CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD THESIS

ISSN 1213-2365

Coupled Learning and Planning for Active 3D Mapping

Tomáš Petříček

petrito1@fel.cvut.cz

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of study: Artificial Intelligence and Biocybernetics

CTU–CMP–2017–06

November 3, 2017

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/petricek/Petricek-TR-2017-06.pdf

Supervisor: Tomáš Svoboda
Supervisor-Specialist: Karel Zimmermann

Received funding from the EU under grant agreements FP7-ICT-247870 NIFTi, FP7-ICT-609763 TRADR, No. 692455 Enable-S3; the Czech Science Foundation under Project GA14-13876S, Project 17-08842S; and the Grant Agency of the CTU in Prague, grant No. SGS11/125/OHK3/2T/13, SGS13/142/OHK3/2T/13, and SGS15/081/OHK3/1T/13

Research Reports of CMP, Czech Technical University in Prague, No. 6, 2017

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Coupled Learning and Planning for Active 3D Mapping

Tomáš Petříček

November 3, 2017

Acknowledgement

I am thankful for the continuous support I received from my supervisors, Tomáš Svoboda and Karel Zimmermann, who guided me through my doctoral studies. The discussions we were having and collaboration on the research presented here were invaluable. Also, I would like to thank all the colleagues with Center of Machine Perception and Department of Cybernetics who create such an inspiring environment for research. My gratitude also goes to the many colleagues I collaborated with on the NIFTI and TRADR projects, where we were developing the search & rescue platform used in experiments presented here.

I gratefully acknowledge receiving funding from the European Union[1] under grant agreements FP7-ICT-247870 NIFTi, FP7-ICT-609763 TRADR, No. 692455 Enable-S3; the Czech Science Foundation under Project GA14-13876S, Project 17-08842S; and the Grant Agency of the CTU in Prague, grant No. SGS11/125/OHK3/2T/13, SGS13/142/OHK3/2T/13, and SGS15/081/OHK3/1T/13.

I hereby certify that the results presented in this thesis were achieved during my own research in cooperation with my supervisors, Tomáš Svoboda and Karel Zimmermann, and the coauthors of the respective publications, notably Vojtěch Šalanský.

[1] Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

Abstract

The thesis presents several results in the area of 3D perception, with focus on combining learning and planning in active 3D mapping.

Autonomous robots, including those deployed in search and rescue operations or autonomous vehicles, must build and maintain accurate representations of the surroundings to operate efficiently and safely in human environment. These representations, or maps, should encompass both low-level information about geometry of the scene and high-level semantical information, including recognized categories or individual objects.

In the first part we propose a method of 3D object recognition based on matching local invariant features, which is further extended for 3D point cloud registration task and evaluated on challenging real-world datasets. The method builds on a multi-stage feature extraction pipeline composed of sparse keypoint detection to reduce complexity of further stages, establishing local reference frames as a means to achieve invariance with respect to rigid transformations without sacrificing descriptiveness of the underlying 3D shape, and a compact description of the shape based on area-weighted normal projections. For a moderate overlap between the laser scans, the registration method provides a superior registration accuracy compared to state-of-the-art methods including Generalized ICP, 3D Normal-Distribution Transform, Fast Point-Feature Histograms, and 4-Points Congruent Sets.

In the second part, two tasks from the area of active 3D mapping are being solved—namely, simultaneous exploration and segmentation with a mobile robot in a search and rescue scenario, and active 3D mapping using a sensor with steerable depth-measuring rays, with applications in autonomous driving. For these tasks, we assume that the localization is provided by an external source.

In the simultaneous exploration and segmentation task, we consider a mobile robot exploring an unknown environment along a known path, using a static panoramic sensor providing RGB and depth measurements, and controlling a narrow field-of-view thermal camera mounted on a pan-tilt unit. The task is to control the sensor along the path to maximize accuracy of segmentation of the surroundings into human body and background categories. Since demanding optimal control does not allow for online replanning, we rather employ the optimal planner offline to provide guiding trajectories for learning a CNN-based control policy in a guided Q-learning framework. A policy initialization is proposed which takes advantage of a special structure of the task and allows efficient learning of the policy.

In the active 3D mapping task, our method simultaneously learns to reconstruct a dense 3D occupancy map from sparse measurements and optimizes the reactive control of depth-measuring rays. We propose a fast prioritized greedy algorithm to solve the control subtask online, which needs to update the cost function in only a small fraction of possible rays in each iteration. An approximation ratio of the algorithm is derived. We experimentally demonstrate, using publicly available KITTI dataset, that accuracy of the 3D improves significantly when learning-to-reconstruct is coupled with the optimization of depth measuring rays.

Resumé

Disertační práce představuje několik výsledků z oblasti 3D vnímání, se zaměřením na kombinaci učení a plánování v oblasti aktivního 3D mapování.

Autonomní roboty, včetně pátracích a záchranných robotů či autonomních aut, musí stavět a udržovat přesné reprezentace okolního světa, aby mohly efektivně a bezpečně pracovat v prostředí společně s lidmi. Tyto reprezentace neboli mapy by měly zahrnovat jak informaci nižší úrovně abstrakce týkající se geometrie scény, tak sémantickou informaci včetně rozpoznaných kategorií či jednotlivých objetků.

V první části navrhujeme metodu rozpoznávání trojrozměrných objektů založené na párování lokálních invariantních příznaků, kterou dále rozšiřujeme pro úlohu registrace množin trojrozměrných bodů a vyhodnocujeme s využitím několika sad náročných reálných dat. Navržená metoda staví na extrakci příznaků sestávající z detekce význačných bodů pro omezení složitosti navazujících kroků, ustanovení lokálních souřadných systémů pro dosažení invariance vzhledem ke shodné transformaci se zachováním rozlišujicích trojrozměrného tvaru, a kompaktním popisem (deskriptorem) daného tvaru, založeném na plochou vážené projekci normálových vektorů povrchu. Pro střední úroveň překryvu mezi laserovými měřeními tato registrační metoda poskytuje vynikající přesnost registrace v porovnání s metodami současného stavu poznání—s metodami Generalized ICP, 3D Normal-Distribution Transformation, Fast Point-Feature Histograms a 4-Points Congruent Sets.

Ve druhé části řešíme dvě úlohy z oblasti aktivního 3D mapování, konkrétně simultánní průzkum a segmentaci s mobilním robotem v pátrací a záchranné misi a aktivní trojrozměrné mapování s využitím dálkového senzoru s řiditelnými měřicími paprsky, s možným využitím v autonomních vozidlech. Pro tyto úlohy předpokládáme, že lokalizaci poskytuje externí systém.

V úloze simultánního průzkumu a segmentace uvažujeme mobilní robot, který prozkoumává neznámé prostředí na přibližně známé trase, využívá statický senzor poskytující barevná (RGB) a hloubková měření a řídí tepelnou kameru s úzkým úhlem pohledu připevněnou otočně-sklopné jednotce. Úloha spočívá v řízení pohybu senzoru podél trasy s cílem maximalizovat přesnost segmentace prostředí do kategorií člověk a pozadí. Protože náročné optimální řízení nedovoluje přeplánovávat online, raději využíváme optimální plánování offline pro generování trajektorií usměrňujících učení řídicí strategie založené založeném na konvoluční neuronové síti prostřednictvím algoritmu guided Q-learning. Navrhujeme využití zvláštní struktury úlohy pro inicializaci parametrů řídicí strategie, která umožňuje efektivní účení.

Pro úlohu aktivního trojrozměrného mapování navrhujeme metodu simultánní rekonstrukce kompletní mapy obsazenosti z řídkých měření a optimalizace reaktivního řízení měřicích paprsků. Pro řešení řídicí podúlohy online dále navrhujeme rychlý prioritní hladový algoritmus, který vyžaduje aktualizaci nákladové funkce v každé iteraci pouze u malého zlomku uvažovaných paprsků. U tohoto algoritmu odvozujeme aproximační poměr. Experimentálně ověřujeme, s využitím veřejně dostupné sady dat KITTI, že přesnost trojrozměrné rekonstrukce se významně zvýší, je-li učení jak mapovat svázáno s plánováním měřicích paprsků.

Contents

List of Figures

List of Tables

List of Algorithms

Notation

a, x, N	scalars (in italics)
$\mathbf{x}$	column vector (in lowercase bold)
$\mathbf{A}$	matrix (in uppercase bold)
$\mathbf{x}^\mathsf{T}$	vector $\mathbf{x}$ transposed, a row vector
$\mathbf{0}$	column vector of zeros of appropriate size
$\begin{bmatrix} a & b & c \end{bmatrix}$	row vector with 3 elements
$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}$	a matrix composed of blocks—matrix $\mathbf{R}$, column vector $\mathbf{t}$, row vector $\mathbf{0}^\mathsf{T}$, and 1
$\|\mathbf{x}\|$	Euclidean norm of vector $\mathbf{x}$, $\sqrt{\mathbf{x}^\mathsf{T}\mathbf{x}}$
$\mathcal{P}$, $\mathcal{S}$	sets
$\{1, \ldots, N\}, \{i\}_{i=1}^N$	set of integers from 1 to N
$\{\mathbf{x}_i\}_{i=1}^N$	set of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$
$(1, \ldots, N)$	tuple (sequence) of integers from 1 to N
$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	the set of natural, integer, and real numbers, respectively
$\operatorname{argmin}, \operatorname{argmax}$	minimizer and maximizer, respectively
$f : \mathcal{X} \to \mathcal{Y}$	function f with domain $\mathcal{X}$ and range $\mathcal{Y}$
$f(x)$	function of x, $f : \mathcal{X} \to \mathcal{Y}$, $x \in \mathcal{X} \subseteq \mathbb{R}$, $f(x) \in \mathcal{Y} \subseteq \mathbb{R}$
$g(\mathbf{x}) = g(x_1, \ldots, x_M)$	scalar function of $\mathbf{x}$, $g : \mathcal{X} \to \mathcal{Y}$, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^M$, $g(\mathbf{x}) \in \mathcal{Y} \subseteq \mathbb{R}$
$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) & \cdots & h_N(\mathbf{x}) \end{bmatrix}^\mathsf{T}$	vector-valued function of $\mathbf{x}$, $\mathbf{h} : \mathcal{X} \to \mathcal{Y}$, $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^M$, $\mathbf{h}(\mathbf{x}) \in \mathcal{Y} \subseteq \mathbb{R}^N$

Abbreviations

1D, 2D, ...	one-dimensional (one dimension), two-dimensional etc.
CNN	convolutional neural network
DAG	directed acyclic graph
DNN	deep neural network
DQN	deep Q-network estimating state-action values
FN, FP, TN, TP	false negative, false positive, true negative, true positive, respectively, or the number of such decisions (see Table 2.2)
FNR, FPR, ...	false negative rate, false positive rate, ..., respectively (see Section 2.4)
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
IL	imitation learning
MILP	mixed-integer linear program
RANSAC	Random Sample Consensus [27]
RGB, ...	color channels red, green, blue, and image data composed thereof; extra data channels may follow—e.g. RGBD for an extra depth channel, RGBT for an extra thermal channel.
RL	reinforcement learning
SES	simultaneous exploration and segmentation
SGD	stochastic gradient descent
SL	supervised learning
SLAM	simultaneous localization and mapping
SVD	singular value decomposition
s.t.	subject to

1. Introduction

Building and maintaining accurate representation of the surroundings, commonly called mapping, is a crucial task underpinning many applications of autonomous systems such as search & rescue robots or autonomous vehicles. A low-level representation in form of a metric or topological map is needed for navigating the environment, a high-level representation, e.g., in form of segmentation of the map into semantic categories, is needed to perform more complex tasks such as searching for human victims or fire sources with search & rescue robots, or following traffic laws with autonomous cars.

Mapping is not possible without the robot simultaneously localizing itself in the map, because prior updating the map with new measurements, these must be first registered into the common reference frame. The task is therefore generally described as simultaneous localization and mapping (SLAM). Indeed, without a reliable SLAM capability, all the use cases of such systems where we cannot hope to obtain precise and complete map in advance would remain unfeasible—this would typically include any use case dealing with a dynamic environment. Its solution, or its parts, to this day often depend on local feature descriptors and their matching to establish correspondences and estimate relative movement of the robot, typically using a robust estimation method from the RANSAC family [27]. Other cues may be used, where available, including odometry or readings from an inertial measurements unit (IMU), to constrain the process of establishing the correspondences prior to using a local search such as iterative closest point (ICP) [12, 15]. Many methods in the past decades dealt with designing methods related to the local-feature machinery [11, 41, 102], including detecting keypoints [95], establishing local reference frames [67] at these keypoints, or creating descriptors of their local neighborhood (see [31] for a survey). Every stage of feature construction may be negatively affected by a non-uniform sampling density if not treated carefully—Tombari *et al.* [95] consider this to be one of the main open issues severely decreasing the performance of existing methods. We focus on feature-matching methods applied to 3D object recognition and point cloud registration addressing this issue in chapters 3 and 4.

For many vision tasks, purposive control of the sensors is provably superior to passive perception only and leads to well-posed and stable problems [5]. Recently, active vision have been employed in combination with expressive deep neural networks in several areas including multi-view object recognition [40] or scene categorization [37]. Also, it has been shown [47] that deep neural-network visuomotor policies can be trained end-to-end to solve real-world manipulation tasks, and that learning task-specific features dramatically improves the success rate of the primary manipulation task. Levine *et al.* [47] use a guided policy search method to transform policy search into a supervised learning problem, with supervision being provided by a simple trajectory-centric reinforcement learning method.

Segmentation of objects in an unknown environment from sensory data gathered by a mobile robot is a common task to be solved in search & rescue missions. Currently, deploying fully autonomous rescue robots is problematic, for example, due to limited trust to such systems from the human team members [43] and due to strict protocols which must be followed. In a typical scenario, a coarse exploration path is provided by either a human operator or a global planner, along which measurements can be collected

and processed to support making mission-related decisions. Since most of the sensors have a limited field of view and exploration time is also limited, the resulting coverage of the environment by the sensors is usually limited and a natural question arises—what should be measured with which sensor to maximize recognition performance of the system? We call this problem *simultaneous exploration and segmentation with incomplete data* (SES). A simplified version of this problem can be formulated as a mixed-integer linear program (MILP), nevertheless, the dimensionality of the planning task is huge and does not allow for real-time replanning when new data arrive—a reactive control policy would be desirable. We address this problem in chapter 5.

Accurate 3D perception is also a key capability of autonomous vehicles as it is a necessary condition for safe operation and following traffic laws. Besides being necessary for full autonomy, it is an essential component for many capabilities in driver-assistance systems in general, such as lane following, emergency braking or predictive active damping. Due to the current rotating lidars being expensive, heavy, and due to the fact that they contain moving parts which are prone to mechanical wear, several manufacturers have announced development of cheaper and smaller solid-state lidars (SSL) without moving parts [2, 3, 76]. Notably, solid-state lidars from Quanergy shall allow controlling individual measuring rays, for example, focusing in most interesting directions with respect to the current traffic situation. This presents a major advantage but also a challenging multidimensional control task of planning around half a million measurements per second. We address this problem in chapter 6 [105].

1.1. Goals of the Thesis

- Extracting local invariant features is negatively affected by a non-uniform sampling density if not treated carefully [95]. One of the goals is thus to address the problem of uneven sampling density inherent in typical range-sensing methods and to propose a method of establishing local descriptors invariant to rigid transformation which addresses this issue.
- Motivated by our experience with developing a mobile search & rescue robot and ICP-based mapping aided by IMU and odometry, we investigate the possibilities of extending the method previously mentioned to address the 3D point cloud registration problem, as a suitable building block of a SLAM pipeline which is not susceptible to initial pose estimates provided by dead-reckoning.
- We consider an instance of the simultaneous exploration and segmentation problem using the search & rescue platform equipped with a thermal camera mounted on a pan-tilt unit. We note, that a simplified version of such a task can be formulated as a mixed-integer linear program (MILP), nevertheless, the dimensionality of the exploration planning task is huge and does not allow for real-time replanning when new data arrive. Our goal is therefore to find a reactive control policy from a limited number of data gathered by the real platform, because capturing such data cannot be done automatically and requires time-consuming manual annotation.
- Propose a method for active 3D mapping for the depth sensors which allow control of individual depth-measuring rays, such as the newly emerging solid-state lidars. That is, a method which simultaneously reconstructs a dense 3D occupancy map from sparse depth measurements and optimizes reactive control of the depth-measuring rays.

1.2. Contribution of the Thesis

- In chapter 3, we present a method for feature-based 3D object recognition in cluttered scenes dealing with the problem of non-uniform sampling density which is inherent in typical range sensing applications. We suggest a method operating on polygonal meshes which overcomes the problem by accounting for the respective surface area in both establishing local frames and creating feature descriptors. The method is able to recognize even highly occluded objects and outperforms state of the art in terms of recognition rate on a standard publicly available dataset.
  The corresponding paper [68] was presented at 21th International Conference on Pattern Recognition (ICPR), 2012 (CORE rating: B).
- In chapter 4, we extend this method further for the task of point cloud registration and evaluate it on challenging real-world datasets, with focus given to evaluation of its individual components. For a moderate overlap between the laser scans, the method provides a superior registration accuracy compared to state-of-the-art methods including Generalized ICP, 3D Normal-Distribution Transform, Fast Point-Feature Histograms, and 4-Points Congruent Sets. Compared to the surface normals, the points as the underlying features yield higher performance in both keypoint detection and establishing local reference frames. Moreover, sign disambiguation of the basis vectors proves to be an important aspect in creating repeatable local reference frames. A novel method for sign disambiguation is proposed which yields highly repeatable reference frames.
  This work was accepted for publication in PLOS ONE [69] (Impact Factor 2016: 2.806, 5 year: 3.394).
- In chapter 5, we consider the problem of pan-tilt sensor control for active segmentation of incomplete multi-modal data. Since demanding optimal control does not allow for online replanning, we rather employ the optimal planner offline to provide guiding samples for learning of a CNN-based control policy in a guided Q-learning framework. The proposed policy initialization and guided Q-learning avoids poor local optima and yields reasonable results from hundreds of roll-outs. The results suggest that the proposed policy outperforms the baseline and is suitable for real-time control.
  Within this work, two multimodal datasets for human segmentation in images were created and made publicly available—one being composed of semi-synthetic images from a structured-light sensor, the other composed of panoramic images captured by a mobile search & rescue platform equipped with a time-of-flight sensor.
  The manuscript was submitted for publication, currently a revised version is under review.
- In chapter 6 [105] we propose an active 3D mapping method for depth sensors, which allow individual control of depth-measuring rays, such as the newly emerging solid-state lidars. The method simultaneously (i) learns to reconstruct a dense 3D occupancy map from sparse depth measurements, and (ii) optimizes the reactive control of depth-measuring rays. To make the first step towards the online control optimization, we propose a fast prioritized greedy algorithm, which needs to update its cost function in only a small fraction of possible rays. The approximation ratio of the greedy algorithm is derived. An experimental evaluation on the subset of the KITTI dataset demonstrates significant improvement in the 3D map accuracy when learning-to-reconstruct from sparse measurements is coupled with the optimization of depth measuring rays.
  Our work was presented [105] at The IEEE International Conference on Computer Vision (ICCV), 2017 (CORE rating: A*).

2. Background and Related Work

In this chapter, we summarize basic notions and previous work which provides background knowledge on various subtopics of the thesis.

First, we present task of point set registration which is related to chapters 3 and 4 and briefly survey:

- methods of estimating aligning pose (motion parameters) in cases when the correspondences are known but the measurements are corrupted with noise (also known as absolute orientation),
- the iterative closest point (ICP) algorithm and its variants as a direct approach to the registration task in case of unknown correspondences,
- random sample consensus (RANSAC) as a means to find the pose in a population of tentative correspondences which contains relatively small number of inliers,
- methods of local invariant features which provide a means to establish tentative correspondences based on shape similarity instead of sole proximity given by the current pose estimate.

Then, we give a brief overview of supervised learning methods which provide a means to learn input-output mapping from a set of training examples, with a focus on training deep neural networks. This relates to chapters 5 and 6 which deal with planning and learning in two active perception tasks.

Chapter Outline

2.1. Registration of 3D Point Sets

2.1.1. Pose from Known Correspondences

Least-squares fitting of two 3D point sets $\mathcal{P}$, $\mathcal{P}'$ with known correspondences is a classic photogrammetric task for which several solutions have been proposed. The problem, also termed *absolute orientation*, lies in finding transformation

$$\operatorname*{argmin}_{\mathbf{T}} \sum_i \|\mathbf{p}'_i - \mathbf{T}(\mathbf{p}_i)\|^2 \tag{2.1}$$

for corresponding points $\mathbf{p}_i \in \mathcal{P}$ and $\mathbf{p}'_i \in \mathcal{P}'$. The solutions to this problem differ mostly in how the transformation is represented, partially also in the class of transformations being considered (rigid transform or similarity). We list the solutions in Table 2.1.

Table 2.1. Solutions to the absolute orientation problem.

Method	Transformation	Representation	Correspondences
Arun *et al.* [7]	rigid transform	orthogonal matrices	points
Horn *et al.* [35]	similarity	orthogonal matrices	points
Horn [34]	similarity	unit quaternions	points
Walker *et al.* [98]	rigid transform	dual quaternions	points, directions

Using orthogonal matrices to represent rotations, Arun *et al.* [7] proposed a solution involving singular value decomposition (SVD) of 3×3 matrix $\mathbf{M} = \sum_i \mathbf{q}_i \mathbf{q}_i'^{\mathsf{T}}$ composed of the corresponding points $\mathbf{q}_i, \mathbf{q}'_i$ with the respective centroids subtracted. An alternative solution using orthogonal matrices was later derived in [35] which involves eigenvalue decomposition of 3×3 symmetric matrix $\mathbf{M}^{\mathsf{T}}\mathbf{M}$. Previously, Horn *et al.* [34] proposed a formulation using unit quaternions to represent rotations where the optimal rotation is found by eigenvalue decomposition of a 4×4 matrix $\mathbf{N}$, which contains sums and differences of the elements from $\mathbf{M}$. The solutions from [35, 34] allows to estimate an extra scale factor s, which gives $\mathbf{p}'_i \approx s\mathbf{R}(\mathbf{p}_i) + \mathbf{t}$. The method of Walker *et al.* [98] allows to incorporate additional correspondences of unit direction vectors $\mathbf{n}_j, \mathbf{n}'_j$, such as surface normals, with overall criterion being (using rotation matrices for brevity)

$$\sum_i \|\mathbf{p}'_i - \mathbf{R}\mathbf{p}_i - \mathbf{t}\|^2 + \sum_j \|\mathbf{n}'_i - \mathbf{R}\mathbf{n}_i\|^2. \tag{2.2}$$

Their method involves computing eigenvalue decomposition of a 4×4 symmetric matrix.

Eggert *et al.* [24] stated that for typical, real-world noise levels, there is no difference in the robustness of the final solutions to (2.1). Despite not being stated explicitly in [24], this may not apply to the case where additional correspondences of direction vectors are available—that case is considered only by the dual-quaternion formulation [98].

2.1.2. Iterative Closest Point—Ad Hoc Correspondences

Method of Iterative Closest Point (ICP) establishes correspondences in an ad hoc fashion between the nearest neighbors in the respective point sets using the current estimate of aligning pose. The procedure of finding nearest neighbors alternates with correcting the pose estimate based on the newly assigned correspondences. We can outline the iterative procedure as follows:

Many variants of the algorithm have been proposed, from the seminal papers on its point-to-point [12] and point-to-plane variants [15], an extension using invariant features to help establish the correspondences [83], to Generalized ICP [81]. A review of registration methods with a particular focus on the ICP family and mobile robotics applications can be found in [71].

2.1.3. Random Sample Consensus—Tentative Correspondences

The Random Sample Consensus (RANSAC) algorithm [27] allows to interpret data containing a significant percentage of gross errors (i.e., outliers). It seeks a model which interprets most of the data, that is, a model with the largest set of inliers called consensus set. In computer vision, such a situation is ubiquitous. Typically, non-Gaussian

Algorithm 2.1. Outline of Iterative Closest Point algorithms.

Require: Point sets $\mathcal{P}$ and $\mathcal{P}'$ to be aligned, initial pose estimate $\mathbf{T}^{(0)}$

1: $t \leftarrow 0$
2: **while** $\neg\text{done}(t, \mathbf{T}^{(t)}(\mathcal{P}), \mathcal{P}')$ **do**
3: $\quad t \leftarrow t + 1$
4: $\quad$ Establish set of correspondences between $\mathbf{p}'_j \in \mathcal{P}'$ and $\mathbf{T}^{(t-1)}(\mathbf{p}_i)$, $\mathbf{p}_i \in \mathcal{P}$ by setting weights $w^{(t)}_{i,j}$ for each pair of points. (Zero weights can be neglected.)
5: $\quad$ Update pose estimate $\mathbf{T}^{(t)} \leftarrow \text{argmin}_{\mathbf{T}} \sum_{i,j} w^{(t)}_{i,j} \|\mathbf{p}'_j - \mathbf{T}(\mathbf{p}_i)\|^2$
6: **end while**
7: **return** $\mathbf{T}^{(t)}$

errors stem from establishing correspondences among noisy data via matching local features. As there are a lot of ambiguities in real-world data when only a closer look is taken and one cannot actively disambiguate alternatives, mismatches are quite frequent and a method robust to such outliers, such as RANSAC, is needed to interpret the data. The basic procedure is outlined in Algorithm 2.2. Several improvements to the original algorithm [27] were proposed [96, 18, 61].

Algorithm 2.2. Outline of the Random Sample Consensus (RANSAC) algorithm.

Require: Data points $\mathcal{P}$, number of iterations $k_{\max}$.

1: Initialize iteration $k \leftarrow 0$.
2: **for** $k \in \{1, \ldots, K\}$ **do**
3: $\quad$ Draw random sample $\mathcal{S}_k \subseteq \mathcal{P}$ of n points, $|\mathcal{S}_k| = n$.
4: $\quad$ Instantiate model $\mathbf{m}_k$ from sample $\mathcal{S}_k$.
5: $\quad$ Find consensus set $\mathcal{S}'_k \subseteq \mathcal{P}$ within some error tolerance of $\mathbf{m}_k$.
6: $\quad$ Iteratively update model $\mathbf{m}'_k$ from consensus set $\mathcal{S}'_k$, and vice versa.
7: **end for**
8: **return** model with the largest consensus set $\mathbf{m}'_{\text{argmax}_{i=1}^{K} |\mathcal{S}'_i|}$

If the inlier ratio w for a single true model is known, the expected number of iterations k needed to select a subset of n good data points is w^{-n}. From another point of view, if we want to ensure with probability z that an outlier-free sample of n data points is drawn at least once in K iterations, that is, $z = 1 - (1 - w^n)^K$, we have to draw $K = \lceil \log(1 - z) / \log(1 - w^n) \rceil$ samples [27].

Note that a lower bound on w can be established online as $w_k = \max_{i=1}^{k} |\mathcal{S}'_i| / |\mathcal{P}| \leq w$. Since the number of samples K needed to obtain an outlier-free sample with probability z is a decreasing function of w, this yields an upper bound $K \leq \lceil \log(1 - z) / \log(1 - w_k^n) \rceil = K_k$, which can be used in practice if inlier ratio w is not known beforehand. Note that from $w_k \leq w_{k+1}$ it follows that $K_k \geq K_{k+1}$ and the iteration can safely be stopped once $k \geq K_k$.

2.2. Local Invariant Features

2.2.1. Keypoint Detection

Keypoint detector searches the surface for points of some distinguished characteristics, which we call *keypoints*. Some methods provide seek also a characteristic scale for each keypoint which typically also identifies the neighborhood from which to extract the fea-

ture. The characteristic scale should be repeatable under the allowed transformations of data. As pointed out by [95], the purpose of the characteristic scale of 3D keypoints differs from that of 2D keypoints. Whereas for 2D keypoints detected in images the characteristic scale is needed to handle inherent scale ambiguity, for 3D keypoints computed from metric scale-unambiguous measurements it identifies the most distinctive neighborhood and enables scale-independent object recognition. Therefore, when recognizing models of a fixed size, scale-space analysis is generally not needed.

2.2.2. Local Reference Frames

Local reference frames, together with a particular parametrization of the neighborhood, are the key means to achieve the desired level of feature invariance. Although the Cartesian coordinate system is the most common today [54, 53, 101, 102], some methods employ only a partial coordinate system [41], and some do not use any local reference frames [21, 79]. Using a full reference frame yields several advantages. First, it does not reduce the information content because 3D position of each point can be fully exploited in the descriptor. Second, when recognizing rigid objects, each tentative correspondence suffices to estimate the object pose. Full reference frame is thus the preferred choice.

2.2.3. Descriptors

According to [94] main proposals for 3D descriptors can be divided into two categories—*signatures* and *histograms*. The first category defines a reference frame in which the local neighborhood is expressed and then encodes surface properties sampled at specific points. The second category accumulates surface properties within individual histogram bins in order to increase robustness of the descriptor. Each category contain both earlier and more recent works—[17, 64, 65, 9, 53, 10] can be considered signatures, [41, 54, 79, 101, 94, 102] can be considered histograms.

2.3. Supervised Learning

Supervised learning, also called learning from examples, establishes mapping $h : \mathcal{X} \to \mathcal{Y}$ from a set of input-output examples $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ by minimizing task-specific loss function $L(y_i, h(\mathbf{x}_i))$ which generalizes to similar but unseen input examples. The terminology differs in various research fields—the inputs may also be called *features*, *predictors*, or *independent variables*, the outputs may be called *labels*, *responses*, or *dependent variables*; in this text we will usually use the terms features and labels as these are common in computer vision literature. The supervision comes from knowing the correct labels y_i for the training data $\mathcal{D}$ which are usually assigned manually by a domain expert.

Statistical decision theory treats $\mathcal{D}$ as an independent identically distributed sample from a joint probability distribution $p_{X,Y}(\mathbf{x}, y) = p_{Y|\mathbf{x}}(y|\mathbf{x})p_X(\mathbf{x})$ and minimizes risk or expected loss

$$R(h) = \mathbb{E}_{X,Y}\{L(y, h(\mathbf{x}))\} = \iint L(y, h(\mathbf{x}))p_{X,Y}(\mathbf{x}, y)\, d\mathbf{x}\, dy. \tag{2.3}$$

Noting that

$$R(h) = \mathbb{E}_X\{\mathbb{E}_{Y|\mathbf{x}}\{L(y, h(\mathbf{x}))\}\} = \mathbb{E}_X\{R(h|\mathbf{x})\}, \tag{2.4}$$

where $R(h|\mathbf{x}) = \mathbb{E}_{Y|\mathbf{x}} L(y, h(\mathbf{x}))$ is called partial risk, it can be seen that the optimal decisions y can be made point-wise, for given $\mathbf{x}$ [33].

We will discuss two instances of the problem in detail:

1. regression analysis with quantitative output $y \in \mathbb{R}$ and Euclidean loss $L(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2$, with population minimizer

$$\operatorname*{argmin}_h R(h) = \mathbb{E}_{Y|\mathbf{x}}\{y\}, \tag{2.5}$$

2. and binary classification with categorical outputs $y \in \{-1, 1\}$ for negative and positive class, respectively, logistic loss $L(y, h(\mathbf{x})) = \log(1+\exp(-yh(\mathbf{x})))$, and population minimizer

$$\operatorname*{argmin}_h R(h) = \log(P_{y|\mathbf{x}}(1|\mathbf{x})/P_{y|\mathbf{x}}(-1|\mathbf{x})) \tag{2.6}$$

$$= \log(P_{y|\mathbf{x}}(1|\mathbf{x})/(1 - P_{y|\mathbf{x}}(1|\mathbf{x}))). \tag{2.7}$$

The minimizers can be found by differentiating partial risk $R(h|\mathbf{x})$ with respect to h and solving for the derivative equal to zero [33].

Because the true distribution $p_{X,Y}(\mathbf{x}, y)$ is unknown, we define the so called empirical risk as the average loss achieved on the data set $\mathcal{D}$, $\hat{R}(h) = 1/n \sum_i^N L(y_i, h(\mathbf{x}_i))$. Since $\mathcal{D}$ is an i.i.d. sample from $p_{X,Y}(\mathbf{x}, y)$, it follows that $\mathbb{E}_{X,Y}\{\hat{R}(h)\} = R(h)$, which lends us the idea of minimizing the empirical risk instead of the true (population) risk, for which we would need to known the probability distribution.

Finally, we define the set of functions $\mathcal{H}$ under consideration using a P-dimensional parameter vector $\boldsymbol{\theta}$, $\mathcal{H} = \{h(\mathbf{x}, \boldsymbol{\theta}) | \boldsymbol{\theta} \in \mathbb{R}^P\}$. The structure of h is selected based on the task at hand—for example it may correspond to a neural network with a fixed structure and number of parameters, with $\boldsymbol{\theta}$ being the weights of the network. Consequently, we may simply write $L(\boldsymbol{\theta}) = 1/N \sum_{i=1}^N L(y_i, h(\mathbf{x}_i, \boldsymbol{\theta}))$ instead of $\hat{R}(h)$.

