods for both heightmap analysis and RGB images evaluation. The ERFNet from [RA18] and their pretrained model for PyTorch is used to analyse RGB images, especially to obtain segmentation features. These features are projected to built voxel map of robot's surroundings. Traversability is estimated using self-supervised learning of CNN, which was trained from scratch.

For reader's better understanding and visualization, traversability estimation diagram is shown in Figure 1.2. Voxel map will be built from point cloud in Section 3.2. Image features will be obtained using ERFNet as explained in Section 3.1 and projected to built voxel map in Section 3.4. Convolutional neural network for estimating traversability will be trained in Chapter 4.

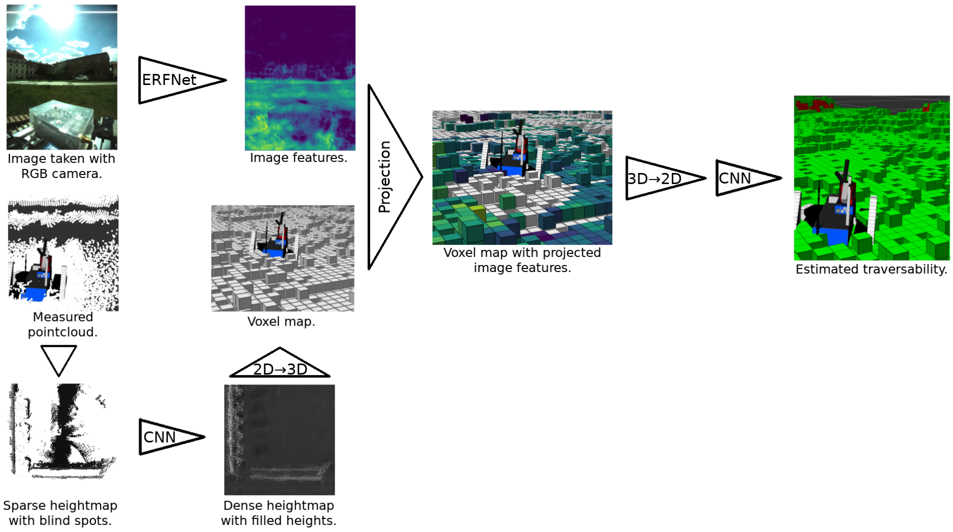


Image taken with RGB camera.
ERFNet
Image features.
Projection
Voxel map with projected image features.
3D→2D
CNN
Estimated traversability.
Measured pointcloud.
Voxel map.
2D→3D
Sparse heightmap with blind spots.
CNN
Dense heightmap with filled heights.

Figure 1.2: Traversability estimation diagram.

Chapter 2

Convolutional Neural Networks

The proposed method for traversability estimation is based on convolutional neural network (CNN). Moreover, 2 pretrained CNNs to obtain dense DEM (Digital Elevation Model) of robot's surroundings and image features are also used.

Convolutional neural networks are considered as state-of-the-art approach for analysing RGB images, grayscales, medical images etc.

2.1 Classification Task

As for classification task, network outputs class probabilities for the whole input data. Baseline architecture usually consists of 2 parts, see Figure 2.1.

First part, also called an encoder, uses convolutional layers in combination with pooling layers, normalization layers and activation (non-linear) layers. This approach leads to reduction of spatial resolution of input but usually with much more channels. Encoder is followed with fully connected part of the network. According to the given problem, output size is defined by the number of classification classes.

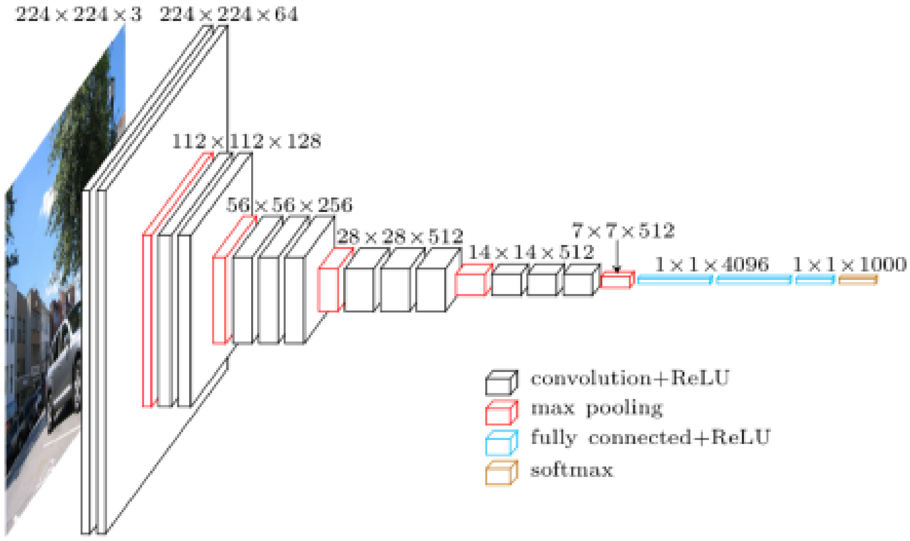


Figure 2.1: Example of classification convolutional neural network VGG16.[1]

State-of-the-art network architectures (GoogLeNet, ResNet) also use much sofisticated layers such as concatenation of inception modules, which help to analyse the same features in various ways. Or skip connections that help with diminishing gradient in learning deeper networks or allow features from the beginning to affect the output directly.

2.2 Segmentation Task

Semantic segmentation problem means that our aim is to classificate image pixel-wise. Output of such convolutional neural network consists of pixel-wise probabilities for each class. First encoding block is usually pretty similar to the one in classification part.

Second block, also called decoder or deconvolutional, upsamples feature maps from the encoder back to input spatial resolution. For this purpose, unpooling layers, upsample blocks or transposed convolution (deconvolutional) layers are used. Again combined with activation layers.

For better understanding and easier visualization see Figure 2.2, which shows common semantic segmentation CNN architecture.

