



DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

3D MAPPING FROM SPARSE LIDAR DATA

3D MAPOVÁNÍ S VYUŽITÍM ŘÍDKÝCH DAT SENZORU LIDAR

PH.D. THESIS
DISERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE

ING. MARTIN VEĽAS

SUPERVISOR
VEDOUĆÍ PRÁCE

PROF. ING. ADAM HEROUT, PH.D.

BRNO 2020

I'd like to dedicate this thesis to my wife Petra
who encouraged me during the times of setbacks.

ABSTRACT

This work deals with the proposal of novel algorithms for sparse 3D LiDAR data processing, including the design of a whole mobile backpack mapping solution. This research was driven by the need for such solutions in the field of geodesy, mobile surveying, and the building construction.

Firstly, there is a proposal of the iterative algorithm for reliable point cloud registration and odometry estimation from 3D LiDAR point clouds. The sparsity and the size of these data are overcome using random sampling by Collar Line Segments (CLS). The evaluation, using standard KITTI dataset, showed superior accuracy over the well known General ICP algorithm.

Convolutional neural networks play an important role in the second method of odometry estimation, which processes encoded LiDAR data in form of 2D matrices. The method is able to run online, while the accuracy is preserved when only translation motion parameters are required. This can be handy when the online preview of mapping is required and the rotation parameters can be reliably provided by e.g. IMU sensor.

Based on the CLS algorithm, mobile backpack mapping solution 4RECON was designed and implemented. Using the calibrated and synchronized pair of Velodyne LiDARS and the deployment of dual antenna GNSS/INS solution, the universal system, providing accurate 3D modeling of both small indoor and large open environments, was developed. Our evaluation proved that the requirements set for this system were fulfilled – relative accuracy up to 5 cm and the average error of georeferencing under 12 cm.

The last pages contain the description and the evaluation of another method based on the convolutional neural networks – designed for ground segmentation of 3D LiDAR point clouds. This method outperformed the current state-of-the-art in this task and represents the way semantics can be introduced into the 3D laser data.

KEYWORDS

Point cloud registration, odometry estimation, backpack laser mapping; Velodyne LiDAR; point cloud; GNSS; IMU; sensor calibration; ground segmentation.

ABSTRAKT

Tato práce se zabývá návrhem nových algoritmů pro zpracování řídkých 3D dat sensorů LiDAR, včetně kompletního návrhu batohového mobilního mapovacího řešení. Tento výzkum byl motivován potřebou takových řešení v oblasti geodézie, mobilního průzkumu a výstavby.

Nejprve je prezentován iterační algoritmus pro spolehlivou registraci mračen bodů a odhad odometrie z měření 3D LiDARu. Problém řídkosti a velikosti těchto dat je řešen pomocí náhodného vzorkování pomocí Collar Line Segments (CLS). Vyhodnocení na standardní datové sadě KITTI ukázalo vynikající přesnost oproti známému algoritmu General ICP.

Konvoluční neuronové sítě hrají důležitou roli ve druhé metodě odhadu odometrie, která zpracovává kódovaná data LiDARu do 2D matic. Metoda je schopna online výkonu, zatímco je zachována přesnost, když požadujeme pouze parametry posunu. To může být užitečné v situacích, kdy je vyžadován online náhled mapování a parametry rotace mohou být spolehlivě poskytnuty např. senzorem IMU.

Na základě algoritmu CLS bylo navrženo a implementováno batohové mobilní mapovací řešení 4RECON. S využitím kalibrovaného a synchronizovaného páru LiDARů Velodyne a s nasazením řešení GNSS/INS s duální anténou, byl vyvinut univerzální systém poskytující přesné 3D modelování malých vnitřních i velkých otevřených prostředí. Naše hodnocení prokázalo, že požadavky stanovené pro tento systém byly splněny – relativní přesnost do 5 cm a průměrná chyba georeferencí pod 12 cm.

Poslední stránky obsahují popis a vyhodnocení další metody založené na konvolučních neuronových sítích – navržených pro segmentaci země v mračcích bodů 3D LiDARu. Tato metoda překonala současný stav techniky v této oblasti a představuje způsob, jakým může být sémantická informace vložena do 3D laserových dat.

KLÍČOVÁ SLOVA

Registrace mračen bodů; odhad odometrie; batohové laserové mapování; Velodyne LiDAR; mračno bodů; GNSS; IMU; kalibrace sensorů; segmentace země.

BIBLIOGRAPHIC CITATION

Ing. Martin Veřas: *3D Mapping from Sparse LiDAR Data*, doctoral thesis Brno, Brno University of Technology, Faculty of Information Technology, 2020.

DECLARATION

I declare that this dissertation thesis is my original work and that I have written it under the guidance of prof. Ing. Adam Herout, Ph.D. and Ing. Michal Španěl, Ph.D.. All sources and literature that I have used during my work on the thesis are correctly cited with complete reference to the respective sources.

Brno, 2020

Ing. Martin Veřas, January 7, 2020

ACKNOWLEDGMENTS

Many thanks to my supervisor Adam Herout and my consultant Michal Španěl for the patience, support, inspiring ideas, and the guidance during my whole study. I would like to thank Jiří Habrovec from Geodrom company for cooperation in the development of our mapping backpack, and my colleagues Zdeněk Materna, Michal Kapinus, Lukáš Polok, Marek Šolony, and Michal Hradiš for valuable advice and help. For English corrections, my thanks belong to my dear friend Zuzana Tulejová.

Personally, my gratitude belongs mostly to my parents who taught me to respect wisdom and education. My dearest thanks belong to my supporting wife Petra, and to God for pretty much everything.

The template for this thesis was derived from classicthesis and generously provided by Lukáš Polok.

CONTENTS

1	INTRODUCTION	1
i	MOTIVATION AND EXISTING SOLUTIONS	3
2	MOTIVATION AND APPLICATIONS OF THE 3D LASER MAPPING	5
3	SENSORS DESCRIPTION	9
3.1	Light Detection and Ranging	9
3.2	Positioning subsystem	14
4	EXISTING SOLUTIONS FOR MOBILE LASER MAPPING	23
4.1	Handheld ZEB-1 and ZEB Revo solutions	24
4.2	Terrestrial solutions (FARO Focus)	31
4.3	Backpack laser mapping solutions	34
4.4	Required algorithms as a part of mobile laser scanning system . . .	37
ii	CORE ALGORITHMS FOR ODOMETRY ESTIMATION	39
5	CLS FOR FAST ODOMETRY ESTIMATION FROM VELODYNE POINT CLOUDS	41
5.1	Abstract	41
5.2	Introduction	41
5.3	Related work	43
5.4	Velodyne point cloud registration	45
5.5	Experimental evaluation	50
5.6	Conclusion	54
6	CNN FOR IMU ASSISTED ODOMETRY ESTIMATION USING VELODYNE LIDAR	55
6.1	Abstract	55
6.2	Introduction	55
6.3	Related Work	56
6.4	Method	58
6.5	Experiments	65
6.6	Conclusion	69
iii	BACKPACK MOBILE MAPPING SOLUTION	71
7	BACKPACK MAPPING WITH CALIBRATED PAIR OF VELODYNE LIDARS	73
7.1	Abstract	73
7.2	Introduction	74
7.3	Related Work	76

7.4	Design of the Laser Mapping Backpack	85
7.5	Experiments	102
7.6	Discussion	113
7.7	Conclusions	114
iv	SEMANTIC GROUND SEGMENTATION	115
8	CNN FOR VERY FAST GROUND SEGMENTATION IN VELODYNE LIDAR	
	DATA	117
8.1	Abstract	117
8.2	Introduction	117
8.3	Related work	119
8.4	Proposed Ground Segmentation Method	122
8.5	Experiments	126
8.6	Conclusion	130
v	SUMMARY	131
9	CONCLUSION	133
9.1	Future work	134
	BIBLIOGRAPHY	135

INTRODUCTION

The laser mobile mapping gets a lot of attraction in recent years as a source of 3D information and geometrical description of buildings, construction sites, natural scenes, etc. It provides not only richer type of information describing the environment in form of full and accurate 3D models, but – most importantly – significantly increases productivity compared to traditional mapping and cartographical techniques in the field of geodesy.

This doctoral thesis and its contribution can be divided into four parts. Firstly, the overview, basic principles and the state-of-the-art in the field of laser mobile mapping will be provided. Besides the list and overall evaluation of existing solutions, and the description of sensors themselves, multiple application domains as the main motivation of this work will be presented.

In the second part, two different methods for odometry estimation and 3D LiDAR data registration are provided. Collar Line Segments (CLS) algorithm represents innovative version of more traditional iterative approaches where the large and sparse LiDAR data are sampled by the line segments and the registration is gradually refined. Direct method is represented by convolutional neural network (CNN) designed for fast estimation of registration parameters from point cloud encoded into 2D depth image. The evaluation of both algorithms showed superior qualities compared to the state-of-the-art methods in terms of precision, online performance, and robustness.

The third and probably the most interesting part presents the development our mobile backpack solution, where the CLS algorithm is deployed as the core element for the point cloud registration. The most significant feature of our backpack is the combination of data from two synchronized and calibrated Velodyne LiDAR scanners, and information from the GNSS/INS subsystem. Thanks to this property, our approach achieved universality for both small indoor and large open outdoor environment, while achieving sufficient accuracy (in order of centimetres), comparable with other, more specialized, solutions.

The last but not least part presents an automatic ground segmentation of 3D LiDAR point clouds using convolution neural networks. This method is an example of the way in which the semantic information can be automatically incorporated into the 3D models. Our segmentation method significantly improved time performance compared to the state-of-the-art, while also providing more stable and accurate results. Moreover, together with previously mentioned algorithm for

odometry estimation using CNNs, the method demonstrated broader usability of convolution networks for LiDAR data processing.

This thesis is written as a compilation of author's previously published papers [98, 99, 100, 96] in years 2016-2019.