2.3.1. Training Neural Networks via Error Back-Propagation

Deep neural networks present computational models composed of multiple processing layers which allow to learn representations of data with multiple levels of abstraction. These methods have outperformed the alternatives in various domains, from speech recognition and object detection to drug discovery and genomics [46].

Deep neural network model can be represented using the same formalism, as a function $h(\mathbf{x}, \boldsymbol{\theta})$ of an input vector $\mathbf{x}$ and model parameters $\boldsymbol{\theta}$. In most simple case, the layers form a chain or directed acyclic graph (DAG) representable by composite function

$$h(\mathbf{x}, \boldsymbol{\theta}) = f_L(\mathbf{f}_{L-1}(\ldots(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x}, \boldsymbol{\theta}_1), \boldsymbol{\theta}_2)\ldots), \boldsymbol{\theta}_{L-1}), \boldsymbol{\theta}_L), \tag{2.8}$$

with $\boldsymbol{\theta}$ being a concatenation of parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L$ of individual layers. The inputs $\mathbf{x}$ are presented to the bottom layer $\mathbf{f}_1$, the top layer f_L yields estimates of the target variable, i.e., the estimated label. Due to historical reasons and assumed similarities with biological nervous system, the values of functions $\mathbf{f}_1, \ldots, f_L$ are called *neural activations*. Increasingly more abstract representation build up in the layers higher in the hierarchy, with the last layer activations corresponding, e.g., to object category.

Rumelhart *et al.* [77] proposed to learn the model parameters via a gradient-descent procedure which efficiently computes the gradient $\nabla L(\boldsymbol{\theta})$ by back-propagating the errors through the network layer by layer. The authors consider multiple layers, each composed of several units of form $f(\mathbf{x}, \boldsymbol{\theta}) = 1/(1+\exp(-\boldsymbol{\theta}^\mathsf{T}\mathbf{x})) \in [0, 1]$, with Euclidean

loss $L(h(\mathbf{x}, \boldsymbol{\theta}), y) = (h(\mathbf{x}, \boldsymbol{\theta}) - y)^2$. It is noted that any function with bounded derivative could be used, nevertheless, using a linear function[1] to combine inputs which is followed by a nonlinearity greatly simplifies the learning procedure [77].

The gradient is evaluated in two passes. In the forward pass, activations are being computed starting from the bottom layer upward, eventually computing the loss $L(\boldsymbol{\theta})$. In the backward pass, the partial derivatives $\partial L/\partial \mathbf{f}_i$ and $\partial \mathbf{f}_i/\partial \boldsymbol{\theta}_i$, are being evaluated starting from the top layer downward, using the activations computed in the forward pass and the partial derivatives from the layer above. In the end, the partial derivatives $\partial L/\partial \boldsymbol{\theta}_1$ are evaluated in the first layer. Such a procedure, now widely adopted in training neural networks, is called *back-propagation*.

Let us represent an update to the parameters in iteration t as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \boldsymbol{\Delta}_t, \tag{2.9}$$

with the initial parameter value $\boldsymbol{\theta}_1$ being chosen randomly. Having computed the gradient, the simplest version of gradient descent is to change each weight proportionally to minus the corresponding partial derivative, which gives

$$\boldsymbol{\Delta}_t = -\alpha_t \nabla L(\boldsymbol{\theta}_t), \tag{2.10}$$

where $\alpha_t > 0$ is an iteration-specific learning rate. Techniques have been proposed to accelerate gradient descent by accumulating a velocity vector in directions of persistent reduction in the objective across iterations, namely, the momentum method [70] and Nesterov's accelerated gradient (NAG) [62], with parameter updates

$$\boldsymbol{\Delta}_t = \mu \boldsymbol{\Delta}_{t-1} - \alpha_t \nabla L(\boldsymbol{\theta}_t), \tag{2.11}$$

$$\boldsymbol{\Delta}_t = \mu \boldsymbol{\Delta}_{t-1} - \alpha_t \nabla L(\boldsymbol{\theta}_t + \mu \boldsymbol{\Delta}_{t-1}), \tag{2.12}$$

respectively, where $\mu \in [0, 1]$ is the momentum coefficient [92].

If the training data set $\mathcal{D}$ is an i.i.d. sample from joint distribution $p_{X,Y}$, so are the subsets $\mathcal{D}_t \subseteq \mathcal{D}$, and as such, they provide estimates of the expected loss. Using these noisy gradients instead of the gradient computed on all training data leads to so-called *stochastic* gradient descent (SGD). It has been observed in practice that with large, redundant data sets the stochastic version is considerably faster than the original, sometimes by orders of magnitude [60, 45]. This has lead to SGD with momentum acceleration being a predominant optimization technique for training deep neural networks models [13].

In computer vision, a convolutional neural network (CNN) [44, 49] is typically a model of choice [46]. As objects are usually decomposable into meaningful parts and these into smaller motifs, which may occur anywhere in the image, it is unnecessary to use fully-connected layers, especially in the lower levels of hierarchy. Instead, the CNN models use collections of learned convolutional kernels for their linear components, which operate only locally on a *receptive field*. Because of their properties and their previously demonstrated capabilities, we used convolutional neural networks in chapter 5 and 6 to process both multimodal image data and top-down 3D occupancy maps, where the position of the objects within the horizontal plane is, in general, not significant, nor is the in-plane rotation.

[1] A bias can be introduced by using an additional input x_0 with constant value 1.

Table 2.2. Confusion matrix for binary classifications with negative and positive classes. The acronyms TN, FP, FN, FP denote the numbers of respective decisions.

Truth \ Prediction	Negative	Positive
Negative	true negative (TN)	false positive (FP)
Positive	false negative (FN)	true positive (TP)

2.4. Performance Metrics

During operation of a recognition system, some recognition trials typically fail. For a binary classification task with negative and positive classes, often denoted 0 and 1 or -1 and 1, respectively, evaluation metrics can be defined based on the binary *confusion matrix* shown in Table 2.2. Correct decisions are on the diagonal.

Among commonly used metrics are precision, recall or true-positive rate (TPR), false-negative rate (FNR), false-positive rate (FPR), and intersection over union (IOU), defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2.13}$$

$$\text{recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.14}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{FP}}, \tag{2.15}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{2.16}$$

$$\text{IOU} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}. \tag{2.17}$$

3. 3D Object Recognition via Matching Local Features

Much attention has recently been given to the methods based on matching local invariant features applied to 3D object recognition [41, 54, 9, 23, 53, 94], surface matching [101, 102], or 3D point cloud registration [79]. One of the reasons is their computational efficiency and robustness to occlusion and clutter. Depth data acquired with a typical range sensor suffer from non-uniform sampling density which comprises problems for many existing methods, namely [41, 53, 101]. These methods deal with this problem usually by resampling the data prior to recognition which introduces extra delay and computational costs.

We suggest an approach in which area of polygons is taken explicitly into account in both establishing local reference frames and creating feature descriptors. This is a natural way to avoid assumptions which do not hold or are hard to enforce. As a mesh representation is often built for other purposes, this may come without any extra cost.

The rest of the chapter is organized as follows. First, we survey related work on feature-based 3D object recognition and areas closely related. Then we describe our method—new local invariant features and their use in 3D object recognition. Finally, we conclude with an evaluation of the method using a publicly available dataset.

Chapter Outline

3.1. Previous Work

Dorai and Jain [21] introduced a representation for 3D free-form objects using area-weighted shape indexes. Their framework, nevertheless, uses a global object descriptors in early matching stage and thus cannot be used for recognition of occluded objects. Moreover, it requires the calculation of principal curvatures which are sensitive to noise [53, 54]. Similarly, Rusu *et al.* [79] also do not use local reference frames but instead compute histograms of invariant point-pair properties.

The spin images of Johnson and Hebert [41] represents a classical approach to object recognition in cluttered scenes which is still used as a benchmark. It establishes a partial

coordinate system and introduces invariance by a specific space parametrization. To describe neighborhood of an oriented point a 2D histogram is created to accumulate occurrences of neighboring points expressed in a cylindrical coordinate system without the polar coordinate.

Most of the recently introduced methods use local reference frames as a means for achieving invariance. Mian *et al.* [54] create local frames from oriented-point pairs and accumulate surface area in a 3D histogram. In their later work [53] a local frame is established from principal axes while feature scale is chosen to maximize the ratio between these principal axes. The descriptor is created as a depth map of a smoothed surface patch.

Corners found in dense 2D normal maps are used as features in [9, 64] to align multiple range views and to recognize objects in cluttered scenes. The local frame is established from the principal curvature directions.

Tombari *et al.* [94] emphasize the need for establishing repeatable local frames and show how sign-disambiguation of basis vectors improves overall performance of the detector. Note that the sign ambiguity affects many methods with local frames based on principal components, e.g. [9, 53, 64].

Zaharescu *et al.* [101, 102] extract scale-invariant features from a triangular mesh with a real-valued function defined over its points. Their feature detector seeks the extrema of the Laplacian in scale space and establishes a local frame from the normal vector and the gradient of the function. Their descriptor is a histogram of oriented gradients projected on orthogonal planes.

Drost *et al.* [23] develop fast voting scheme using cheap point-pair features. Tentative correspondences vote for a reference point on the model and a 1D rotation to recover full rigid transform.

3.2. Invariant Features Construction

By *features* we mean local descriptions of data which can be matched with each other. The key requirements on 3D features are invariance under specified class of transformations and robustness to noise. Specifically for recognition of rigid objects, we require features to be invariant under rigid transformations so that features computed for the same parts of the object viewed from different viewpoints match.

Our method operates on polygonal meshes. We regard mesh $\mathcal{M}$ as a tuple $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, with vertices $\mathcal{V} = \{\mathbf{v}_i\}$ and faces $\mathcal{F} = \{f_i\}$. Each vertex is a 3D point, $\mathbf{v}_i \in \mathbb{R}^3$, each face is an n-tuple of vertex indices. For a face f_i, $\mathbf{c}_i \in \mathbb{R}^3$ denotes its centroid, a_i its area, and $\mathbf{n}_i \in \mathbb{R}^3$ denotes its normal vector. The normal vector of triangle $f_i = (1, 2, 3)$ can be computed as $\mathbf{n}_i = \frac{(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)\|}$.

3.2.1. Repeatable Local Frames

Importance of unique and unambiguous local frames in the process of feature extraction and description was emphasized in [94] which shows that using ambiguous local frames decreases performance of the descriptor.

Non-uniform sampling may negatively affect every stage of feature construction if not treated carefully. The local frame is often established from principal components of points [53, 94], with all points $\mathbf{v}_i$ contributing equally to the corresponding covariance matrix $\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{v}_i - \boldsymbol{\mu})(\mathbf{v}_i - \boldsymbol{\mu})^\mathsf{T}$. Obviously, with local variations in sampling density the covariance matrix and the corresponding local frame will change. In such

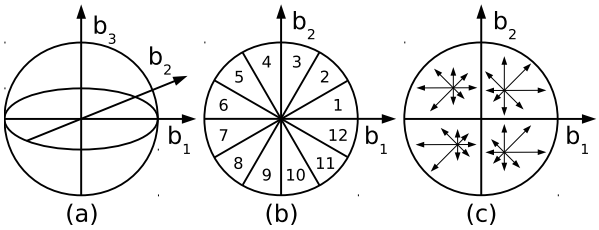


Figure 3.1. Feature descriptor—histogram of normal projections with 12 orientation bins. (a) Spherical support with local reference frame, (b) 12 orientation bins, (c) 4 spatial bins.

a case, an easy solution would be to weight each point by the surface area which it is an exemplar for.

To obtain basis vectors and create a repeatable local frame which is robust to noise we apply a procedure similar to [25] which estimates the local image orientation. In our case, we seek a direction which agrees with most of the surface normals. Since we assume the surface normals to be constant over individual polygons, we use their surface areas as weights and maximize over $\mathbf{b}$

$$\sum_{i:f_i \in N_s(\mathbf{p})} a_i (\mathbf{b}^\mathsf{T} \mathbf{n}_i)^2 = \mathbf{b}^\mathsf{T} \mathbf{C} \mathbf{b}, \tag{3.1}$$

subject to $\|\mathbf{b}\| = 1$, where $\mathbf{C} = \sum_{i:f_i \in N_s(\mathbf{p})} a_i \mathbf{n}_i \mathbf{n}_i^\mathsf{T}$, $\mathbf{p}$ is the feature center, s a scale factor, and $N_s(\mathbf{p})$ the set of neighboring faces $N_s(\mathbf{p}) = \{f_i : \|\mathbf{c}_i - \mathbf{p}\| < s\}$.

The two eigenvectors of $\mathbf{C}$ corresponding to the two largest eigenvalues in decreasing order give first two basis vectors $\mathbf{b}_1$, $\mathbf{b}_2$. The first one, $\mathbf{b}_1$, is the minimizer of (3.1) and defines the dominant orientation of the surface normals. The second one can be seen as the dominant orientation of normal projections onto plane $\mathbf{b}_1^\mathsf{T} \mathbf{x} = 0$.

To obtain an unambiguous local frame we need to disambiguate sign of the basis vectors. We follow a procedure based on [14] and change the sign of the first two basis vectors $\mathbf{b}_k$, $k = 1, 2$, to make

$$\sum_{i:f_i \in N_s(\mathbf{p})} a_i \operatorname{sign}(\mathbf{b}_k^\mathsf{T} \mathbf{n}_i) \tag{3.2}$$

non-negative. The third orthogonal basis vector $\mathbf{b}_3$ is then obtained as a cross product of the first two, $\mathbf{b}_3 = \mathbf{b}_1 \times \mathbf{b}_2$.

The procedure described above yields a unique and unambiguous local frame. It is, nevertheless, susceptible to noise. For a near-planar surface only the first basis vector is repeatable enough, the latter two will change arbitrarily in presence of noise. A similar situation will occur in sign-disambiguation with (3.2) close to zero. To improve robustness to noise we require both $\min(\lambda_1/\lambda_2, \lambda_2/\lambda_3)$, where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of $\mathbf{C}$ in decreasing order, and $|\sum_{i:f_i \in N_s(\mathbf{p})} a_i \operatorname{sign}(\mathbf{b}_k^\mathsf{T} \mathbf{n}_i)|$ to be higher than some thresholds.

3.2.2. Feature Descriptor

To abstract from small spatial or angular displacements of the local reference frame we choose a histogram-based descriptor, similar to *MeshHOG* of [101]—see Figure 3.1 for an illustration.

The histogram essentially accumulates projections of face normals. Each area-weighted face normal $a_i \mathbf{n}_i$ and its centroid $\mathbf{c}_i$ within $N_s(\mathbf{p})$ is projected onto three orthogonal

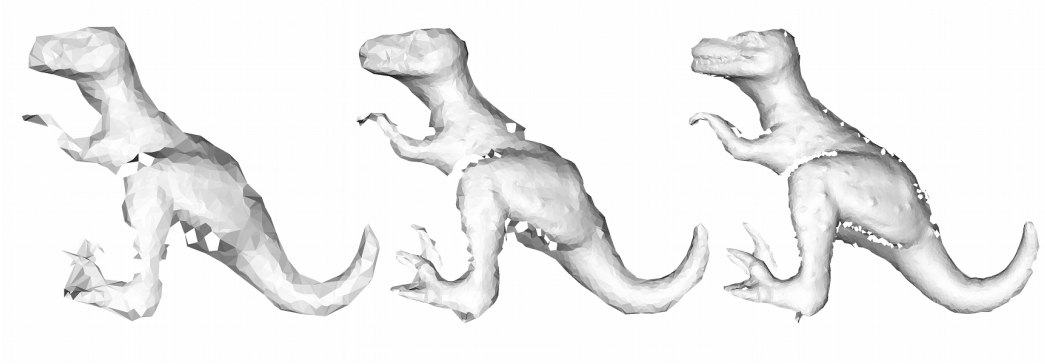


Figure 3.2. Example model—triangular meshes obtained from subsampling measurement point clouds. Model measurements subsampled to (left) $|\mathcal{V}_m| = 1.0$, (middle) $|\mathcal{V}_m| = 3.2$, and (right) $|\mathcal{V}_m| = 10.0$.

planes axis-aligned with basis vectors $\mathbf{b}_k$. The angular coordinate of the projected normal determines weights of orientation bins, coordinates of the centroid projection determine weights of spatial bins. Each vote, weighted by the radial coordinate of the normal projection, is interpolated linearly between two nearest orientation bins and bilinearly between 4 spatial bins. Three plane projections, 4 spatial and 12 orientation bins yield a descriptor of overall length $3 \times 4 \times 12 = 144$.

To make the descriptor scale-invariant, all the polygon areas could simply be normalized by the feature scale squared, i.e. a_i/s^2 would be used as a weight instead of a_i. However, we assume a fixed set of scales in our experiments. Unlike [101], we do not need to normalize the descriptor afterwards because surface area itself is meaningful and unambiguous quantity.

3.3. Object Recognition via Matching Local Features

The recognition method consists of off-line learning and on-line recognition. In the off-line learning phase features are computed densely for the model. In the on-line recognition phase features at the same scales are computed for the scene—but sparsely, at random locations. The scene features are then matched with those of the model to form a set of tentative correspondences, each one giving a preliminary estimate of the object pose. The final object hypothesis is generated by a consensus-based procedure.

3.3.1. Model Description using Partial Views

Since objects are to be recognized in range images sensed from a single viewpoint, the model description is also created from such views to account for self occlusion seen in real scenes. For our experiments we rendered 16 views for each model from cameras placed uniformly on a circle around the model, assuming common object positions. These views were then subsampled by randomly choosing a subset of points $\mathcal{V}_m$ and triangulated to obtain a mesh. Such a model view with varying mesh resolution is shown in Figure 3.2. The number of points in a model view is denoted by $|\mathcal{V}_m|$ and is given in thousands.

We avoid costly scale-space analysis by choosing feature scales prior to recognition based on the model size, similarly to [9]. The scales for a model with diameter d are chosen within interval $[d/32, d/16]$ from a discrete set of scales which is shared between

all models. Resulting features are then matched with only those computed at the same intrinsic scale which reduces the number of potential mismatches.

Since partial views overlap, some features computed on these overlapping views might be duplicate. We discard these redundant features simply by enforcing minimum distance $s/8$ between each pair of features, s being a feature scale.

3.3.2. Scene Description and Recognition

In the on-line recognition phase, the scene is first subsampled by randomly choosing a subset of points $\mathcal{V}_s$ and triangulated to obtain a mesh. In the following, $|\mathcal{V}_s|$ denotes the number of points in a scene and is given in thousands. Then, the features are computed at a subset of vertices and matched with the model ones using the Euclidean distance.

For each scene feature, the two nearest model features are found and a tentative correspondence with the first one is established if $d_1/d_2 < \tau$, d_1 and d_2 being distances to the model features, and τ some threshold. The purpose of this test, originally described in [50], is to discard correspondences which are likely to be incorrect and which may lead to an incorrect pose estimate. Incorrect correspondences also slow down the process of pose estimation and thus it is beneficial to discard them early. We set $\tau = 0.8$ in all our experiments which discards approximately 90% of incorrect correspondences and keeps 60% of the correct ones.

Each such correspondence gives an estimate of object pose, i.e., rotation $\mathbf{R}$ and translation $\mathbf{t}$ which aligns model $\mathbf{M}$ to scene $\mathbf{S}$ following $\mathbf{S} = \mathbf{R}\mathbf{M} + \mathbf{t}$. The estimate is obtained as

$$\mathbf{R} = \mathbf{B}_s\mathbf{B}_m^\mathsf{T}, \mathbf{t} = \mathbf{p}_s - \mathbf{R}\mathbf{p}_m, \tag{3.3}$$

where $\mathbf{B}_s = \begin{bmatrix}\mathbf{b}_1^s & \mathbf{b}_2^s & \mathbf{b}_3^s\end{bmatrix}$, $\mathbf{B}_m = \begin{bmatrix}\mathbf{b}_1^m & \mathbf{b}_2^m & \mathbf{b}_3^m\end{bmatrix}$ are matrices with the basis vectors, $\mathbf{p}_s$, $\mathbf{p}_m$ the locations of the scene feature and the model feature, respectively.

The consensus set of $(\mathbf{R}, \mathbf{t})$ consists of all estimates $(\mathbf{R}_i, \mathbf{t}_i)$ with $\alpha(\mathbf{R}_i^\mathsf{T}\mathbf{R}) < 12°$ and $\|\mathbf{t} - \mathbf{t}_i\| < d/10$, where $\alpha(\mathbf{Q})$ is the rotation angle of $\mathbf{Q}$ in the axis-angle representation and d the model diameter. The final object hypothesis is found as the estimate with the largest consensus set.

3.4. Experiments and Results

In our experiments we use the publicly available dataset from Mian *et al.* [53, 54], consisting of 5 models and 50 scenes acquired with a laser scanner. Each scene contains from 4 to 5 objects with known ground truth for object poses and occlusion, see Figure 3.3 for an example. We exclude the model of *rhino* from our evaluation as in [9, 23, 53, 54] to allow direct comparison with prior work. Also, we manually corrected the ground truth for scene 6 which accounts for an improvement of 0.5% in overall recognition rate.

3.4.1. Object Recognition in Real Scenes

We carry out the experiment of [54, 53] where the models are being recognized one by one. If the final object hypothesis agrees with the ground truth, the recognition trial is considered successful, otherwise it is a failure. Since there is no threshold for successful alignment given in [53, 54], we use the same as [23], i.e. $d/10$, where d is the model diameter, for the translation and 12° for the rotation.

The experiments were repeated for various mesh resolutions to study the effects of subsampling. For all resolutions 1000 scene features were extracted at each feature scale.

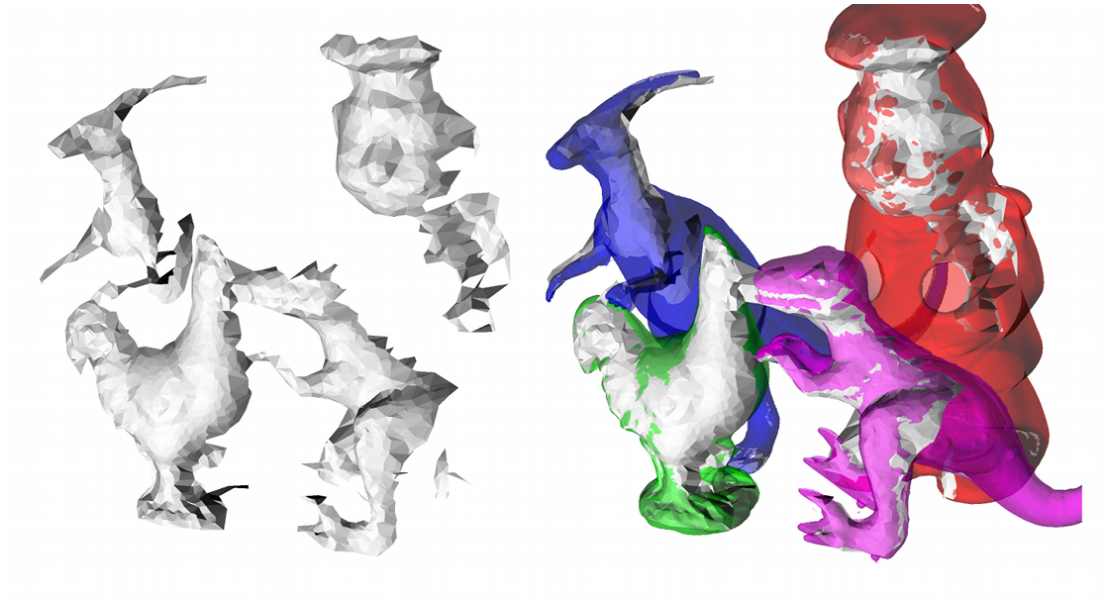


Figure 3.3. Cluttered scene subsampled to $|\mathcal{V}_s| = 3.2$ (left) prior to object recognition and (right) with recognized objects overlaid.

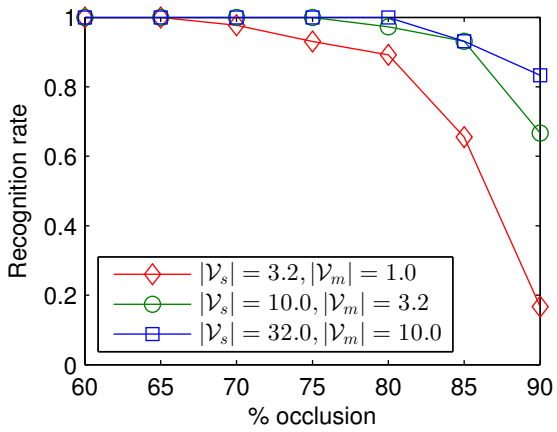


Figure 3.4. Recognition rate to occlusion for varying point cloud resolution.

We also evaluate the effects of occlusion, as defined in [41]. Using the highest resolution $|\mathcal{V}_s| = 32.0, |\mathcal{V}_m| = 10.0$, we achieved the overall recognition rate of 98.4% and 100% recognition rate for objects with occlusion up to 84%. For the second highest resolution, $|\mathcal{V}_s| = 10.0, |\mathcal{V}_m| = 3.2$, the recognition rate decreased to 98.8% for objects up to 84% occlusion—see Figure 3.4. This is comparable or better than results reported on this dataset so far—Bariya and Nishino report 97.5% [9], Drost *et al.* 97% [23]. In Table 3.1 a more detailed comparison is given.

All experiments were performed on Intel Core i7-740QM at 1.73GHz with 4GB RAM, using a MATLAB implementation. Run times per object for various mesh resolutions are reported in Table 3.2, where the preprocessing denotes subsampling the data, triangulating the points, and computing surface normals. We list run times of [23] for easy comparison, though their method was implemented in Java. Except the lowest resolution, our method outperforms [23] $\tau_d = 0.025$ in terms of recognition rate and run time. Still, we assume that a significant speed-up is possible through optimization

Table 3.1. Feature-based object recognition—comparison of recognition rates.

Method	$< 80\%$	$< 84\%$	Overall
Johnson and Hebert [41]	-	87.8% [54]	-
Drost *et al.* [23] $\tau_d = 0.04$	-	89.2%	-
Mian *et al.* [53]	$> 95\%$	-	-
Mian *et al.* [54]	-	96.6%	-
Drost *et al.* [23] $\tau_d = 0.025$	-	97.0%	-
Bariya and Nishino [9]	-	97.5%	-
$\lvert\mathcal{V}_s\rvert = 3.2, \lvert\mathcal{V}_m\rvert = 1.0$	95.5%	92.9%	87.8%
$\lvert\mathcal{V}_s\rvert = 10.0, \lvert\mathcal{V}_m\rvert = 3.2$	99.2%	98.8%	97.3%
$\lvert\mathcal{V}_s\rvert = 32.0, \lvert\mathcal{V}_m\rvert = 10.0$	100.0%	100.0%	98.4%

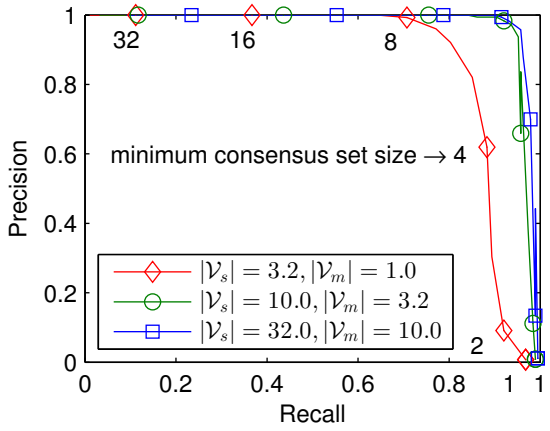


Figure 3.5. Precision to recall for varying point cloud resolution..

and parallelization.

3.4.2. Object Detection

To illustrate what the performance might be in a more generic scenario we evaluate our method in terms of precision-recall by varying the minimum consensus set size. In this experiment, after generating an object hypothesis, its consensual correspondences are removed from further processing and another largest consensus set is sought. Object hypotheses are being generated until there is no sufficiently large consensus set of correspondences.

Resulting precision-recall curves for various mesh resolutions are given in Figure 3.5. Considering the middle mesh resolution and required precision level 95%, 93.6% recall can be achieved. From the other side, allowing at most 5% objects to be missed, 93.7% object hypotheses will be correct.

3.5. Conclusion

We addressed the problem of non-uniform sampling which is inherent in typical range sensing methods. Operating on polygonal meshes, the proposed method overcomes the problem by exploiting surface area in both establishing local frames and creating feature descriptors. On a standard publicly available dataset [53, 54] we achieved recognition

Table 3.2. Run time per object for varying model resolution.

Method	Preproc.	Recog.	Total [s]
Drost *et al.* [23] $\tau_d = 0.04$	-	-	1.97
Drost *et al.* [23] $\tau_d = 0.025$	-	-	85
$\lvert\mathcal{V}_s\rvert = 3.2, \lvert\mathcal{V}_m\rvert = 1.0$	0.2	10.8	11
$\lvert\mathcal{V}_s\rvert = 10.0, \lvert\mathcal{V}_m\rvert = 3.2$	1.3	16.7	18
$\lvert\mathcal{V}_s\rvert = 32.0, \lvert\mathcal{V}_m\rvert = 10.0$	12.2	31.1	43

rates superior to those previously reported, successfully recognizing all objects with occlusion up to 84%. The results indicate that proper weighting should be employed throughout the whole process of feature extraction and description.

4. Point Cloud Registration—Evaluation on Challenging Datasets

Point cloud registration has many applications including mobile robotics, object modeling, and object recognition and pose estimation. It is a crucial step of the most commonly used methods for Simultaneous Localization and Mapping (SLAM), whether operating on the data from laser scanners or consumer-electronics RGB-D sensors, which have become widely available.

A variant of the Iterative Closest Points (ICP) algorithm is often employed to solve the task— see [12, 15] for the seminal papers on its point-to-point and point-to-plane formulations, respectively, or [81] for a generalization of these two methods. Despite many advantages of the algorithm, including real-time operation in some settings, the ICP algorithm has several drawbacks. Being an iterative local minimization method, it is sensitive to the initial alignment of the point clouds to be registered and their mutual overlap. As shown by [72], an inaccurate initial alignment or a low overlap between laser scans may deteriorate the accuracy of registration severely.

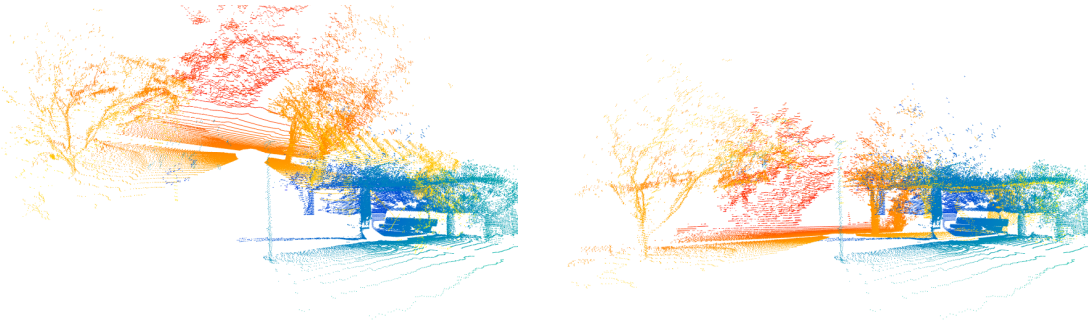
In order to overcome the limitations of the ICP algorithm, methods to establish global correspondences based local feature descriptors were suggested, such as [79, 53]. Since ICP performs very well if started within the basin of convergence, the coarse alignment obtained from these global methods often serves as an initial guess for ICP [53]. In modeling of objects from their partial views, ICP has been used to verify established correspondences and to refine registration provided from these [55].

Several alternative approaches have also been proposed. Instead of the special-purpose ICP formulation, Fitzgibbon [28] approached the registration problem as a general non-linear optimization which allowed to incorporate robust estimation via a Huber kernel. In 3D Normal-Distribution Transform (3D-NDT) [51], the surface is represented by a Gaussian Mixture Model and registration is also carried out by standard methods from numerical optimization. 4-points congruent sets (4PCS) are sought and matched in [4]. Despite the fast matching procedure proposed in the paper, for n input points the number of all possible coplanar 4-tuples is still $\mathcal{O}(n^4)$, which presents a major issue, especially for scans with large planar regions. The computational efficiency of this method was later addressed in [93] by creating 4-tuples from sparse local features instead of points.