[1] Taken from D. Frossard, *VGG in Tensorflow*, `https://www.cs.toronto.edu/~frossard/post/vgg16/vgg16.png`

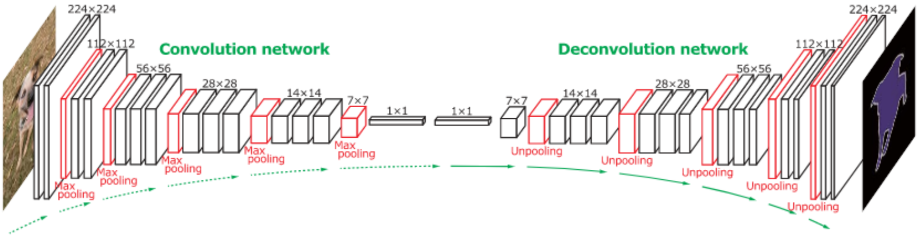


Figure 2.2: Example of semantic segmentation convolutional neural network.[2]

State-of-the-art segmentation CNN (DeepLab) uses other modern techniques. The first is the atrous (dilated) convolutional layer, which is similar to convolutional layer, but with spacing between the weights in kernel. The second method uses conditional random fields (CRF) on top of output of the CNN.

Unfortunately, for extraction of image features DeepLab cannot be used due to its high feed forward time. Fortunately, ERFNet provides much faster result on comparable level of precision.

[2] Taken from [NHH15]

Chapter 3

Projection of Features

In this thesis, the goal is to estimate traversability from depth sensing and from camera images. It is essential to assign image features to corresponding point on heightmap grid. Figure 3.1 shows the fusion of LiDAR and camera measurements.

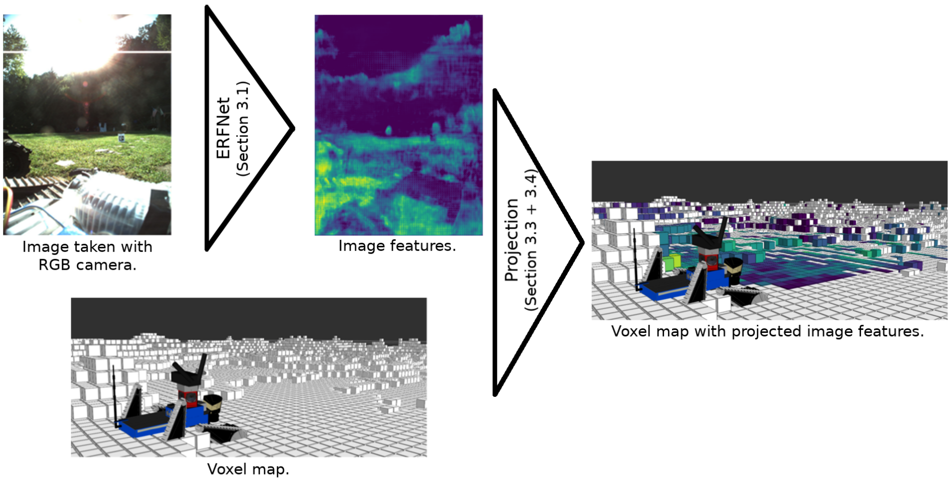


Image taken with RGB camera.
ERFNet (Section 3.1)
Image features.
Projection (Section 3.3 + 3.4)
Voxel map with projected image features.
Voxel map.

Figure 3.1: Image features projection to voxel map diagram.

3.1 Image Features

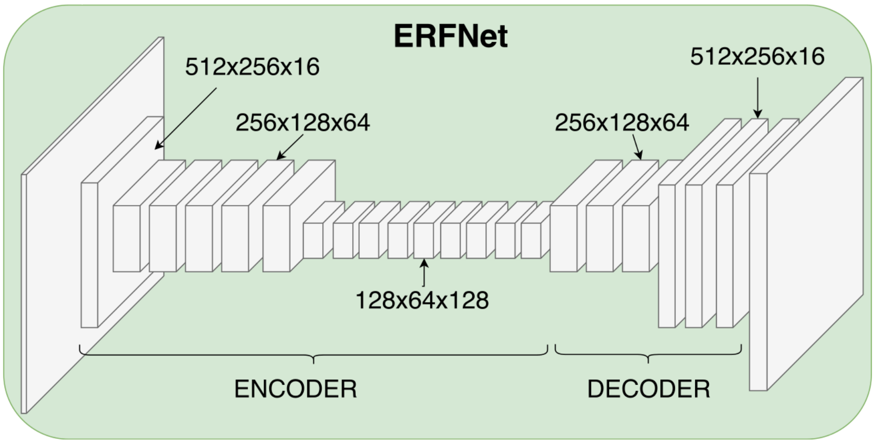


Figure 3.2: ERFNet architecture with spatial resolutions according to input of shape 1024 × 512 × 3.[1]

As proposed in Section 1.3, image features provided by ERFNet from [RA18] will be used. Its architecture is showed in Figure 3.2. Because our goal is not to segment our images to classes used in [RA18], ERFNet will be cut before its very last upsampling layer and only the encoded and decoded features will be used.

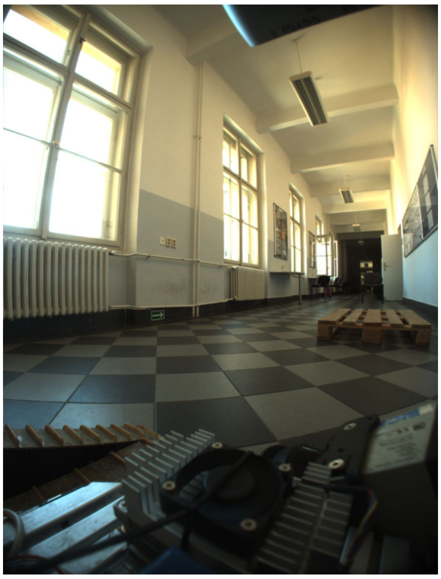


Figure 3.3: Example of image from robot camera.

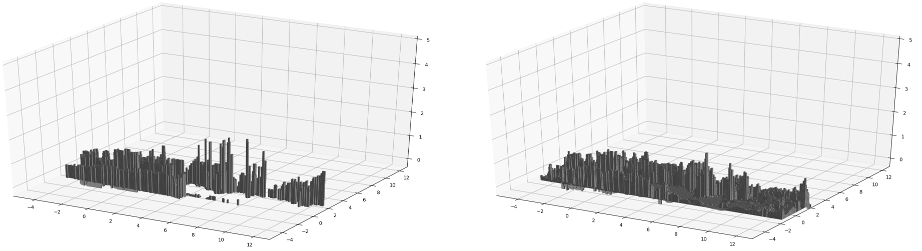
Example of input image obtained from one of robot's RGB camera is shown in Figure 3.3.