Part I

MOTIVATION AND EXISTING SOLUTIONS

MOTIVATION AND APPLICATIONS OF THE 3D LASER MAPPING

The process of laser scanning is successfully deployed in many applications for automatic or semi-automatic 3D reconstruction of both indoor and outdoor environment. Resulting 3D models in the form of point clouds are further used as an input for generation of 3D CAD models, for the measurements and the visualizations, generation of building documentation, the estimation of the amount of material available or required, etc. Information and figures 2.1-2.6 presented in this chapter regarding applications and requirements on mobile mapping system was provided as a courtesy of Geodrom¹ company. This firm provides services of mobile laser mapping in geodetic application since 2012, having significant experiences in this area which were shared with us.

Some of the outcomes of implemented projects based on the mobile laser scanning will be demonstrated in this section. The input data in form of point clouds are created by existing commercial solutions. Fig. 2.1 shows the project, where large seven-floor hospital building was scanned with a mobile LiDAR system. Then, this 3D point cloud was semi-automatically transferred into the CAD model by measuring both indoor and outdoor construction elements. The output was used for visualization and many other tasks of Building Information Modeling (BIM), which captures physical and functional characteristics for facility management, energetic optimization, and the digital archiving of the building. This 3D documentation, capturing the actual state of the building, is an input for planning and projecting the reconstructions of the building by the architects.

Moreover, if the fast scanning process of 3D mapping is available, the models can be built continuously during the construction. If the 3D building designs are also a part of the project documentation, early detection of the differences and errors in construction is possible.

In Fig. 2.2, the mobile laser scanning was used to acquire 3D data of the single flat. The data (again in a form of a point cloud) were used to perform precise measurements and generate the footprint, including all relevant building elements and their dimensions. The model was used for documentation and projecting the reconstruction of the apartment. Moreover, a 3D graphical model for visualization was produced from point cloud data.

¹ <http://www.geodrom.cz/>

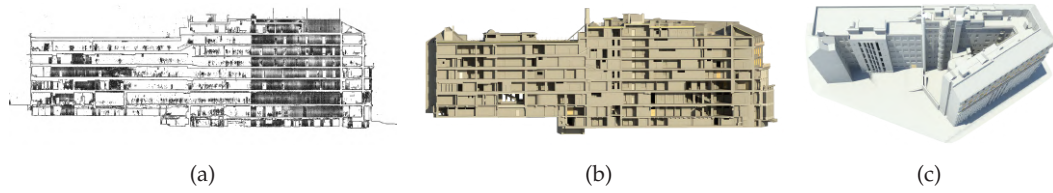


Figure 2.1: Point cloud acquired by the laser scanning (a), the slice of the 3D CAD model (b) and the 3D visualisation (c).



Figure 2.2: Top view of the laser point cloud (a), the footprint of the flat generated from this scan (b) and the 3D models for visualization (c), (d).

The mobile laser mapping provided fast documentation and estimation of the cubature of given bulk material, stored in the mining site as shown in Fig. 2.3a. Estimation of the cubature (amount) of the material is performed continuously – periodically, or after each time the material is shipped in or out – to check, if the total amount of material matches the documented incomes and outcomes.

Laser scanning was also used for planning and creation of the embankment for shooting gallery – the “hill” of dirt in Fig. 2.3b as a protection behind the targets. After a 3D model of existing terrain was generated from the laser data, the amount of dirt, which needs to be shipped-in, was estimated.

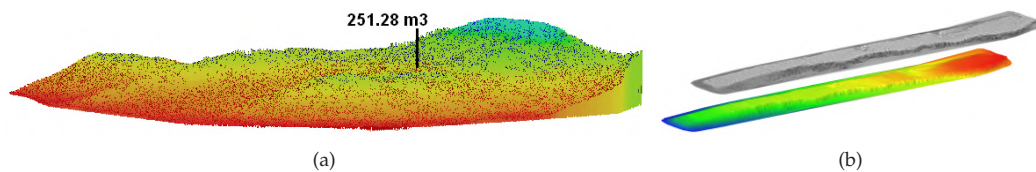


Figure 2.3: The scan of the heap of the bulk material (a) and the estimation of total cubature (volume, amount, . . .) of the material (b)

Laser scanning was also successfully used for the terrain mapping of the outdoor (e.g. forest) environment. This project was realized for the city council as a precise documentation of inaccessible terrain for a forest revitalization.

From the scans in Fig. 2.4a, the Digital Terrain Model (DTM, Fig. 2.4b) was built by ground segmentation and removal of redundant objects (trees, bushes, cabins,

etc.). From this model, a precise local contour map and a terrain profile in Fig. 2.4c was automatically generated.

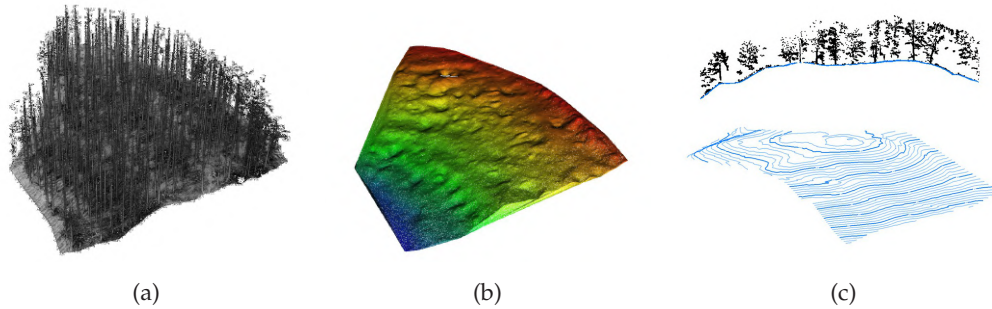


Figure 2.4: The laser scan of the forest environment (a), the terrain model generated by the ground detection in this scan (b) and the automatically generated 3D contour map (c).

Estimation of DTM was also an essential part in the task of documentation of the forest vegetation – especially the amount of timber available for logging. When the ground terrain was classified in a point cloud model (Fig. 2.5a), a slice within a certain height (usually 1.3 m) above the ground was extracted and the tree trunks were detected as shown in Fig. 2.5b. Moreover, not only the number of trees in the area could be counted but also a diameter of each individual tree was estimated. From these data, the biomass and the amount of timber can be monitored. The data provide a reference for verification of total amount of timber after logging and potential detection of theft. Moreover, additional data segmentation based on customer requirements is available.

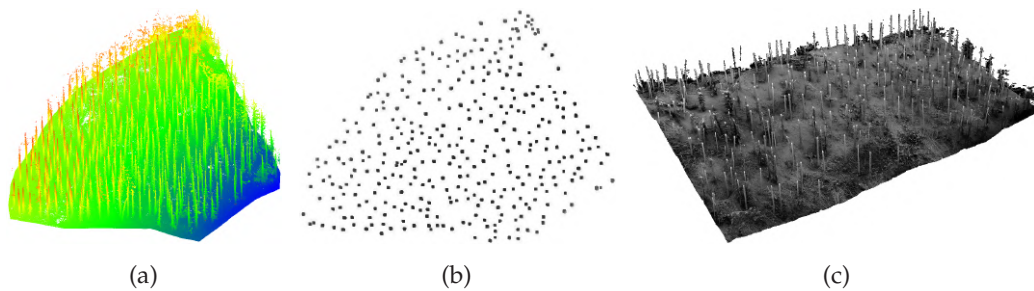


Figure 2.5: The 3D scan of the forest obtained by the LiDAR sensor (a), the slice (b) of the scan within certain height threshold over the terrain for the estimation of timber mass. The slice from the terrain model from ground up to this threshold is displayed in (c).

For many challenging objects, the task of documentation and blueprints generation, representing the actual state of a property, would be infeasible without a measurement of 3D data directly using mobile laser scanning. The complex structure of the historical roof construction, supporting and other building elements (e.g. chimneys) were successfully captured by this methodology in a form of the

point cloud in Fig. 2.6a. This model was created as a part of documentation for the planning of reconstructions. It also captures the actual state of the building for digitization and preserving the cultural heritage.

The other example of such a challenging task was a 3D modelling of a walking bridge in natural environment as shown in Fig. 2.6b. The actual state of the bridge structure can be used as a supporting material for the reconstruction project and the development of new construction elements.

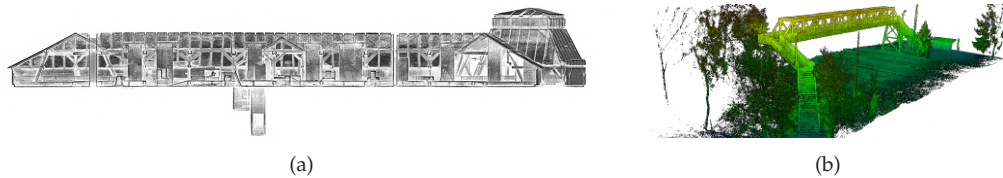


Figure 2.6: The scans of the objects with challenging and complex shapes – the roof structure (a) and the walking bridge (b).

In all projects of documentation, monitoring, planning etc., described above, the essential inputs are the models in form of point clouds. All original models were obtained by laser scanning as the most common source of precise 3D data nowadays.

Based on the applications mentioned above, requirements on a mobile mapping system can be summarized as follows:

- quick data acquisition (the hundreds of square meters per day),
- high mobility and adaptation for a variety of environments (complex or enclosed areas),
- indoor mapping where the GPS signal is not available,
- accuracy in order of centimeters, and
- automation of data processing.

These requirements on mapping system naturally reflects requirements on services demanded by the marked within feasible cost and delivery time. For post processing of resulting point cloud models, multiple commercial software systems are available: PointCab², TerraSolid³ or Autodesk⁴ solutions.