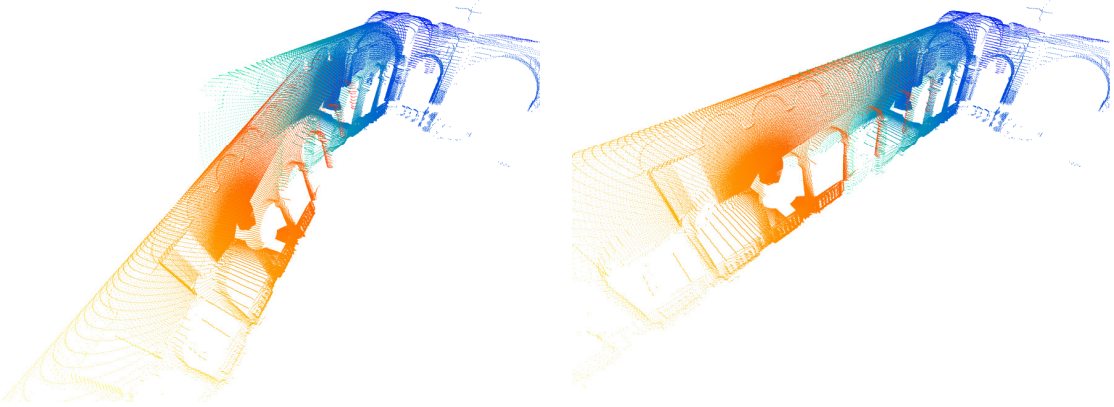
Even though many registration algorithms have been proposed, their fair comparison is still difficult due to a lack of datasets which would capture variety of scenes robots may encounter in the real world. A notable contribution to this area is due to [73, 72] which provide an experimental protocol using six medium-sized datasets with accurate ground-truth poses, capturing diverse environments, both indoor and outdoor, ranging from an apartment to a woodland area. This experimental protocol constitutes a basis for our evaluation. An example pair of reading and reference point clouds are shown in Fig 4.1. The same protocol has previously been used in evaluating 3D-NDT in [36, 52].

The contribution presented in this chapter is threefold.

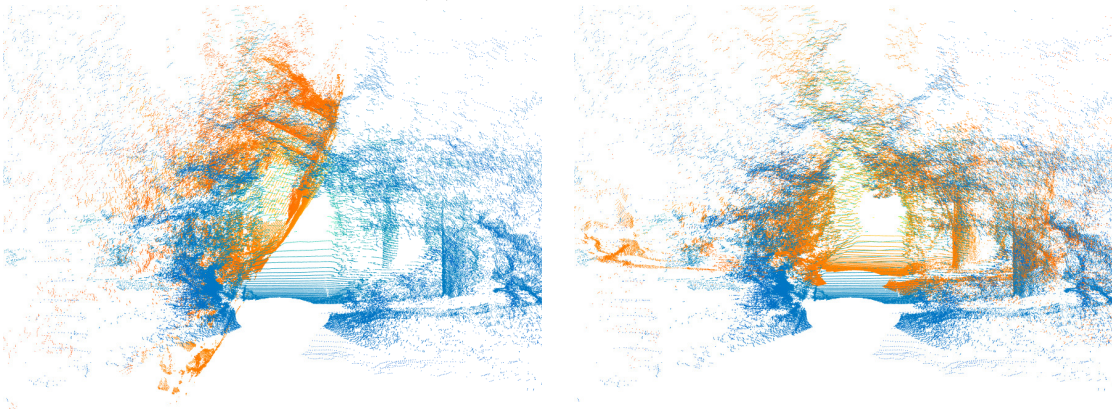
- We extend the local features from [68] by introducing keypoint detection and modifying the underlying method for establishing local reference frames. The method is evaluated on challenging real-world datasets, showing that for a moderate overlap



(a) Gazebo, overlap 0.39



(b) ETH, overlap 0.59



(c) Wood, overlap 0.89

Figure 4.1. Data from the experimental protocol. Reading and reference point clouds (left) prior registration and (right) aligned according to ground truth. The reference is displayed in blue, the reading in orange tones.

between the laser scans, it provides a superior registration accuracy compared to four local methods [12, 15, 81, 51] and another three global methods [4, 93, 79].
- Underlying components of the method, namely the keypoint detection and the local reference frames, are evaluated with respect to the task, along with the effects of their respective parameters, and general suggestions are given concerning specific design choices.
- For local reference frames, we compare three methods for sign disambiguation of the basis vectors. One of these methods is novel and achieves better repeatability than the general method of [14] used in the Signature of Histograms of Orientations (SHOT) [94]. The results also justifies using the sensor position for sign disambiguation in situations when it is known.

Chapter Outline

4.1. Methods

4.1.1. Feature-Based Registration

We formulate the registration task according to [72]. Given two point clouds, *reading* $\mathcal{P}_1 \subset \mathbb{R}^3$ and *reference* $\mathcal{P}_0 \subset \mathbb{R}^3$, the task is to find a rigid transformation $\mathbf{T}_{0\leftarrow 1}$ such that $\mathbf{p}_0 = \mathbf{T}_{0\leftarrow 1}(\mathbf{p}_1)$ for corresponding points $\mathbf{p}_1 \in \mathcal{P}_1$, $\mathbf{p}_0 \in \mathcal{P}_0$. In homogeneous coordinates, this is a linear transformation

$$\begin{bmatrix} \mathbf{p}_0 \\ 1 \end{bmatrix} = \mathbf{T}_{0\leftarrow 1} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{0\leftarrow 1} & \mathbf{t}_{0\leftarrow 1} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix}, \tag{4.1}$$

with $\mathbf{R}_{0\leftarrow 1}$ being a 3-by-3 rotation matrix, $\mathbf{t}_{0\leftarrow 1}$ a 3-by-1 translation vector, and $\mathbf{0}^\mathsf{T}$ a 1-by-3 zero vector. Points $\mathbf{p}_i$ can be assigned additional properties, such as surface normal $\mathbf{n}_i$, saliency h_i, or a descriptor $\mathbf{d}_i$, where the subscript denotes the index of the corresponding point.

The task is directly related to finding correspondences from the reading to the reference. From a set of tentative correspondences, found by matching local descriptors of the data, the transformation can be estimated using a robust estimator, such as Random Sample Consensus (RANSAC) [27].

We first introduce a general framework for point cloud registration based on matching local invariant features. Within such a framework, underlying components of the method are then evaluated.

4.1.2. Keypoint Detection

Keypoints are selected as extrema of a saliency measure, which determines the kind of structures being sought in the data and directly affects repeatability and robustness of the detection. Fixed-scale and adaptive-scale detectors can be distinguished [95]—the former are given scale as a parameter, the latter seek characteristic scales within a scale-space representation of the data, which need to be constructed for these purposes. We will not consider scale-adaptive detectors for the registration task since the scale is not ambiguous with the data from calibrated sensors, relevant scale changes are unlikely to occur in reality, and because seeking characteristic scale introduces an additional source of errors which affect all the following stages. We restrict feature matching to include only features of the same scale.

Local extrema are obtained via non-maxima suppression where only the keypoints with locally maximal saliency are retained. Specifically, a keypoint at point $\mathbf{p}$ with saliency h is kept only if $h \geq h_i$ for all $i \in \mathcal{N}_\sigma(\mathbf{p})$, where $\mathcal{N}_\sigma(\mathbf{p})$ is a set of point indices within the σ-neighborhood of $\mathbf{p}$. Points $\mathbf{p}$ with $|\mathcal{N}_\sigma(\mathbf{p})| < 10$ are excluded from keypoint detection.

We consider two types of keypoint detectors. The first uses the covariance matrix of points,

$$\mathbf{C}_\mathrm{p} = \frac{1}{\sum_i w_i} \sum_{i \in \mathcal{N}_s(\mathbf{p})} w_i (\mathbf{p}_i - \mu)(\mathbf{p}_i - \mu)^\mathsf{T}, \tag{4.2}$$

where $\mu = \frac{1}{\sum_i w_i} \sum_i w_i \mathbf{p}_i$, the second type uses the covariance matrix of normals,

$$\mathbf{C}_\mathrm{n} = \frac{1}{\sum_i w_i} \sum_{i \in \mathcal{N}_s(\mathbf{p})} w_i \mathbf{n}_i \mathbf{n}_i^\mathsf{T}, \tag{4.3}$$

where w_i are weights assigned to individual points, $\mathcal{N}_s(\mathbf{p})$ is a set of neighboring points of $\mathbf{p}$, $\{i \mid \|\mathbf{p}_i - \mathbf{p}\| \leq s\}$.

The eigenvalues of these covariance matrices will be denoted by $\lambda_1, \lambda_2, \lambda_3$ in their decreasing order, and their corresponding eigenvectors by $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$. We consider the following saliency measures as functions of the eigenvalues: $\min(\frac{\lambda_1}{\lambda_2}, \frac{\lambda_2}{\lambda_3})$, λ_1, λ_2, λ_3, $\frac{\lambda_1}{\lambda_2}$, $\frac{\lambda_2}{\lambda_3}$.

Despite an intuitive geometrical meaning of the saliency measures, this may not directly correspond to their quality in terms of repeatability of the corresponding keypoints. Therefore, we evaluate several such measures in order to select the most suitable for the task at hand.

Some of these saliency measures have previously been used. For example, [104] uses the smallest eigenvalue λ_3 of $\mathbf{C}_\mathrm{p}$ and several methods based on $\mathbf{C}_\mathrm{n}$ have been implemented in the Point Cloud Library [78], including the one using λ_3, which can be seen as a direct extension of [85] but replacing the image gradients with the surface normals. We provide an experimental evaluation which compares them with possible alternatives and justifies their usage with challenging real-world data.

For the keypoints found in this stage we establish local reference frames and compute the descriptors.

4.1.3. Local Reference Frames

Local reference frames are the key means to achieve the desired level of descriptor invariance. Although the Cartesian coordinate system is the most common today [104, 94, 68],

there are methods using a single reference axis [41], or no local frames at all [79]. Using reference frames yields several advantages. First, the three-dimensional distribution of points can be captured by the descriptor to increase its discriminative power. Second, each feature correspondence can provide an estimate of the transformation between the laser scans.

As noted by [94], although many methods rely on repeatable local frames, the importance of its particular choice is underrated. A common approach followed by many methods is to establish the basis of the reference frame from the eigenvectors of the feature covariance matrices as defined above.

As discussed in [14], singular value decomposition (SVD) of a matrix is unique only up to a reflection of each pair of singular vectors $\mathbf{u}_i, \mathbf{v}_i$ since $\sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T} = \sigma_i(-\mathbf{u}_i)(-\mathbf{v}_i)^\mathsf{T}$ for every pair of singular vectors. The same applies to eigenvalue decomposition of real symmetric matrices. Disambiguating the sign of the eigenvectors is thus needed to obtain a unique and unambiguous reference frame [94]. Right-handedness of the reference frame is then enforced by setting one of the basis vectors to the cross product of the remaining two.

Zhong [104] uses the eigenvectors of the point covariance matrix $\mathbf{C}_p$ but does not disambiguate their signs. Tombari *et al.* [94] use the eigenvectors of the point covariance matrix $\mathbf{C}_p$, replacing $\boldsymbol{\mu}$ by the feature position $\mathbf{p}$ and using $w_i = 1 - \|\mathbf{p}_i - \mathbf{p}\|/s$ for the weights, s being the scale, and follow the general procedure of [14] to disambiguate signs. The eigenvectors of $\mathbf{C}_n$ are used in [68], with weights w_i assigned based on the surface area of the respective polygons.

Throughout this paper, $\mathbf{Q}_i$ denotes the orthonormal basis of the local reference frame associated with point $\mathbf{p}_i$, and contains the eigenvectors $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$ in its columns. We consider three different methods of sign disambiguation, applied individually to each eigenvector $\mathbf{q}$.

- The first, used in [94, 68], changes the sign of $\mathbf{q}$ to make $\sum_i \operatorname{sign}(\mathbf{p}_i - \mathbf{p})^\mathsf{T}\mathbf{q}$ positive—we refer to this method as *support*.
- The second, denoted *mean*, reverses the sign of $\mathbf{q}$ if $(\mu - \mathbf{p})^\mathsf{T}\mathbf{q} < 0$, where $\mathbf{p}$ is the feature position and μ the centroid of the points within the local neighborhood defined above. This method has not been used, to our knowledge, to establish local reference frames.
- The third, denoted *sensor*, assumes the sensor origin $\mathbf{s}$ is known and reverses the sign if $(\mathbf{s}-\mathbf{p})^\mathsf{T}\mathbf{q} < 0$. This is a commonly used method for ensuring consistent orientation of estimated surface normals when the sensor position is known, yet it is less common in disambiguating all axes of local reference frames.

4.1.4. Feature Descriptor

We use the descriptor of [68] with 3×3 in-plane spatial bins and 8 polar bins. The descriptor is created by projecting the points within the neighborhood and their corresponding normals onto three planes spanned by pairs of the basis vectors, and accumulating the projections into histograms. Each oriented point casts weighted votes into the two nearest polar bins, given by the normal projection, and into the four nearest spatial bins, given by the point projection. The weights are proportional to relative proximity to each histogram bin and inversely proportional to the local surface sampling density (the area of the corresponding polygon was used in [68]). See Fig 4.2 for an illustration, and [68] for more details regarding the descriptor and its application to object recognition.

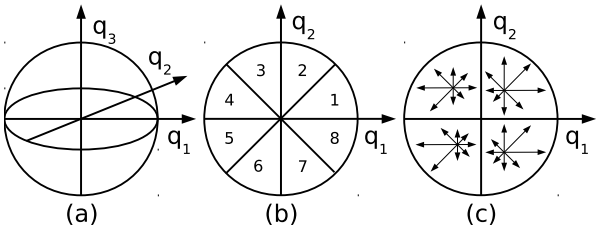


Figure 4.2. Feature descriptor—histogram of normal projections with 8 orientation bins. (a) spherical support with local reference frame, (b) 8 orientation bins, (c) 4 spatial bins.

4.1.5. Pose from Correspondences

Invariant descriptors are used to establish tentative correspondences from reading to reference. Let $\mathbf{p}_1$, $\mathbf{p}_0$ be a pair of points from reading and reference, respectively, and $\mathbf{d}_1$, $\mathbf{d}_0$ their associated descriptors. Then a correspondence is established if $\mathbf{d}_0$ is the among three nearest neighbors of $\mathbf{d}_1$ from the set of the reference descriptors and $\mathbf{d}_1$ is among three nearest neighbors of $\mathbf{d}_0$ from the set of the reading descriptors.

The pose is estimated from the set of tentative correspondences via Locally Optimized RANSAC [19]—pairs of correspondences, drawn randomly from the set, generate model poses and the pose maximizing the number of consensual correspondences is sought. The maximum number of iterations is estimated online, by setting the probability of missing the inlier set to $\eta = 1/100$ (see [19] for details). To generate the model poses, and to refine the pose from consensual correspondences, we use the method of [98] to find

$$\underset{\mathbf{R}_{0\leftarrow 1},\mathbf{t}_{0\leftarrow 1}}{\operatorname{argmin}} \sum_i \|\mathbf{R}_{0\leftarrow 1}\mathbf{p}_{1,i} + \mathbf{t}_{0\leftarrow 1} - \mathbf{p}_{0,i}\|^2 + \|\mathbf{R}_{0\leftarrow 1}\mathbf{n}_{1,i} - \mathbf{n}_{0,i}\|^2 \tag{4.4}$$

for matching point positions $\mathbf{p}_{0,i}$, $\mathbf{p}_{1,i}$ and normal vectors $\mathbf{n}_{0,i}$, $\mathbf{n}_{1,i}$. Before creating the model pose, we check that feature distances are consistent among both the point clouds and ignore the inconsistent samples.

Correspondences of the surface normals were used to generate model poses (from a pair of correspondences) and to locally optimize the model when up to five consensual correspondences were available. With more correspondences, the benefit of these terms vanished and minimizing the criterion based solely on the corresponding positions yielded more accurate pose estimates. Inlier threshold for a correspondence to be considered consensual was twice the scale of non-maxima suppression.

4.1.6. Data Set and Experimental Protocol

The laser registration datasets of [73] are used for experimental evaluation of the method and its components. The datasets were recorded with a laser rangefinder mounted on a tilting platform. For each scan the ground-truth position and orientation was obtained using a theodolite. The datasets contain both indoor and outdoor scenes, structured (an apartment, buildings) and unstructured environment (woodland area, a mountain plain), and dynamic elements with varying time spans (intra-scan and inter-scan motions, seasonal changes).

An experimental protocol for evaluation of point cloud registration methods is introduced in [72] by selecting pairs of scans from the datasets. From each dataset, 35 pairs of laser scans were selected to ensure approximately uniform coverage of the scan overlap from 0.3 to 0.99. The overlap is defined as the ratio of points from $\mathcal{P}_1$ for which a matching point exists in $\mathcal{P}_0$.

For each pair of point clouds 3×64 perturbations from the ground-truth alignment were generated to serve as initial alignments, 64 from each of the three Gaussian distributions with increasing variance. This establishes three classes of registration tasks with increasing difficulty, called *easy*, *medium*, and *hard* poses and constitutes a common ground for assessing sensitivity to initial alignment. As the feature-based methods are mostly insensitive to initial alignment of the point clouds we only use the first *hard* pose for each pair for their evaluation.

After transforming the reading point cloud to the initial pose, both point clouds are preprocessed as follows. First, the points with distance to the sensor less than 1 m or greater than 20 m are removed. Then, the point clouds are subsampled to achieve a maximum sampling density about $100\,\mathrm{m}^{-2}$ and surface normal is estimated at each point kept by fitting a plane to its 15 nearest neighbors before subsampling. The normals are reoriented to point towards the sensor.

The datasets are particularly challenging due to several reasons. Our results suggest that the main difficulty comes with a low overlap between some of the point cloud pairs and sometimes prevailing repetitive structures, especially in the *ETH* dataset. Variations due to viewpoint change, sampling and noise seem to be relatively high compared to those induced by the variability in the scene itself.

Another difficulties comes with the sensing device—large parts of the scene are occluded by the moving platform itself, namely the poles on which the prisms are mounted. Tilting the laser also causes a very nonuniform sampling density, which increases towards the axis of rotation. Nevertheless, these are all difficulties which might need to addressed in applications and therefore we consider this to be a good benchmark for evaluation of registration methods.

4.2. Results

Prior to evaluating the registration method as a whole following the protocol of [72], we evaluate keypoint detection and local reference frames using a small number of laser scans and fix their parameters. The following parameter choices are assessed:

- type of the features used (points, normals),
- scale of the keypoint detection and the local reference frames,
- weights assigned to the features (normalized distance from feature point [94], surface area [68]),
- method of sign disambiguation (support, mean, sensor),
- pairs of basis vectors to ensure right-handedness of the local reference frames ($\mathbf{q}_1 \times \mathbf{q}_2$, $\mathbf{q}_2 \times \mathbf{q}_3$, $\mathbf{q}_3 \times \mathbf{q}_1$).

4.2.1. Repeatability of Keypoint Detection

For keypoint detection we measure relative keypoint repeatability similarly to [95], as the ratio of the repeatable keypoints to all keypoints extracted from the reading. A keypoint is said to be repeatable if, after being transformed into the reference by the ground-truth transformation $\mathbf{T}_{0\leftarrow 1}$, its nearest neighbor among the keypoints detected in the reference is closer than some threshold. We set this threshold to be the same as the scale of non-maxima suppression.

As discussed in [56], the density of the extracted keypoints may affect the repeatability score—trivially extracting all the points would yield high repeatability. Thus, we include an experiment similar to the "Quantity Bias" experiment from [95], where only a limited number of the most salient keypoints are extracted to evaluate the keypoint

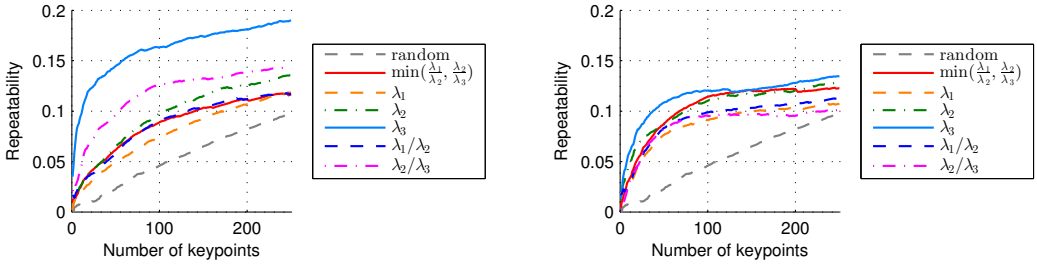


Figure 4.3. Repeatability of keypoints for various saliency measures. Repeatability of keypoints from (left) points and (right) normals for each saliency measure.

detector. The repeatability score is also computed for the same number of randomly extracted points to assess the ratio of keypoints being matched by accident.

Regarding the feature weights, we have not found any to be significantly better than the others across all scales, both in keypoint detection and local reference frames, and therefore only the unit weights are considered further. From scales ranging from 0.25 m to 1.0 m, 0.35 m provides best performing parameter combinations and is therefore selected to report the quantitative results below. This scale is also used in the point cloud registration experiments.

For each feature type, we report the results for the saliency measures in Fig 4.3. The saliency measure λ_3 provides the best results for both feature types, with a large margin for points. It selects the regions where the minimum variance of the features in any direction is locally maximum, informally speaking, where the features spread in all directions most evenly. Note that the relative order of the saliency measures tends to be stable with the increasing number of selected keypoints, with only a few exceptions.

4.2.2. Repeatability of Local Reference Frames

To evaluate the repeatability of the local reference frames, several metrics have been proposed—[94] measures the mean cosine of the corresponding axes, [67] aligns the z axes before measuring the cosine of the x axes to decorrelate the two measurements. In our experiments, we apply the same metric we use to quantify the rotation error of the registration itself.

Specifically, if $\mathbf{Q}_1$ and $\mathbf{Q}_0$ are the bases of the corresponding local reference frames from the reading and the reference, respectively, and $\mathbf{R}_{0\leftarrow 1}$ is the ground-truth rotation, the displacement of the reference frames is computed as

$$e_{\mathrm{q}} = \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_{0\leftarrow 1}\mathbf{Q}_1\mathbf{Q}_0^{\mathsf{T}}) - 1}{2}\right). \tag{4.5}$$

This measures the minimum angle of rotation needed to align the two bases and provides an upper bound on displacements of individual axes.

For the selected pairs of laser scans, 250 points are randomly selected from their overlapping parts where the local reference frames are established. The displacement e_{q} is then computed for such corresponding reference frames.

As mentioned above, we further consider only the unit weights as other alternatives do not provide significant advantage. From scales 0.5 m to 2.0 m, the larger ones were found to provide more repeatable local reference frames and were also most frequent among the best performing combinations. All the results below are given for the scale fixed to 2 m.

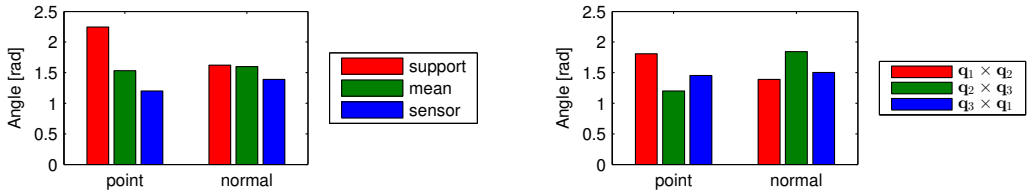


Figure 4.4. Local frame displacement. Average displacement e_q of the corresponding local reference frames for (left) sign disambiguation methods and (right) pairs of disambiguated vectors ensuring right-handedness of the basis.

The average displacements for the sign disambiguation methods and the pairs of disambiguated vectors are shown in Fig 4.4. Note that we show the results with the remaining parameters having their optimal values—for example, the result for points and sign disambiguation based on the sensor uses $\mathbf{q}_2 \times \mathbf{q}_3$ to comply with the right-hand rule but the similar result for normals uses $\mathbf{q}_1 \times \mathbf{q}_2$ as these eigenvectors are easier to disambiguate for this feature type.

Fig 4.4 (top) shows that the sensor origin provides the strongest hint for the sign for both feature types, for points with a large margin, and therefore should be preferred to the others. In situations where the sensor origin is not available, the *mean* method outperforms the general method of [14, 94], here denoted *support*, when using points as features. For normals, these two methods perform comparably.

Fig 4.4 (bottom) shows that the most repeatable direction, including the sign, corresponds to the surface normal or a related direction, i.e., the 3rd basis vector for points, and the 1st basis vector for normals. The sign of this direction should therefore always be disambiguated directly, using an appropriate method, and not be given by the cross product of the remaining vectors.

4.2.3. Registration

In this section, we evaluate our method using the experimental protocol described above and compare it to state-of-the-art methods [81, 51, 79, 4, 93].

Let $\hat{\mathbf{T}}_{0\leftarrow 1}$ be the estimate of the ground-truth transformation $\mathbf{T}_{0\leftarrow 1}$ which aligns the reading with the reference. To assess the quality of the registration, [72] defines the residual transformation

$$\Delta\mathbf{T} = \hat{\mathbf{T}}_{0\leftarrow 1}\mathbf{T}_{0\leftarrow 1}^{-1} = \hat{\mathbf{T}}_{0\leftarrow 1}\mathbf{T}_{1\leftarrow 0} = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{4.6}$$

and computes registration errors e_r and e_t from its rotational and translational components, $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$, such that

$$e_\mathrm{r} = \arccos\left(\frac{\mathrm{tr}(\Delta\mathbf{R}) - 1}{2}\right), \tag{4.7}$$

$$e_\mathrm{t} = \|\Delta\mathbf{t}\|, \tag{4.8}$$

where $\mathrm{tr}(\Delta\mathbf{R})$ denotes the trace of $\Delta\mathbf{R}$. The rotation error e_r corresponds to the angle of rotation in the axis-angle representation. To allow an interpretation in terms of accuracy and precision, [72] suggests to use robust error statistics, namely the 50th, 75th and 95th percentiles of empirical distributions of the errors, referred to as A50, A75, A95. We follow this convention in our evaluation.

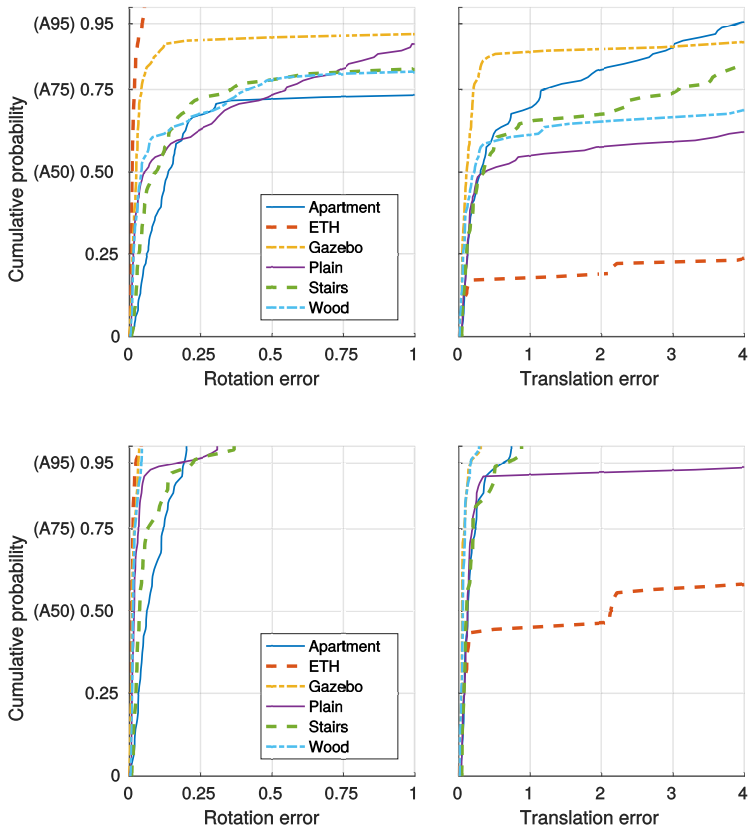


Figure 4.5. Point cloud registration accuracy. Distribution of (left) rotation and (right) translation errors for (top) all reading-reference pairs from hard poses and for (bottom) the pairs with overlap at least 0.75. A50, A75, and A95 denote the 50th, 75th, and 95th percentiles.

From local methods, we include in comparison Generalized ICP (G-ICP) [81] and 3D Normal-Distribution Transform (3D-NDT) [51] which is applied in a coarse-to-fine fashion with voxel sizes 2 m, 1 m and 0.5 m. For a complementary evaluation of 3D-NDT, please refer to [89, 20, 36, 52]. From global methods, we include another method based on matching local features, namely the Fast Point-Feature Histograms (FPFH) with Sample Consensus Initial Alignment (SAC-IA) [79], and two alternative approaches, namely 4-Points Congruent Sets (4PCS) [4] and its keypoint-based variant (K-4PCS) [93].

In general, same preprocessing steps were used as for the proposed method except for 4PCS and K-4PCS for which we had to limit the number of input points to reduce running time, by using maximum density of $4\,\mathrm{m}^{-2}$ and limiting maximum number of points to 500. For computing FPFH we used the common feature scale 2 m, SAC-IA used three tentative correspondences for each feature and minimum feature distance of 1 m to generate model poses. All the state-of-the-art methods were implemented in Point Cloud Library [78].

The rotation and translation errors, e_r and e_t, for the samples from the *hard* poses are summarized in Table 4.1 (the best result for given percentile is typeset in bold). We also include the baseline results from [72] for the point-to-point (Point) [12] and point-to-plane (Plane) [15] ICP variants to allow easy comparison. Moreover, Fig 4.5 shows the full distributions of the errors achieved by the proposed method.

Table 4.1. Quantile statistics of registration errors.