[1] Taken and slightly modified from [RA18]

The image spatial resolution will be reduced to half of the original size because of cutting off the last upsampling layer. In addition, for better feed forward time of ERFNet, the input image size will be reduced to half. Cameras on TRADR robot provide resolution 1232 × 1616 with 3 RGB channels, we will change it to 616 × 808 with 3 RGB channels and features obtained with ERFNet will have spatial resolution 308 × 404 with 16 feature channels.

3.2 Voxel Map

LiDAR provides only sparse measurement and that is the reason why DEM includes holes with unknown information in the beginning, as can be seen in Figure 3.4. For creating dense DEM convolutional neural network that fills blind spots provided by Ing. Vojtěch Šalanský is used.



(a) : Visualization of sparse DEM.

(b) : Visualization of dense DEM.

Figure 3.4: Comparison of sparse (a) and dense (b) DEM.

To build a 3D voxel map, the 2D DEM is converted to 3D. Every point in grid contains information about height, so each point needs to be set at that height and voxels below will be filled. Such built voxel map is visualized in Figure 3.5.

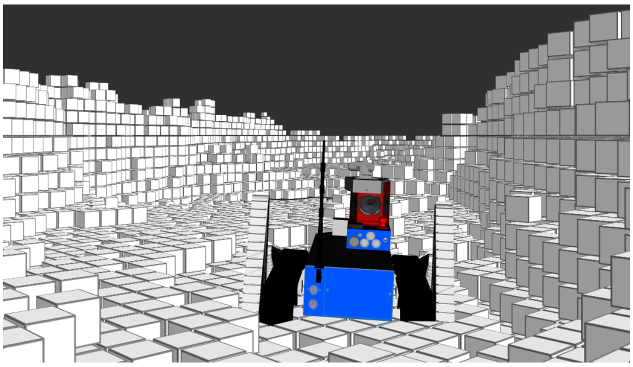


Figure 3.5: Voxel map visualized in ROS visualization tool RViz.

3.3 Camera Model

For image features assignment a camera model for demonstration of how 3D point in world coordinates maps to 2D point on image plane needs to be created.

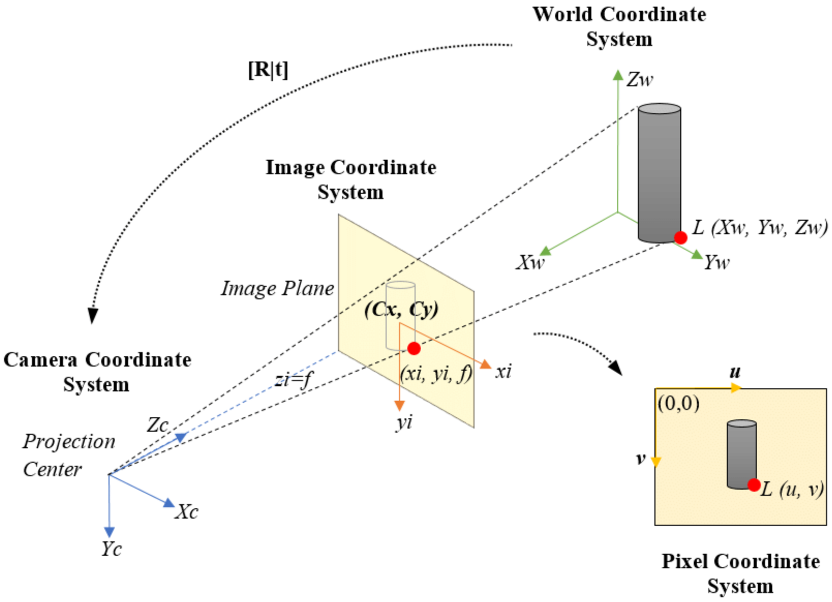


Figure 3.6: Projection of 3D object on a 2D image plane using Pinhole Camera Model.[2]

According to [HS17] and [Kit17], the simplest model is Pinhole Camera Model. The goal is to find transformation between world coordinates and image plane of camera, see Figure 3.6. Mathematically speaking, $\mathbf{C}$ must satisfy the condition as follows:

$$P = \mathbf{C}P_w, \tag{3.1}$$

where P is 2D point on image plane of camera and P_w is 3D point in world coordinates, both in homogenous coordinates. Camera matrix $\mathbf{C}$ can be decomposed to 2 matrices. Intrinsic matrix $\mathbf{K}$ and extrinsic matrix $\mathbf{X}$. Intrinsic matrix $\mathbf{K}$ is unique for every camera type, because it contains camera parameters.

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

where f is focal length of camera, c_x and c_y is principal point.

[2] Taken from [OGC17]

Extrinsic parameters depends on transformation between camera coordinate system and world coordinate system. Thus it contains rotation part **R** and translation part **t**.

$$\mathbf{X} = \left[\mathbf{R} \mid \mathbf{t}\right] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right] \tag{3.3}$$

Knowing both intrinsic matrix **K** and extrinsic matrix **X**, the mathematical model of camera was created as follows:

$$P = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right] P_w = \mathbf{K}\mathbf{X}P_w = \mathbf{C}P_w. \tag{3.4}$$

3.4 Ray-Tracing Method

For assignment of image features obtained in Section 3.1 ray-tracing rendering technique will be used. Ray-tracing is method of 3D rendering in computer vision, which is also very common in games graphics.

We have already built camera model in Section 3.3 and voxel map of surroundings of robot in Section 3.2. We will combine these to shoot ray from camera origin through pixel in image plane to find its intersection with voxel map, as can be seen in Figure 3.7.