² <https://www.pointcab-software.com/>

³ <http://www.terrasolid.com/>

⁴ <https://www.autodesk.com/>

SENSORS DESCRIPTION

In this chapter, a closer look will be given to the laser, positioning and the inertial sensors as essential components of these complex mobile mapping systems. Besides the basics and the overview of state-of-the-art in Light Detection and Ranging (LiDAR), information regarding positional and inertial subsystems are presented in this section .

3.1 LIGHT DETECTION AND RANGING

"LiDAR is the single best way of getting integral information about the world around you."

Douglas Thornton, Battelle Memorial Institute

Nowadays remote sensing is based on LiDAR sensors, performing non-destructive measurements of the environment using laser sensing. Applications vary from construction sites monitoring [31] and piping planning in the existing building, to sensing of ground topography, measurement of the structure and the function of vegetation, or forest attributes [55].

Basic and necessary components of LiDAR system are: high frequency laser-emitting diode and the photodetector receiver [83]. The category, the power of laser transmitter, and the size of receiver determine maximal operating distance of LiDAR system. Commonly, these components are accompanied with GNSS (Global Navigation Satellite System) receiver and IMU (Inertial Measurement Unit) sensor to estimate the position and the orientation of LiDAR system for data alignment.

There are two categories of laser LiDAR systems: topographic and bathymetric. While topographic LiDAR uses near infra-red lasers of higher sampling frequencies, the bathymetric sensors operate with green light transmitter on lower frequencies. Beside higher rates, topographic LiDAR uses lasers with lower power consumption, enables larger measurement distances and better accuracy. On the other hand, the green light used in bathymetric LiDARs is able to penetrate water level [83, 71].

There is also a difference in their range measurement principle: phase-shift or time-of-flight. In phase-shift scanners, the distance between irradiated object and the sensor is measured by estimation of a phase shift between emitted and received laser signal. It enables high accuracy, acquisition speed and therefore vast amount

of collected data points, but it is suitable for nearer scenes (up to 100 m). The visible wavelengths are used in general, but some LiDAR sensors (e.g. FARO) use infrared light. Time-of-flight laser scanners estimate the distance by measuring the time between emission of a laser signal and reception of reflected return. These scanners use infrared wavelengths and enable measurements of large scenes (beyond 100 m) [18].

The other advantage of time-of-flight sensors is the ability of multiple returns or full waveform output (Fig. 3.1), while phase-shift LiDAR enables only estimation of a single discrete return (3D point) on a leading edge of return signal - first object reflectance. The possibility to “see through” the scattered objects (e.g. trees) is a significant advantage of laser mapping comparing with approaches using traditional RGB cameras. In case of airborne forrest mapping, camera imaging is only able to capture the canopy without modeling the ground terrain. This is overcome by deploying time-of-flight LiDAR system as shown in Fig. 3.1. The full-waveform outputs enable modeling of the structure in mapping environment - e.g. biomass, biodiversity analysis, or resource availability in mapping of natural environments [5].

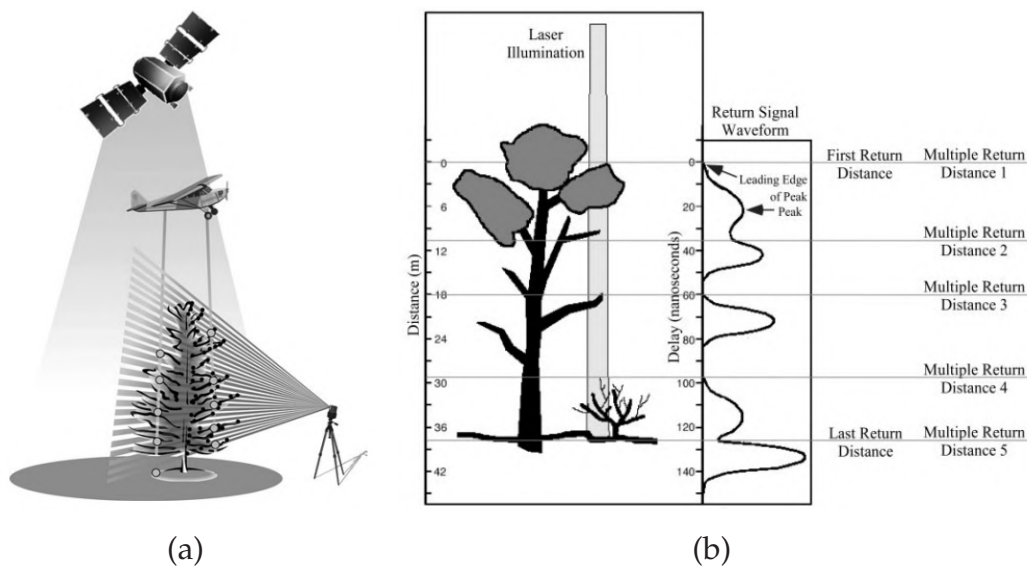


Figure 3.1: When airborne or terrestrial laser scanning (a) [94] is used for scanning of forest environment (b) [55], the laser illuminates canopy, tree branches, bushes and finally the ground. Measurements representing distances to all these objects are acquired by single laser illumination.

3.1.1 Laser Scanning Systems

From data acquisition point of view, there are three types of laser measurement methods, utilized in 3D modeling: Airborne Laser Scanning (ALS), Vehicle (VLS)

and Terrestrial Laser Scanning (TLS) – each requiring different approaches, different algorithms, and with a different suitability for different application [38].

In Airborne Laser Scanning systems, LiDAR scanners are mounted on small airplanes or relatively cheap unmanned aerial vehicles (UAV), like quadcopter or plane drones. ALS enables mapping of large environments within a short period of time, while commonly achieves worse positional precision and resolution. Since measured distances are commonly hundreds of meters because of plane's altitude, emitted light attenuates when penetrating through the atmosphere as a function of humidity, temperature, temperature, and other factors. These are the additional factors affecting the measurements comparing with TLS or VLS approaches [38, 27, 46].

In Vehicle Laser Scanning systems, size, weight, and also high durability of hardware stops to be an issue. Large and heavy robust LiDAR system can be safely mounted atop of a common vehicle. In the situations when GNSS positional system is failing (tunnels, covered parking lots, etc.) or is not available, the additional wheel odometry aiding can be used for reasonably precise estimation of the position, since the movement of the vehicle is usually smooth without sudden quick changes in speed or orientation. However, these systems are limited for environments traversable by vehicles, or outdoor mobile robotic platform.

Terrestrial Laser Scanning is suitable for situations, where high positional accuracy and data density is a priority. Also, it is not limited for clear outdoor environments as airborne scanning, or traversability by wheel platform as vehicle scanning. Usually, these sensors are mounted atop of a tripod stand or a human carried mobile platform, accompanied with GNSS receiver in case of outdoor mapping. The process of data acquisition is more demanding than in previous approaches, since the operator has to move the scanner into convenient viewpoints all around the scene in order to capture the whole environment [18].

3.1.2 *Velodyne and the other 3D LiDAR scanners*

LiDAR scanner, including laser emitter and receiver, is the core element of all previously mentioned systems [1]. First, LiDAR optical technology was used in 1960s in lunar laser ranging, satellite remote sensing, oceanography and in the atmospheric research. In 1990s, the LiDAR sensors operating in frequencies 2 – 25 kHz were used in topographic mapping applications. Since 2007 as a result of DARPA autonomous driving Grand Challenges, development and production of LiDAR scanners rapidly grows. They became the source of precise 3D information, since the traditional camera based computer vision systems (e.g. stereovision) lacked necessary precision. [2]

Originally, the scanners capturing data in a single 2D plane (so called 2D rangefinders) were available. These devices consist of a single emitter-receiver pair accompanied with optical system – usually moving mirrors – to aim the laser signal across the scanning plane as demonstrated in Fig. 3.2a. The common horizontal field of view varies from 90° to 270° . Such scanners are commonly used in industry (e.g. quality inspection in production lines), collision avoidance systems, or simple mapping solutions (e.g. ZEB [30]).

In many applications of mobile mapping, localisation, autonomous vehicle navigation, or infrastructure surveying, high resolution data, capturing the environment all around the sensor, are required and single plane measurements are not sufficient. There are different approaches extending 2D rangefinders into 3D laser scanners [2].

As the simplest solution, additional mechanical elements tilt or move up-and-down the system of laser electronics and optical parts. In these so called "winking" or "nodding" LiDARs, 3D information is captured, but on the other hand, azimuthal resolution and data density is significantly reduced.

In "flash LiDAR" units, a large area is simultaneously illuminated and per-pixel range information is captured by 2D focal plane array. These sensors are not widely used, since they are difficult to manufacture and their field of view and range is quite limited. There are stationary systems, which use flash LiDARs providing high quality 3D images but they take several minutes to collect necessary data. In the future, these scanners can potentially replace mechanical laser systems, which are limited by moving parts.

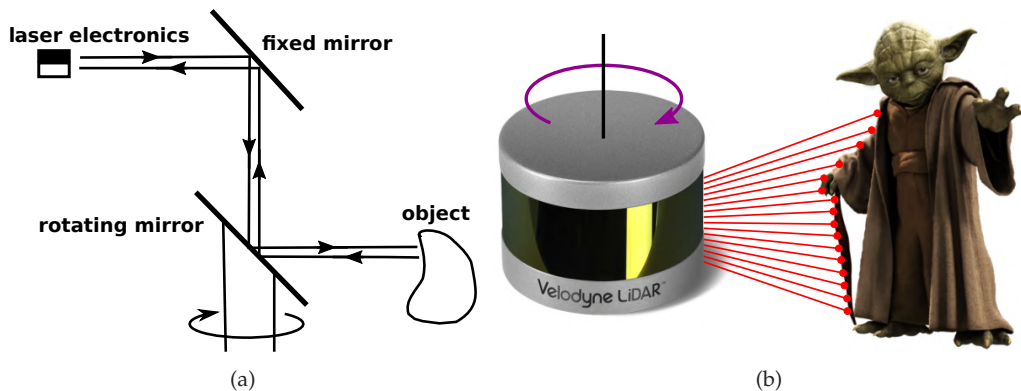


Figure 3.2: Scanning principle of 2D and Velodyne 3D LiDAR. The pair of mirrors (fixed and rotating) reflects each single emitted laser ray into the different directions of a single plane (a). This enables observation of the whole environment around the sensor – 360° horizontal field of view – while vertical field of view is limited (e.g. $\pm 15^\circ$ for VLP-16 model). Multiple laser beams (16 beams for the VLP-16 model in the example) scan the scene simultaneously (b).