		Apartment			ETH			Gazebo			Plain			Stairs			Wood		
		A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95	A50	A75	A95
Rotation hard poses	Point [12, 72]	1.04	1.60	**2.53**	0.97	1.73	3.05	0.58	1.20	2.59	0.46	0.99	2.09	1.10	1.64	2.53	0.97	1.44	2.35
	Plane [15, 72]	1.01	1.72	2.95	1.31	2.09	3.11	0.58	1.31	2.88	0.50	1.09	3.05	1.48	1.91	2.94	1.05	1.56	2.53
	G-ICP [81]	0.73	1.74	3.10	0.74	1.43	3.03	0.38	1.13	2.92	0.07	0.86	2.97	0.56	1.63	3.07	0.57	1.26	2.72
	3D-NDT [51]	0.98	1.72	3.08	0.51	1.49	2.55	**0.02**	1.27	2.54	0.26	1.38	3.10	0.14	1.60	3.01	0.59	1.43	2.58
	FPFH [79]	0.51	1.96	3.13	0.03	0.05	0.64	0.13	0.30	1.38	0.16	**0.51**	**0.95**	0.44	2.31	2.99	0.28	**0.61**	2.42
	4PCS [4]	1.67	2.89	3.09	0.06	0.08	0.14	0.10	0.15	**1.07**	0.21	1.66	3.04	0.40	2.52	3.06	0.30	2.23	3.00
	K-4PCS [93]	1.81	2.94	3.12	0.13	0.35	3.03	0.40	2.17	2.98	0.29	0.77	2.69	0.99	2.68	3.05	2.09	2.81	3.10
	Ours	**0.14**	**0.35**	2.81	**0.01**	**0.02**	**0.04**	0.02	**0.04**	1.70	**0.05**	0.66	1.35	**0.10**	**0.23**	**1.83**	**0.04**	0.61	**2.28**
Translation hard poses	Point [12, 72]	1.29	1.99	**3.24**	3.84	7.06	14.77	1.58	2.79	**4.57**	2.02	3.14	**6.33**	1.81	2.78	4.75	2.32	**3.73**	**6.82**
	Plane [15, 72]	1.35	2.18	3.66	4.18	8.55	19.56	1.87	3.33	6.95	2.35	4.13	8.85	2.05	3.28	5.50	2.79	4.52	7.86
	G-ICP [81]	1.23	2.32	3.73	3.66	7.17	16.94	1.79	4.57	7.39	1.20	**3.09**	6.37	1.58	3.30	6.97	2.59	4.70	8.82
	3D-NDT [51]	1.24	2.22	3.58	**2.37**	**4.24**	**7.38**	**0.09**	2.76	5.02	2.02	3.69	7.01	1.30	2.62	**4.74**	1.72	4.15	7.54
	FPFH [79]	0.67	2.33	4.25	4.61	8.60	14.15	0.96	1.87	7.06	2.54	6.25	8.20	1.65	5.55	8.19	2.66	6.10	8.32
	4PCS [4]	1.91	3.03	4.62	4.43	9.74	14.27	0.79	1.31	5.88	2.86	9.06	15.37	1.49	5.63	10.08	2.57	6.78	10.09
	K-4PCS [93]	2.10	3.02	4.94	7.08	9.75	14.72	3.52	6.69	10.47	2.25	6.16	12.24	1.89	6.70	10.57	6.66	10.06	12.26
	Ours	**0.30**	**1.12**	4.04	6.33	10.63	14.42	0.13	**0.20**	5.21	**0.46**	6.79	9.37	**0.38**	**2.56**	7.49	**0.19**	6.20	10.53
Rotation hard poses overlap ≥ 0.75	Point [12, 72]	0.82	1.55	2.50	0.37	1.54	3.05	0.36	1.10	2.58	0.40	0.87	1.99	0.98	1.62	2.60	0.83	1.36	2.21
	Plane [15, 72]	0.81	1.64	3.00	0.02	1.85	3.12	0.40	1.16	2.70	0.42	0.93	3.05	1.40	1.82	2.94	0.89	1.46	2.46
	G-ICP [81]	0.16	1.63	3.10	0.60	1.24	3.06	0.02	1.01	2.95	0.01	0.72	2.90	0.02	1.57	3.08	0.01	1.18	2.74
	3D-NDT [51]	**0.04**	1.70	3.09	**0.01**	1.45	2.50	0.01	1.16	2.28	**0.01**	1.18	3.12	**0.01**	1.55	3.03	**0.01**	1.24	2.60
	FPFH [79]	0.11	0.30	1.68	0.02	0.05	0.06	0.03	0.06	0.12	0.04	0.07	0.48	0.06	0.10	2.91	0.13	0.16	1.16
	4PCS [4]	0.52	1.81	2.95	0.06	0.07	0.13	0.08	0.10	0.35	0.07	0.21	0.92	0.20	1.76	2.67	0.19	0.23	2.90
	K-4PCS [93]	0.43	2.91	3.10	0.08	0.28	3.04	0.07	0.17	2.16	0.15	0.31	2.08	0.34	2.73	3.04	0.39	2.98	3.12
	Ours	0.06	**0.12**	**0.20**	0.01	**0.01**	**0.02**	**0.01**	**0.02**	**0.03**	0.01	**0.04**	**0.08**	0.04	**0.08**	**0.21**	0.01	**0.02**	**0.04**
Translation hard poses overlap ≥ 0.75	Point [12, 72]	0.86	1.57	2.39	1.95	**3.30**	7.00	0.98	1.90	3.48	1.29	1.88	**3.06**	1.30	2.18	3.55	1.45	2.32	3.52
	Plane [15, 72]	0.83	1.70	2.68	2.19	4.30	9.54	1.23	2.70	5.87	1.40	2.40	4.95	1.42	2.47	4.08	1.86	2.84	4.60
	G-ICP [81]	0.41	1.72	2.83	2.39	5.51	13.70	0.06	3.47	6.50	0.07	1.58	4.22	0.06	2.42	6.53	0.09	3.02	7.83
	3D-NDT [51]	**0.06**	1.69	3.09	2.08	3.70	**5.38**	**0.04**	2.17	4.82	**0.05**	2.46	5.46	**0.04**	1.99	3.50	**0.03**	2.12	4.74
	FPFH [79]	0.27	0.49	2.18	**0.43**	4.47	7.32	0.26	0.37	0.75	0.39	0.81	3.52	0.22	0.50	7.86	0.36	0.81	2.22
	4PCS [4]	0.94	2.34	3.41	0.88	3.43	7.46	0.70	0.90	1.58	0.93	2.45	7.50	0.84	2.09	8.82	1.25	1.52	8.58
	K-4PCS [93]	0.69	1.57	4.90	4.56	5.28	8.11	1.02	2.18	5.48	1.33	1.85	5.30	1.34	5.53	10.57	1.93	8.21	11.42
	Ours	0.15	**0.25**	**0.68**	1.16	4.31	6.33	0.04	**0.12**	**0.29**	0.10	**0.21**	3.74	0.12	**0.20**	**0.76**	0.05	**0.10**	**0.27**

The table lists the 50th, 75th, and 95th percentiles (denoted A50, A75, and A95 respectively) of the rotation error e_r (in radians) and the translation error e_t (in meters). The two bottom blocks are restricted to the point cloud pairs with relative overlap at least 0.75. The results of the ICP variants [12, 15] are due to [72].

All methods considered in this paper fail in many cases, sometimes producing pose estimates which are further from the ground truth than the initial poses. Such results would most likely be unsatisfactory for any SLAM application. We also list the results for the reading-reference pairs with the overlap ratio at least 0.75 as the insufficient overlap seems to be the main cause why the methods based on matching invariant features are failing—see the bottom half of Table 4.1. Across higher overlap ratios, these methods yield good results in majority of cases.

Interesting fail cases are obtained with the *ETH* dataset—despite all methods failing to provide a reasonable translation estimate in most cases (A50 $\geq 2.37\,\mathrm{m}$), the feature-based methods consistently provide very accurate estimates of rotation. Moreover, our method even achieves the highest rotation accuracy on this dataset, contrary to a rather low translation accuracy. This is due to the regular structure of the environment with many repetitive patterns with similar orientation—even if these features are mismatched with each other, the rotation can still be estimated correctly from their correspondences. See Fig 4.6 for a visualization of consensual feature correspondences.

Average registration errors across all datasets are summarized in Table 4.2, together with running times. The proposed method provides the most accurate estimates of rotation for all reading-reference pairs with hard initial poses, while 3D-NDT provides the most accurate estimates of translation. For relative overlap at least 75 percent, nevertheless, the proposed method provides superior accuracy in both rotation and translation. Average running time of our method, 14 s, is less than 6× higher than that of the fastest local method, which is the point-to-point ICP, and about 3× lower than that of the second fastest global method, which is K-4PCS. Having been implemented in Matlab, our method can still benefit from further optimizations achieved by using a compiled language. Other methods were implemented in C++ as a part of the Point Cloud Library [78].

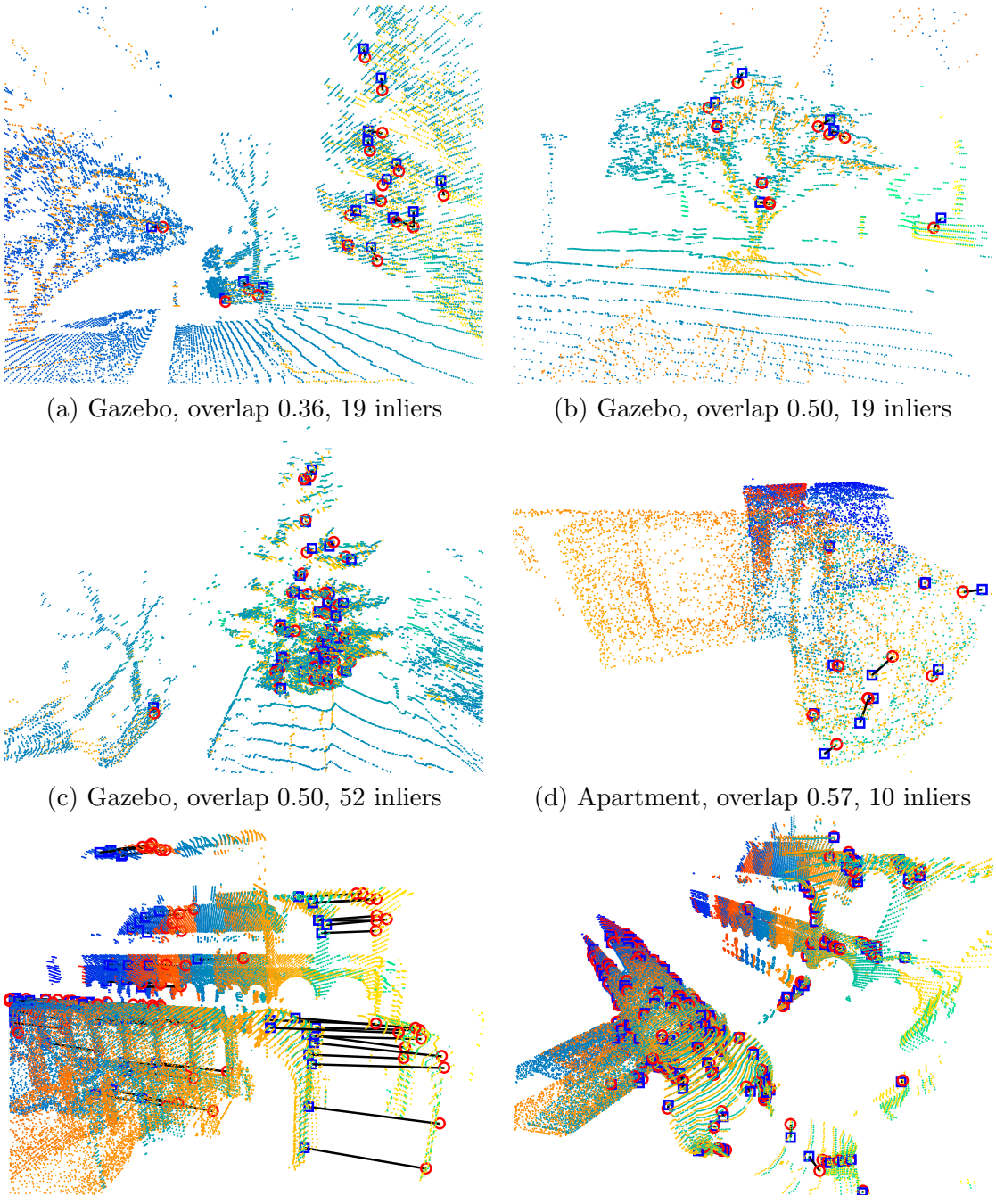
Table 4.2. Average errors and running times of registration methods.

Method	Rotation	Translation	Time
Point [12, 72]	0.97 (0.85)	2.60 (1.52)	**2.5** ± 2.0 s
Plane [15, 72]	1.05 (0.91)	3.09 (1.89)	4.1 ± 2.9 s
G-ICP [81]	0.83 (0.71)	2.71 (1.80)	3.1 ± 2.9 s
3D-NDT [51]	0.84 (0.70)	**1.98** (1.29)	4.7 ± 3.4 s
FPFH [79]	0.59 (0.19)	3.13 (1.00)	119.5 ± 80.1 s
4PCS [4]	0.85 (0.45)	3.57 (1.67)	47.0 ± 272.8 s
K-4PCS [93]	1.21 (0.87)	4.67 (2.80)	55.8 ± 154.3 s
Ours	**0.32** (**0.03**)	2.66 (**0.58**)	14.0 ± 18.8 s

All errors listed are on *hard* poses, rotation error e_r is in radians, translation error e_t in meters, time shown is mean ± standard deviation. Errors for point cloud pairs of relative overlap at least 75 percent are listed in brackets. The best result is printed in bold. All methods but ours are implemented in C++ as part of the Point Cloud Library [78], ours is implemented in Matlab.

4.3. Conclusion

In this chapter, we extended the local features of [68] by introducing keypoint detection and using a more robust method of the underlying local reference frames. The method



(a) Gazebo, overlap 0.36, 19 inliers

(b) Gazebo, overlap 0.50, 19 inliers

(c) Gazebo, overlap 0.50, 52 inliers

(d) Apartment, overlap 0.57, 10 inliers

(e) ETH, overlap 0.67, 99 inliers

(f) ETH, overlap 0.92, 206 inliers

Figure 4.6. Consensual feature correspondences for the proposed method. The reading and reference point clouds are displayed in orange and blue tones, respectively, aligned with each other using the ground-truth pose. Black lines connect the corresponding features from the consensual set, i.e., the inliers, marked by red circles and blue squares. The markers would be concentric in case of a perfect match. Reading-reference pairs are shown in order of increasing overlap, listing the dataset, their overlap, and the number of inliers for each. Mostly accurate pose estimates are shown, except in (e), which shows an inaccurate translation estimate due to repetitive structures in the ETZ dataset.

was evaluated on a set of challenging real-world datasets [73]. We compared the method to state of the art—two local-optimization-based methods (Generalized ICP [81] and 3D Normal-Distribution Transform [51]) and three global-search-based methods (Fast Point-Feature Histograms with SAC [79] and two variants of the 4-Points Congruent Set method [4, 93]). The experimental protocol from [72] provided a sufficient level of difficulty for both classes.

Failures of the feature-based methods, ours and [79], are mostly due to low overlap between the point clouds and repetitive structures which prevail in some of the scenes, especially in the *ETH* dataset. Nevertheless, for overlap ratios above 0.75 the proposed method achieves the highest accuracy and could also be used to initialize the local methods which achieves high accuracy when an initial estimate within a basin of convergence is provided.

The evaluation of its underlying components suggests that the points constitute a more solid base for both detecting keypoints and establishing local reference frames than the surface normals. Local maxima of the smallest eigenvalue of the feature covariance matrix provide most repeatable keypoints for both of the feature types. Note that this corresponds to the well-known method image-based keypoint detector of [85].

Sign disambiguation of the basis vectors proved to be a very important aspect in creating repeatable local reference frames. For situations in which the sensor position is not known, we proposed a novel method which achieves better repeatability than the general method of [14] used in the SHOT descriptor [94]. The results also confirmed that the sensor position, when it is known, provides a very informative clue for sign disambiguation and justified its usage therein. Another conclusion can be made regarding which vectors should ensure a right-handed coordinate system—vectors close to surface normal are the easiest to disambiguate and should thus be used preferably.

We see many possibilities for improving the overall accuracy of registration which can be addressed in future work, namely

- introducing a verification step to ensure that the geometric constraints are met and the open-space assumption is not violated,
- detecting repetitive structures to reduce mismatched features, or
- using higher-level knowledge to identify, recognize, and match distinguished objects in the scene.

5. Guiding Simultaneous Exploration and Segmentation

Segmentation of objects in an unknown environment from sensory data captured by a mobile robot is crucial for many applications including search & rescue (SAR) missions. In a typical SAR scenario, a human operator or a global planner provides a coarse exploration path along which the measurements are to be collected, registered and processed. Since most of the sensors have a limited field of view and the exploration time is a common issue, resulting coverage of the environment by the sensors is often incomplete, which may decrease the performance of object detection. When sensors are located on pan-tilt units, the dimensionality of the exploration planning task is huge and does not allow for real-time replanning when new data arrive. We propose a novel reactive control of body-mounted pan-tilt sensors for accurate classification of data gathered along a given exploration path. We call this problem *simultaneous exploration and segmentation with incomplete data* (SES).

The proposed method is demonstrated on the problem of human segmentation on a mobile SAR robot, which is equipped with a static panoramic `RGBD` sensor and a pan-tilt thermal camera (`T`) with a small field-of-view, see Fig. 5.2. Since temperature is an important cue for detecting humans in SAR, a segmentation-friendly control of the pan-tilt unit is needed for compensating the limited sensor coverage and maintain accurate segmentation. We design a (re)active human body segmentation algorithm in which deep convolutional neural networks (CNNs) simultaneously segment humans in incomplete `RGBDT` data and control pan of the thermal camera to minimize segmentation error.

CNNs have recently been shown to be a powerful representation for both classification [42] and control [47]. However, the success of CNNs is usually conditioned either by (i) a large number of labeled training examples [42, 58, 48], or (ii) a careful initialization [49, 47]. We show that in contrast to a general reinforcement learning task, the structure of SES allows for an efficient policy initialization.

In particular, we first extend Long's segmentation CNN [49] by depth and both depth and thermal modalities, and retrain it on our own human/background-annotated `RGBDT` dataset. These segmentation CNNs are further used for self-supervised training of a control sub-network, on data without any annotation, which estimates potential impact of thermal measurements on the classification error. The control sub-network is further extended by sub-sampling and fully connected layers and trained to predict the long-term impact of possible thermal-camera motions on the classification error. To train the control CNN efficiently, we propose a guided Q-learning algorithm, which uses optimal trajectories estimated by a Mixed Integer Linear Programming (MILP) planner to guide the exploration of the Q-learning and consequently avoids poor local optima.

Contributions of this chapter are four-fold: (i) We make two `RGBDT` datasets with annotated humans publicly available, one being composed of semi-synthetic images from a structured-light sensor, and the other one composed of panoramic images captured by a mobile search & rescue platform equipped with a time-of-flight sensor. (ii) We show how a pretrained segmentation network [49] can be extended by depth and thermal modalities. (iii) We propose guided Q-learning and show that it outperforms non-guided

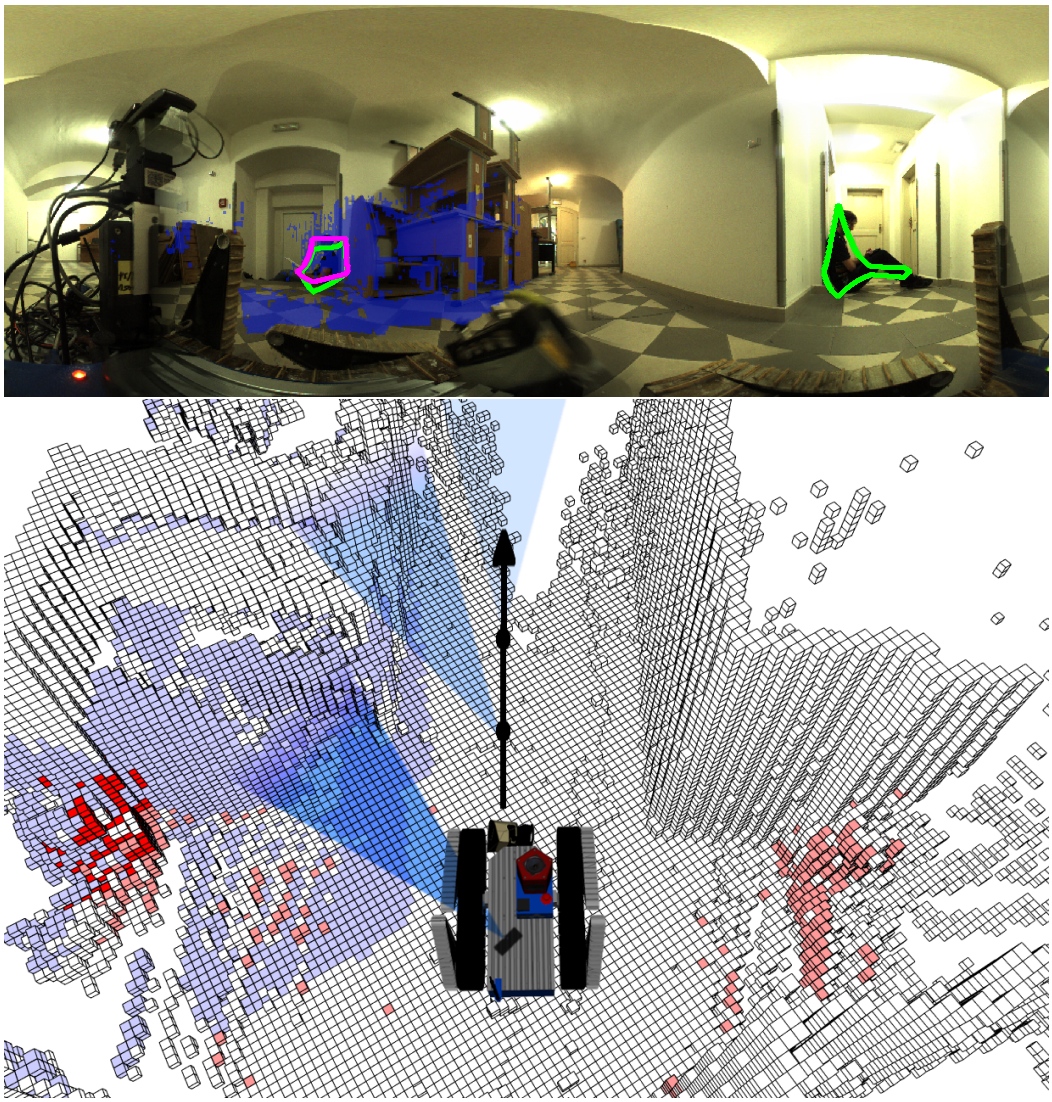


Figure 5.1. (top) Panoramic RGB image with segmented humans outlined by green and magenta contours, as given by the CNN-based segmentation models using either only RGBD data or the data with an additional thermal modality. The reprojected thermal measurements collected up to the current time are emphasized by blue overlay. (bottom) Reconstructed and segmented voxel map with accumulated thermal measurements displayed in blue color. Light red denotes the voxels marked as corresponding to human based on RGBD data only, dark red denotes the voxels marked as human based on the data with additional temperature measurements. Robot path with positions is denoted by black arrow with dots and selected thermal viewpoints are outlined by blue cones. Thermal camera is controlled to maximize the long-term sum of $\Delta\epsilon$.

Q-learning of Mnih *et al.* [58]. (iv) We suggest self-supervised policy initialization for instances of the SES problem.

Chapter Outline

5.1. Previous Work

A very recent overview [8] provides a comprehensive survey of the active perception research in the last four decades. The overview also suggests a basic taxonomy of active perception by defining the essential active pentuple *why, what, how, when, where*.

Doumanoglou *et al.* [22] use two robotic arms for folding an unknown piece of cloth. Since the type of cloth is crucial for the folding strategy, it is recognized from `RGBD` data (Kinect). One view is usually insufficient; the robotic arms have to act purposively. The arms turn the cloth around to provide an alternative view. The turning action is implicitly learned with decision forests.

Jia *et al.* [39] propose a framework that actively decides the next best view (NBV) for the object recognition tasks. It evaluates similarity based on an implicit shape model, a prior for the model, and a prior for the views. The problem is converted into a classification problem and a boosting algorithm is learned for combining the three information sources.

The active visual segmentation approach proposed by Mishra *et al.* [57] understands the activity very much differently from us. The authors propose an automatic segmentation method *given* a fixation on an object or a scene part. An initial fixation is further refined by choosing certain points on the skeleton of the segmented object.

Shubina and Tsotsos [86] propose a strategy for finding a target object in an unknown 3D world within a fixed time budget. Both the search space of object locations and that of robot positions is tessellated, into a 3D and 2D grid, respectively, and the sensed sphere [100] is used to represent surroundings of the camera. The execution time of each action includes robot movement, and image acquisition and analysis. Since Ye and Tsotsos [100] proved this task be NP-hard, the authors propose a greedy two-stage strategy which first selects where to look next, and then where to move next.

Andreopoulos *et al.* [6] share many concepts with [86], notably the concept of 3D search space grid, here called target confidence map. Their work adds an obstacle map and a multi-view visual detector. The core contribution is a probabilistic update of

both the target confidence and the obstacle map. The planning is greedy—next best view and position (of a humanoid robot) is selected.

Johns *et al.* [40] propose an active multi-view recognition method. They decompose the multi-view object classification task into a set of independent two-view classification tasks, each dealing with a single image pair. In this setting, the number of pairs for classification increases quadratically with the number of views, which becomes impractical for longer trajectories. They also show how to use this pair-wise decomposition in a trajectory optimization aimed at maximizing recognition accuracy—they establish an undirected graph from neighboring views, the unobserved views are assigned estimated cross-entropy scores for future pair-wise classification, and a path of a given length maximizing sum of the scores is found using graph search. After each new view observed, the scores are updated and the trajectory re-planned.

A deep Q-network (DQN) proposed by Mnih *et al.* [58] can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning. Being tested on the challenging domain of classic Atari 2600 games, the method outperforms all previous algorithms and achieves a level comparable to that of a professional human games tester. In this work, we show that training DQN policies can benefit from being provided with guiding samples obtained from an optimal planner.

Levine *et al.* [47] develop a guided policy search algorithm which allows learning deep CNN policies that map raw image observations directly to torques at the robot's motors. They evaluate the method on a range of real-world manipulation tasks, such as screwing a cap onto a bottle. In contrast to [58], [47], the reward used in our method is tightly coupled with the segmentation error.

Palmero *et al.* [66] present a new `RGB`-depth-thermal dataset with annotated human bodies which is similar to the dataset we publish in this work as for the represented modalities and object of interest. Our dataset, nevertheless, exhibits higher variability of background scenes and human poses, motivated by search & rescue scenarios. Their method [66] of human body segmentation relies on background subtraction using a learned Gaussian mixture model and the camera being static which is not applicable in our settings.

Recently Jayaraman and Grauman [37] proposed to use the reinforcement learning for active object and scene categorization, in which a learned CNN policy successively selects viewpoints of `RGB` camera to minimize categorization error. In contrast to this task, we solve the task of active 3D segmentation from incomplete `RGBDT` data captured online in a structured 3D environment. Hence, the learned policy has to infer both (i) the expected segmentation errors and (ii) the occlusions preventing future acquisition of thermal data. To tackle such complex task we propose self-supervised initialization and provide optimal trajectories to guide the reinforcement learning.

5.2. Problem Definition

The sensory suite of our mobile robot consists of (i) the Point Grey Ladybug 3 panoramic camera providing `RGB` images, (ii) the SICK LMS-151 laser scanner on a rotating mount providing depth measurements `D` and (iii) the thermal camera Micro-Epsilon thermoIMAGER TIM 160 with a small field of view mounted on a pan-tilt unit and providing thermal measurements `T`. The robot follows a known short-horizon path consisting of several discrete positions into an unknown environment. As the robot explores the environment, it simultaneously builds a 3D voxel map of occupancy and localizes itself within the map. In addition to that, temperature of some voxels can be measured by

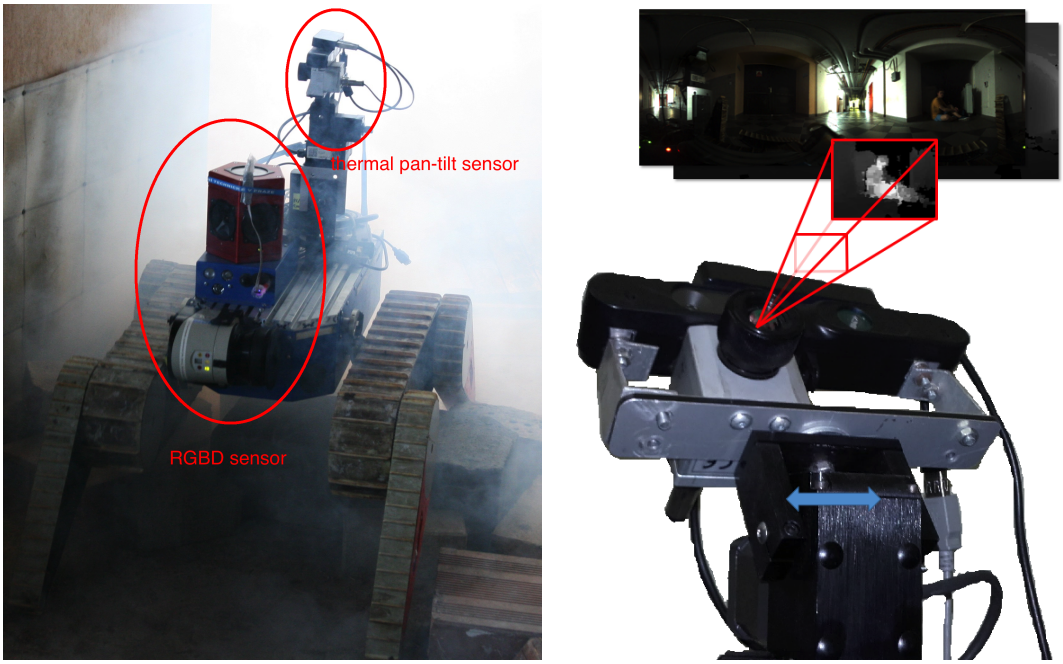


Figure 5.2. Skid-steer search & rescue robot with (i) a panoramic RGBD sensor consisting of an omnidirectional camera and a rotation laser scanner, (ii) a narrow-FOV thermal (T) sensor, mounted on a pan-tilt unit.

the thermal camera. Our instance of the SES problem is defined as the classification of all voxels visible from the robot and simultaneous control of the thermal camera which yields low classification error.

The proposed pipeline is outlined in Fig. 5.3. At each position, the voxel map of occupancy and temperature is reprojected into the RGB camera coordinate frame to create depth and thermal image, respectively, of the same resolution as the RGB images. Concatenation of the RGB image with depth image D is denoted by $\mathbf{x}$, the thermal image is denoted by $\mathbf{z}$.

The probability of human presence/absence in particular pixels is estimated by two segmentation networks. The first segmentation network $S_\theta(\mathbf{x})$ provides estimates without using any temperature measurements, the second segmentation network $S_\psi(\mathbf{x}, \mathbf{z})$ use the available temperature measurements. Network parameters are denoted by θ and ψ, respectively. Outputs of these networks, $\hat{\mathbf{y}}(\theta)$ and $\hat{\mathbf{y}}(\psi)$, are projected by mapping P onto the existing 3D voxel map to update the respective probability estimates in the corresponding voxels, denoted by $\hat{Y}(\theta)$ and $\hat{Y}(\psi)$. While $\hat{\mathbf{y}}$ or $\hat{Y}$ denote probabilities, additive log odds updates are used internally in form of $\sigma^{-1}(\hat{Y}_i) = \log(\hat{Y}_i/(1-\hat{Y}_i))$.

Motion of the thermal camera is determined by state-action value function network $Q_\omega(X, u)$ with parameters ω, which assigns Q-values $Q_1, \ldots, Q_N$ to N discrete control actions. At each state X, the best available action $u^* = \arg\max_u Q_u$ is chosen to control the motion of the thermal camera. The state X is defined later in Section 5.3.2. The proposed measuring-classification-control loop is summarized in Fig. 5.1.

Let us denote $\mathcal{V}(i_1, \ldots, i_K)$ the set of the voxels visible by the thermal camera from viewpoints $i_1, \ldots, i_K$ captured at K positions along the path (see Fig. 5.1). We assume that the motion dynamics of the thermal camera is constrained and that viewpoint i_k at time k is given as $i_k = f(i_{k-1}, u_k)$, where f is the motion model and u_k is a control action at time k.

Algorithm 5.1. The active segmentation algorithm.

1: Capture `RGB`, `D`, and `T` data and update the corresponding 3D voxel maps.
2: Construct $\mathbf{x}$ and $\mathbf{z}$ from the `RGB` camera image and the current voxel maps of occupancy and temperature.
3: Estimate local pixel-wise human probability
4: $\hat{\mathbf{y}}(\psi) = S_\psi(\mathbf{x}, \mathbf{z})$, $\hat{\mathbf{y}}(\theta) = S_\theta(\mathbf{x})$.
5: Update the corresponding voxel maps $\hat{Y}(\psi)$ and $\hat{Y}(\theta)$ using mapping P.
6: Estimate new control $u^* = \arg\max_u Q_\omega(X, u)$.
7: Simultaneously move the robot towards the next position on the exploration path and the thermal camera by control signal u^* towards the viewpoint to be captured at the next position.
8: Repeat from the beginning.

The learning is defined as a search for parameters θ, ψ, and ω which minimize the cross-entropy loss $\mathcal{H}(Y, \hat{Y}(\theta, \psi, i_1 \dots i_K))$ between estimated global voxel map $\hat{Y}(\theta, \psi, i_1 \dots i_K)$ at final position K and ground-truth voxel map Y subject to the motion constrains of the thermal camera,

$$\begin{aligned} &\arg\min_{\psi,\theta,\omega} \sum_v \mathcal{H}\left(Y_v, \hat{Y}_v(\theta, \psi, i_1 \dots i_K)\right) \\ &\text{s.t. } i_k = f\left(i_{k-1}, u_k(\omega)\right) \ \forall_{k\in\{1,\dots,K\}}, \end{aligned} \tag{5.1}$$

where Y_v, $\hat{Y}_v$ denotes elements (voxels) of voxel map Y, $\hat{Y}$, respectively, and initial viewpoint i_0 is a constant assumed to be known in advance.