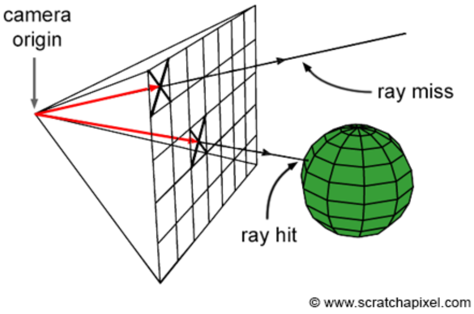


Figure 3.7: Tracing ray from camera origin through image plane.[3]

Because ray-tracing method is computationally demanding, the ray will not be shot through every pixel of the image plane. There will only be 3600 rays shot in 1 image. Because we use 5 cameras, it is 18000 traced rays in

[3] Taken from An Overview of the Ray-Tracing Rendering Technique on Scratchapixel, `www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-overview`

total. Rays are traced through linearly spread out pixels in upper part of image plane. We do not shoot through the bottom pixels, because there robot sees only itself, which is also shown in Figure 3.3.

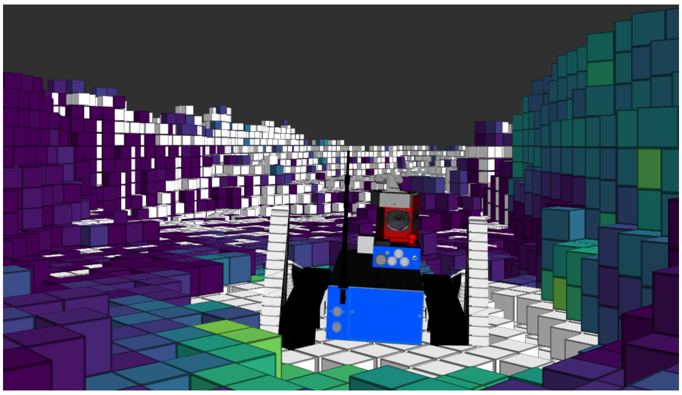


Figure 3.8: Projected image features to built voxel map.

Figure 3.8 shows projection of image features to corresponding heights. Blue or green boxes are coloured based on features projected to them, white blocks are without projection of image features.

For implementing ray-tracing in our traversability estimation, voxel_map Python package provided by Ing. Tomáš Petříček, Ph.D, was used.

Chapter 4

Self-Supervised Learning

As proposed in Section 1.3, self-supervised learning will be used in order to estimate traversability of robot's surroundings from RGB camera images and heightmap. Used data were measured during several human-operated paths through some environments.

We will work with both indoor environments and outdoor environments. Indoor environments consist of paths driven around hallways and staircases. Outdoor environment is a combination of mines and common outdoor terrains, like grass, pavements or roads. Firstly, these environments will be trained separately and afterwards combined for training as much general neural network as possible.

4.1 Labels Estimation

In order to train any neural network, labels for our input training data must be obtained. Ideal approach would be assign labels manually to precisely decide whether is area traversable or not.

Another possibility is to simulate, where would the robot end up after placing it on a given surface and decide whether its position is safe enough to also traverse such surface. Such decision is mainly done by pitch and roll angles.

Simulation of this problem is provided by my supervisor doc. Ing. Karel Zimmermann, Ph.D. It is based on avoiding collision with surface and also minimizing potential energy of robot. Unfortunately, it struggles with time as well as manual labelling. Our heightmap has resolution 256 × 256, and every point of this heightmap should be simulated. One simulation takes about 300 ms which results in approximately 5 and a half hours per heightmap.

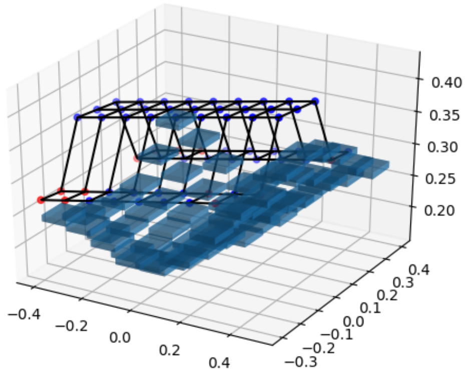
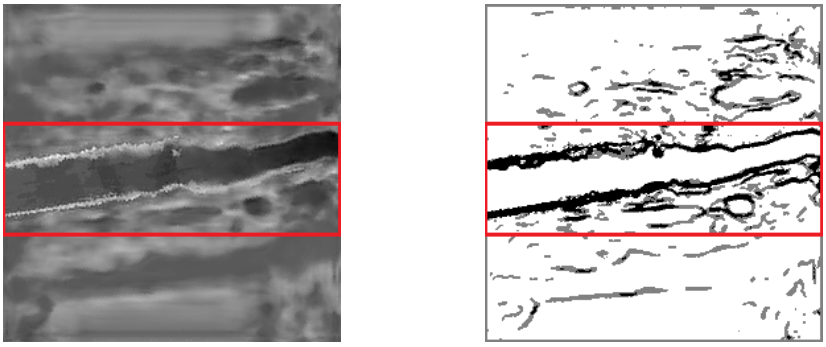


Figure 4.1: Visualization of robot's position estimated on given patch of heightmap, with robot drawn as a grid skeleton.

Unfortunately, we are unable to spend hours of manual labelling or computing simulation of robots position. More straightforward method for labels estimation is proposed, which uses linear regression on local areas of heightmap. The plane is fitted to the closest 13 points of heightmap to estimate slope φ of the plane. We define a traversable threshold $\theta_T = 0.5$ rad and non-traversable threshold $\theta_N = 0.8$ rad. If $|\varphi| < \theta_T$ such point is determined as a traversable. If $|\varphi| > \theta_N$ such point is determined as a non-traversable. Otherwise we do not estimate if it is traversable or not and it will not be used for training.

Furthemore, we explicitly say that places which were truly driven through by operator are surely traversable. These methods are combined to get both true negatives based on very steep slope, e. g. walls, and true positives, which are flat enough or were overcome in past.



(a) : Visualization of heightmap as bird-view grayscale.

(b) : Visualization of estimated labels (white - traversable, black - non-traversable, grey - uncertain).

Figure 4.2: Estimated labels (b) to its input heightmap (a) with marked visible area of sight of robot (red rectangle).

As can be seen in Figure 4.2, traversability is estimated nearly correctly. Black parts (non-traversable) correspond to walls, and white parts (traversable) correspond to flat way in mine.

4.2 Training CNN

With labelled dataset the last step in workflow is to design and train convolutional neural network to estimate traversability of robots environment. Input consists of data prepared as described in Chapter 3 and labels estimated in Section 4.1.