Solid State Hybrid Lidars, like the Velodyne scanners, combine the spinning system providing 360° horizontal view with the solid-state detector with multiple laser beams [2], aligned regularly across the vertical field of view as shown in Fig. 3.2b. In currently available models, 16, 32, or 64 laser beams cover the $20^\circ - 40^\circ$ vertical angle. Velodyne provides data in form of a point cloud, where each point consists of the 3D position, returned intensity information and the index of a laser beam, which provided the measurement. The example of 3D laser scan can be found in Fig. 3.3.

The output light is focused by the lenses and after striking the object, a portion of reflected light passes the UV filter before the detection by the laser receiver. This decreases the energy introduced by sun and reduces the noise and the sensitivity, especially in outdoor environments. Velodyne is Time of Flight sensor, therefore it is able to operate in the single or dual return mode (see Fig. 3.1 for the details), providing first, strongest or both laser return measurements.

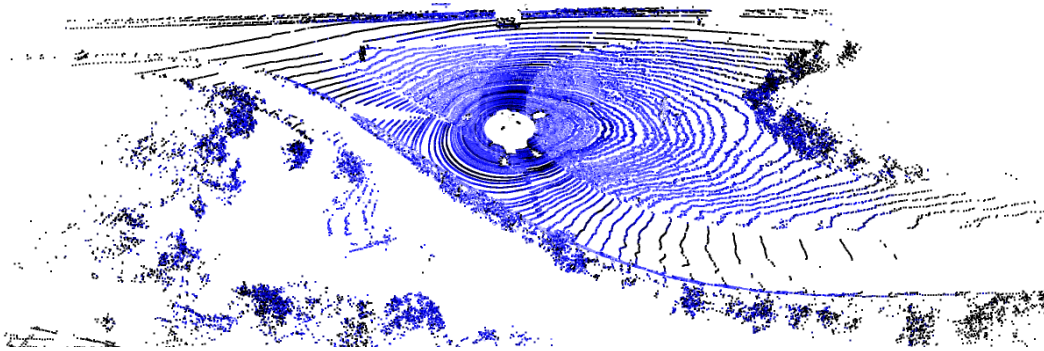


Figure 3.3: Example of Velodyne 3D LiDAR scan in crossroad environment. Data were captured by autonomous driving platform equipped by Velodyne HDL-64.

The units rotate at 600 RPM (10 Hz) by default and this angular speed can be increased up to 900 RPM. However, because the points per second rate is constant – repetition rate 20 kHz for each laser beam – increasing the frequency of rotation reduces the angular resolution. In some cases, a higher RPM also degrades the precision because of higher vibrations of a laser unit. Maximal operational range reaches up to 80 – 100 meters, while the accuracy of measurements should not exceed 2 cm. The ability of such a long range measurements limits the miniaturization of the unit, because the effective range is proportional to the diameter of optical lenses. Table 3.1 provides an overview of different Velodyne models and the properties including size, weight, and performance in points per second.

The company receives huge investments from companies aiming for autonomous driving (e.g. Ford, Baidu) and announces significant increases in production – 1 million of LiDARs in year 2018. These factors will probably significantly reduce the price of sensors, making 3D laser sensors accessible for wider range of applications.





	Model	Size (WxH, cm)	Weight	Laser beams	Vertical FOV	Points per sec.	Price \$
	VLP-16 (Puck)	7.5 × 10	830 g	16	30°	300k	5k
	VLP-16 (Puck Lite)	7.5 × 10	590 g	16	30°	300k	6k
	VLP-16 (Puck Hi-Res)	7.5 × 10	830 g	16	20°	300k	8k
	VLP- 32C (Ultra Puck, new)	N/A	N/A	32	40°	600k	50k
	HDL- 32E	8.6 × 14.5	1 Kg	32	40°	700k	40k
	HDL- 64E	23 × 28.3	13.6 Kg	64	27°	2.2M	70k

Table 3.1: Specifications of available Velodyne 3D LiDAR models¹. The availability Ultra Puck model was announced just recently and the specifications are not publicly available yet. Also, the prices are significantly falling with increasing production and therefore are only approximate.

3.2 POSITIONING SUBSYSTEM

To perform 3D mapping and create a complete model of the environment of interest, the laser scans have to be aligned into the common coordinate frame. This task is straightforward, when the trajectory of the moving platform and the changes in orientation (jointly denoted as an odometry) are known.

The estimation of odometry is a tricky part and there are multiple ways of solving this problem under different conditions and with different types of hardware available. When only laser data (or camera images, depth images, etc.) are avail-

¹ <https://velodynelidar.com/>

able, there are multiple solutions to estimate odometry using SLAM (Simultaneous Localisation And Mapping) techniques [98, 107, 105, 88, 99]. Common requirement of these systems is the presence of features and distinguishing objects (walls, tables, trees, ...) in the environment.

When these conditions are not met – e.g. in large fields, meadows, parks or empty parking lots – pure SLAM solutions are failing and aiding sensors are necessary. Common practice is the deployment of GNSS (Global Navigation Satellite Systems), providing global positional information, IMU (Inertial Measurement Unit) sensors for local motion estimation, magnetometers, wheel odometers, etc. These modules of the positional subsystem will be briefly introduced in the following sections.

Of course, the deployment of aiding sensors is not a universal solution. In many cases, these sensors are failing due to improper conditions: missing or inadequate reception of the satellite signal in GNSS solutions, slippery or unstable surface for wheel odometry, or a magnetic field interference when magnetometers are used to estimate the azimuth to the north.

3.2.1 GNSS

As mentioned before, GNSS module provides of global positioning. Beside the traditional, and fully operational GNSS like GPS and GLONASS [59, 26, 60] (developed, owned and maintained by US and Russian military forces respectively) also new systems for precise positioning have emerged. Both, the european civil system Galileo and the Chinese project BeiDou will consist of 30 new satellites each, once they are fully operational. With more than 70 satellites already launched, there will be about 120 satellites available with all 4 systems (GPS, GLONASS, Galileo and BeiDou) when fully deployed.

Multiple satellite systems provide independence on the single system, which can be shut down or blocked in a case of major incident, and moreover, it also enables multi-GNSS approaches for positioning. Availability of more satellites improves their visibility, spatial geometry, dilution of precision, and therefore also convergence, accuracy, and reliability of precise positioning. The fusion of measurements from different satellite systems is performed internally in the positioning systems, contained in plenty of devices from the smartphones to professional navigation, or geodetic solutions, and it is out of the scope of this work.

3.2.1.1 *Corrections and precision of GPS positioning*

With a single GPS receiver, the approach is referred to as the absolute positioning [26]. Significant positional inaccuracies (error 5 – –20 meters) are caused by an

error in satellite orbit, errors of both the transmitter and the receiver clock, atmospheric (ionospheric and tropospheric) errors, multipath etc.

When the additional reference GPS receiver with known position is available, a so called differential positioning with better accuracy is available. The idea is that receivers placed within a certain distance will be similarly affected by most of errors listed above. Corrections can be estimated at the reference station and sent to a so called "rover" to improve its measurements.

The distance from the receiver to the satellite can be estimated by the *code phase* estimation – the phase difference between received and generated digital PRN sequence. This approach is simpler, but also less accurate, and the estimated distance is referred to as an pseudorange. A more precise solution is the estimation of the *carrier phase* between the received radio signal and the signal generated by internal oscilloscope [26, 53]. This is more tricky, since there is no way of directly estimating the number of complete cycles of the carrier signal in the phase – only the fraction of cycle is certain. This is called *integer ambiguity*.

Using the pseudo-range corrections on the level of PRN code phase (or code shift) is called *Differential GPS (DGPS)* and it improves precision of the estimated position up to 1 – 5 meters. In 1985, the Radio Technical Commission for Marine Services (RTCM) proposed a standard for transmitting these corrections over the radio for real-time positioning in the marine navigation.

On the other hand, the deployment of the carrier phase corrections requires a solution of integer ambiguity at first. The number of total carrier cycles is the additional unknown in the system of equations. Firstly, this approximation by the float number is used resulting in the sub-meter positional precision. After a sufficient number of observations from satellites is reached and the solution converges to a number of full carries cycles, the precision in order of centimeters (2 – 3 cm) is achieved. Precision is also affected by the distance between the rover and the reference station.

The carrier phase corrections can be accessed offline after the measurements are completed and the correct positions are estimated in the post-processing stage. In many cases, there is a demand for online corrections, when the precise position is required in real time (marine or airborne navigation, ground marks estimation by construction engineers, etc.). The standardized way of the online carrier phase corrections exchange is known as *Real Time Kinematics (RTK)* [53].

For using the RTK corrections, a base station of known position with second GPS receiver has to be set-up. The transmission of corrections is usually performed via radio connection. There are also many companies providing these data from real, and also from so called virtual base stations, all around the world. The charges for

using these data are relatively small, and the data are easily accessible via mobile internet connection with low requirements on data bandwidth.

3.2.2 Inertial Measurement Unit (IMU)

Beside the positional information provided by GNSS receivers, the orientation of the moving platform carrying the scanner has to be estimated in order to align captured LiDAR scans. Inertial sensors – accelerometers and gyroscopes – are able to estimate linear acceleration and angular velocity respectively [103]. Integrating these measurements, both the positional information and the orientation with respect to the initialization is estimated with a high output rate. Usually, a unit is constructed from 6 inertial sensors: 3 orthogonal accelerometers and 3 orthogonal gyroscopes each aligned with different axis (front-back, right-left, and top-down direction).