This optimization problem is solved by approximately as successive minimization over θ, ψ, and ω. Optimization over ψ and θ is approximated by minimizing the cross entropy of pixel-wise updates $\hat{y}_i(\theta)$, $\hat{y}_i(\psi)$ with respect to pixel-wise ground-truth y_i using a dataset of annotated images (combined 5.4.1 and 5.4.2), which is tackled by Stochastic Gradient Descent (SGD) as two independent tasks

$$\arg\min_\theta \sum_i \mathcal{H}\left(y_i, \hat{y}_i(\theta)\right), \tag{5.2}$$

$$\arg\min_\psi \sum_i \mathcal{H}\left(y_i, \hat{y}_i(\psi)\right). \tag{5.3}$$

Substituting $\mathcal{H}\left(Y_v, \hat{Y}_v(\theta)\right)$ with $\mathcal{H}_v(\theta)$ and $\mathcal{H}\left(Y_v, \hat{Y}_v(\psi)\right)$ with $\mathcal{H}_v(\psi)$, optimization over ω simplifies as follows

$$\begin{aligned} &\arg\min_\omega \sum_{v\in\mathcal{V}(i_1,\dots,i_K)} \mathcal{H}_v(\psi) + \sum_{v\notin\mathcal{V}(i_1,\dots,i_K)} \mathcal{H}_v(\theta) \\ &\text{s.t. } i_k = f(i_{k-1}, u_k(\omega)) \ \forall_{k\in\{1,\dots,K\}} \\ &= \arg\min_\omega \sum_{v\in\mathcal{V}(i_1,\dots,i_K)} \mathcal{H}_v(\psi) - \sum_{v\in\mathcal{V}(i_1,\dots,i_K)} \mathcal{H}_v(\theta) + \sum_{v\in\mathcal{V}} \mathcal{H}_v(\theta) \\ &\text{s.t. } i_k = f(i_{k-1}, u_k(\omega)) \ \forall_{k\in\{1,\dots,K\}} \\ &= \arg\max_\omega \sum_{v\in\mathcal{V}(i_1,\dots,i_K)} \underbrace{\mathcal{H}_v(\theta) - \mathcal{H}_v(\psi)}_{\Delta\mathcal{H}_v(\theta,\psi)} \\ &\text{s.t. } i_k = f(i_{k-1}, u_k(\omega)) \ \forall_{k\in\{1,\dots,K\}}, \end{aligned} \tag{5.4}$$

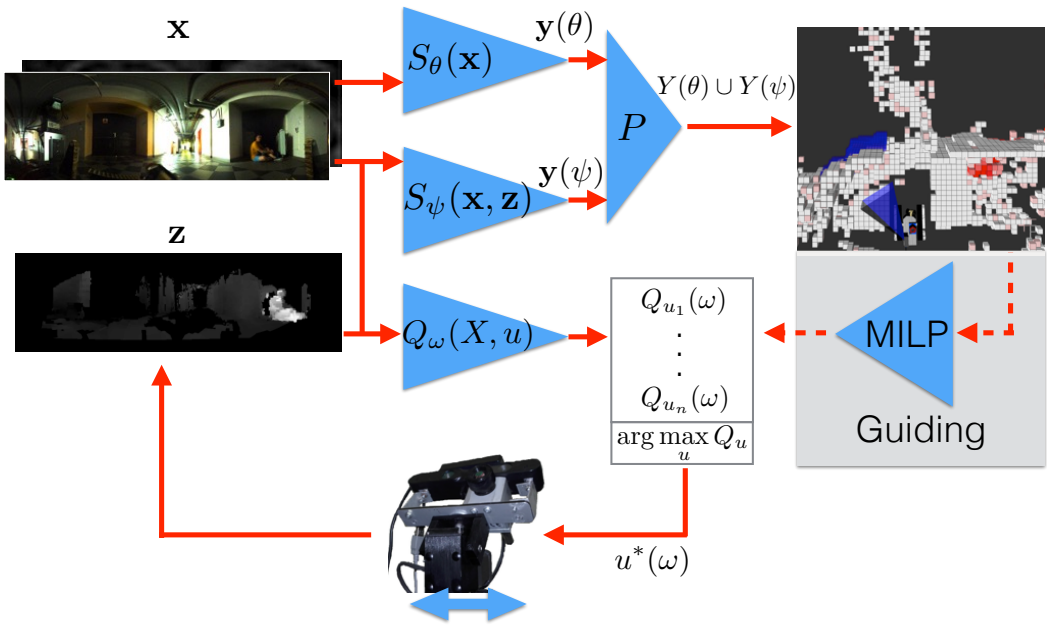


Figure 5.3. Learning outline. Human presence/absence in particular pixels is determined by two segmentation networks $S_\theta(\mathbf{x})$ and $S_\psi(\mathbf{x},\mathbf{z})$. Motion of the thermal camera is controlled by state-action value function network $Q_\omega(X,u)$. While learning of the segmentation networks is tackled by SGD, learning of the Q-network is guided by the optimal Q-values provided by the MILP-based planner.

where difference

$$\mathcal{H}\left(Y_v,\hat{Y}_v(\theta)\right)-\mathcal{H}\left(Y_v,\hat{Y}_v(\psi)\right)=\Delta\mathcal{H}_v(\theta,\psi) \tag{5.5}$$

denotes the reduction of the cross-entropy loss in voxel v when the temperature becomes known at this particular voxel—we call this quantity *gain*. Optimization step (5.4) is the most complicated one due to the motion and budget constraints which bind the control $u_1(\omega),\ldots,u_K(\omega)$ over the whole horizon K. Consequently, we propose the guided Q-learning algorithm for optimization of ω, which is detailed in Section 5.3.

5.3. Learning of the Control Network

If (i) the visibility of all voxels in all viewpoints along the robot path is available in advance, (ii) the gain is known for all voxels, and (iii) the control signals are discrete, then the optimal control corresponds to the weighted maximum coverage problem with limited budget and motion constraints. Such formulation is an instance of the following

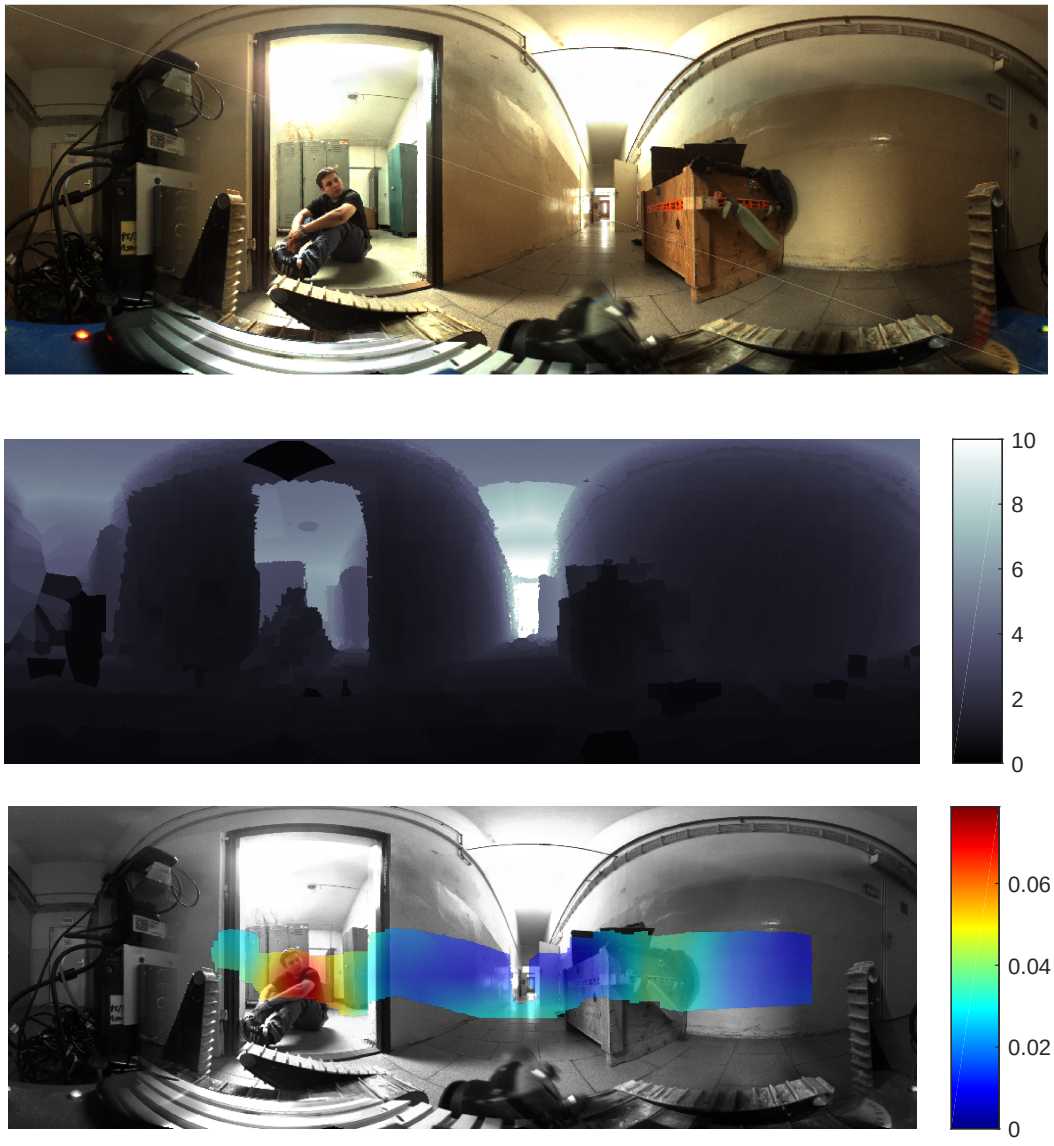


Figure 5.4. The self-initialized network $\mathcal{H}_{\omega_1}(\mathbf{x})$ estimates the expected per-pixel gain derived in Eq. (5.7) (right) from the RGBD input (left). Approximate pixel coverage for each viewpoint is known at each position. The coverage with the estimated gain is shown in color.

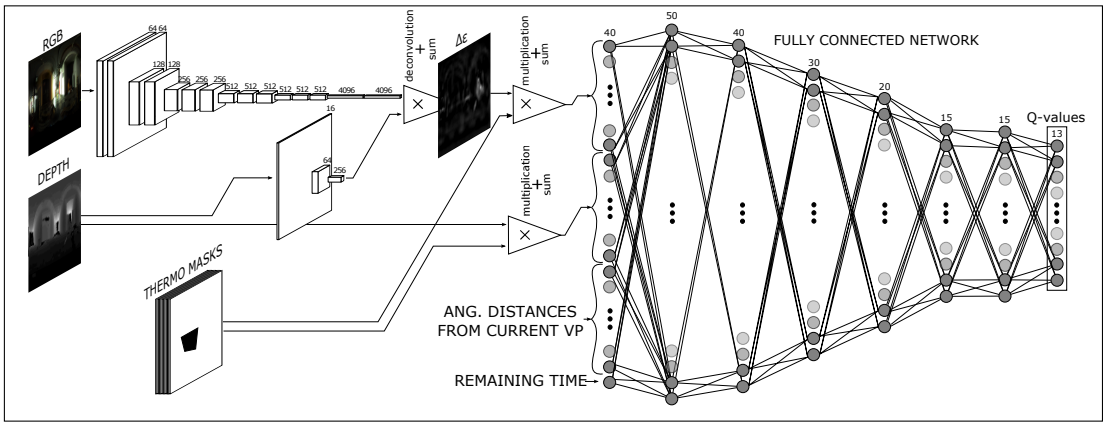


Figure 5.5. Deep CNN control policy overview. The policy is composed of two subnetworks, RGBD→$\Delta\epsilon$ and TD$\Delta\epsilon$→Q, with an interconnecting subsampling layer in the middle.

Mixed Integer Linear Program (MILP):

$$\begin{aligned} \arg\max_{\mathbf{u},\mathbf{v}} \quad & \mathbf{v}^\mathsf{T} \Delta\mathcal{H}(\psi,\theta) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{u} \geq \mathbf{v} \\ & \mathbf{B}\mathbf{u} = \mathbf{1} \\ & \mathbf{C}\mathbf{u} \leq \mathbf{1} \\ & \mathbf{v} \in [0,1]^V \\ & \mathbf{u} \in \{0,1\}^{KN}, \end{aligned} \tag{5.6}$$

where $\mathbf{A}$ is a sparse binary matrix which captures visibility of the voxels in the available viewpoints along the planning horizon, $\mathbf{B}$ is a sparse binary matrix determined by the budget constraints (single viewpoint per position), $\mathbf{C}$ is a sparse binary matrix which captures the motion constraints, V is the number of voxels in the map, K is the planning horizon (i.e., the number of positions along the path), N is the number of the available viewpoints (actions). However, since an unknown environment is typically explored, neither the map nor the gain $\Delta\mathcal{H}_v$ are known in a testing scenario, which makes direct online optimization impossible.

On the other hand, complete voxel maps with corresponding voxel gains are available for the annotated training sequences. Since a local gradient optimization of ω would require recurrent estimation of the gain with respect to the considered horizon K, which is both computationally demanding and prone to get stuck in a poor local minimum, we instead use MILP to directly optimize the control $\mathbf{u}$ on the training sequences. Optimal Q-values eventually guide the learning of parameters ω, see Sec. 5.3.2 for details.

Since the raw sensory measurements are high-dimensional, learning of deep Q-value network $Q_\omega(X,u)$ from randomly initialized weights would require a huge amount of training samples. To avoid such a demanding training procedure, we suggest to divide the $Q_\omega(X,u)$ network into two sub-networks: (i) the $\Delta\mathcal{H}_{\omega_1}(\mathbf{x})$ network predicting an approximation of $\Delta\mathcal{H}$ from $\mathbf{x}$ and (ii) the $q_{\omega_2}(\Delta\mathcal{H},X)$ network which predicts the Q-values from the gain $\Delta\epsilon$ and state X. These networks are first trained independently and then concatenated and fine-tuned as the $Q_\omega(X,u)$ network (see Fig. 5.5). Learning of the Q-value network is summarized in the three following steps.

1. Initialize the Q-value network by training the gain predicting sub-network $\Delta\mathcal{H}_{\omega_1}$ from supervised and self-supervised $\Delta\mathcal{H}$ annotations. In the supervised setting, $\Delta\mathcal{H}$ annotations are just the difference of segmentation cross entropies (Eq. 5.5). In the self-supervised setting, $\Delta\mathcal{H}$ annotations are estimated as Kullback-Leibler divergence of the outputs of segmentation networks on arbitrary not annotated data, see Sec. 5.3.1 for details. The $\Delta\mathcal{H}_{\omega_1}$ sub-network predicts the expected reduction of the cross-entropy loss as a result of measuring temperature at particular pixels.

2. Learn Q-value network $q_{\omega_2}(\Delta\mathcal{H},X)$ by the proposed guided Q-learning algorithm. The guided Q-learning first use the MILP planner to estimate optimal trajectories which maximize $\Delta\epsilon$-weighted coverage of voxels from the explored environment. These trajectories are used to normalize the Q-values and to guide the exploration. Learned policy approximates these optimal trajectories and consequently minimize the segmentation error, see Section 5.3.2 for details.

3. Connect these subnetworks into the final Q-value network $Q_\omega(X,u) = q_{\omega_2}(\Delta\mathcal{H}_{\omega_1}(\mathbf{x}),X)$ and fine-tune its parameters ω. Note, that the fine-tuned Q_ω network does not predict $\Delta\epsilon$ anymore.

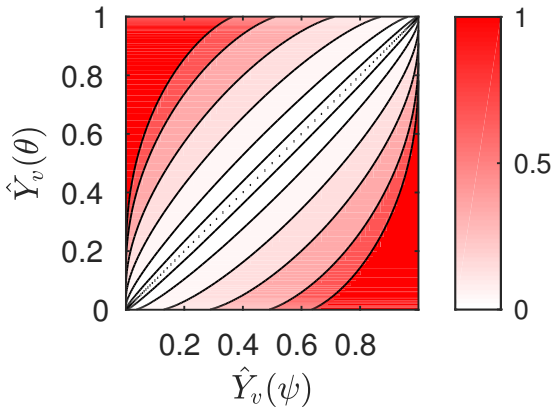


Figure 5.6. Values of the expected gain. $E_{Y_v \sim B(\hat{Y}_v(\psi))}\{\Delta\mathcal{H}(\psi,\theta)\}$ as a function of probability estimates $Y_v(\psi)$ and $Y_v(\theta)$ from segmentation.

5.3.1. Self-Supervised Policy Initialization

When annotations Y are available, supervised learning of the gain predicting network $\Delta\mathcal{H}_{\omega_1}$ is straightforward. We collect training pairs $(\mathbf{x}, \mathcal{H}(\mathbf{y}, \hat{\mathbf{y}}(\theta)) - \mathcal{H}(\mathbf{y}, \hat{\mathbf{y}}(\psi)))_k$ for fixed parameters θ and ψ, and learn a regression network minimizing the Euclidean loss. In addition to this, we also suggest a self-supervised learning setup, in which arbitrary not annotated `RGBDT` data can be used. In this setting, we approximate the gain using outputs of segmentation networks $\hat{\mathbf{y}}(\theta)$ and $\hat{\mathbf{y}}(\psi)$ as the expected difference of the cross entropy losses under the best current estimate $\hat{\mathbf{y}}(\psi)$ of truth labels as follows

$$\begin{aligned}\mathbb{E}_{y_i \sim B(\hat{y}_i(\psi))}\{\Delta\mathcal{H}(\psi,\theta)\} &= \mathbb{E}_{y_i \sim B(\hat{y}_i(\psi))}\{\mathcal{H}(y_i, \hat{y}_i(\theta)) - \mathcal{H}(y_i, \hat{y}_i(\psi))\} \\ &= \mathcal{H}(\hat{y}_i(\psi), \hat{y}_i(\theta)) - \mathcal{H}(\hat{y}_i(\psi)) \\ &= \mathcal{H}(\hat{y}_i(\psi)) + D_{\mathrm{KL}}(\hat{y}_i(\psi) \| \hat{y}_i(\theta)) - \mathcal{H}(\hat{y}_i(\psi)) \\ &= D_{\mathrm{KL}}(\hat{y}_i(\psi) \| \hat{y}_i(\theta)) \end{aligned} \tag{5.7}$$

where $B(p)$ is the Bernoulli distribution with parameter p, $\mathcal{H}(p)$ is the entropy of such a Bernoulli distribution, and $\mathcal{H}(p,q)$ and $D_{\mathrm{KL}}(p\|q)$ denote the cross entropy and Kullback-Leibler divergence, respectively, of the respective distributions. Values of the expected gain are shown in Fig. 5.6. Predicted gain for a testing image is shown in Fig. 5.4.

5.3.2. Guided Q-Learning

The second sub-network q_{ω_2} (as well as the whole Q-value network Q_ω during fine-tuning) is trained by the proposed guided Q-learning method, which is summarized in Fig. 5.2. Let us define extended state

$$X_k = (\mathbf{x}_k, \mathbf{z}_k, \mathbf{m}_k, \angle(I, i_k), K - k)\,, \tag{5.8}$$

where $\angle(I, i_k)$ denotes the angular distance of all viewpoints I from current viewpoint i_k, $K - k$ is the remaining number of the positions, $\mathbf{m}_k$ denotes the thermal masks determining coverage of pixels by temperature for the allowed viewpoints $i \in I$. Given the state, we can perform actions u, which control the thermal camera viewpoint selected at the next position. Reward R' for performing the action is given by the gain of newly covered voxels.

The algorithm successively collects training transitions from available maps and learns Q-value regression network $Q_\omega(X, u)$ with weights ω. The Q-value network assigns the expected gain of covered voxels when action u is applied in state X and then controlling optimally.

The guided Q-learning first estimates gain for all voxels. The optimal control u^* of the thermal camera and the optimal gain coverage q^* is determined by solving the corresponding MILP instance from the current state (see line 2). Then it evaluates the sum of gains q' achievable for all possible controls u' by successively applying each control u' and solving the corresponding MILP instance from the following state X' (see lines 4–6). All these transitions (X_k, u', Q) are stored in the dataset $\mathcal{D}$ (see line 6). We have considered (and experimentally evaluated, see Fig. 5.9) three different types of Q-values:

1. *raw* sum of covered $\Delta\mathcal{H}$-values: q',
2. *absolute loss* in the sum of covered $\Delta\mathcal{H}$-values: $q' - q^*$,
3. *relative loss* in the sum of covered $\Delta\mathcal{H}$-values: q'/q^*.

Eventually, either the optimal control u^* or Q-value-driven control $\arg\max_u Q(X_k, u)$ is applied (see lines 8–9) and the process continues from the following state X_{k+1}. When a sufficient number of transitions is collected, SGD is performed on weights ω of the regression network $Q_\omega(X, u)$, until the validation error stops decreasing (see line 11).

In contrast to the standard Q-learning, the guided Q-value network is not forced to predict the absolute sum of $\Delta\mathcal{H}$ which is often loosely connected with features observed in the current state. Guided Q-learning predicts rather the expected impact on the optimality. Another advantage stems from guiding the exploration of the state-action space close to the optimal trajectories. In the experiments, guiding probability p linearly decreases from 1 towards 0.

Algorithm 5.2. The guided Q-learning algorithm.

Require: Initial viewpoint i_0

1: **for** $k \in \{1, \ldots, K\}$ **do**
2: $(q^*, u^*) \leftarrow \texttt{MILP}(X_k)$ ▷ Optimal control
3: **for** $u' \in \{1, \ldots, N\}$ **do**
4: $(X',\ R') \leftarrow \texttt{act}(X_k, u')$ ▷ Apply action u.
5: $q' \leftarrow R' + \texttt{MILP}(X')$ ▷ Optimum from X'
6: $\mathcal{D} \leftarrow \mathcal{D} \cup \left(X_k, u', \frac{q'}{q^*}\right)$
7: **end for**
8: $u_k \leftarrow \begin{cases} u^* & \text{with prob. } p \\ \arg\max_u Q_\omega(X_k, u) & \text{with prob. } 1-p \end{cases}$
9: $X_{k+1} \leftarrow \texttt{act}(X_k, u_k)$
10: **end for**
11: $\omega \leftarrow \texttt{SGD}(Q_\omega,\ \mathcal{D})$.

5.4. Learning of the Multimodal CNN Models

Convolutional neural networks are expressive models which allow efficient element-wise prediction for inputs of variable size. They are composed of multiple processing layers forming a directed acyclic graph. The *bottom* layer has the source data as its input, the *top* layer yields the target prediction or a task-specific scalar loss for training.

For two spatial dimensions, which is the case for images, a single output element at coordinates i, j can be described as a function of input elements within the *receptive field*,

$$\mathbf{y}_{i,j} = f_{s,k}(\{\mathbf{x}_{si+\Delta i, sj+\Delta j}\}_{0 \leq \Delta i \leq k, 0 \leq \Delta j \leq k}), \tag{5.9}$$

where f is the function the layer represents, s is an integer stride in the spatial dimensions, and k is the extent of the receptive field (i.e., the *kernel*). The function f can be the inner product (of $\mathbf{x}$ with layer weights) in case of the convolution layers, a maximum over the receptive field for the pooling layers, or a nonlinear scalar function in case of the activation layers.

The loss serves as the optimization criterion which is commonly minimized by Stochastic Gradient Descent (SGD) with momentum [70]. To train the image-based models, we use SGD with Nesterov's accelarated gradient (NAG) [62, 92] which yields weights update from Eq. (2.12). The segmentation models use the multinomial logistic loss for training, the regression model uses the Euclidean loss.

All the models having `RGB` as input reuse the 16-layer VGG net [88] as adapted and fine-tuned by [49], namely the *FCN-32s* variant. Since annotated depth and thermal data are much scarcer, and no suitable pretrained models are available for these modalities, we employ smaller models, with similar structure but having four times less output channels in each convolutional layer to prevent overfitting. The architecture for a single modality, depth or thermal, is summarized in Table 5.1. The multimodal models are then composed by summing up the outputs of the (last) deconvolution layers, directly before the final softmax layer.

First, we train the segmentation networks using extra modalities—one using depth, the other using depth with the thermal modality. These are then concatenated with the pre-trained `RGB` segmentation network [49] and fine-tuned to provide the S_θ and S_ψ networks used in the experimental evaluation in Sec. 5.5.3. Outputs $\hat{\mathbf{y}}(\theta)$ and $\hat{\mathbf{y}}(\psi)$ are used to train gain-predicting network $\Delta\mathcal{H}_{\omega_1}$, once with ground-truth labels $\mathbf{y}$ to predict $\Delta\mathcal{H}(\theta, \psi)$ directly and once with not annotated data to predict its estimate in form of the Kullback-Leibler divergence from Eq. (5.7). Finally, the gain-predicting network is merged with the control sub-network q_{ω_2} and fine-tuned on guiding trajectories.

For learning parameters of the models, we use training subsets from the two following datasets, where we replaced the missing measurements in case of the depth and thermal modalities by their nearest valid neighbors. The validation subset of the panoramic dataset 5.4.2 were used for early stopping and to select models for test. The reported results in Sec. 5.5.2 and 5.5.3 are obtained on the test sequences from the panoramic dataset.

5.4.1. Semi-Synthetic Human Body Dataset

In order to obtain a large number of images with accurate ground-truth segmentation for training and evaluation we chose to create a semi-synthetic dataset [1] in the following way. First, positive examples with humans in various poses were captured in the lab, in front of the green screen to simplify their annotation. Second, background images were captured in a real-life environment, both outdoor and indoor, without the need to constraint the scene conditions much. We used Asus Xtion PRO LIVE to capture the `RGBD` data and IMAGER TIM 160 to capture the thermal data `T`. Finally, semi-synthetic

[1] http://ptak.felk.cvut.cz/tradr/data/human_seg/

Table 5.1. CNN architecture for depth and thermal modalities.

Layer	Type	Kernel	Stride	Channels
1/1	Convolution + ReLU	3×3	1	16
1/2	Convolution + ReLU	3×3	1	16
1/3	Max. pooling	2×2	2	16
				(→ 1/2 size)
2/1	Convolution + ReLU	3×3	1	32
2/2	Convolution + ReLU	3×3	1	32
2/3	Max. pooling	2×2	2	32
				(→ 1/4 size)
3/1	Convolution + ReLU	3×3	1	64
3/2	Convolution + ReLU	3×3	1	64
3/3	Convolution + ReLU	3×3	1	64
3/4	Max. pooling	2×2	2	64
				(→ 1/8 size)
4/1	Convolution + ReLU	3×3	1	128
4/2	Convolution + ReLU	3×3	1	128
4/3	Convolution + ReLU	3×3	1	128
4/4	Max. pooling	2×2	2	128
				(→ 1/16 size)
5/1	Convolution + ReLU	3×3	1	128
5/2	Convolution + ReLU	3×3	1	128
5/3	Convolution + ReLU	3×3	1	128
5/4	Max. pooling	2×2	2	128
				(→ 1/32 size)
6/1	Convolution + ReLU	7×7	1	1024
6/2	Dropout (0.5)			1024
6/3	Convolution + ReLU	1×1	1	1024
6/4	Dropout (0.5)			1024
6/5	Convolution	1×1	1	2
6/6	Deconvolution	64×64	32	2
				(→ original size)
7	Softmax		1	2

images were composed by placing annotated humans onto the background images, using the depth information to avoid implausible configurations and to impose realistic occlusions. For a pair of images, object configurations (i.e., rotation, translation, and scale) were sampled from a uniform distribution until a plausible configuration was found, as measured by an ad-hoc criterion which rewards contact at boundary pixels and penalizes object pixels behind the background. The process is illustrated by Fig. 5.7, showing the source images and the resulting composition.

The source images were split into training, validation, and test sets prior to composition. The number of images in every group is summarized in Table 5.2.

Table 5.2. Number of images in the semi-synthetic segmentation data set.

Data set	Training	Validation	Test
Human	1617	539	539
Background	369	123	122
Composed	4022	1381	1294

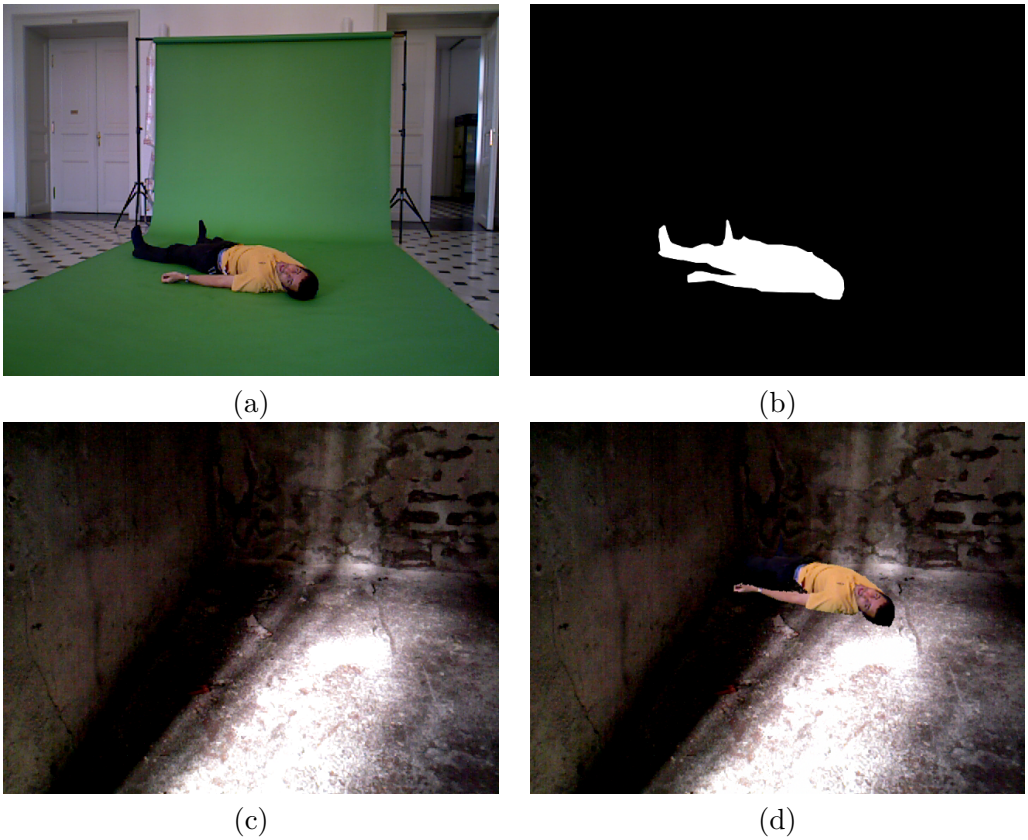


Figure 5.7. Semi-synthetic human body dataset. (a) An image of human body image of a human with (b) the ground-truth segmentation; (c) a background image; (d) a semi-synthetic image composed from the source images.

Table 5.3. Number of images (sequences) in the panoramic segmentation data set from the search & rescue platform.

Data set	Training	Validation	Test
Human / Background	225 (15)	60 (4)	60 (4)

5.4.2. Panoramic Human Body Dataset

The panoramic human body dataset [2] was captured indoors using the mobile search & rescue platform depicted in Fig. 5.2. During data capture, the robot localized itself using the ICP-based SLAM from [72, 87]. We recorded 24 sequences in total (see Table 5.3 for summary) with the robot following a straight path during which it was stopping regularly to capture data, including the thermal images from 13 viewpoints. The data allow to generate instances of the simultaneous exploration and segmentation task outlined above. The panoramic `RGB` images from the Ladybug 3 camera are 1024×512 pixels in size, the depth and thermal images are rendered in the same resolution from captured data and corresponding voxel maps—see Fig. 5.11 for an example.

5.5. Experiments

The experiments are divided into a synthetic evaluation (Section 5.5.1), which mainly shows the influence of different hardware and learning setups, and real (Section 5.5.3) which compares the behavior of the learned policy and the greedy algorithm on the search & rescue platform.

5.5.1. Synthetic Experiments

This section provides the comparison of the proposed guided Q-learning method `GQ-policy` in terms of the total $\Delta\epsilon$ of covered voxels. We provide the comparison on 64 randomly generated maze-like maps for the following methods:

- `greedy` reactive control similar to [86], which at each position choses the viewpoint maximizing $\Delta\epsilon$ of voxels
- `Q-policy` reactive control learned by Q-learning similar to [58].
- `optimal` control estimated as a solution of MILP by the CPLEX solver. It creates a theoretical upper bound for the case in which the map, gain and visibility of all voxels along the whole robot's path is known in advance. This method is mainly used to normalize the results and make maps with significantly different sum of gains comparable.
- `optimal-incomplete` control estimated as repeated optimization of MILP by the CPLEX solver on the so far available incomplete map. It requires to update the map and recompute the visibility of voxels and re-plan the trajectory at each robot's position.
- the `A`* control estimated as a `A`*-like search of the optimal trajectory, which solves the same task as the MILP for the `optimal` control, but the number of expanded nodes is limited 10^5. Again, it is assumed that the map, $\Delta\epsilon$ and visibility of all voxels along the whole robot's path is known in advance.

`GQ-policy` and `Q-policy` policies are modeled by the CNN with the same number of hidden and output layers and neurons, only the number of inputs is different if influence of possible features is evaluated. Considered features are denoted as follows:

[2] http://ptak.felk.cvut.cz/tradr/data/active_seg

Table 5.4. Comparison of all methods. The hardware setup corresponds to the one used in real experiments (Section 5.5.3)

Method	rs$\Delta\epsilon$	
	mean	variance
GQ-policy ($\Delta\epsilon$)	0.807	0.012
GQ-policy ($\Delta\epsilon$+D)	0.846	0.012
GQ-policy ($\Delta\epsilon$+D+$\Delta\epsilon$cog)	0.884	0.006
optimal-incomplete	0.847	0.010
greedy	0.657	0.013
Q-policy ($\Delta\epsilon$+D+$\Delta\epsilon$cog)	0.722	0.013
A* with 10^5 nodes	0.943	0.003
optimal	1.000	0.000

Each row corresponds to the results achieved by particular method on 64 synthetically generated testing maps.