4.2.1 CNN architecture

For our purpose of estimating traversability, convolutional neural network with convolutional layers, max-pooling layers and upsampling layers is designed. Full architecture is visualized in Figure 4.3. Input is tensor with shape $256 \times 256 \times 17$ according to used heightmap plus 16 image features channels. Output is $256 \times 256 \times 2$ tensor with 2 channels of class probabilities, first channel is probability of traversability for each point in heightmap, second channel is probability of non-traversability.

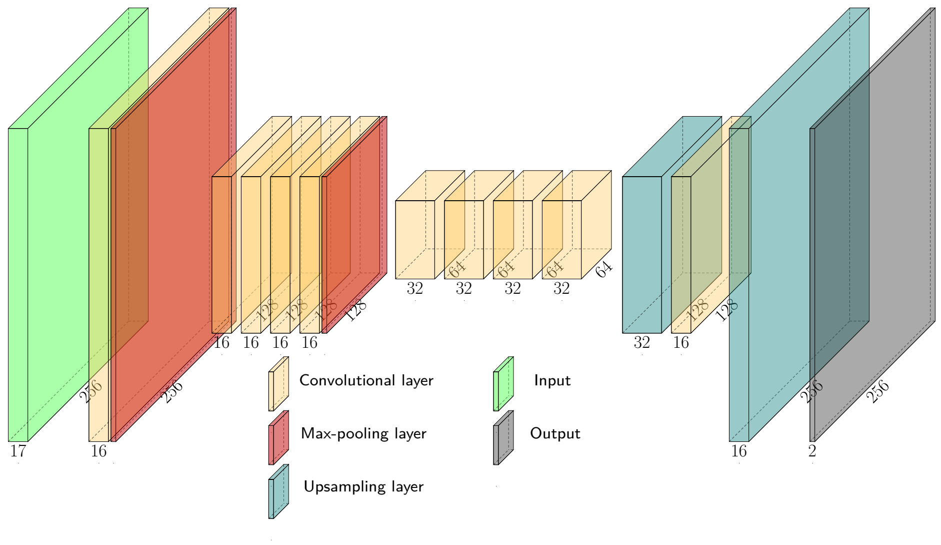


Figure 4.3: Architecture of CNN used for traversability estimation.

4.2.2 Training

Python module PyTorch is used for building and training the designed CNN. As an optimizer Adam optimizer with learning rate $\alpha = 0.001$ was used.

As a loss function we used common segmentation task loss called Cross-Entropy loss, given by formula:

$$H(\mathbf{w}) = \sum_{i}^{N} -y_i \log \mathbf{s}_{y_i}(\mathbf{f}(\mathbf{x}_i, \mathbf{w})), \tag{4.1}$$

where $\mathbf{s}(\mathbf{f}(\mathbf{x}, \mathbf{w}))$ stands for softmax function in order to get class probability. The problem in this thesis is defined as two class, the loss function is actually Binary Cross-Entropy loss, where $N = 2$ in (4.1).

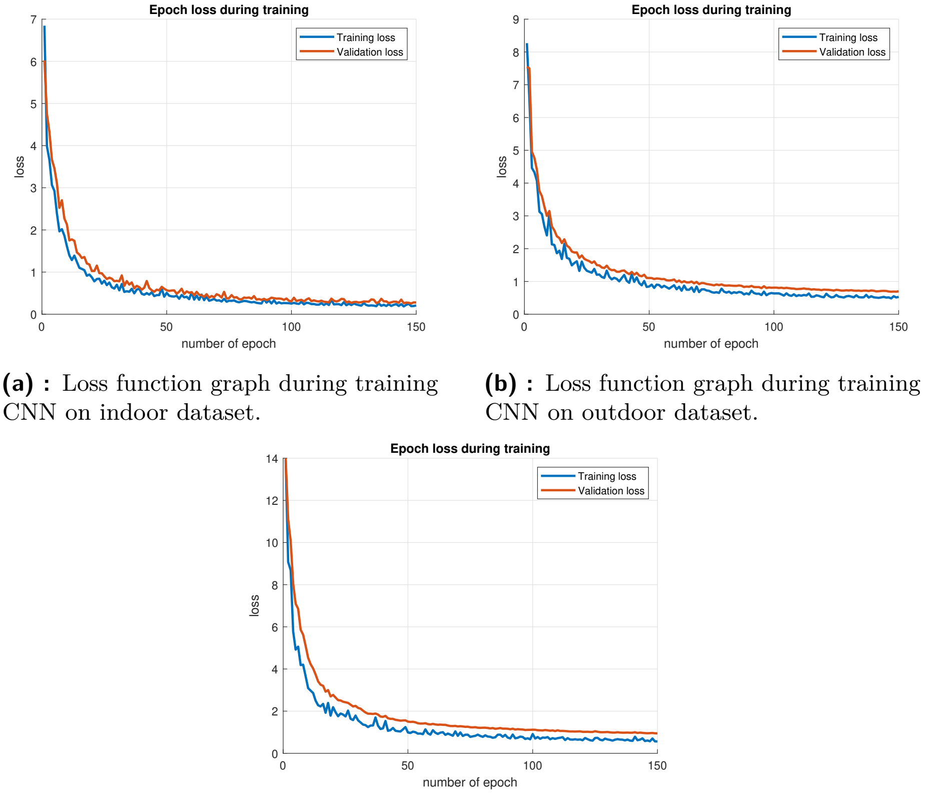
Designed CNN was trained on 3 datasets, indoor dataset, outdoor dataset and combination of indoor and outdoor dataset.

Indoor dataset The indoor dataset consists in total of 1253 training sets of data and 162 validation sets of data. Data are obtained from bagfiles of movement along hallways with easy obstacles such as palette, and staircases in both directions.

Outdoor dataset As for outdoor dataset we prepared 1465 training and 153 validation sets of data from bagfiles from mines and outdoor environment such as roads, pavements, grass including high grass with obstacles as well.

Combination of indoor and outdoor dataset For combination both indoor and outdoor dataset were merged together which results in 2718 training and 315 validation sets of data.

Loss on validation data follows loss on training data, as shown in Figure 4.4. It means that the designed CNN does not extremely overfit to training set. Further training with more epochs was done, but there was not any useful improvement.



(a) : Loss function graph during training CNN on indoor dataset.

(b) : Loss function graph during training CNN on outdoor dataset.