There are two basic types of inertial systems, which differ in a reference frame, which accelerometers and gyroscopes operate in. Stable platform systems are aligned with global reference systems, since the inertial sensors are mounted on the stable platform isolated from external rotations using gimbals. On the other hand, in strapdown systems the inertial sensors are mounted rigidly on the platform. Therefore, the data are measured in a local body frame rather than a global frame. In order to compensate gravity, firstly the outputs of rate gyroscopes are integrated into the orientation and used for alignment into the global frame. Then, corrected acceleration data (without the gravity effect) are double integrated to obtain a valid position with respect to the initial one. The whole schema of the strapdown system computation is described also in Fig. 3.4. Most of current systems are strapdown, rather than gyros with a stable platform, because of better stability, higher angular speed capabilities and lower g-sensitivity [8].

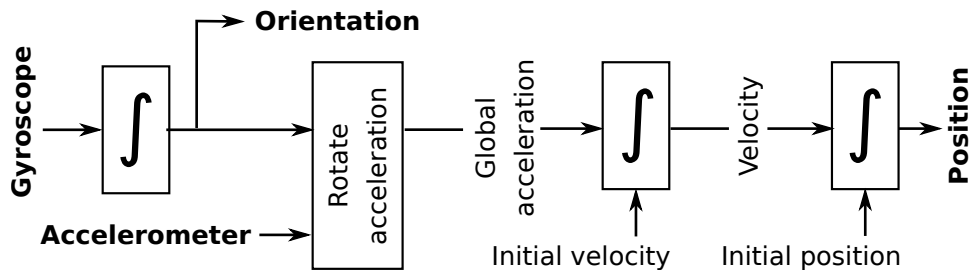


Figure 3.4: The operational schema of strapdown IMU. The integrated angular velocity (the orientation) is used to correctly project the outputs of accelerometer into the global coordinate frame. On contrary, stable platform IMU operates in global frame directly and the outputs of the gyroscopes and the accelerometers are processed completely independently without the projection of acceleration into the global axes and also without the orientation feedback.

IMU sensors can be also divided into multiple groups, according to the physical principle of the gyroscope inside. The most common types are *fiber-optics gyroscopes* (FOG) and *MEMS* (*Micro-Electro-Mechanical Systems*) gyroscopes.

The optical ones are based on the theoretical principle of Sagnac effect, discovered at the beginning of the 20th century [70]. According to this principle, propagation of the light emitted into the closed-loop path depends on the rate of external rotation. The core of FOG contains the coil of optical fibers [103]. The length of fiber varies from hundreds of meters to kilometers [8]. Into this coil, two laser beams are fired from the opposite directions. The one propagating in the direction of the rotation takes a longer path, while the path is shorter for the beam fired from the opposite direction. This principle is described in Fig. 3.5. Within the coil, beams are combined, and they interfere due to Sagnac effect. The external rotation introduces a phase shift and causes changes in the intensity of combined beams proportionally to the magnitude of angular velocity.

The most important advantages of this type of gyroscope are high scalability (for higher rotation rates), since it is possible to increase the length of an optic fiber on the coil, high precision, and construction without moving parts [8]. For manufacturing, standard telecom-technology components (optic fibers, transmitter, receiver, etc.) are used and therefore are easier to assemble [70].

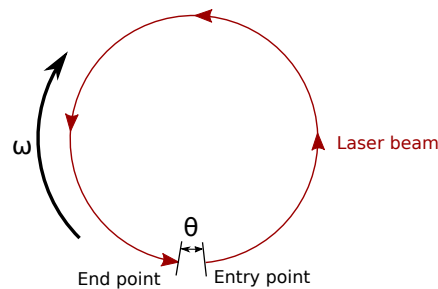


Figure 3.5: Sagnac effect within the loop of optical fibers causes the beam (dashed circle) in the direction of rotation ω to experience longer path than the beam in the opposite direction (solid circle). As an result, phase shift Θ is introduced into the combined signal.

MEMS gyroscopes are built using single silicon micromachining technology, they are easier and cheaper to manufacture, because they do not need a high-precision assembly [103]. Although their precision can not match the accuracy of the optical units, MEMS are smaller, consume lower power and they are constructed more robustly. The gyroscope contains a mass vibrating along the drive axis. Upon external rotation, the mass undergoes also the secondary vibration introduced by the Coriolis effect.

3.2.3 The GNSS-IMU cooperation in the positioning subsystem

The outputs of IMU sensors significantly drift over time, and therefore, lower rate aiding sensors like GNSS or magnetometers are needed to provide stable data. They are used to fix accumulated errors and to provide initial values by direct measurements of the position and the orientation [8, 91]. The combination of a high rate and less accurate IMU outputs with more stable and lower frequency aiding data compensates drawbacks of each sensor, when used separately. The fusion of data provided by inertial and global navigation sensors is commonly performed by a reliable Kalman filtering or linear quadratic estimation [50], which preserves the reliability of the navigation under varying conditions and the qualities of input.

The system of coupled sensors – GPS, IMU and alternatively additional sensors, like odometers or magnetometers – provides complex positional information, including rotation with respect to a certain global reference frame. On the other hand, individual sensors (accelerometers, gyroscopes, magnetometers, LiDARs, etc.) provide measurements within their local (body) coordinate frames. In order to combine these data, transformation into a common coordinate system has to be performed [91, 50].

Inertial sensors – accelerometers and gyroscopes – provide the acceleration and the angular speed measurements within a local body frame (see Fig. 3.6) with respect to the current position and the orientation of the sensor. As mentioned before, in order to fuse the data from GPS, inertial measurements must be at first transformed into some geodetic reference frame used internally by the navigation GPS-IMU subsystem.

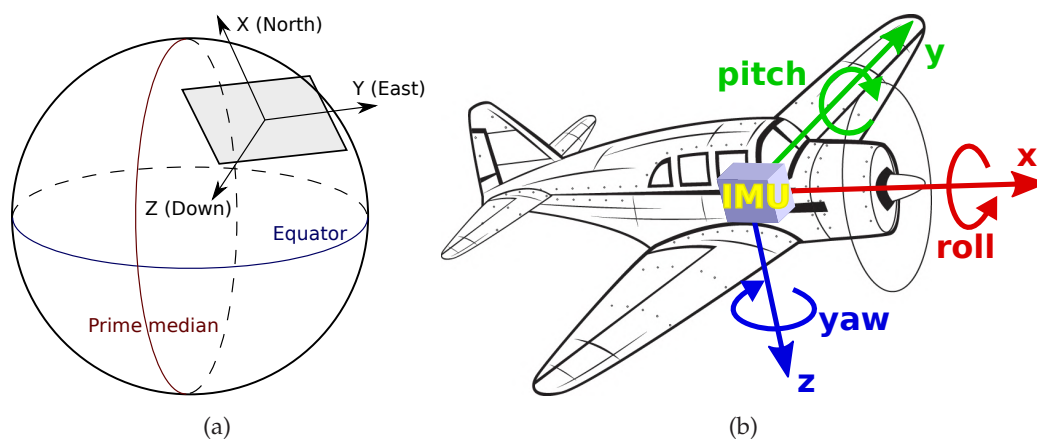


Figure 3.6: Local geodetic (navigation) frame NED (North-East-Down) (a) given the surface normal plane in the certain position. The IMU attached to the measures the values of acceleration (X-Y-Z) and the orientation (Roll-Pitch-Yaw) (b) with respect to the local body coordinate frame.

This transformation is usually represented by the rotation matrix R , also denoted as Direction Cosine Matrix (3.1), (3.2), computed from Euler angles roll ϕ , pitch θ , and yaw ψ , represented by matrices R_ϕ , R_θ , and R_ψ . Multiplication of the point or the vector in a body frame by this matrix will transform this point/vector into the Local Geodetic frame (NED) in Fig. 3.6a [91].

$$R = R_\psi R_\theta R_\phi \quad (3.1)$$

$$R = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (3.2)$$

Depending on the number of known Euler angles, the IMU operates as a vertical gyroscope or AHRS system. When only a roll and a pitch (jointly denoted as the attitude) are known, IMU is not able to estimate the heading azimuth in a horizontal plane. When all angles including yaw (heading) can be estimated, IMU operating as an AHRS (Attitude and heading reference system) and the fusion with GPS data (e.g. in the Kalman filtering) is possible [91, 50].

3.2.3.1 Heading estimation

The estimation of attitude is a relatively simple task, since accelerometers measure the gravity vector in a body coordinate frame, when keeping still, or under a constant motion. Aligning this vector (e.g. in the calibration stage) with the vector $[0, 0, 1]$ in the NED geodetic frame is a straightforward mathematical task and it is equivalent to the estimation of the roll and pitch angles.

The harder part is the *estimation of heading* (yaw angle) in order to perform a full alignment with a geodetic reference frame. There are multiple possibilities for this estimation given the available hardware and the properties of the application [91]:

- In the situations, where the mobile platform is moving constantly forward, the GPS course can be set as the heading orientation.
- With the additional second GPS receiver, so called GPS True heading can be estimated from a dual antenna system. The antennas of the receiver are placed in a fixed position with respect to the carrying platform in a significant distance from each other (meters or at least tens of centimeters). An example of such a system is the backpack ROBIN, described in the Sec. 4.3.2 and shown in Fig. 4.5b.
- Performing a calibration procedure with high accelerations, when the changes in the GPS position can be aligned with the velocity vector, provided by the integration of the accelerometer output (after a gravity compensation).

- Additional sensors like magnetometers, which are able to estimate the direction to the north. These sensors are quite sensitive to disturbance by the source of magnetic field and require proper magnetic calibration.