D is sub-sampled layer of pixel depths, $\Delta\epsilon$ is sub-sampled layer of per-pixel-$\Delta\epsilon$ multiplied by depth D, which makes it proportional to the sum of per-voxel-$\Delta\epsilon$ in particular viewpoints. Eventually $\Delta\epsilon\text{cog} \approx \frac{\sum \text{D}\cdot\Delta\epsilon}{\sum \Delta\epsilon}$ is center of gravity of $\Delta\epsilon$, which provides the approximate depth in which the voxels with significant $\Delta\epsilon$ are located. Note that for real experiments (Section 5.5.3) the GQ-policy ($\Delta\epsilon$+D) was used. Table 5.4 compares all these methods, especially for GQ-policy the influence of alternative features is shown. The performance is measured by the relative sum of covered per-voxel-$\Delta\epsilon$-values (rs$\Delta\epsilon$) defined as

$$\text{rs}\Delta\epsilon = \frac{\text{achieved\_sum\_of\_}\Delta\epsilon}{\text{optimal\_sum\_of\_}\Delta\epsilon}. \tag{5.10}$$

Proposed GQ-policy clearly outperforms the greedy algorithm. However we observed that in some cases, the greedy works slightly better than GQ-policy, therefore we also show histogram of differences in rs$\Delta\epsilon$ defined as:

$$\text{drs}\Delta\epsilon = \text{rs}\Delta\epsilon(\text{GQ-policy}) - \text{rs}\Delta\epsilon(\text{greedy}).$$

The histogram, shown on Fig. 5.8, reveals that in only 7% of testing maps the performance is mildly decreased, while in 73% performance is improved by more than 10%.

Fig. 5.9 shows that learning the GQ-policy with *relative* Q-values (see Section 5.3.2) outperforms learning with *absolute* or *not normalized* Q-values, see Section 5.3.2 for Q-values definition. Consequently proposed GQ-policy is learned with *relative* Q-values in all experiments.

We also evaluates the influence of action discretization and range within which the thermal camera operates. The corresponding results are summarized in Table 5.5. The action action discretization is given by the number of distinguished viewpoints. Range 180° corresponds to the thermal camera operating in two frontal quadrants. Range 360° corresponds to the thermal camera operating in all four quadrants with allowed turn over.

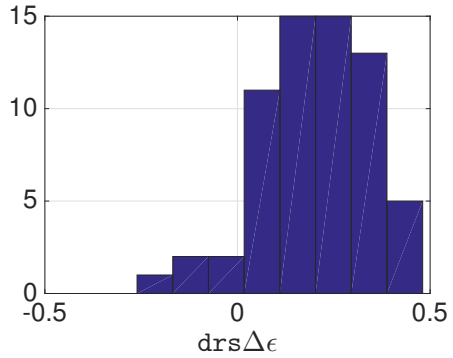


Figure 5.8. Histogram of $\texttt{drs}\Delta\epsilon$: only in 7% of testing maps the performance of the proposed `GQ-policy` is slightly worse than `greedy`.

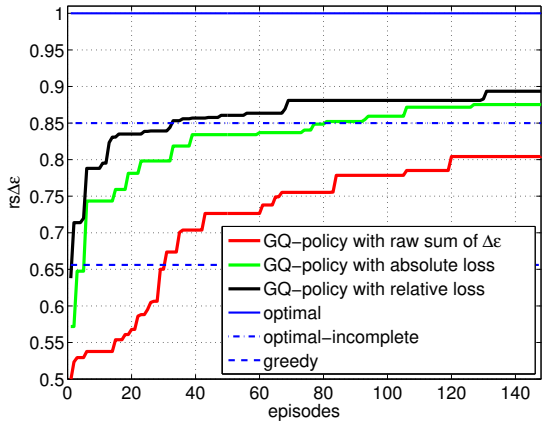


Figure 5.9. Relative sum of $\Delta\epsilon$ as a function learning episodes.

5.5.2. Learning the Image-Based CNN Models

Using SGD with NAG, we performed 10^5 parameter updates with momentum coefficient $\mu = 0.99$, linearly decaying learning rate from $\alpha = 10^{-4}$ to zero, and a single example per batch. An additional L_2 regularization on weights was used with coefficient $\lambda = 5 \times 10^{-4}$. The parameters of the models learned from scratch were initialization using the procedure from [30].

The parameters of the segmentation networks S_θ and S_ψ, and the gain-predicting network $\Delta\mathcal{H}_{\omega_1}$ were selected to minimize the loss on the validation set. [3] The ROC curves for the fine-tuned S_θ and S_ψ networks are shown in Fig. 5.10. As can be seen, the additional thermal modality provides an increase in true positive rate (i.e., recall) of approximately 10% for a wide range of false positive rate. The CNN-based models were implemented in the *Caffe* framework [38].

5.5.3. Real Experiments

The control policies were also evaluated on the mobile search & rescue platform and test sequences from the panoramic dataset described in Sec. 5.4.2. As in the synthetic experiments, the robot was following a straight path discretized into 14 positions at which viewpoints were to be selected. Viewpoint i_k at position k was selected based on

[3] Namely the parameters θ from iteration 8×10^3, ψ from 14×10^3, ω_1 for true $\Delta\mathcal{H}$ prediction from iteration 64×10^3, and ω_1 for self-initialized D_{KL} prediction from iteration 90×10^3 were selected.

Table 5.5. Influence of the action discretization and range.

Method	rs$\Delta\epsilon$	
	mean	variance
GQ-policy 7 viewpoints, 180°	0.853	0.014
GQ-policy 13 viewpoints, 180°	0.846	0.012
GQ-policy 25 viewpoints, 180°	0.821	0.012
GQ-policy 24 viewpoints, 360°	0.853	0.008
greedy 7 viewpoints, 180°	0.772	0.020
greedy 13 viewpoints, 180°	0.657	0.013
greedy 25 viewpoints, 180°	0.676	0.013
greedy 24 viewpoints, 360°	0.628	0.016

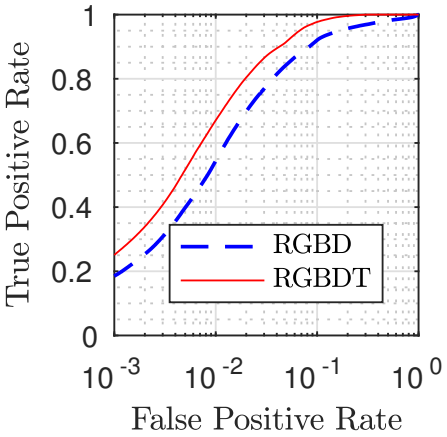


Figure 5.10. ROC curves for the two segmentation networks, S_θ using RGBD and S_ψ using RGBDT as input, evaluated on the panoramic test dataset.

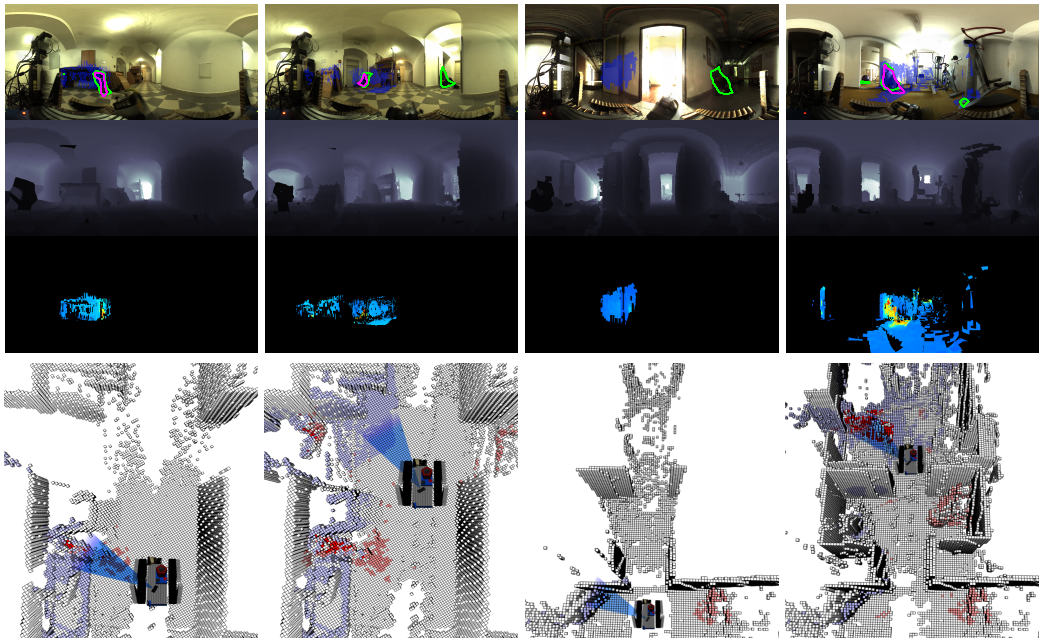


Figure 5.11. Panoramic images and corresponding voxel maps from two experiments on test data with the mobile search & rescue platform. (top) Three panoramic images: (i) `RGB` image with humans delineated by green and magenta contours, as given by segmentation from the $S_\theta(\mathbf{x})$ and $S_\psi(\mathbf{x}, \mathbf{z})$ networks, respectively, and blue overlay denoting the accumulated temperature measurements, (ii) depth image and (iii) thermal image, both rendered from the voxel map. (bottom) Reconstructed and segmented voxel map with accumulated thermal measurements in blue. Light red denotes the voxels marked as human by the $S_\theta(\mathbf{x})$ network based on the `RGBD` data only, dark red denotes the voxels marked as human by the $S_\psi(\mathbf{x}, \mathbf{z})$ network based on the data with additional temperature measurements.

the observations from the preceding position $k - 1$.

We compared the following control policies:

- *RGBD* uses only the segmentation from $S_\theta(\mathbf{x})$ and thus no thermal measurements. It provides a loose lower bound on the performance since the additional thermal modality provides an important cue with respect to the segmentation task and improves the performance in general, no matter what views are selected.
- *DQN* provides reactive control similar to Mnih *et al.* [58] with the double DQN extension from [32] and the prioritized experience replay from [80].
- *Greedy* D_{KL} corresponds to the $\Delta\mathcal{H}_{\omega_1}$ network predicting the gain obtained via self-initialization. The predicted pixel-wise gain is accumulated by viewpoint kernels and the maximum within the motion constraints is selected for the next action.
- GQ_0 D_{KL} corresponds to the self-initialized Q_ω network.
- GQ_1 $\Delta\mathcal{H}$ corresponds to the Q_ω network fine-tuned on the guiding trajectories ($p = 1$) with ω_1 previously trained to predict true gain $\Delta\mathcal{H}$.
- *Optimal* uses additional information of true $\Delta\mathcal{H}$ to plan the optimal trajectory by solving instances of MILP.

DQN usually needs millions of examples to achieve satisfying results. The computational complexity of our task does not allow to sample such a number of training data. Consequently, we modified some parameters to accommodate our setting. [4] The

[4] Training parameters of DQN:

optimization was carried out in the Tensorflow library [1] using SGD with gradient clipping to maximum norm of 10. The DQN network used the same architecture as our Q_ω network but without normalizing the gain prior to the fully-connected control sub-network as it must predict absolute expected rewards. The gain-predicting sub-network was initialized with the same parameters ω_1 as GQ_1 $\Delta\mathcal{H}$ prior to fine-tuning, the control sub-network was initialized with random weights according to [30]. During learning, 10^4 experience examples were gathered in total. Finally, the model achieving the highest rewards on the validation sequences was selected for testing.

Our control policies GQ_0 D_{KL} and GQ_1 $\Delta\mathcal{H}$ were initialized using the model parameters learned in Sec. 5.5.1 and 5.5.2. The GQ_1 $\Delta\mathcal{H}$ network were further fine-tuned on 2198 training examples from optimal plans provided by the CPLEX solver as solutions to the corresponding instances of MILP. From the guiding trajectories, 15 were of full length (14 viewpoints to plan) and 29 were of varying length ≥ 5 generated from the same source data. To reduce the planning time, planning horizon $K = 6$ were used, which still allowed to plan one full sweep ahead. The model with the lowest error [5] obtained on 578 guiding examples from 4 validation source sequences was selected for the comparison.

Since the *RGBD* and *Optimal* policies provide loose bounds on the performance from both sides, we are actually interested in evaluating the relative performance with respect to these bounds. In Fig. 5.12, we thus normalize the true positive rate with respect to the bounds provided by the *RGBD* and *Optimal* policies.

Using temperature as an additional modality generally improves the performance and the extent of such improvement varies with policy, due to different thermal images captured. The *DQN* policy is on par with the *Greedy* D_{KL} policy in most of the FPR range, being outperformed by both the self-initialized policy GQ_1 D_{KL} and the policy GQ_1 $\Delta\mathcal{H}$ fine-tuned on guiding trajectories.

5.6. Conclusion

We have proposed a guided self-supervised learning method of a deep policy network used for active segmentation. The method was evaluated on a real robotic platform, where the learned policy controls the motion of a thermal camera mounted on a pan-tilt unit to achieve low segmentation error. We have experimentally verified that the proposed learning method outperforms other approaches.

batch size	1	replay memory size	10^3
learning rate	10^{-4}	replay start size	50
gradient momentum	0.99	initial exploration prob.	0.9
target network update freq.	100	final exploration	0.1
discount factor	0.99	final exploration frame	5000

[5] Namely the parameters ω from iteration 88×10^3.

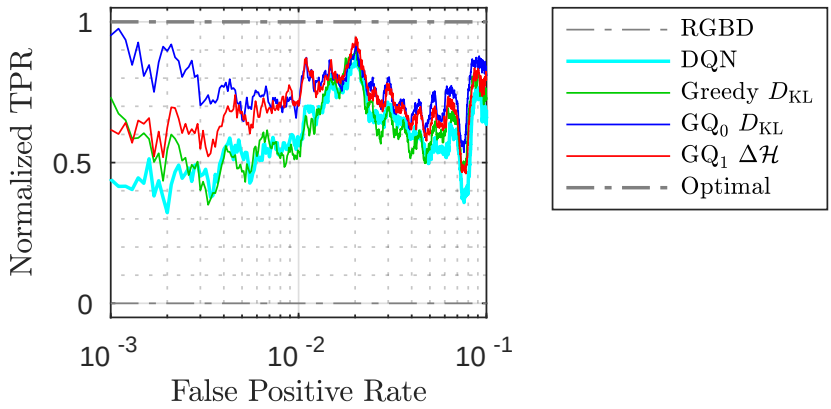


Figure 5.12. ROC curves for resulting human-background segmentation of the voxel maps from 20 instances of the simultaneous exploration and segmentation task. The instances were generated from 4 full test sequences by randomly selecting starting position k, viewpoint i_k, and planning horizon K. True positive rate is normalized with respect to the loose bounds provided by the *RGBD* policy using no thermal measurements and the *Optimal* policy using additional information about true gain $\Delta\mathcal{H}$ to plan optimal trajectories.

6. Coupled Learning and Planning for Active 3D Mapping

Development of autonomous vehicles such as self-driving cars or ground robots has attracted substantial attention of the robotics community in the last few years. One of the reasons is that an accurate 3D perception, which is an essential component for many fundamental capabilities such as emergency braking, predictive active damping or self-localization from offline maps [82], has finally become possible. The autonomous vehicles will require a sensor providing high resolution and long range 3D measurements. Since state-of-the-art rotating lidars are very expensive, heavy and contain moving parts prone to mechanical wear, several manufacturers have announced the development of cheaper, lighter, smaller and motionless solid-state lidars (SSL), which should become available soon. For example, Quanergy Systems have demonstrated a prototype of SSL with target cost of $250 at automotive scale production [2], followed by Innoviz [3] or Velodyne [76].

In contrast to rotating lidars, the SSL uses an optical phased array as a transmitter of depth measuring light pulses. Since the built-in electronics can independently steer pulses of light by shifting its phase as it is projected through the array, the SSL can focus its attention on the parts of the scene important for the current task. Task-driven reactive control steering hundreds of thousands of rays per second using only an on-board computer is a challenging problem, which calls for highly efficient parallelizable algorithms. As a first step towards this goal, we propose an active mapping method for SSL-like sensors, which simultaneously (i) learns to *reconstruct a dense 3D voxel-map* from sparse depth measurements and (ii) optimize the reactive *control of depth-measuring rays*, see Figure **??**. The proposed method is evaluated on a subset of the KITTI dataset [29], where sparse SSL measurements are artificially synthesized from captured lidar scans, and compared to a state-of-the-art 3D reconstruction approach [16].

The main contribution of this chapter lies in proposing a computationally tractable approach for very high-dimensional active perception task, which couples learning of the 3D reconstruction with the optimization of depth-measuring rays. Unlike other approaches such as active object detection [37] or segmentation [57], SSL-like reactive control has significantly higher dimensionality of the state-action space, which makes a direct application of unsupervised reinforcement learning [37] prohibitively expensive. Keeping the on-board reactive control in mind, we propose prioritized greedy optimization of depth measuring rays, which in contrast to a naïve greedy algorithm re-evaluates only 1/500 rays in each iteration. We derive the approximation ratio of the proposed algorithm. The method is compared with the state-of-the-art 3D reconstruction approach [16] on a publicly available dataset [29].

The 3D mapping is handled by an iteratively learned convolution neural network (CNN), as CNNs proved their superior performance in [16, 99]. The iterative learning procedure stems from the fact that both (i) the directions in which the depth should be measured and (ii) the weights of the 3D reconstruction network are unknown. We initialize the learning procedure by selecting depth-measuring rays randomly to learn an initial 3D mapping network which estimates occupancy of each particular voxel. Then,

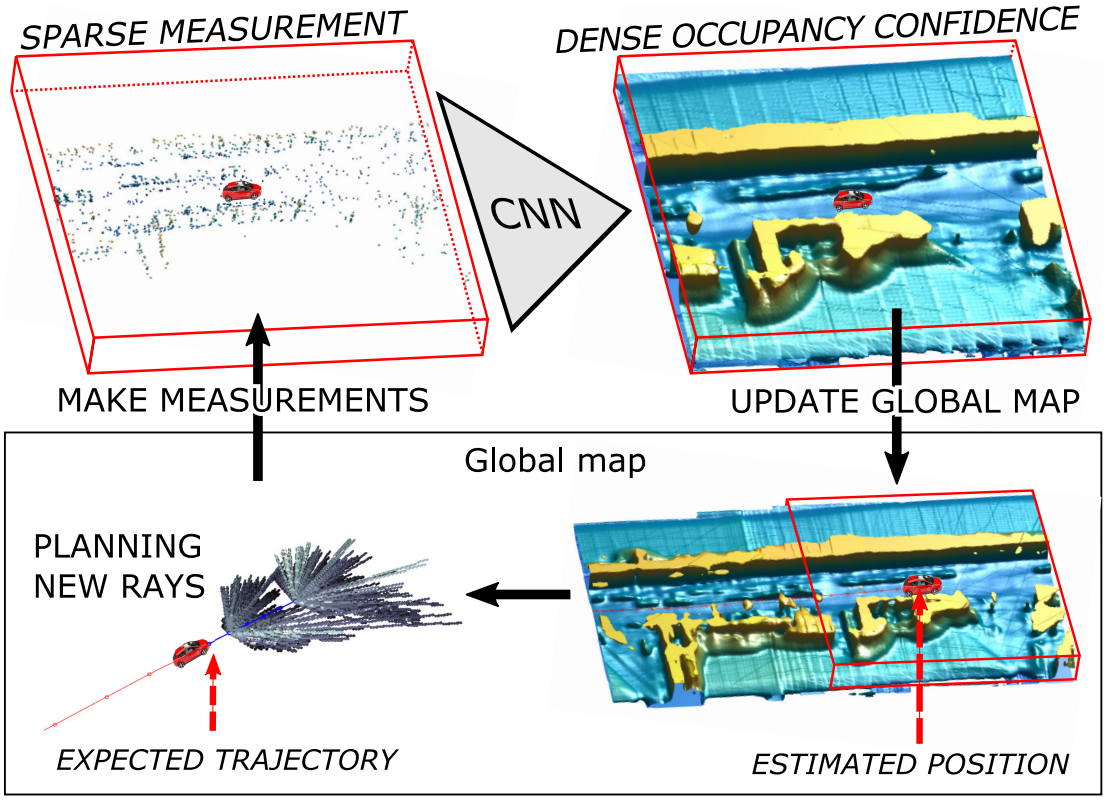


Figure 6.1. Active 3D mapping with solid state lidar. Iteratively learned deep convolutional network reconstructs local dense occupancy map from sparse depth measurements. The local map is registered to a global occupancy map, which in turn serves as an input for the optimization of depth-measuring rays along the expected vehicle trajectory. The dense occupancy maps are visualized as isosurfaces.

using this network, depth-measuring rays along the expected vehicle trajectory can be planned based on the expected reconstruction (in)accuracy in each voxel. To reduce the training-planning discrepancy, the mapping network is re-learned on optimized sparse measurements and the whole process is iterated until validation error stops decreasing.

Chapter Outline

6.1. Previous Work

High performance of image-based models is demonstrated in [90], where a CNN pooling results from multiple rendered views outperforms commonly used 3D shape descriptors in object recognition task.

Several volumetric and multi-view network architectures for object classification are compared by Qi *et al.* [74]. The authors focus on closing a performance gap between these two approaches and investigate several techniques towards this goal, such as data augmentation, or using 2D convolution with elongated kernels for projecting volumetric representation into a 2D image. We choose a similar approach in designing the mapping network.

Choy *et al.* [16] proposed a unified approach for single and multi-view 3D object reconstruction which employs a recurrent neural architecture. Their recurrent neural network architecture learns a map from sequences of images to object shapes in terms of 3D occupancy grid ($32 \times 32 \times 32$ voxels). Despite providing competitive results in the object reconstruction domain, the architecture is not suitable for dealing with high-dimensional outputs due to its high memory requirements and would need significant modifications to train with full-resolution maps which we use. We provide a comparison of this method to ours in Sec. 6.5.3, in a limited setting.

Model-fitting methods such as [84, 91, 75] rely on a manually-annotated dataset of models and assume that objects can be decomposed into a predefined set of parts. Besides that these methods are suited mostly for man-made objects of rigid structure, fitting of the models and their parts to the input points is computationally very expensive (e.g., minutes per input for [84, 91]) and prevents its usage within our active mapping scenario. Decomposition of the scene into plane primitives as in [59] does not scale well with scene size (quadratically due to candidate pairs) and could not most likely deal with the level of sparsity we encounter.

Geometrical and physical reasoning comprising stability of objects in the scene is used by Zheng *et al.* [103] to improve object segmentation and 3D volumetric recovery. First, solid 3D primitives are recovered from point cloud, and then the unstable objects are grouped with the physically stable ones to minimize an energy function which includes a penalty for object (in)stability, size, geometric complexity etc. The proposed volumetric recovery is based on implicit algebraic models and the assumption of objects being aligned with coordinate axes which seems unrealistic in practice. Their assumption of objects being aligned with coordinate axes which seems unrealistic in practice. Moreover, it is not clear how to incorporate learned shape priors for complex real-world objects which were shown to be beneficial for many tasks (e.g., in [63]). Firman *et al.* [26] use a structured-output regression forest to complete unobserved geometry of tabletop-sized objects. The regressor, learned from already available volumetric elements, casts votes (termed *voxlets*) into a volumetric representation which keeps track of signed distance to the surface from each voxel. The marching-cubes algorithm is used to convert the signed-distance representation into a polygonal mesh.

A generative model proposed by Wu *et al.* [99], termed Deep Belief Network, learns joint probability distribution $p(\mathbf{x}, y)$ of complex 3D shapes $\mathbf{x}$ across various object categories y. Their model assumes that all cameras are registered in a common reference frame so that 2.5D images can be converted to a 3D occupancy grid ($30 \times 30 \times 30$ voxels). The authors suggest to use the model for Next-Best-View prediction via rendering view hypotheses by Gibbs sampling and selecting the view maximizing mutual information between class label y and the newly observed voxels conditioned on current observation.

End-to-end learning of stochastic motion control policies for active object and scene

categorization is proposed by Jayaraman and Grauman [37]. Their CNN policy successively proposes a distibution of camera views to capture with RGB camera to minimize categorization error. The authors suggest that active vision requires an agent to be able to reason about the effects of the actions it executes on its internal representation, and use a look-ahead error as an unsupervised regularizer on the classification objective.

Andreopoulos *et al.* [6] solve the problem of an active search for an object in a 3D environment. While they minimize the classification error of a single yet apriori unknown voxel containing the searched object, we minimize the expected reconstruction error of all voxels. Also, their action space is significantly smaller than ours because they consider only local viewpoint changes at the next position while the SSL planning chooses from tens of thousands of rays over a longer horizon. Similarly to us, the world is modeled by a confidence voxel map. Their action space is, nevertheless, significantly smaller than ours because only local viewpoint changes are considered while the SSL planning chooses from tens of thousands of rays. Also, they minimize the classification error of a single yet apriori unknown voxel containing the searched object, while we minimize the expected reconstruction error of all voxels. Finally, they plan the sensor motion greedily only over the single position, while we search for a solution with a longer horizon.

6.2. Overview of the Active 3D Mapping

We assume that the vehicle follows a known path consisting of L discrete positions and a depth measuring device (SSL) can capture at most K rays at each position. The set of rays to be captured at position l is denoted J_l.

We denote $\mathbf{Y}$ the global ground-truth occupancy map, $\hat{\mathbf{Y}}$ its estimate, and $\mathbf{X}$ the map of the sparse measurements. All these map share common global reference frame corresponding to the first position in the path. For each of these maps there are local counterparts $\mathbf{y}_l, \hat{\mathbf{y}}_l$, and $\mathbf{x}_l$, respectively. Local maps corresponding to position l all share a common reference frame which is aligned with the sensor and captures its local neighborhood of size $64\text{m} \times 64\text{m} \times 6.4\text{m}$ discretized into $320 \times 320 \times 32$ voxels. The global ground-truth map $\mathbf{Y}$ is used to synthesize sensor measurements $\mathbf{x}_l$ and to generate local ground-truth maps $\mathbf{y}_l$ for training.

The active mapping pipeline, consisting of a measure-reconstruct-plan loop, is depicted in Fig. ?? and detailed in Alg. 6.1. Neglecting sensor noise, the set of depth-

Algorithm 6.1. Active 3D mapping.

1: Initialize position $l \leftarrow 0$ and select depth-measuring rays randomly.
2: Measure depth in the directions selected for position l and update global sparse measurements $\mathbf{X}$ and dense reconstruction $\hat{\mathbf{Y}}$ with these measurements.
3: Obtain local measurements $\mathbf{x}_l$ by interpolating $\mathbf{X}$.
4: Compute local occupancy confidence $\hat{\mathbf{y}}_l = \mathbf{h}_\theta(\mathbf{x}_l)$ using the mapping network $\mathbf{h}_\theta$.
5: Update global occupancy confidence $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} + \hat{\mathbf{y}}_l$.
6: Plan depth-measuring rays along the expected vehicle trajectory over horizon L given reconstruction $\hat{\mathbf{Y}}$.
7: Repeat from line 2 for next position $l \leftarrow l + 1$.

measuring rays obtained from the planning, the measurements $\mathbf{x}_l$, and the resulting reconstruction $\hat{\mathbf{Y}}$ can all be seen as a deterministic function of mapping parameters θ and $\mathbf{Y}$. If we assume that ground-truth maps $\mathbf{Y}$ come from a probability distribution,

both learning of θ and planning of the depth-measuring rays approximately minimize common objective

$$\mathbb{E}_{\mathbf{Y}}\left\{\mathcal{L}\left(\mathbf{Y}, \hat{\mathbf{Y}}\left(\theta, \mathbf{Y}\right)\right)\right\}, \tag{6.1}$$

where

$$\mathcal{L}\left(\mathbf{Y}, \hat{\mathbf{Y}}\right) = \sum_i w_i \log\left(1 + \exp(-Y_i \hat{Y}_i)\right) \tag{6.2}$$

is the weighted logistic loss, $Y_i \in \{-1, 1\}$ and $\hat{Y}_i \in \mathbb{R}$ denote the elements of $\mathbf{Y}$ and $\hat{\mathbf{Y}}$, respectively, corresponding to voxel i. In learning, $w_i \geq 0$ are used to balance the two classes, *empty* with $Y_i = -1$ and *occupied* with $Y_i = 1$, and to ignore the voxels with unknown occupancy. We assume independence of measurements and use, for corresponding voxels i, additive updates of the occupancy confidence $\hat{Y}_i \leftarrow \hat{Y}_i + h_i(\mathbf{x}_l)$ with $h_i(\mathbf{x}_l) \approx \log\left(\Pr\left(Y_i = 1|\mathbf{x}_l\right)/\Pr\left(Y_i = -1|\mathbf{x}_l\right)\right)$. $\Pr\left(Y_i = 1|\mathbf{x}_l\right)$ denotes the conditional probability of voxel i being occupied given measurements $\mathbf{x}_l$ and $\sigma\left(\hat{Y}_i\right) = 1/\left(1 + e^{-\hat{Y}_i}\right)$ is its current estimate.

6.3. Learning of 3D Mapping Network

The learning is defined as approximate minimization of Equation 6.1. Since (i) the result of planning $\mathbf{x}_l\left(\theta, \mathbf{Y}\right)$ is not differentiable with respect to θ and (ii) we want to reduce variability of training data[1], we locally approximate the criterion around a point θ^0 as

$$\mathbb{E}_{\mathbf{Y}}\left\{\sum_l \mathcal{L}\left(\mathbf{y}_l, \mathbf{h}_\theta\left(\mathbf{x}_l\left(\theta^0, \mathbf{Y}\right)\right)\right)\right\} \tag{6.3}$$

by fixing the result of planning in $\mathbf{x}_l(\theta^0, \mathbf{Y})$. We also introduce a canonical frame by using the local maps instead of the global ones, which helps the mapping network to capture local regularities. The learning then becomes the following iterative optimization

$$\theta^t = \arg\min_\theta \mathbb{E}_{\mathbf{Y}}\left\{\sum_l \mathcal{L}\left(\mathbf{y}_l, \mathbf{h}_\theta\left(\mathbf{x}_l\left(\theta^{t-1}, \mathbf{Y}\right)\right)\right)\right\}, \tag{6.4}$$

where minimization in each iteration is tackled by Stochastic Gradient Descent. Learning is summarized in Alg. 6.2.

Algorithm 6.2. Learning of active mapping.

1: Initialize $t \leftarrow 0$ and obtain dataset $\mathcal{D}_0 = \{(\mathbf{x}_l, \mathbf{y}_l)\}_l$ by running the pipeline with the rays being selected randomly, instead of using the planner.
2: Train the mapping network on $\mathcal{D}_t$ to obtain $\mathbf{h}_{\theta^t}$ with parameters θ^t.
3: Obtain $\mathcal{D}_{t+1} = \left\{\left(\mathbf{x}_l\left(\theta^t, \mathbf{Y}\right), \mathbf{y}_l\right)\right\}_l$ by running Alg. 6.1 and using $\mathbf{h}_{\theta^t}$ for mapping.
4: Set $t \leftarrow t + 1$ and repeat from line 2 until validation error stops decreasing.

Note, that in order to achieve (i) local optimality of the criterion and (ii) statistical consistency of the learning process (i.e., that the training distribution of sparse measurements $\mathbf{x}_l$ corresponds to the one obtained by planning), one would have to find a fixed point of Equation 6.4. Since there are no guarantees that any fixed point exists, we instead iterate the minimization until validation error is decreasing.

[1] We introduce a canonical frame by using the local maps instead of the global ones, which helps the mapping network to capture local regularities.