(c) : Loss function graph during training CNN on combination of indoor and outdoor dataset.

Figure 4.4: Binary Cross-Entropy loss function during training CNN.

Chapter 5

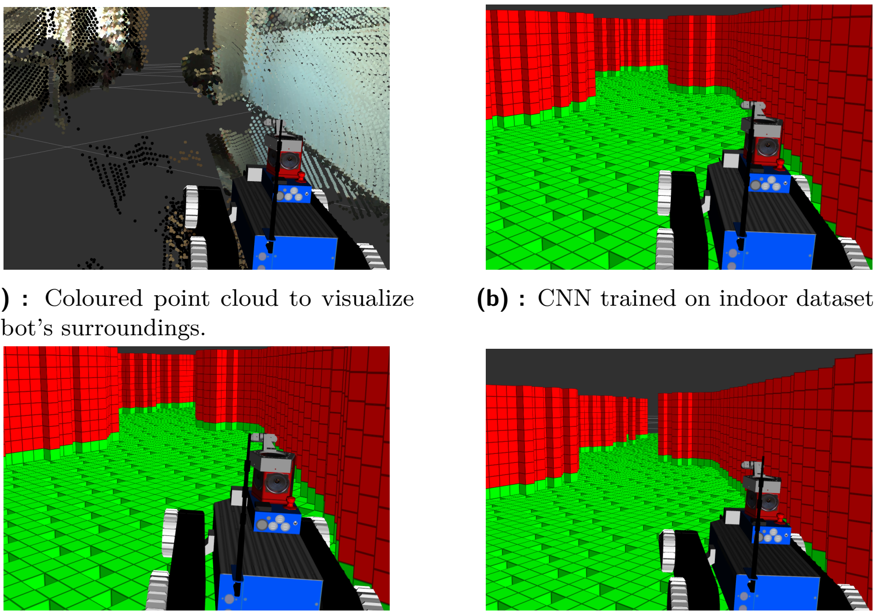
Experimental Results

In this chapter, the trained convolutional neural networks from Section 4.2 will be tested in various environments which are the same as in which the CNN was trained.

5.1 Flat Terrains

Let's start with easy samples, such as hallways or just flat terrain.

Hallway case is easy to evaluate in terms of traversability. All CNNs made correct decision as can be seen in Figure 5.1, despite the fact, that there were obstacles, such as chair, which can be seen as a non-traversable patch directly in front of robot, or palette, which is currently under robots body as a traversable obstacle.



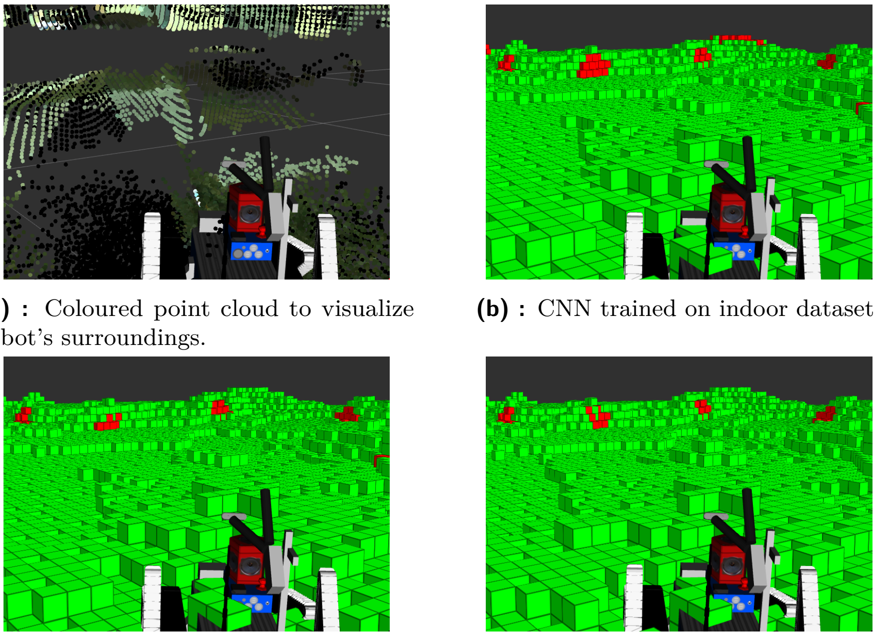
(a) : Coloured point cloud to visualize robot's surroundings.

(b) : CNN trained on indoor dataset.

(c) : CNN trained on outdoor dataset.

(d) : CNN trained on combination of indoor and outdoor dataset.

Figure 5.1: Visualization of traversability in hallway. Green voxels refer to traversable, red to non-traversable.



(a) : Coloured point cloud to visualize robot's surroundings.

(b) : CNN trained on indoor dataset.

(c) : CNN trained on outdoor dataset.

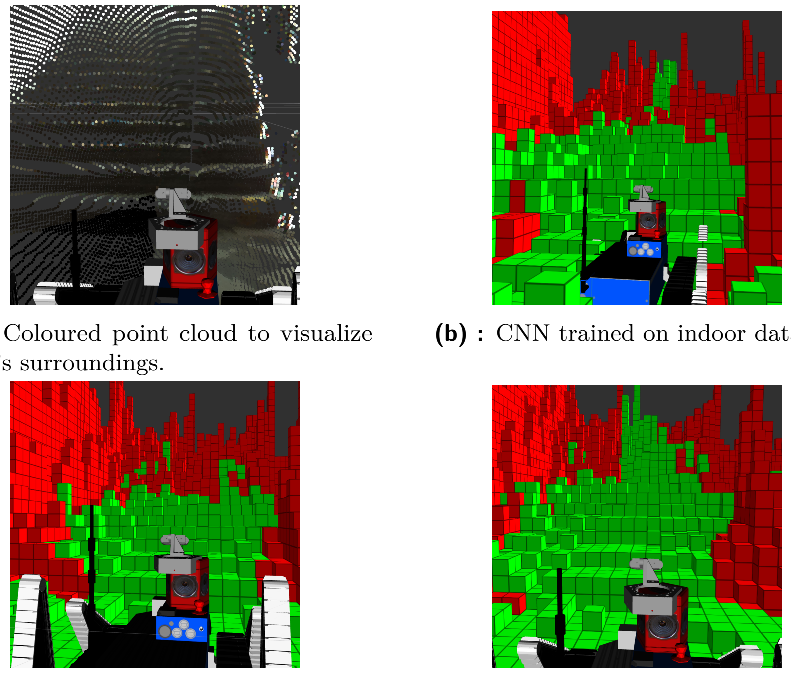
(d) : CNN trained on combination of indoor and outdoor dataset.