Both the alignment of reference frames and the data fusion is usually performed internally, within GNSS-IMU navigation solutions, nowadays provided by many companies (e.g. OxTS, Novatel, SBG, Advanced Navigation, IMAR, etc.).

EXISTING SOLUTIONS FOR MOBILE LASER MAPPING

In today's 3D mapping solutions, a laser scanner plays an inevitable role because of its precision, direct estimation of the 3D information (compared to the stereo vision), and independence from illumination conditions. Many companies develop their different solutions, fulfilling different requirements on accuracy, mobility, and time consumption of the scanning process. We provided an overview of multiple existing LiDAR mobile mapping solutions in our previous publication [100] and it can be found in Table 7.1 in Chapter 7.

There are also some highly challenging and even pathological environments, such as long tunnel-like environments, or large, flat, and featureless spaces (large parking lots, fields, meadows, etc.) [90]. For these scenes, the 3D reconstruction and SLAM with purely LiDAR system would fail without an aid of some additional sensors (GNSS sensors, odometer, IMU, magnetometers, etc.).

Specialization of a particular solution and therefore the key differences in available solutions are given by the platform for mounting of a laser scanner (tripod, backpack, or handheld solution) and by the way of extending the simple 2D rangefinders into the 3D laser scanners – by the system of rotating mirrors, or by the flexible spring. Besides this, the sensors vary in the price, size, accuracy and the requirements on scanning process.

The representatives of these mobile mapping systems will be presented in the following chapters. Beside these solutions, there are also mobile solutions, when laser scanning system is mounted on a vehicle platform and falling into the category of VLS, Vehicle Laser Scanners (e.g. solutions of RIEGL¹ company). They are not limited in terms of weight and lack of mobile platform robustness. Also, assumptions, regarding the movement, can be made, and additional sensor inputs can be used. The vehicle drives smoothly and follows the Ackermann steering principle, which constraints the trajectory and simplifies the estimation of the vehicle odometry [86]. Moreover, the wheel odometer provides additional inputs for the movement parameters estimation. Unfortunately, these solutions are strictly limited for easily traversable environments and can not be used in staircases, difficult terrain, etc. Therefore they cannot be considered as fully mobile mapping solutions.

¹ <http://www.riegl.com/>

4.1 HANDHELD ZEB-1 AND ZEB REVO SOLUTIONS

In 2012, the concept of the Zebedee scanner (Fig. 4.1a) has been presented [13]. It took multiple model generations until the scanner evolved in 2013 into a commercial product ZEB-1 (Fig. 4.1b) and a more recent model ZEB-REVO (Fig. 4.1c) of the GeoSlam company [64, 15].

ZEB-1 was used and tested in many scenarios [90]. During the few hours of walking, the data for the 3D model of a small village was captured for the reconstruction of the cultural heritage. This small handheld design is suitable for a detailed reconstruction of small areas, indoor mapping and it was deployed in a mapping of an underground mining system and also in scanning of an outdoor forest structure.

In traditional mapping systems, the extension of 2D rangefinders into 3D scanners is achieved by an additional moving platform, performing a known and deterministic motion, changing the pose (especially the orientation) of the 2D scanner. Additionally, the stabilizers are used to reduce the high frequency motions or vibrations caused by terrain or mechanical obstacles. On the contrary, in Zebedee (and ZEB-1) the motion of the platform is amplified by a flexible spring for non-deterministic and smooth changes of position and orientation, enabling a 3D range sensing [13]. The spring tends to amplify low and medium frequency motions and suppress high frequencies and vibrations. This quality can be fine tuned by adjusting the length, the toughness of the spring, and the changes in the mass of the mounted sensors.

Besides the time-of-flight laser rangefinder (Hokuyo UTM-30LX), the IMU sensor with high frequency accelerometers and gyroscopes (MicroStrain 3DM-GX2) is placed within a sensor head. It serves for the initialisation of trajectory, and the deviations from the measured IMU data are used as a part of a loss function in the optimization process. Hokuyo rangefinder covers 270° of a horizontal field of view with a 30 m indoor and 15 m outdoor maximum range.

ZEB solutions produce 3D reconstructions with only 4 degrees of freedom (DoF) [20]. Since the IMU sensor with accelerometers and gyroscopes is used, the attitude (roll and pitch angles) is estimated, and only the initial XYZ location and the heading (yaw angle or the bearing to north) are unknown. For all DoF estimation, additional sensors would be required: GNSS sensor for positional information, and a way for a true heading estimation, for example a magnetometer or a dual antenna GNSS receiver.

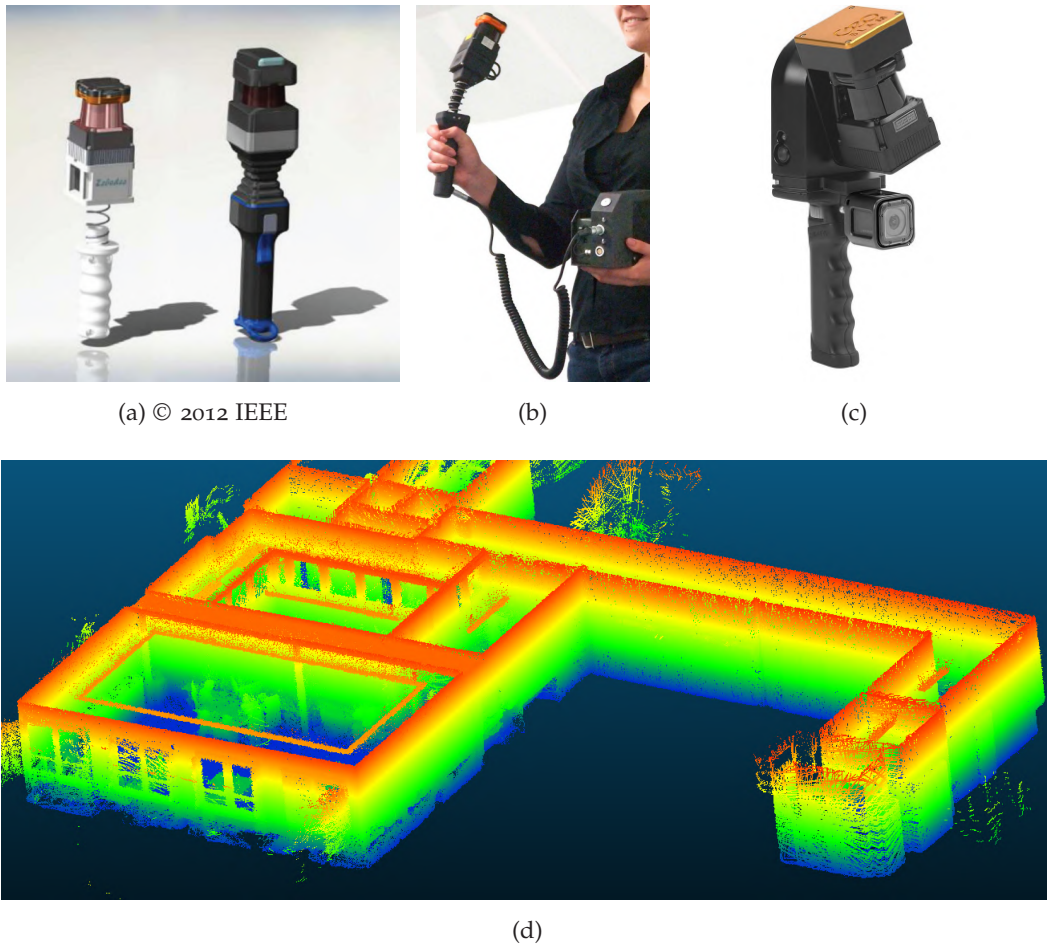


Figure 4.1: The original concept of Zebedee scanner (a) [13] and the evolution into the commercial products. The spring mount for extension of the laser rangefinder into the 3D scanner was used in ZEB-1 (b). In ZEB-REVO (c), the rangefinder is continuously rotated around horizontal front axis for scanning in 3D. The height-colored example of 3D model of the office environment created by ZEB-1 is shown in (d). Images (b) and (c) were taken from the manufacturer’s website².

4.1.1 Algorithm overview

For the estimation of an elapsed trajectory, a SLAM algorithm for a continuously spinning 2D laser [11] from the same authors is used. Algorithm follows similar strategy as widely-used ICP algorithm (Iterative Closest Point) repeating matching and optimization step. In the first step, the surface elements (or surfels) are estimated from spatially and temporarily close points.

Firstly, the voxel grid is build over data. Within well populated voxels, surface parameters are computed from centralized second-order momentum matrix of 3D points coordinates. Only the points falling into the fixed time window are processed together. Corresponding surfels are matched by K-nearest neighbour search

² <https://geoslam.com/>

of 6D space of the surfel positions and normals. In optimization step, the corrections of surfels positions and orientations are estimated jointly minimizing following criteria: distance of matched surfels, deviations of IMU measurements, and initial condition constraints enforcing continuity with previous trajectory segment.

Zebedee/ZEB-1 SLAM algorithm computes elapsed trajectory in 3 steps. Firstly, the initial guess is estimated by integration IMU reading from accelerometers and gyroscopes. Then, as described above, the open-loop trajectory is computed, using iterative algorithm described above, where matching and optimization step is repeated within a certain time window. Finally, global registration for reaching closed-loop solution uses the same algorithm for all data, instead of computing small increments in a limited time window. If the error accumulated by the incremental solution does not exceeds certain level, this global registration produces correct alignment of recorded data.

4.1.2 ZEB-1 performance and usability evaluation

In order to evaluate performance, accuracy, pros, and cons of ZEB-1 solution, the comparison with precise terrestrial laser scanner Leica C10 was made [90]. While a Leica scanner achieves millimetre accuracy and fine resolution, the scanning process is time and effort demanding while it requires manual transporting of the scanner and sophisticated data post-processing (see chapter 4.2). Therefore it is ideal for ground truth acquisition and as a baseline solution in terms of the output quality.