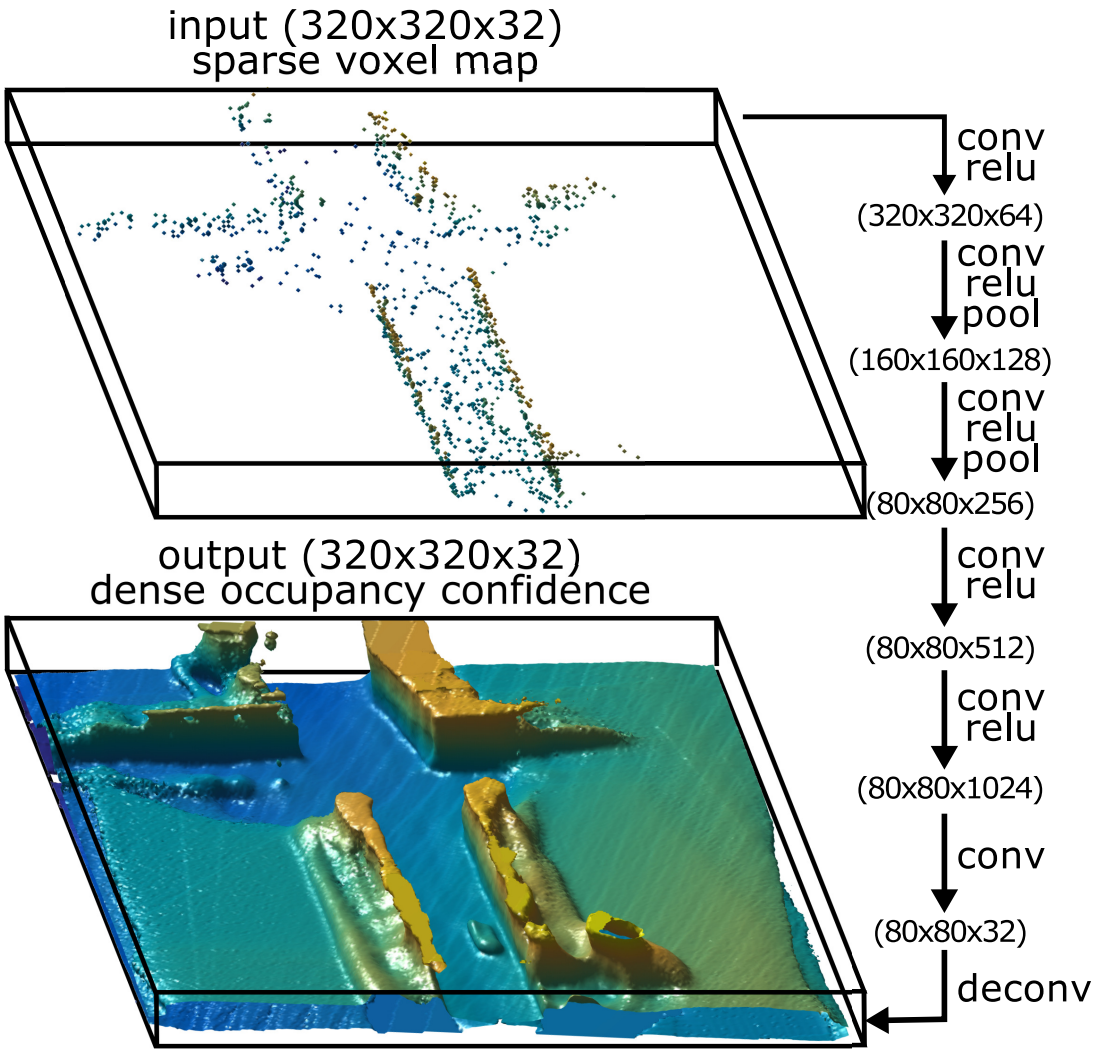


Figure 6.2. Architecture of the mapping network. (top) An example input with sparse measurements, showing only the occupied voxels. (bottom) The corresponding reconstructed dense occupancy confidence after thresholding. (right) Schema of the network architecture, composed of the convolutional layers, denoted *conv*, linear rectifier units, denoted *relu*, pooling layers, denoted *pool*, and upsampling layers, denoted *deconv*.

6.3.1. Structure of Mapping Network

The mapping network consists of 6 convolutional layers with 5×5 kernels followed by linear rectifier units (element-wise $\max\{x, 0\}$) and, in 2 cases, by max pooling layers with 2×2 kernels and stride 2, see Fig. 6.2. In the end, there is an fourfold upsampling layer so that the output has same size as input. The network was implemented in *MatConvNet* [97].

6.4. Planning of Depth Measuring Rays

Planning at position l searches for a set of rays J, which approximately minimizes the expected logistic loss $\mathcal{L}(\mathbf{Y}, \mathbf{h}_{\theta^t}(\mathbf{x}_{l+L}))$ between ground truth map $\mathbf{Y}$ and reconstruction obtained from sparse measurements $\mathbf{x}_{l+L}$ at the horizon L. The result of planning is the set of rays J which will provide measurements for a sparse set of voxels. This set of voxels is referred to as *covered* by J and denoted as $C(J)$. While the mapping network is trained *offline* on the ground-truth maps, the planning have to search the subset of rays *online* without any explicit knowledge of the ground-truth occupancy $\mathbf{Y}$. Since it is not clear how to directly quantify the impact of measuring a subset of voxels on the reconstruction $\mathbf{h}_{\theta^t}(\mathbf{x}_{l+L})$, we introduce simplified reconstruction model $\hat{\mathbf{h}}(J, \hat{\mathbf{Y}})$, which predicts the loss based on currently available map $\hat{\mathbf{Y}}$. This model conservatively assumes that the reconstruction in covered voxels $i \in C(J)$ is correct (i.e., $\mathcal{L}\big(Y_i, \hat{h}_i\big(J, \hat{\mathbf{Y}}\big)\big) = 0$) and the reconstruction of not covered voxels $i \notin C(J)$ does not change (i.e., $\mathcal{L}\big(Y_i, \hat{h}_i\big(J, \hat{\mathbf{Y}}\big)\big) = \mathcal{L}(Y_i, \hat{Y}_i)$). Given this reconstruction model, the expected loss simplifies to the following:

$$\sum_i \mathcal{L}\left(Y_i, \hat{h}_i\left(J, \hat{\mathbf{Y}}\right)\right) = \sum_{i \notin C(J)} \mathcal{L}\left(Y_i, \hat{Y}_i\right). \tag{6.5}$$

Since the ground-truth occupancy of voxels is apriori unknown, neither the voxel-wise loss nor the coverage are known. We model the expected loss in voxel i as

$$\mathcal{L}\left(Y_i, \hat{Y}_i\right) \approx \mathbb{E}_{Y_i \sim \mathrm{B}(\sigma(\hat{Y}_i))}\left\{\mathcal{L}\left(Y_i, \hat{Y}_i\right)\right\} = \mathcal{H}\left(\mathrm{B}\left(\sigma\left(\hat{Y}_i\right)\right)\right) = \epsilon_i, \tag{6.6}$$

where $\mathcal{H}\left(\mathrm{B}(p)\right)$ is the entropy of the Bernoulli distribution with parameter p, denoting the probability of outcome 1 from the possible outcomes $\{-1, 1\}$. The vector of concatenated losses ϵ_i is denoted $\boldsymbol{\epsilon}$.

The length of particular rays is also unknown, therefore coverage $C(J)$ of voxels by particular rays cannot be determined uniquely. Therefore, we introduce probability p_{ij} that voxel i will not be covered by ray $j \in J$. This probability is estimated from currently available map $\hat{\mathbf{Y}}$ as the product of (i) the probability that the voxels on ray j which lie between voxel i and the sensor, $R_j^-(i)$, are unoccupied and (ii) the probability that at least one of the following voxels or the voxel i itself, $R_j^+(i)$, are occupied,

$$p_{ij} = 1 - \prod_{u \in R^-(i)} \left(1 - \sigma_u\left(\hat{\mathbf{Y}}\right)\right)\left(1 - \prod_{u \in R^+(i)} \left(1 - \sigma_u\left(\hat{\mathbf{Y}}\right)\right)\right). \tag{6.7}$$

If ray j does not intersect voxel i, then $p_{ij} = 1$. The vector of probabilities p_{ij} for ray j is denoted $\mathbf{p}_j$. Assuming that rays J are independent measurements, the expected loss is modeled as $\boldsymbol{\epsilon}^\mathsf{T} \prod_{j \in J} \mathbf{p}_j$.

The planning searches for the set $J = J_1 \cup \cdots \cup J_L$ of subsets $J_1 \dots J_L$ of depth-measuring rays for the following L positions, which minimize the expected loss, subject to budget constraints $|J_1| \leq K, \dots |J_L| \leq K$

$$J^* = \arg\min_J \boldsymbol{\epsilon}^\mathsf{T} \prod_{j \in J} \mathbf{p}_j, \tag{6.8}$$

$$\text{s.t. } |J_1| \leq K, \dots |J_L| \leq K, \tag{6.9}$$

where $|J_l|$ denotes cardinality of the set J_l.

This is a non-convex combinatorial problem[2] which needs to be solved online repeatedly for millions of potential rays. We tried several convex approximations, however the high-dimensional optimization has been extremely time consuming and the improvement with respect to the significantly faster greedy algorithm was negligible. As a consequence of that, we have decided to use the greedy algorithm. We first introduce its simplified version (Alg. 6.3) and derive its properties, the significantly faster prioritized greedy algorithm (Alg. 6.4) is explained later.

We denote the list of available rays at position l as V_l. At the beginning, the list of all available rays is initialized as follows $V = V_1 \cup \cdots \cup V_L$. Alg. 6.3 successively builds the set of selected rays J. In each iteration the best ray j^* is selected, added into J and removed from V. The position from which the ray j^* is chosen is denoted l^*. If the budget K of l^* is reached, all rays from V_{l^*} are removed from V.

In order to avoid multiplication of all selected rays at each iteration, we introduce the vector $\mathbf{b}$, which keeps voxel loss. Vector $\mathbf{b}$ is initialized as $\mathbf{b} = \boldsymbol{\epsilon}$ and whenever ray j is selected, voxel losses are updated as follows $\mathbf{b} = \mathbf{b} \odot \mathbf{p}_j$, where $\odot$ denotes element-wise multiplication.

Algorithm 6.3. Greedy planning of depth measuring rays.

Require: Set of available rays V and budget K

$J \leftarrow \emptyset$ ▷ Initialization
$\mathbf{b} \leftarrow \boldsymbol{\epsilon}$
while $\neg(V = \emptyset)$ **do**
  $j^* \leftarrow \arg\min_{j \in V} \mathbf{b}^\mathsf{T}\mathbf{p}_j$ ▷ Add the best ray.
  $J \leftarrow J \cup j^*$
  $\mathbf{b} \leftarrow \mathbf{b} \odot \mathbf{p}_j$ ▷ Update voxel costs.
  $V \leftarrow V \setminus j^*$ ▷ Remove j^* from V.
  if $|J_{l^*}| = K$ **then**
    $V \leftarrow V \setminus V_{l^*}$ ▷ Close the position.
  end if
end while
return Set of selected rays J

The rest of this section is organized as follows. Section 6.4.1 shows the upper bound for the approximation ratio of the greedy algorithm. Section 6.4.2 introduces the prioritized greedy algorithm, which in each iteration needs to re-evaluate the cost function $\mathbf{b}^\mathsf{T}\mathbf{p}_j$ only for a small fraction of rays.

6.4.1. Approximation Ratio of the Greedy Algorithm

We define the approximation ratio of a minimization algorithm to be $\rho = \frac{f}{\text{OPT}}$, where f is the cost function achieved by the algorithm and OPT is the optimal value of the cost

[2] In our experiments, the number of possible combinations is greater then 10^{2000}.

function. Given ρ, we know that the algorithm provides solution whose value is at most ρ OPT. In this section we derive the upper bound of the approximation ratio $\mathrm{UB}(\rho)$ of Algorithm 6.3. Fig. 6.3 shows values of $\mathrm{UB}(\rho)$ for different number of positions L.

The greedy algorithm successively selects rays that reduce the cost function the most. To show how cost function differs from OPT, an upper bound on the cost function need to be derived. Let us suppose that in the beginning of an arbitrary iteration we have voxel losses given by vector $\mathbf{b}$, the following lemma states that for arbitrary voxel i, there always exists a ray j, that reduces the cost function to $\sum_i b_i(1-\frac{1}{K})+\frac{\mathrm{OPT}}{K}$, where

$$\mathrm{OPT} = \mathbf{1}^\mathsf{T} \prod_{j=1}^{K} \mathbf{p}_j = \mathbf{1}^\mathsf{T} \mathbf{p}^{\mathrm{OPT}} \tag{6.10}$$

is the unknown optimum value of the cost function which is achievable by K rays $\mathbf{p}_1 \dots \mathbf{p}_K$.

Lemma 6.4.1. *If for some rays* $\prod_{j=1}^{K} p_{ij} = p_i^{\mathrm{OPT}}$ *then*

$$\forall_{\mathbf{0}\leq\mathbf{b}\leq\mathbf{1}} \exists_j \sum_{i=1}^{V} p_{ij} b_i \leq \sum_{i=1}^{V} b_i \left(1 - \frac{1}{K}\right) + \frac{\mathrm{OPT}}{K}. \tag{6.11}$$

Proof. We know that there is optimal solution consisting from K rays. Without loss of generality we assume that $\prod_{j=1}^{K} p_{ij} = p_i^{\mathrm{OPT}}$ holds for first K rays, then

$$\forall_i \sum_{j=1}^{K} p_{ij} \leq K - 1 + p_i^{\mathrm{OPT}}. \tag{6.12}$$

This holds for an arbitrary positive scaling factor b_i, therefore

$$\forall_i \sum_{j=1}^{K} p_{ij} b_i \leq (K - 1 + p_i^{\mathrm{OPT}}) b_i. \tag{6.13}$$

We sum up inequalities over all voxels i

$$\sum_{i=1}^{V} \sum_{j=1}^{K} p_{ij} b_i \leq \sum_{i=1}^{V} (K - 1 + p_i^{\mathrm{OPT}})\, b_i. \tag{6.14}$$

We switch sums in the left-hand side of the inequality to obtain addition of K terms

$$\sum_{i=1}^{V} p_{i1} b_i + \dots + \sum_{i=1}^{V} p_{iK} b_i \leq \sum_{i=1}^{V} (K - 1 + p_i^{\mathrm{OPT}})\, b_i. \tag{6.15}$$

Hence, we know that at least one of these K terms has to be smaller than or equal to $\frac{1}{K}$ of the right-hand side

$$\begin{aligned} \exists_j \sum_{i=1}^{V} p_{ij} b_i \leq \frac{1}{K} \sum_{i=1}^{V} (K - 1 + p_i^{\mathrm{OPT}}) b_i = \sum_{i=1}^{V} b_i (1 - \frac{1}{K}) + \frac{1}{K} \sum_{i=1}^{V} p_i^{\mathrm{OPT}} b_i \leq \\ \leq \sum_{i=1}^{V} b_i (1 - \frac{1}{K}) + \sum_{i=1}^{V} \frac{p_i^{\mathrm{OPT}}}{K} = \sum_{i=1}^{V} b_i (1 - \frac{1}{K}) + \frac{\mathrm{OPT}}{K} \end{aligned} \tag{6.16}$$

□

Especially, if there is only one position, all optimal K rays $\mathbf{p}_1 \dots \mathbf{p}_K$ are either already selected or still available. This assumption allows to derive the following upper bound on the cost function of the greedy algorithm f^K after K iterations for $L = 1$.

Theorem 6.4.1. *Upper bound* $\mathrm{UB}(f^K) \geq f^K$ *of the greedy algorithm after* K *iterations is*

$$\mathrm{UB}(f^K) = E\frac{1}{e} + \mathrm{OPT}\left(1 - \frac{1}{e}\right), \tag{6.17}$$

where $E = \sum_{i=1}^{V} \epsilon_i$ *and* e *is Euler number.*

Proof. We prove the upper bound by complete induction. In the beginning no ray is selected, per-voxel loss is $b_i^0 = \epsilon_i$ and the value of the cost function $f^0 = \sum_{i=1}^{V} b_i^0 = E$. Using Lemma 6.4.1, we know that there exists ray j such that $\sum_{i=1}^{V} p_{ij} b_i^0 \leq \sum_{i=1}^{V} b_i^0 (1 - \frac{1}{k}) + \frac{\mathrm{OPT}}{K}$, therefore we know that

$$f^1 = \sum_{i=1}^{V} p_{ij} b_i^0 \leq \sum_{i=1}^{V} b_i^0 \left(1 - \frac{1}{K}\right) + \frac{\mathrm{OPT}}{K} = E\left(1 - \frac{1}{K}\right) + \frac{\mathrm{OPT}}{K}. \tag{6.18}$$

The greedy algorithm continues by updating the per-voxel loss $b_i^1 = b_i^0 p_{ij}$.

In the second iteration there are two possible cases: (i) we have either used the optimal ray in the first iteration, then the situation is better and we know there is $(K - 1)$ rays which achieves optimum, or (ii) we have not selected the optimal ray in the first iteration, therefore we have still K rays which achieves the optimum. Since the cost function reduction in the latter case gives the upper bound on the cost function reduction in the former one, we assume that there is still k optimal rays available, therefore there exists ray j such that

$$\begin{aligned} f^2 = \sum_{i=1}^{V} p_{ij} b_i^1 \leq \sum_{i=1}^{V} b_i^1 \left(1 - \frac{1}{k}\right) + \frac{\mathrm{OPT}}{K} \leq \\ \leq E\left(1 - \frac{1}{K}\right)^2 + \frac{\mathrm{OPT}}{K}\left(\left(1 - \frac{1}{K}\right) + 1\right). \end{aligned} \tag{6.19}$$

In the third iteration, similarly

$$\begin{aligned} f^3 = \sum_{i=1}^{V} p_{ij} b_i^2 \leq f^2 \left(1 - \frac{1}{K}\right) + \frac{\mathrm{OPT}}{K} \leq \\ \leq E\left(1 - \frac{1}{K}\right)^3 + \frac{\mathrm{OPT}}{K}\left(\left(1 - \frac{1}{K}\right)^2 + \left(1 - \frac{1}{K}\right) + 1\right). \end{aligned} \tag{6.20}$$

We assume that the following holds

$$f^{t-1} \leq E\left(1 - \frac{1}{K}\right)^{t-1} + \frac{\mathrm{OPT}}{K} \sum_{u=0}^{t-2} \left(1 - \frac{1}{K}\right)^u. \tag{6.21}$$

and prove the inequality for f^t. Using the assumption (6.21) and Lemma 6.4.1, the

following inequalities hold

$$f^t \leq \sum_{i=1}^{V} b_i^{t-1} \left(1 - \frac{1}{K}\right) + \frac{\text{OPT}}{K} \leq$$

$$\leq \left[E\left(1 - \frac{1}{K}\right)^{t-1} + \frac{\text{OPT}}{K} \sum_{u=0}^{t-2} \left(1 - \frac{1}{K}\right)^u\right] \left(1 - \frac{1}{K}\right) + \frac{\text{OPT}}{K}$$

$$= E \underbrace{\left(1 - \frac{1}{K}\right)^t}_{\alpha_t^K} + \text{OPT} \underbrace{\frac{1}{K} \sum_{u=0}^{t-1} \left(1 - \frac{1}{K}\right)^u}_{\beta_t^K}. \tag{6.22}$$

Since $\alpha_t^K + \beta_t^K = 1$ [3] and $\alpha_K = \left(1 - \frac{1}{K}\right)^K \leq \frac{1}{e}$, the upper bound for cost function of the greedy algorithm in Kth iteration is

$$f^K \leq E\frac{1}{e} + \text{OPT}\left(1 - \frac{1}{e}\right). \tag{6.23}$$

□

Theorem 6.4.1 reveals that the approximation ratio of the greedy algorithm $\rho = \frac{f^K}{\text{OPT}}$ after K iterations has following upper bound:

$$\rho \leq \frac{\text{OPT}\left(\frac{E}{\text{OPT}}\frac{1}{e} + \left(1 - \frac{1}{e}\right)\right)}{\text{OPT}} \leq \frac{E}{\text{LB}(\text{OPT})e} + \left(1 - \frac{1}{e}\right). \tag{6.24}$$

We can simply find LB(OPT) by considering for each voxel the best K rays independently.

So far we have assumed that the greedy algorithm chooses only K rays and that all rays are available in all iterations. Since there are L positions and the greedy algorithm can choose only K rays at each position, some rays may be no longer available when choosing $(K+1)$th ray. In the worst case possible, the rays from the most promising position will become unavailable. Since we have not chosen optimal rays we can no longer achieve OPT. Nevertheless, we can still choose from rays which achieve a new optimum.

We introduce $\overline{\text{OPT}}_v$ as the optimum achievable after closing v positions. Obviously $\overline{\text{OPT}}_0 = \text{OPT}$. Let us assume that, when the first position is closed we cannot lose more than R_1, therefore $\overline{\text{OPT}}_1 = \text{OPT} + R_1$. Without any additional assumption, R_1 could be arbitrarily large. We discuss potential assumptions later. Similarly $\overline{\text{OPT}}_2 = \text{OPT} + R_1 + R_2$, and $\overline{\text{OPT}}_v = \text{OPT} + \sum_{l=1}^{v} R_l$. The following theorem states the upper bound for f^{LK} as a function of $\overline{\text{OPT}}_v$.

Theorem 6.4.2. *Upper bound* $\text{UB}(f^{LK}) \geq f^{LK}$ *of the greedy algorithm after* LK *iterations is*

$$\text{UB}(f^{LK}) = E\frac{1}{e} + \sum_{u=0}^{L-1} \gamma_u \overline{\text{OPT}_u}, \tag{6.25}$$

where $\gamma_u = \left(1 - \sqrt[L]{\frac{1}{e}}\right)\left(\sqrt[L]{\frac{1}{e}}\right)^{L-1-u}$.

[3] Using the geometric series summation formula, $\beta_t^K = \frac{1}{K}\sum_{u=0}^{t-1}\left(1 - \frac{1}{K}\right)^u = (1-a)\sum_{u=0}^{t-1} a^u = 1 - a^t = 1 - \left(1 - \frac{1}{K}\right)^t = 1 - \alpha_t^K$ for $a = \left(1 - \frac{1}{K}\right)$.

Proof. We start from the result (6.22) shown in the proof of Theorem 6.4.1. Since there is LK rays achieving optimum $\overline{\text{OPT}_0} = \text{OPT}$, the cost function f^K in Kth iteration is bounded as follows:

$$f^K \leq E \underbrace{\left(1 - \frac{1}{LK}\right)^K}_{\alpha_K^{LK}} + \overline{\text{OPT}_0} \underbrace{\frac{1}{LK} \sum_{u=0}^{K-1} \left(1 - \frac{1}{LK}\right)^u}_{\beta_K^{LK}}. \tag{6.26}$$

In the $(K+1)$th iteration, there are two possible cases: (i) rays from some position l become not available and there is $K(L-1)$ rays available which can achieve a new optimum which is not higher than $\overline{\text{OPT}_1}$ or (ii) all rays are available and there is still LK rays which achieve $\overline{\text{OPT}_0} = \text{OPT}$. Noticing that the upper bound is increasing in $\overline{\text{OPT}_0}$ and L, we can cover both cases by considering there is still LK rays which achieves $\overline{\text{OPT}_1}$, therefore

$$\begin{aligned} f^{K+1} &\leq (E\alpha_K^{LK} + \overline{\text{OPT}_0}\beta_K^{LK})(1 - \frac{1}{LK}) + \frac{\overline{\text{OPT}_1}}{LK} = \\ &= E\alpha_{K+1}^{LK} + \overline{\text{OPT}_0}\beta_K^{LK}(1 - \frac{1}{LK}) + \frac{\overline{\text{OPT}_1}}{LK}. \end{aligned} \tag{6.27}$$

We can now continue up to the iteration $2K$ in which the upper bound is as follows:

$$f^{2K} \leq E\alpha_{2K}^{LK} + \overline{\text{OPT}_0}\beta_K^{LK}\alpha_K^{LK} + \overline{\text{OPT}_1}\beta_K^{LK}. \tag{6.28}$$

For $(2K+1)$th iteration the situation is similar as for $(K+1)$th iteration. In order to cover both cases, we consider that there is LK rays which achieves $\overline{\text{OPT}_2}$ and continue up to the $3k$th iteration, which yields the following upper bound:

$$f^{3K} \leq E\alpha_{3K}^{LK} + \overline{\text{OPT}_0}\beta_K^{LK}\alpha_{2K}^{LK} + \overline{\text{OPT}_1}\beta_K^{LK}\alpha_K^{LK} + \overline{\text{OPT}_2}\beta_K^{LK}. \tag{6.29}$$

Finally after LK iterations the upper bound is

$$\begin{aligned} f^{LK} &\leq E\alpha_{LK}^{LK} + \beta_K^{LK} \sum_{u=0}^{L-1} \alpha_{(L-1-u)K}^{LK} \overline{\text{OPT}_u} \leq \\ &\leq E\frac{1}{e} + \sum_{u=0}^{L-1} \left(1 - \sqrt[L]{\frac{1}{e}}\right) \left(\sqrt[L]{\frac{1}{e}}\right)^{L-1-u} \overline{\text{OPT}_u}. \end{aligned} \tag{6.30}$$

The last inequality stems from the fact that $(\alpha_K^{LK})^L = \alpha_{LK}^{LK} \leq \frac{1}{e}$ and that $\alpha_K^{LK} + \beta_K^{LK} = 1$. □

Finally we derive the upper bound of the approximation ratio $\rho = f^{LK}/\text{OPT}$.

Theorem 6.4.3. *Upper bound of the approximation ratio is*

$$\rho \geq \frac{E}{\text{LB}(\text{OPT})}\frac{1}{e} + \sum_{u=0}^{L-1} \gamma_u \left(1 + \frac{\sum_{v=1}^{u} R_v}{\text{LB}(\text{OPT})}\right) \tag{6.31}$$

where $\text{LB}(\text{OPT})$ *is lower bound of the* OPT.

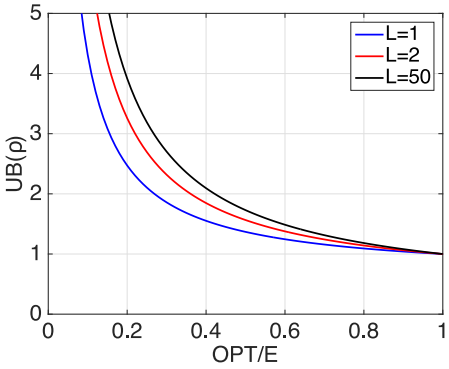


Figure 6.3. UB(ρ) as a function of $\frac{\text{OPT}}{E}$ ratios with $R_v \leq \frac{V}{L}$.

Proof.

$$\begin{aligned}\rho = \frac{f^{LK}}{\text{OPT}} \leq \frac{\text{UB}(f^{LK})}{\text{OPT}} = \frac{E\frac{1}{e} + \sum_{u=1}^{L} \gamma_u \overline{\text{OPT}}_u}{\text{OPT}} = \\ = \frac{\text{OPT}(\frac{E}{\text{OPT}}\frac{1}{e} + \sum_{u=1}^{L} \gamma_u \frac{\overline{\text{OPT}}_u}{\text{OPT}})}{\text{OPT}} = \\ = \frac{E}{\text{OPT}}\frac{1}{e} + \sum_{u=1}^{L} \gamma_u \frac{\text{OPT} + \sum_{v=1}^{u} R_v}{\text{OPT}} \leq \\ \leq \frac{E}{\text{LB}(\text{OPT})}\frac{1}{e} + \sum_{u=0}^{L-1} \gamma_u \left(1 + \frac{\sum_{v=1}^{u} R_v}{\text{LB}(\text{OPT})}\right)\end{aligned} \tag{6.32}$$

□

The approximation ratio depends on the OPT, if OPT $= 0$ then $\rho = \infty$, if OPT $= E$ then $\rho = 1$. If we make an assumption that each position covers only $\frac{1}{L}$ fraction of voxels, then $R_v \leq \frac{V}{L}$. Fig. 6.3 shows values of LB(ρ) for different ratios of $\frac{\text{OPT}}{E}$ for this case.

6.4.2. Prioritized Greedy Planning

In practice we observed a significant speed up of the greedy planning (Alg. 6.3) by imposing prioritized search for $\arg\min_j \mathbf{b}^\mathsf{T}\mathbf{p}_j$. Namely, let us denote Δ_j^k the decrease of the expected reconstruction error achieved by selecting ray j in iteration k,

$$\Delta_j^k = \sum_i (b_i^{k-1} - b_i^k) = \sum_i b_i^{k-1}(1 - p_{ij}), \tag{6.33}$$

and show that it is non-increasing. For $p_{ij}, p_{ij'} \in [0, 1]$ and $b_i^{k-1} \geq 0$ it follows that

$$b_i^{k-1}(1 - p_{ij}) \geq b_i^{k-1} p_{ij'}(1 - p_{ij}). \tag{6.34}$$

Summing the inequalities for all voxels i, we get

$$\Delta_j^k = \sum_i b_i^{k-1}(1 - p_{ij}) \geq \sum_i b_i^{k-1} p_{ij'}(1 - p_{ij}) = \Delta_j^{k+1} \tag{6.35}$$

for an arbitrary ray j' selected in iteration k. Note that $\Delta_j^k \geq \Delta_j^{k+a}$ for any $a \geq 1$.

Now, when we search for j maximizing Δ_j^k in decreasing order of $\Delta_j^{k-a_j}$, $a_j \geq 1\ \forall j$, we can stop once $\Delta_j^k > \Delta_{j'}^{k-a_{j'}}$ for the next ray j' because none of the remaining rays can be better than j. Moreover, we can take advantage of the fact that all the remaining rays including j remained sorted when updating the priority for the next iteration. The proposed planning is detailed in Alg. 6.4.

The number of re-evaluations of Δ_j in Alg. 6.4 was approximately 500× smaller than in Alg. 6.3. Despite the sorting took about a 1/10 of the computation time, the prioritized planning was about 30× faster and took 0.3s on average using a single-threaded implementation.

Algorithm 6.4. Prioritized greedy planning of depth measuring rays.

Require:
Set of rays $\mathcal{V} = \{1, \ldots, N\}$ at positions $\mathcal{L}$, budget K, voxel costs $\mathbf{b}$, probability vectors $\mathbf{p}_j\ \forall j \in \mathcal{V}$, mapping from ray to position $\lambda\colon \mathcal{V} \mapsto \mathcal{L}$

$\mathcal{J}_\ell \leftarrow \emptyset\ \forall \ell \in \mathcal{L}$ ▷ No rays selected
$\Delta_j \leftarrow \infty\ \forall j \in \mathcal{V}$ ▷ Force recompute.
$S \leftarrow (1, \ldots, N)$ ▷ Sequence of ray indices, $S(n)$ denotes the nth element in the sequence, $S(m{:}n)$ the subsequence from the mth to the nth element.
while $S \neq \emptyset$ **do**
  for $n \in (1, \ldots, |S|)$ **do**
    $\Delta_{S(n)} \leftarrow \mathbf{b}^\mathsf{T}(\mathbf{1} - \mathbf{p}_{S(n)})$
    if $n < |S| \wedge \Delta_{S(n)} \geq \Delta_{S(n+1)}$ **then**
      break
    end if
  end for
  Sort subsequence $S(1:n)$ s.t. $\Delta_{S(n')} \geq \Delta_{S(n'+1)}$
  Merge sorted subsequences $S(1:n-1)$ and $S(n:|S|)$
  $j^* \leftarrow S(1), l^* \leftarrow \lambda(j^*)$
  $\mathcal{J}_{\ell^*} \leftarrow \mathcal{J}_{\ell^*} \cup \{j^*\}$ ▷ Add the best ray.
  $\mathbf{b} \leftarrow \mathbf{b} \odot \mathbf{p}_{j^*}$ ▷ Update voxel costs.
  if $|\mathcal{J}_{\ell^*}| = K$ **then**
    $S \leftarrow S \setminus \{j : \lambda(j) = \ell^*\}$ ▷ Close the position.
  else
    $S \leftarrow S \setminus \{j^*\}$ ▷ Remove j^* from S.
  end if
end while
return Selected rays $\mathcal{J}_\ell$ at every position $\ell \in \mathcal{L}$

6.5. Experiments

6.5.1. Dataset

All experiments were conducted on selected sequences from categories *City* and *Residential* from the KITTI dataset [29]. We first brought the point clouds (captured by the Velodyne HDL-64E laser scanner) to a common reference frame using the localization data from the inertial navigation system (OXTS RT 3003 GPS/IMU) and created the ground-truth voxel maps from these. The voxels traced from the sensor origin towards

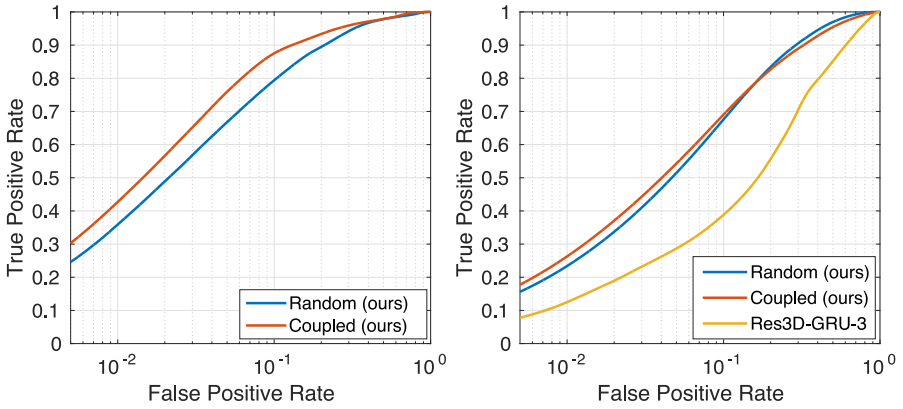


Figure 6.4. ROC curves of occupancy prediction from active 3D mapping on test sets. (left) *Random* denotes the global occupancy $\hat{\mathbf{Y}}$ obtained by using $\mathbf{h}_{\theta^0}$ with random sparse measurements, *Coupled* the occupancy obtained by using $\mathbf{h}_{\theta^3}$ with the prioritized greedy planning. The voxels which are more than 1m from what could possibly be measured are excluded, together with the false positives which can be attributed to discretization error (in 1-voxel distance from an occupied voxel). (right) *Random* denotes the local occupancy maps $\hat{\mathbf{y}}_l$ obtained by using $\mathbf{h}_{\theta^0}$, *Coupled* the maps obtained by using $\mathbf{h}_{\theta^1}$, and *Res3D-GRU-3* denotes the reconstruction obtained by the network adapted from [16].

each measured point were updated as empty (their occupancy confidence was decreased, $y_i \leftarrow y_i - 1$) except for the voxels incident with any of the end points which were updated as occupied (their occupancy confidence was increased by the same amount for each incident end point, $y_i \leftarrow y_i + 1$). The dynamic objects were mostly removed in the process since the voxels belonging to these objects were also many times updated as empty while moving. All maps used axis-aligned voxels of edge size 0.2 m.