Figure 5.2: Visualization of traversability on flat terrain. Green voxels refer to traversable, red to non-traversable.

Figure 5.2 shows visualization of traversability on a flat terrain which should be entirely traversable. As well as in case of hallways, all CNNs provide correct estimation. CNNs trained on outdoor and on combined dataset are a little bit more accurate in comparison with CNN trained on indoor dataset. There are few points estimated incorrectly, which is probably due to imperfectly filled heights. But those errors occur pretty far from robot's current position and they would be corrected during exploration lately.

5.2 Staircases

The designed network will be tested on a harder terrain - staircases, as well.



(a) : Coloured point cloud to visualize robot's surroundings.

(b) : CNN trained on indoor dataset.

(c) : CNN trained on outdoor dataset.

(d) : CNN trained on combination of indoor and outdoor dataset.

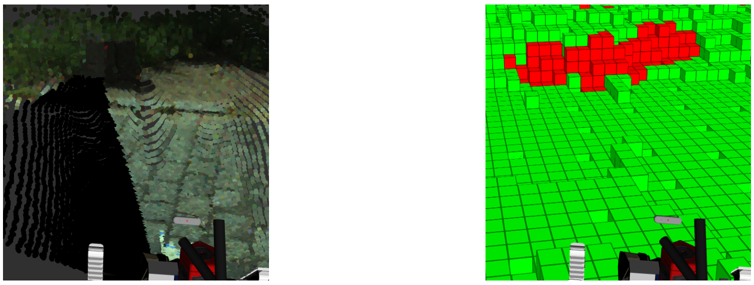
Figure 5.3: Visualization of traversability on staircases. Green voxels refer to traversable, red to non-traversable.

Staircases themselves are harder to analyse. For example, they can be traversed only in a specific direction, which we do not take in consideration in this thesis. Even filling up heightmap is problematic, as our LiDAR does not see, where the end of stairs is and what follows next. For this reason, it is easy to determine wrong heights to be analysed, which is shown in Figure 5.3.

The result of traversability estimation on staircases is visualized in Figure 5.3. We can see, that all CNNs outputs stairs as a traversable, but indoor and combined CNNs outperform outdoor CNN. That is expected, as outdoor CNN was not trained on stairs at all. In upper parts of staircases, best estimation provides combined CNN even with big height differences, which are suppressed thanks to image features.

5.3 Prepared Experiment

The desired goal was to be able to distinguish similar heights with different image features. As an experiment we used high grass next to rocks with almost the same height. Due to the problem's complexity the results will be discussed separately for each trained CNN.

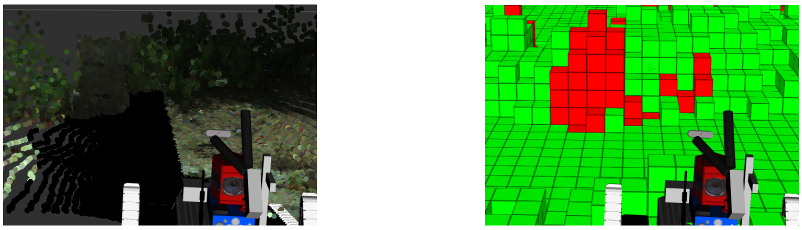


(a) : Coloured point cloud to show position of rock and grass.

(b) : Estimated traversability from indoor CNN.

Figure 5.4: Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on indoor dataset.

CNN trained on indoor dataset performed as expected, it estimates grass and rocks non-traversable as they are too high to be traversed, see Figure 5.4. It is caused by training, where this CNN has not been trained on grass nor rocks features.

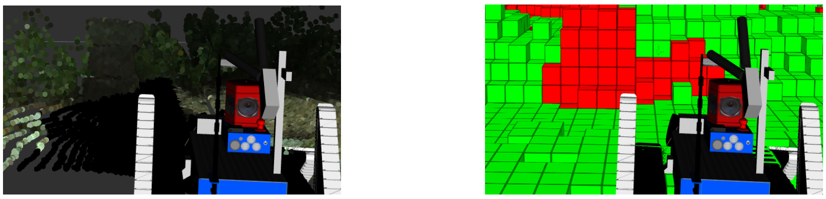


(a) : Coloured point cloud to show position of rock and grass.

(b) : Estimated traversability from outdoor CNN.

Figure 5.5: Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on outdoor dataset.

CNN trained on outdoor dataset provides much better results. It distinguished rocks from grass and outputs high grass as traversable and rocks as non-traversable, which is visualized in Figure 5.5. We can also see that traversability estimation is not that perfect, as there are a few voxels estimated as non-traversable even though they should be traversable.

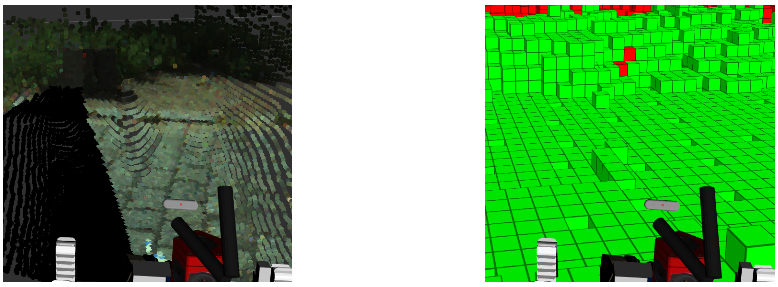


(a) : Coloured point cloud to show position of rock and grass.

(b) : Estimated traversability from combined CNN.

Figure 5.6: Visualization of traversability on experiment with grass and rocks with the same height. CNN trained on combined dataset.

The general convolutional neural network trained both on indoor and outdoor data provides result of this experiment somewhere between the other 2 CNNs. In Figure 5.6 we can see it outputs rock as non-traversable, which is correct. But as for grass, it outputs it as traversable and also non-traversable. Such result is more successful than indoor CNN, but also a bit worse than outdoor CNN.