On the other hand, ZEB solutions can be used by a person after a short introduction of the technology. The data are collected while a person is walking in the environment which should be reconstructed. This process is faster compared to terrestrial solutions. The post-processing can be also performed by the user in the additional proprietary desktop software, but it is also possible to submit collected ZEB data to the online system of GeoSlam company. For both the desktop software and the post-processing service, further payments are charged – one time payment for the software or payments per each data sequence processed by online service (the amount depends on the data size).

There are also limitations in ZEB-1 design, especially in the motion transferred from the operator holding the sensor. If the spring and the sensor head is being still while the person is moving (e.g. person slowly turns or walks too smoothly), the scanner is degraded into a simple rangefinder and it is likely that the SLAM algorithm fails. Also in cases of very high dynamics, an error is possible to drift extensively, and inaccurate model is built due to a failing global registration. Also – from the experiences of data collection operators – the acquisition procedure is

uncomfortable and the process is becoming physically painful approximately after 30 minutes of manual "swinging" the sensor head.

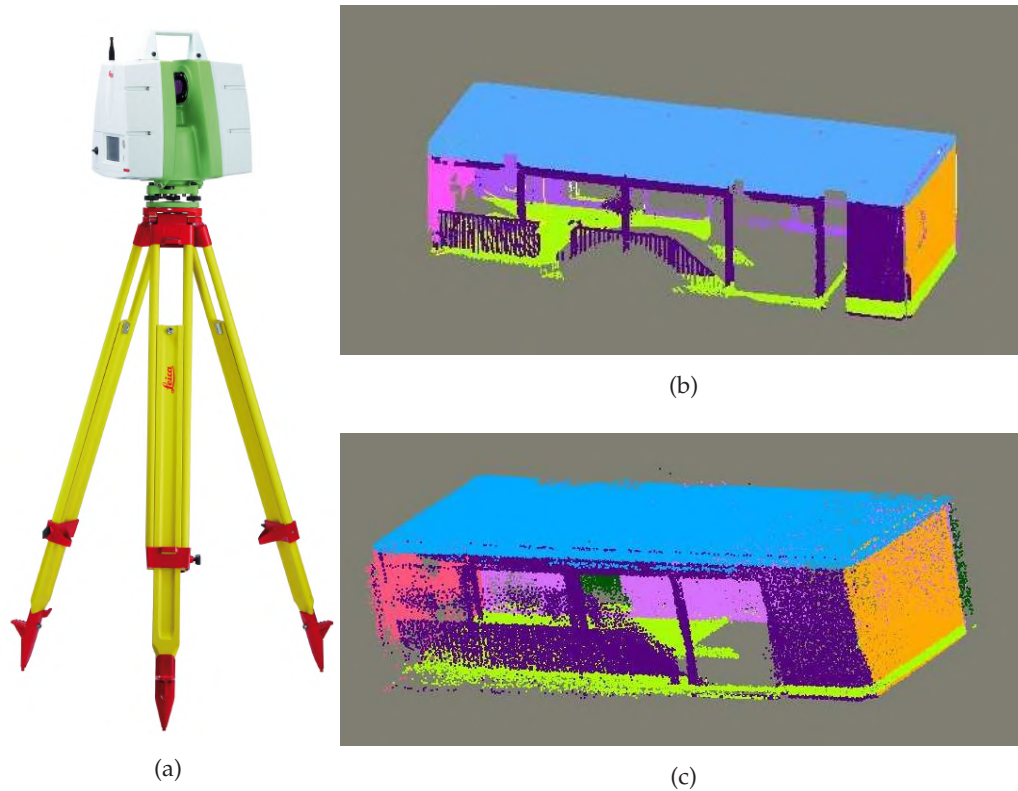


Figure 4.2: Terrestrial laser scanner Leica C10 (a) (image taken from manufacturer's website³) used as a reference system for evaluation of ZEB-1. Both solutions were used for 3D reconstruction of the same indoor environment of the fire brigade building. Denser, more accurate and less noisy result is achieved by Leica C10 scanner (b) than by ZEB-1 solution (c) [90]. These differences were expected and they are also visually observable. Note: point clouds were colored by the wall segmentation.

Sirmacek et al. [90] evaluated the performance and the accuracy of the ZEB-1 sensor in terms of visual recognition of the objects in 3D model and also prepared a quantitative analysis.

When the user of the 3D reconstruction (e.g. architect, mechanical engineer) visually inspects the model or tries to recognize important features, in the output of the ZEB mapping, the objects in a decimetre scale are visible – staircase handles, window and door boundaries, etc. However, smaller objects like decorations, small holes in a ceiling, or cracks in wooden parts are lost in the noise and inaccuracies. If such information is necessary, more precise sensors (terrestrial lasers like Leica C10 or FARO Focus) have to be deployed. Visual recognition of objects is worsened by the lack of returned laser's intensity reading, which correlates with the color of the object, in Hokuyo LiDAR. This model can be only artificially colored by

³ <https://leica-geosystems.com/>

the height, timestamps, surface normals or the segmentation. Other scanners like Velodyne and terrestrial LiDARs (e.g. FARO, Leica) provide this value and the terrestrial lasers usually also capture RGB data for the texturing laser measurements by the true surface color.

The points density varies in the interval of 1.000 to 18.000 of points/m². This represents an average distance 0.8 – 3 cm between the points and one can not expect finer details to be captured in the output reconstruction. On the flat surface patches (20 × 20 cm patches used in the experiments), the standard deviation of 3D points varies within an interval of 4 – 9 mm which corresponds to the Hokuyo LiDAR specification of the standard deviation below 10 mm, for the measurements up to 10 m. In the other experiment with a ground truth provided by Leica C10 scanner, both for the Leica and the ZEB-1, the CAD model was built, and corner to corner distances were measured. The maximal error of these distances reached 3.8 cm for the ZEB-1 compared to the Leica baseline. For a room of 6 × 4 m, the difference between the real floor area and the area estimated by the ZEB-1 CAD model was 0.4 m². Such accuracy would be probably acceptable for the needs of real-estate assessments or approximate planning of the reconstruction. For more precise measurements (e.g. measuring window glass), such accuracy would not be acceptable.

The important aspect of the solution is naturally its price. For a ZEB-1 scanner, the price is approximately 26.000 €, but further charges for a post-processing service, or a one-time investment in a post-processing software and training of the operator is necessary.

4.1.3 *Specification and performance of the ZEB-REVO solution*

Further evolution of the ZEB-1 laser scanner, introduced in 2013, resulted in the development of the ZEB-REVO model in Fig. 4.1c, released in March 2016 [15]. The key difference between these models is an additional built-in motor for continuous rotation of the incorporated Hokuyo UTM-30LX-F rangefinder [30].

The motor of sensory head rotates the LiDAR in 0.5 Hz frequency, generating significantly lower motion than the spring mechanism used in the previous ZEB-1 scanner. In order to increase the point density and even the distribution of 3D points, a rangefinder with a higher frame rate was deployed too [20]. Instead of the Hokuyo UTM-30LX providing 40 frame-lines per second in ZEB-1, UTM-30LX-F of ZEB-REVO scans 100 lines per second with lower angular resolution. Therefore, both solutions generate the same number of approximately 43.000 3D points per second.

The regular distribution of the measurements is necessary for preserving the performance of the SLAM algorithm (Zebedee algorithm is described in Section sec:zeb-algorithm), since only well populated voxel sectors can be used for a surfels generation and for an alignment estimation [15]. Although the algorithm behind the ZEB-REVO has not been particularly disclosed, since the Zebedee scanner deploys exactly the SLAM solution for continuously spinning rangefinder, one can assume that the algorithm remained the same, or at least the core features were preserved.

A significant advantage of the ZEB-REVO scanner, compared to the previous spring-like solution ZEB-1, is the independence on the transferred motion from the moving platform or from the operator carrying the scanner. This enables both the stationary parts in the trajectory, smooth motions of the mount and also the placement on the fixed position (e.g. on the wall for continuous monitoring of the environment). Total weight of the solution is 5.1 – 1.0 kg for the sensory head and 4.1 kg for the backpack casing and its content [30]. Although the additional motor for continuous LiDAR rotation eliminated the need for manual "swinging" of the sensor head, improved comfort in data acquisition could be compromised by almost 2.5-times heavier handheld part (the sensory head for ZEB-1 weights only 0.41 kg [13]).

Specifications claim maximum range of 30 m in an ideal (indoor) environment and the optimal performance for range 15 – 20 also outdoors [29, 30, 95]. The measurement noise should remain below 10 – 30 mm and the absolute positional accuracy, after 10 min scanning process, between 3 – 30 cm. The sensor head rotates in 5 Hz frequency covering $360^\circ \times 270^\circ$ field of view.

The cost of hardware including a basic one year support is 34.000 €. For the additional training of the operators, data processing services or software, further payments are charged. for example, additional RGB camera can be purchased and mounted on the sensory head. It provides supplementing imagery and also enables colouring of the 3D points for a better visual inspection of the reconstructed environment.

4.1.4 *Problematic environments and usage guidelines*

In general, ZEB scanners achieve the best scanning results in open indoor scenes with well distinguished feature objects [30]. In featureless environments, door transitions, and stairwells, precautions have to be made to obtain good quality results.

The optimal size of featuring objects should be proportional to the scanning range in 1 : 10 ratio (e.g. 0.5 m objects for 5 m distances). In the absence of such conditions, augmentation of the scene could be necessary e.g. by placing the boxes

around or keeping a short scanning range (optimal range is below 10 m). Other solution could be pointing the scanner towards the featuring objects while the operator transits the environment. Similar procedure could be necessary when transitioning (e.g. through the doors) to a different environment. When features of both environments could not be observed at once, the operator should turn before the transition and move backwards, while pointing towards a previously observed environment. The operator should repeatedly scan the same features, take a slow walking pace, limit the survey to 30 minutes and avoid scanning the moving objects. Especially in featureless environments, moving objects could impact the results badly. It is also a good practice not to keep other people close: optimally, at least 20 m range should be preserved.