For generating the sparse measurements, we consider an SSL sensor with the field of view of 120° horizontally and 90° vertically discretized in $160 \times 120 = 19200$ directions. At each position, we select $K = 200$ rays and ray-trace in these directions until an occupied voxel is hit or the maximum distance of 48m is reached. Only the rays which end up hitting an occupied voxel produce valid measurements, as is the case with the time-of-flight sensors. Local maps $\mathbf{x}_l$ and $\mathbf{y}_l$ contain volume of 64m × 64m × 6.4m discretized into $320 \times 320 \times 32$ voxels.

6.5.2. Active 3D Mapping

In this experiment, we used 17 and 3 sequences from the *Residential* category for training and validation, respectively, and 13 sequences from the *City* category for testing. We evaluate the iterative planning-learning procedure described in Sec. 6.3. For learning the mapping networks, we used learning rate $\alpha = 10^{-3}(1/8)^{\lceil i/10 \rceil}$ based on epoch number i, batch size 1, and momentum 0.99. Networks $\mathbf{h}_{\theta^0}, \ldots, \mathbf{h}_{\theta^3}$ were trained for 20 epochs. Validation performance stopped improving after 3 planning-learning iterations.

The ROC curves shown in Fig. 6.4 (left) are computed using ground-truth maps $\mathbf{Y}$ and predicted global occupancy maps $\hat{\mathbf{Y}}$. The performance of the $\mathbf{h}_{\theta^3}$ network (denoted *Coupled*) significantly outperforms the $\mathbf{h}_{\theta^3}$ network (*Random*), which shows the benefit of the proposed iterative planning-mapping procedure. Examples of reconstructed global occupancy maps are shown in Fig. 6.5. Note that the valid measurements covered around 3% of the input voxels.

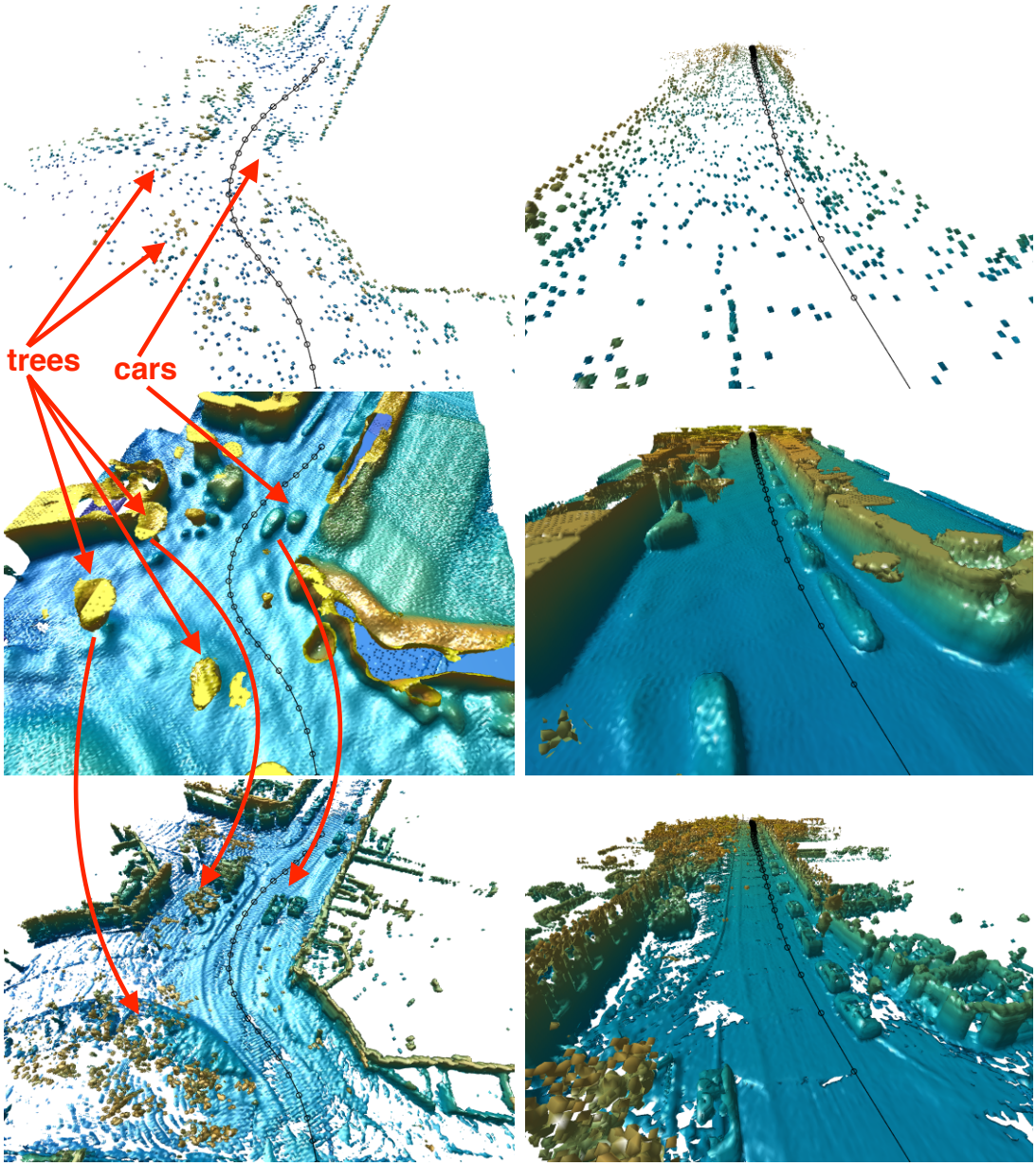


Figure 6.5. Examples of global map reconstruction. (top) Sparse measurement maps $\mathbf{X}$. (middle) Reconstructed occupancy maps $\hat{\mathbf{Y}}$ in form of isosurface. (bottom) Ground-truth maps $\mathbf{Y}$. The black line denotes trajectory of the car.

6.5.3. Comparison to a Recurrent Image-Based Architecture

We provide a comparison with the image-based reconstruction method of Choy *et al.* [16]. Namely, we modify the residual network with Gated Recurrent Units (GRU), *Res3D-GRU-3*, to use sparse depth maps of size 160×120 instead of RGB images. The sensor pose corresponding to the last received depth map was used for reconstruction. The number of views were fixed to 5, with $K = 200$ randomly selected depth-measuring rays in each image. For this experiment, we used 20 sequences from the *Residential* category—18 for training, 1 for validation and 1 for testing. Since the *Res3D-GRU-3* architecture is not suited for high-dimensional outputs due to its high memory requirements, we limit the batch size to 1 and the size of the maps to $128 \times 128 \times 32$, which corresponds to $16 \times 16 \times 4$ recurrent units. Our mapping network was trained and tested on voxel maps instead of depth images.

The corresponding ROC curves, computed from local maps $\mathbf{y}_l$ and $\hat{\mathbf{y}}_l$, are shown in Fig. 6.4 (right). Both $\mathbf{h}_{\theta^0}$ and $\mathbf{h}_{\theta^1}$ networks outperforms the *Res3D-GRU-3* network. We attribute this result mostly to the fact that our method is implicitly provided the known trajectory, while the *Res3D-GRU-3* network is not. Another reason may be the ray-voxel mapping which is also known implicitly in our case, compared to [16].

6.6. Conclusions

We have proposed a computationally tractable approach for the very high-dimensional active perception task. The proposed 3D-reconstruction CNN outperforms a state-of-the-art approach by 20% in recall, and it is shown that when learning is coupled with planning, recall increases by additional 8% on the same false positive rate. The proposed prioritized greedy planning algorithm seems to be a promising direction with respect to on-board reactive control since it is about $30\times$ faster and requires only $1/500$ of ray evaluations compared to a naïve greedy solution.

7. Conclusion and Future Work

We have contributed to several areas of 3D vision, namely, 3D object recognition and 3D point cloud registration based on matching local invariant features (chapters 3 and 4), and active vision where we combined learning with planning to create a control policy (chapters 5 and 6).

The method of local invariant features addresses the issue of non-uniform sampling density inherent in range-sensing methods. In 3D object recognition, it outperformed the competitors in the time of publication [68]. In 3D point cloud registration, the method is shown [69] to provide advantages over ICP-based registration in cases with at least moderate overlap ($\geq 75\%$) of the reading and reference point clouds. It also provides a superior performance compared to other state-of-the-art methods of global registration, with competitive running times.

Within the active vision area, we addressed the problem of simultaneous exploration and segmentation with incomplete data, for which we introduced a method of self-initialized policy learning (chapter 5). We have shown that a simplified version of the task can be solved as a mixed-integer linear program (MILP) to obtain the optimal sensor trajectory. This is, nevertheless, computationally very demanding and therefore not suited for online replanning after every new measurement. Instead, we proposed to use these optimal trajectories as a supervision for learning a reactive CNN-based policy. We have demonstrated two modes of policy initialization to limit the amount of labeled data from experiments with the real robot—the first uses the multimodal segmentation models with a set of labeled multimodal images, the second uses only a set of unlabeled multimodal images with the given segmentation models.

We also proposed a computationally tractable approach to active 3D mapping, which couples learning of the 3D reconstruction with the optimization of depth-measuring rays. Unlike other active vision tasks, this task has significantly higher dimensionality of the state-action space, which renders unsupervised reinforcement learning prohibitively expensive to use. To solve the planning subtask online, we proposed a fast prioritized greedy algorithm, for which we also derived an approximation ratio. Using the publicly available KITTI dataset, we have demonstrated that accuracy of the reconstruction improves when learning to reconstruct is coupled with planning new measurements.

Using an off-line planner to provide the training set for training a reactive control policy seems to be a very promising research direction which helps to avoid some problems connected with conventional reinforcement learning methods, such as the huge number of episodes the agent must observe before achieving desirable levels of performance. For the active 3D mapping task, an alternative approach would be to train a reactive policy instead of using the planner, together with the 3D mapping network. This is, nevertheless, challenging due to the problem of high dimensionality of the actions (there is around half a million rays to be planned each second) and the constraints which need to enforced (the limited budget).

A. Author's Publications

A.1. Publications Related to Thesis

A.1.1. Impacted Journal Articles

- Tomas Petricek and Tomas Svoboda. Point cloud registration from local feature correspondences—evaluation on challenging datasets. *PLOS ONE*, 2017. doi:10.1371/journal.pone.0187943.
  Authorship: 50%

A.1.2. Conference Papers Excerpted by ISI

- Tomas Petricek and Tomas Svoboda. Area-weighted surface normals for 3D object recognition. In *Pattern Recognition (ICPR), 2012 21th International Conference on*, pages 1492–1496, Nov. 2012.
  Authorship: 50%
- Karel Zimmermann, Tomas Petricek, Vojtech Salansky, and Tomas Svoboda. Learning for active 3D mapping. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.
  Authorship: 35%

A.1.3. Others

- Petricek, T. and Salansky, V. and Zimmermann, K. and Svoboda, T. Guided reinforcement learning for simultaneous exploration and segmentation. Manuscript submitted for publication (revised version under review).
  Authorship: 25%

A.2. Other Publications

A.2.1. Impacted Journal Articles

- G.J.M. Kruijff, I. Kruijff-Korbayová, S. Keshavdas, B. Larochelle, M. Janíček, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, M.A. Neerincx, R. Looije, N.J.J.M Smets, T. Mioch, J. van Diggelen, F. Pirri, M. Gianni, F. Ferri, M. Menna, R. Worst, T. Linder, V. Tretyakov, H. Surmann, T. Svoboda, M. Reinštein, K. Zimmermann, T. Petříček, and V. Hlaváč. Designing, developing, and deploying systems to support humanrobot teams in disaster response. *Advanced Robotics*, 28(23):1547–1570, 2014. doi:10.1080/01691864.2014.985335.
  Authorship: 3.7%

A.2.2. Conference Papers Excerpted by ISI

- K. Zimmermann, P. Zuzanek, M. Reinstein, T. Petricek, and V. Hlavac. Adaptive traversability of partially occluded obstacles. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3959–3964, May 2015. doi:10.1109/ICRA.2015.7139752.
  Authorship: 20%

A.2.3. Conference Papers Excerpted by Scopus

- Mario Gianni, Panagiotis Papadakis, Fiora Pirri, Ming Liu, Francois Pomerleau, Francis Colas, Karel Zimmermann, Tomas Svoboda, Tomas Petricek, Geert Kruijff, Harmish Khambhaita, and Hendrik Zender. A unified framework for planning and execution-monitoring of mobile robots. In Nilufer Onder Sanem Sariel-Talay, Stephen F. Smith, editor, *Automated Action Planning for Autonomous Mobile Robots: Papers from the AAAI Workshop (WS-11-09)*, pages 39–44, Menlo Park, USA, August 2011. AAAI Press.
  Authorship: 8%
- G.J.M. Kruijff, M. Janíček, S. Keshavdas, B. Larochelle, H. Zender, N.J.J.M. Smets, T. Mioch, M.A. Neerincx, J.V. Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlaváč, T. Svoboda, T. Petříček, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, S. Pratzler-Wanczura, and M. Sulk. Experience in system design for human-robot teaming in urban search and rescue. In Kazuya Yoshida and Satoshi Tadokoro, editors, *Field and Service Robotics*, volume 92 of *Springer Tracts in Advanced Robotics*, pages 111–125. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-40686-7_8.
  Authorship: 3.23%

A.3. Citations of Author's Publications

Self-citations of any author are excluded.

- Mario Gianni, Panagiotis Papadakis, Fiora Pirri, Ming Liu, Francois Pomerleau, Francis Colas, Karel Zimmermann, Tomas Svoboda, Tomas Petricek, Geert Kruijff, Harmish Khambhaita, and Hendrik Zender. A unified framework for planning and execution-monitoring of mobile robots. In Nilufer Onder Sanem Sariel-Talay, Stephen F. Smith, editor, *Automated Action Planning for Autonomous Mobile Robots: Papers from the AAAI Workshop (WS-11-09)*, pages 39–44, Menlo Park, USA, August 2011. AAAI Press.
  - E. Boukas, I. Kostavelis, A. Gasteratos, and G. C. Sirakoulis. Robot guided crowd evacuation. *IEEE Transactions on Automation Science and Engineering*, 12(2):739–751, April 2015.
  - F. Chen, L. Wang, J. Lu, F. Ren, Y. Wang, X. Zhang, and C. Xu. A smart cloud robotic system based on cloud computing services. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 316–321, Nov 2016.
  - I. Kostavelis, A. Gasteratos, E. Boukas, and L. Nalpantidis. Learning the terrain and planning a collision-free trajectory for indoor post-disaster environments. In *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6, Nov 2012.
  - Thomas Reinbacher and César Guzmán-Alvarez. *Template-Based Synthesis of Plan Execution Monitors*, pages 451–461. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
  - H. Zhang, W. Chen, J. Wang, and K. Li. Human interest oriented heterogeneous multi-robot exploration under connectivity constraints. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 976–981, Dec 2013.
- G.J.M. Kruijff, I. Kruijff-Korbayová, S. Keshavdas, B. Larochelle, M. Janíček, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, M.A. Neerincx, R. Looije, N.J.J.M Smets, T. Mioch, J. van Diggelen, F. Pirri, M. Gianni, F. Ferri, M. Menna, R. Worst, T. Linder, V. Tretyakov, H. Surmann, T. Svoboda,

M. Reinštein, K. Zimmermann, T. Petříček, and V. Hlaváč. Designing, developing, and deploying systems to support humanrobot teams in disaster response. *Advanced Robotics*, 28(23):1547–1570, 2014. doi:10.1080/01691864.2014.985335.

  - Karsten Berns, Atabak Nezhadfard, Massimo Tosa, Haris Balta, and Geert De Cubber. Unmanned ground robots for rescue tasks. In *Search and Rescue Robotics - From Theory to Practice*, chapter 04. InTech, Rijeka, 2017.
  - Geert De Cubber, Daniela Doroftei, Konrad Rudin, Karsten Berns, Anibal Matos, Daniel Serrano, Jose Sanchez, Shashank Govindaraj, Janusz Bedkowski, Rui Roda, Eduardo Silva, and Stephane Ourevitch. Introduction to the use of robotic tools for search and rescue. In *Search and Rescue Robotics - From Theory to Practice*, chapter 01. InTech, Rijeka, 2017.
  - Shashank Govindaraj, Pierre Letier, Keshav Chintamani, Jeremi Gancet, Mario Nunez Jimenez, Miguel ngel Esbr, Pawel Musialik, Janusz Bedkowski, Irune Badiola, Ricardo Gonalves, Antnio Coelho, Daniel Serrano, Massimo Tosa, Thomas Pfister, and Jose Manuel Sanchez. Command and control systems for search and rescue robots. In *Search and Rescue Robotics - From Theory to Practice*, chapter 08. InTech, Rijeka, 2017.
  - M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke. Autonomous mav navigation in complex gnss-denied 3D environments. In *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–7, Oct 2015.
  - Juntong Qi, Dalei Song, Hong Shang, Nianfa Wang, Chunsheng Hua, Chong Wu, Xin Qi, and Jianda Han. Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake. *Journal of Field Robotics*, 33(3):290–321, 2016.
  - Carmine Tommaso Recchiuto and Antonio Sgorbissa. Post-disaster assessment with unmanned aerial vehicles: A survey on practical implementations and research approaches. *Journal of Field Robotics*, pages n/a–n/a.
  - Carmine Tommaso Recchiuto and Antonio Sgorbissa. *The Project PRISMA: Post-Disaster Assessment with UAVs*, pages 199–211. Springer International Publishing, Cham, 2017.

- G.J.M. Kruijff, M. Janíček, S. Keshavdas, B. Larochelle, H. Zender, N.J.J.M. Smets, T. Mioch, M.A. Neerincx, J.V. Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlaváč, T. Svoboda, T. Petříček, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, S. Pratzler-Wanczura, and M. Sulk. Experience in system design for human-robot teaming in urban search and rescue. In Kazuya Yoshida and Satoshi Tadokoro, editors, *Field and Service Robotics*, volume 92 of *Springer Tracts in Advanced Robotics*, pages 111–125. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-40686-7_8.
  - H. Al-Tair, J. Dias, and M. Al-Qutayri. Towards collaborative support system for teamwork between robots and human in hazard incidents. In *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 96–97, Dec 2013.
  - Ahmad Baranzadeh and Andrey V. Savkin. A distributed control algorithm for area search by a multi-robot team. *Robotica*, 35(6):14521472, 2017.
  - Michael Brunner, Torsten Fiolka, Dirk Schulz, and Christopher M. Schlick. Design and comparative evaluation of an iterative contact point estimation method for static stability estimation of mobile actively reconfigurable robots. *Robotics and Autonomous Systems*, 63(Part 1):89 – 107, 2015.
  - Micael S. Couceiro, David Portugal, and Rui P. Rocha. A collective robotic architecture in search and rescue scenarios. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 64–69, New York, NY, USA, 2013. ACM.
  - Markus Fröhle, Themistoklis Charalambous, Henk Wymeersch, Siwei Zhang, and Armin Dammann. *Formation Control of Multi-Agent Systems with Location Uncertainty*, pages 197–215. Springer International Publishing, Cham, 2017.
  - Shoutao Li, Lina Li, Gordon Lee, and Hao Zhang. A hybrid search algorithm for swarm robots searching in an unknown environment. *PLOS ONE*, 9(11):1–10, 11 2014.
  - Michael Mortimer, Ben Horan, and Mehdi Seyedmahmoudian. Building a relationship between robot characteristics and teleoperation user interfaces. *Sensors*, 17(3), 2017.
  - Huang Peng, Guangming Song, Jian You, Ying Zhang, and Jie Lian. An indoor navigation service robot system based on vibration tactile feedback. *International Journal of Social Robotics*, 9(3):331–341, Jun 2017.
  - L. Pfotzer, S. Ruehl, G. Heppner, A. Roennau, and R. Dillmann. Kairo 3: A modular reconfigurable robot for search and rescue field missions. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pages 205–210, Dec 2014.
  - R. P. Rocha, D. Portugal, M. Couceiro, F. Arajo, P. Menezes, and J. Lobo. The chopin project: Cooperation between human and robotic teams in catastrophic incidents. In *2013*

IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pages 1–4, Oct 2013.

- Gabriel Rodrigues de Campos, Dimos V. Dimarogonas, Alexandre Seuret, and Karl H. Johansson. Distributed control of compact formations for multi-robot swarms. *IMA Journal of Mathematical Control and Information*, page dnw073, 2017.
- B. Soni and A. Sowmya. Victim detection and localisation in an urban disaster site. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2142–2147, Dec 2013.
- Y. Tan. *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*. Advances in Computational Intelligence and Robotics. IGI Global, 2015.
- Fereshta Yazdani, Benjamin Brieber, and Michael Beetz. *Cognition-Enabled Robot Control for Mixed Human-Robot Rescue Teams*, pages 1357–1369. Springer International Publishing, Cham, 2016.
- H. Zhang, W. Chen, J. Wang, and K. Li. Human interest oriented heterogeneous multi-robot exploration under connectivity constraints. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 976–981, Dec 2013.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, GA, 2016. USENIX Association. 52

[2] Evan Ackerman. Quanergy announces $250 solid-state LIDAR for cars, robots, and more. *IEEE Spectrum*, 7 January 2016. 2, 54

[3] Evan Ackerman. Israeli startup innoviz promises $100 solid-state automotive lidar by 2018. *IEEE Spectrum*, 9 August 2017. 2, 54

[4] Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *ACM Trans. Graph.*, 27(3):85:1–85:10, August 2008. 19, 21, 27, 28, 29, 30, 32

[5] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, Jan 1988. 1

[6] A. Andreopoulos, S. Hasler, H. Wersing, H. Janssen, J. K. Tsotsos, and E. Korner. Active 3d object localization using a humanoid robot. *IEEE Transactions on Robotics*, 27(1):47–64, Feb 2011. 35, 57

[7] K.S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):698–700, 1987. 5

[8] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, Feb 2017. 35

[9] P. Bariya and K. Nishino. Scale-hierarchical 3d object recognition in cluttered scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1657 –1664, june 2010. 7, 11, 12, 14, 15, 16, 17

[10] Prabin Bariya, John Novatnack, Gabriel Schwartz, and Ko Nishino. 3d geometric scale variability in range images: Features and descriptors. *International Journal of Computer Vision*, 99:232–255, 2012. 10.1007/s11263-012-0526-7. 7

[11] Paul J. Besl and Ramesh C. Jain. Invariant surface characteristics for 3d object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33(1):33–80, 1986. 1

[12] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992. 1, 5, 19, 21, 28, 29, 30

[13] Léon Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer, Berlin, Heidelberg, 2012. 9

[14] R. Bro, E. Acar, and Tamara G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008. 13, 21, 23, 27, 32

[15] Yang Chen and Grard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992. 1, 5, 19, 21, 28, 29, 30

[16] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 628–644, Cham, 2016. Springer International Publishing. 54, 56, 68, 70

[17] Chin Seng Chua and Ray Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25:63–85, 1997. 10.1023/A:1007981719186. 7

[18] O. Chum and J. Matas. Randomized ransac with t(d,d) test. In *Proceedings of the British Machine Vision Conference*, pages 43.1–43.10. BMVA Press, 2002. doi:10.5244/C.16.43. 6

[19] Ondej Chum, Ji Matas, and Josef Kittler. Locally optimized ransac. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 236–243. Springer Berlin Heidelberg, 2003. 24

[20] Arun Das, James Servos, and Steven Lake Waslander. 3D scan registration using the normal distributions transform with ground segmentation and point cloud clustering. In *2013 IEEE International Conference on Robotics and Automation*, May 2013. 28

[21] C. Dorai and A.K. Jain. Cosmos-a representation scheme for 3d free-form objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(10):1115–1130, oct 1997. 7, 11

[22] Andreas Doumanoglou, Tae-Kyun Kim, Xiaowei Zhao, and Sotiris Malassiotis. Active random forests: An application to autonomous unfolding of clothes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 644–658. Springer International Publishing, 2014. 35

[23] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 998–1005, 2010. 11, 12, 15, 16, 17, 18

[24] D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, 1997. 5

[25] XiaoGuang Feng and P. Milanfar. Multiscale principal components analysis for image local orientation estimation. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 1, pages 478 – 482 vol.1, nov. 2002. 13

[26] M. Firman, O. M. Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5431–5440, June 2016. 56

[27] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. xiii, 1, 5, 6, 21

[28] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(1314):1145–1153, 2003. ¡ce:title¿British Machine Vision Computing 2001¡/ce:title¿. 19

[29] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. 54, 67

[30] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010. 49, 52

[31] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016. 1

[32] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2094–2100. AAAI Press, 2016. 51

[33] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. 8

[34] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987. 5

[35] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127–1135, Jul 1988. 5

[36] Aleš Hrabalík. 3d point cloud registration, experimental comparison and fusing range and visual data. Technical report, Czech Technical University in Prague, 2014. 19, 28

[37] Dinesh Jayaraman and Kristen Grauman. Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 489–505. Springer International Publishing, Cham, 2016. 1, 36, 54, 57

[38] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM. 49

[39] Zhaoyin Jia, Yao-Jen Chang, and Tsuhan Chen. A general boosting-based framework for active object recognition. In *Proceedings of the British Machine Vision Conference*, pages 46.1–46.11. BMVA Press, 2010. doi:10.5244/C.24.46. 35

[40] Edward Johns, Stefan Leutenegger, and Andrew J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, 2016. 1, 36

[41] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999. 1, 7, 11, 16, 17, 23

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 33

[43] G.J.M. Kruijff, M. Janíček, S. Keshavdas, B. Larochelle, H. Zender, N.J.J.M. Smets, T. Mioch, M.A. Neerincx, J.V. Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlaváč, T. Svoboda, T. Petříček, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, S. Pratzler-Wanczura, and M. Sulk. Experience in system design for human-robot teaming in urban search and rescue. In Kazuya Yoshida and Satoshi Tadokoro, editors, *Field and Service Robotics*, volume 92 of *Springer Tracts in Advanced Robotics*, pages 111–125. Springer Berlin Heidelberg, 2014. 1

[44] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 9

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 9

[46] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 8, 9

[47] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016. 1, 33, 36

[48] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *CoRR*, abs/1603.02199, 2016. 33

[49] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 9, 33, 44

[50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 10.1023/B:VISI.0000029664.99615.94. 15

[51] Martin Magnusson. *The Three-Dimensional Normal-Distributions Transform—an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, rebro University, 2009. 19, 21, 27, 28, 29, 30, 32

[52] Martin Magnusson, Narunas Vaskevicius, Todor Stoyanov, Kaustubh Pathak, and Andreas Birk. Beyond points: Evaluating recent 3d scan-matching algorithms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3637, 2015. 19, 28

[53] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89:348–361, 2010. 7, 11, 12, 15, 17, 19

[54] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601, 2006. 7, 11, 12, 15, 17

[55] A. S. Mian, M. Bennamoun, and R. A. Owens. Automatic correspondence for 3D modeling: An extensive review. *International Journal of Shape Modeling*, 11(02):253–291, 2005. 19

[56] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1):43–72, 2005. 25

[57] A. K. Mishra, Y. Aloimonos, L. F. Cheong, and A. Kassim. Active visual segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):639–653, April 2012. 35, 54

[58] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 33, 35, 36, 47, 51

[59] Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103:1–103:12, July 2015. 56

[60] U. A. Muller, A. Gunzinger, and W. Guggenbuhl. Fast neural net simulation with a dsp processor array. *IEEE Transactions on Neural Networks*, 6(1):203–213, Jan 1995. 9

[61] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. Napsac: High noise, high dimensional robust estimation - it's in the bag. In *Proceedings of the British Machine Vision Conference*, pages 44.1–44.10. BMVA Press, 2002. doi:10.5244/C.16.44. 6

[62] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o($1/k^2$). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983. 9, 44

[63] D. T. Nguyen, B. S. Hua, M. K. Tran, Q. H. Pham, and S. K. Yeung. A field model for repairing 3d shapes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5676–5684, June 2016. 56

[64] J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007. 7, 12

[65] John Novatnack and Ko Nishino. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision ECCV 2008*, volume 5304 of *Lecture Notes in Computer Science*, pages 440–453. Springer Berlin / Heidelberg, 2008. 7

[66] Cristina Palmero, Albert Clapés, Chris Bahnsen, Andreas Møgelmose, Thomas B. Moeslund, and Sergio Escalera. Multi-modal rgb–depth–thermal human body segmentation. *International Journal of Computer Vision*, pages 1–23, 2016. 36

[67] A. Petrelli and L. Di Stefano. On the repeatability of the local reference frame for partial shape matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2244–2251, nov. 2011. 1, 26

[68] Tomas Petricek and Tomas Svoboda. Area-weighted surface normals for 3D object recognition. In *Pattern Recognition (ICPR), 2012 21th International Conference on*, pages 1492–1496, Nov. 2012. 3, 19, 22, 23, 25, 30, 71

[69] Tomas Petricek and Tomas Svoboda. Point cloud registration from local feature correspondences—evaluation on challenging datasets. *PLOS ONE*, 2017. doi:10.1371/journal.pone.0187943. 3, 71

[70] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. 9, 44

[71] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015. 5

[72] Franois Pomerleau, Francis Colas, Roland Siegwart, and Stphane Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013. 19, 21, 24, 25, 27, 28, 29, 30, 32, 47

[73] Franois Pomerleau, Ming Liu, Francis Colas, and Roland Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, 2012. 19, 24, 32

[74] C. R. Qi, H. Su, M. Niener, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, June 2016. 56

[75] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2493, June 2015. 56

[76] Philip E. Ross. Velodyne announces a solid-state lidar. *IEEE Spectrum*, 19 April 2017. 2, 54

[77] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, Oct. 1986. 8, 9

[78] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, May 2011. 22, 28, 30

[79] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. 7, 11, 19, 21, 23, 27, 28, 29, 30, 32

[80] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *ICLR*, volume abs/1511.05952, 2016. 51

[81] A. V. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: Science and Systems V*, Seattle, USA, June 2009. 5, 19, 21, 27, 28, 29, 30, 32

[82] Heiko Seifa and Xiaolong Hub. Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. *Autonomous Robots*, 2(2):159–162, 2016. 54

[83] G.C. Sharp, S.W. Lee, and D.K. Wehe. Icp registration using invariant features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):90–102, Jan 2002. 5

[84] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Trans. Graph.*, 31(6):180:1–180:11, November 2012. 56

[85] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600, 1994. 22, 32

[86] Ksenia Shubina and John K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010. Special issue on Intelligent Vision Systems. 35, 47

[87] J. Simanek, M. Reinstein, and V. Kubelka. Evaluation of the ekf-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Transactions on Mechatronics*, 20(2):985–990, April 2015. 47

[88] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 44

[89] T. Stoyanov, Martin Magnusson, and A.J. Lilienthal. Point set registration through minimization of the l2 distance between 3d-ndt models. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5196–5201, 2012. 28

[90] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, Dec 2015. 56

[91] Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Trans. Graph.*, 34(6):175:1–175:11, October 2015. 56

[92] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1139–1147. JMLR Workshop and Conference Proceedings, May 2013. 9, 44

[93] PW Theiler, JD Wegner, and K Schindler. Markerless point cloud registration with keypoint-based 4-points congruent sets. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):283–288, 2013. 19, 21, 27, 28, 29, 30, 32

[94] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6313 of *Lecture Notes in Computer Science*, pages 356–369. Springer Berlin / Heidelberg, 2010. 7, 11, 12, 21, 22, 23, 25, 26, 27, 32

[95] Federico Tombari, Samuele Salti, and Luigi DiStefano. Performance evaluation of 3D keypoint detectors. *International Journal of Computer Vision*, 102:198–220, 2013. 1, 2, 7, 22, 25

[96] P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000. 6

[97] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 689–692, New York, NY, USA, 2015. ACM. 60

[98] Michael W. Walker, Lejun Shao, and Richard A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, 1991. 5, 24

[99] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, June 2015. 54, 56

[100] Yiming Ye and John K. Tsotsos. Sensor planning for 3d object search. *Comput. Vis. Image Underst.*, 73(2):145–168, February 1999. 35

[101] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 373 –380, june 2009. 7, 11, 12, 13, 14

[102] Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Keypoints and local descriptors of scalar functions on 2d manifolds. *International Journal of Computer Vision*, 100:78–98, 2012. 10.1007/s11263-012-0528-5. 1, 7, 11, 12

[103] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S. C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3127–3134, June 2013. 56

[104] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696, 27 2009-oct. 4 2009. 22, 23

[105] Karel Zimmermann, Tomas Petricek, Vojtech Salansky, and Tomas Svoboda. Learning for active 3D mapping. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2017. 2, 3