(a) : Coloured point cloud to show position of rock and grass.

(b) : Estimated traversability from combined CNN.

Figure 5.7: Visualization of wrong traversability on experiment with grass and rocks with the same height. CNN trained on combined dataset.

Occasionally networks output completely wrong results as shown in Figure 5.7.

To sum up, outdoor and combined CNN are able to distinguish rocks from grass, but height information is still pretty powerful feature and makes them decide wrong in some cases. Indoor CNN was not successful in rocks from grass distinction, on the other hand, it is correct because there were no rocks or grass in the indoor training dataset.

Chapter 6

Discussion and Conclusion

6.1 Discussion

Three CNNs with the same architecture (Section 4.2.1) were trained but on different datasets - indoor, outdoor and combination of indoor and outdoor data. Results of each CNN are visualized and evaluated in Chapter 5.

All CNNs can be used for easy terrain traversability estimation, where they provide precise results, as evaluated in Section 5.1. On the other hand, for such terrain such complex approach is not needed and for example, only heightmap information could have been used for these. More challenging terrain such as staircases, analysed in Section 5.2, uses additional measurements obtained from RGB cameras to be able to "correct" heights errors with image features. Especially for staircases, both indoor and combined CNN can be used, as they are both trained on such terrain.

The most challenging problem with same heights, but different traversability, is for example the experimental situation with high grass and rocks described in Section 5.3. This problem requires RGB image features to be able to distinguish grass from rock. Outdoor and combined CNN was trained on grass and rocks as well, so they provide some distinction. For robot's safety an output threshold can be set in order to define how certain output of CNN is desired. For this reason, false negatives (traversable spots estimated inaccurately as non-traversable) can be preferred and such approach can be considered as more conservative.

For real usage, probably most useful would be the general CNN trained both indoor and outdoor as it provides pretty reasonable results. The CNN trained on indoor dataset can be also used for more cautious approach.

There are few reasons, why the networks designed in this thesis lack higher precision. Firstly, with image features we would need bigger datasets. Image features analysis is pretty complex task and even with pretrained segmentation ERFNet more training data is needed to train better CNN. Secondly, the estimation of labels for the input data was rough. With precise labelling, more precise outputs of network can be expected.

Another imperfection in our approach is filling heights to blind spots in depth measuring. Even though we use also image features, we still rely on heights heavily. For example, high rocks (as used in our experiment) are not traversable, but the same rocks could be traversable if they were flat.

6.2 Conclusion

In this thesis, a workflow for traversability estimation from heightmap and RGB images obtained from LiDAR and RGB cameras using convolutional neural network was proposed. We trained those from scratch and described results on several terrains.

Satisfying results were achieved on easier terrains such as hallways, mines or flat ground. We also achieved interesting results on challenging terrains such as staircases or high grass, where our networks worked with some imperfections.

Can the designed networks be used for real terrain traversability estimation? Our models of networks are not perfect, but they could be used for simpler terrains safely. For harder problems we should probably firstly fine-tune networks for such environment, if it is possible. Otherwise we should desire a higher certainty on the output of CNN for more conservative approach, in order not to damage our robot. Also time requirements are pretty high for our method, so it is not useful for fast decisions about traversability without powerful hardware including GPU on our robot.

6.3 Future Work

Some points of possible improvements were already discussed in Section 6.1. We will continue with that in this section.

For improving precision of CNN bigger and more accurate dataset is required. Dataset should contain all possible environments, which can robot face in future tasks. Another improvement could be in data, where filled heights to blind spots are not correct enough, e. g. staircases. That could be fixed with fine-tuning CNN used for filling heights.

Input data are followed with labels, which should be also improved. Ideally human-labelled or at least labelled as described in Section 4.1 using simulation of position of robot. But in combination with bigger dataset, this would be very time demanding.

Bibliography

[BVS$^+$13] I. Bogoslavskyi, O. Vysotska, J. Serafin, G. Grisetti, and C. Stachniss, *Efficient traversability analysis for mobile robots using the Kinect sensor*, 09 2013, pp. 158–163.

[CGGGG18] R. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, *Learning Ground Traversability From Simulations*, IEEE Robotics and Automation Letters **3** (2018), 1695–1702.

[HS17] K. Hata and S. Savarese, *CS231A Course Notes 1: Camera Models*, https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf (2017).

[JGH08] Q. Cao J. Gu and Y. Huang, *Rapid Traversability Assessment in 2.5D Grid-based Map on Rough Terrain*, International Journal of Advanced Robotic Systems **5** (2008), 1–6.

[Kit17] K. Kitani, *Camera Matrix*, http://www.cs.cmu.edu/~16385/s17/Slides/11.1_Camera_matrix.pdf (2017).

[MB12] D. Maier and M. Bennewitz, *Appearance-Based Traversability Classification in Monocular Images Using Iterative Ground Plane Estimation*, 10 2012.

[MBS11] D. Maier, M. Bennewitz, and C. Stachniss, *Self-supervised Obstacle Detection for Humanoid Navigation Using Monocular Vision and Sparse Laser Data*, 2011 IEEE International Conference on Robotics and Automation, May 2011, pp. 1263–1269.

[NHH15] H. Noh, S. Hong, and B. Han, *Learning Deconvolution Network for Semantic Segmentation*, 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1520–1528.

[OGC17] L. Ortiz, L. Gonçalves, and E. Cabrera, *A Generic Approach for Error Estimation of Depth Data from (Stereo and RGB-D) 3D Sensors*, 05 2017.

[RA18] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, *ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation*, IEEE Transactions on Intelligent Transportation Systems **19** (2018), no. 1, 263–272.

[Seb19] B. Sebastian, *Traversability Estimation Techniques for Improved Navigation of Tracked Mobile Robots*, https://vtechworks.lib.vt.edu/handle/10919/94629 (2019).

[SRL$^+$19] V. Suryamurthy, S. Raghavan, A. Laurenzi, N. Tsagarakis, and D. Kanoulas, *Terrain Segmentation and Roughness Estimation using RGB Data: Path Planning Application on the CENTAURO Robot*, 09 2019.

[WDR$^+$19] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, *Where Should I Walk? Predicting Terrain Properties From Images Via Self-Supervised Learning*, IEEE Robotics and Automation Letters **PP** (2019), 1–1.