Since the error of estimated trajectory could significantly drift already after a short period of time, closing the loop after re-surveying of a previous position is necessary for keeping the reconstruction accurate. As a minimal requirement, starting and finishing positions of scanning trajectory should be the same. Beside this, the operator should re-visit as many known places with significant features as possible.

4.1.5 *Performance in real applications*

There have been few external evaluations of the ZEB-REVO solution in applications of indoor mobile mapping [64], monitoring of slope instability [95], or mapping an underground quarry [20]. The precision in terms of registration accuracy, with respect to the baseline obtained by a precise terrestrial laser, or the noise as the distance to best fitting plane, were computed.

In the indoor mapping evaluation [64], a 3D reconstruction obtained by a Leica P20 terrestrial laser was created and used as a ground truth. Same rooms were also mapped by ZEB-REVO and the resulting registration were registered with the precise Leica model (ground truth). In the experiments, the RMS (Root Mean Square) values less than 1 cm were achieved. Moreover, in 3D models, plane segments were detected and distances of each point with respect to the best fitting plane were computed. For the ZEB model, standard deviations of these distances were 11 mm (and 5 mm for Leica baseline reconstruction). These evaluations also showed that a higher error was generated in areas near doors and transitions between rooms. This corresponds with the guidelines for data acquisition (see above) which recommend precautions for such areas.

In the task of slope monitoring [95], ZEB-REVO was used as a complementary source of data, where measurements of terrestrial LiDAR were missing. The complementation of data is shown in Fig. 4.3. In comparison with other sources of 3D

data, like a structure from a motion of camera images or airborne laser scanning, mobile laser mapping provides a fast, cheap, and relatively precise way of missing data.

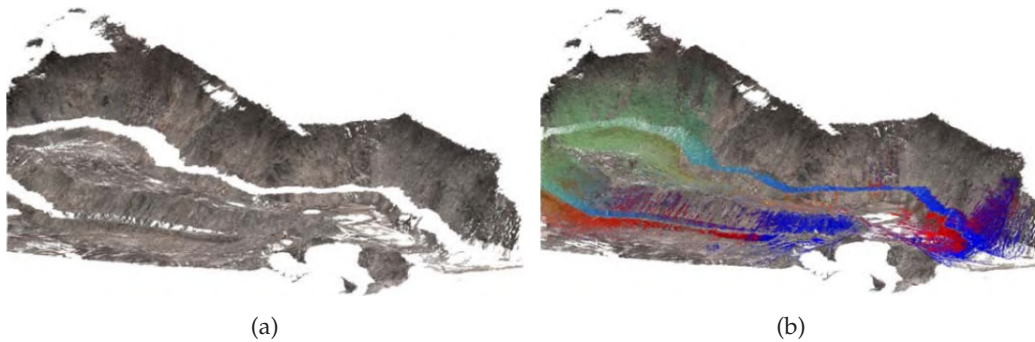


Figure 4.3: The reconstruction of the mountain slope obtained by terrestrial laser scanner (a) with missing parts (white) caused by the occlusions. These areas were scanned by ZEB-REVO mobile LiDAR (colored areas in (b)) in order to complete the model of monitored slope. [95]

In the first step, traditional terrestrial laser scanner is used for precise reconstruction of a terrain from different positions at the bottom of the slope. These frames were registered using the ICP approach with 1 mm mean absolute error. However, many areas remain unscanned (white areas in Fig. 4.3a) because of occlusions with terrain elevations. Complementary point cloud models (coloured in Fig. 4.3b) are created by the ZEB-REVO mobile scanner, since the manipulation with a large terrestrial scanner in the missing areas would be infeasible. And vice-versa: using the ZEB solution for mapping the whole mountain slope is not possible because of a limited outdoor range of approximately 15 m. To complete the reconstruction, the registration of terrestrial and mobile scanner models, with alignment error below 10 mm, was performed.

ZEB-REVO was also tested in mapping of two experimental underground quarries created by French Ministry of Environment [20]. In these experiments, reconstructions with average point density 13.350 and 5.501 points per m^2 (point spacing 8.7 and 13.5 mm per point) were created. Point precision 25.5 and 32.2 mm in terms of the standard deviation with respect to the best fitting plane on perfectly (artificial) planar surfaces were computed.

4.2 TERRESTRIAL SOLUTIONS (FARO FOCUS)

Another way of mobile mapping is a deployment of terrestrial mapping systems usually mounted atop the tripod stand. There are many phase-based scanners like Leica C10 in Fig. 4.2a, FARO Focus solutions in Fig. 4.4a, or time-of-flight sensors

like Sick [44]. Mobility of these solutions is simulated by physical positioning of the tripod into different locations, capturing the individual overlapping frames which have to be aligned in postprocessing. The most significant advantage of terrestrial LiDARs is high accuracy in order of millimeters. The output reconstruction of the office environment (previously mapped also by ZEB-1 in Fig. 4.1b) is shown in Fig. 4.4b including markings of all of the positions where the scanner was placed.

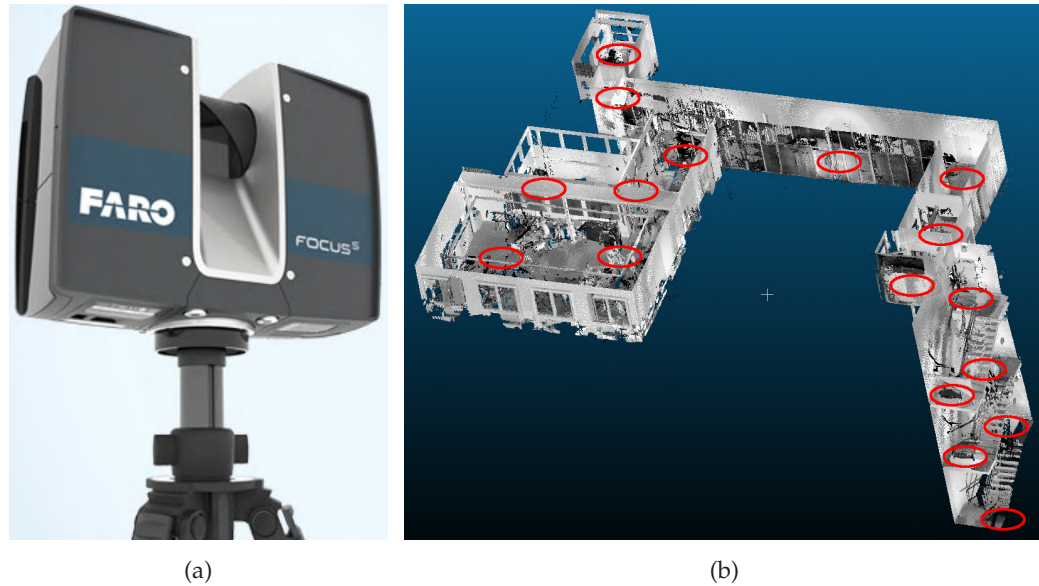


Figure 4.4: The example of terrestrial laser scanner head (FARO Focus in this case) mounted atop of the tripod stand (a) (image taken from manufacturer’s website⁴). In the middle of the head, rotating cylinder with the system of mirrors reflects the laser ray into different directions capturing 305° vertical field of view. Moreover, whole head is slowly turning around covering 360° horizontally. The result of 3D mapping of the office environment with such laser scanner can be found in (b). All positions where the sensor was placed are marked by red circles.

When such terrestrial scanner is used, one has to plan the measuring process, identifying the proper viewpoints, so that whole reconstructed area would be captured [90]. During the data acquisition process, the operator has to move the scanner and the tripod platform into these positions and wait until the scanning procedure is done. Usually, a 360° horizontal and approximately a 270 – 320° vertical field of view is captured (270° for Leica C10, 305° for FARO Focus⁴). After all scans are made, the operator has to align them in external program. This whole process is quite time-consuming and this necessary assistance of a human operator for the whole process increases the cost of the 3D reconstruction. Also, in some models and difficult mapping scenarios, additional artificial features have to be used to augment the environment. These objects, in a form of small spheres or retro-

⁴<https://www.faro.com>

reflective targets, are used for estimation of correct registration of different point cloud scans [63].

Regarding the cost of the scanners themselves, a price for a basic FARO Focus model 70M starts at 20.000 €, and for higher models the price can be doubled (40.000 €).

Another advantage of FARO like scanners is the presence of laser intensity measurements in the output data, which can be used for the point cloud colouring Fig. 4.4b. Such data are missing in e.g. ZEB-1 reconstruction. By fusion with on-board camera, RGB coloured point cloud is available, when reasonable lightning conditions are met.

4.2.1 *Applications of terrestrial scanners in mobile mapping*

Apart from more typical applications like mapping the buildings or indoor environments as shown in Fig. 4.4, terrestrial laser scanners were used in many other domains to provide 3D information for reconstruction, monitoring change detection etc. One of the tasks, where the large scale terrestrial laser data were supplemented by hand-held ZEB measurements for slope monitoring, was previously presented and described in Fig. 4.3.

In the monitoring of forest inventory [63], the number, positions, and diameters of trees were estimated. The LiDAR technology is quite convenient for estimation of these parameters in the larger forest area where manual per-tree measurements would have significant time requirements.

While the maximal range for terrestrial scanner is about hundreds of meters, the range span between the measurements is much lower because of occlusions even in sparse forest. Moreover, observations of a tree from different sides make it easier to recognize in the joint point-cloud reconstruction. Although, denser scanning results in a higher time consumption of data acquisition, larger amount of data, and higher requirements for frames alignment.

After the reconstruction is completed, the Digital Terrain Model (DTM) is computed for the estimation of a terrain slope and a ground height. The slice of remaining data in the certain height above the ground is processed by clustering and circle fitting for the tree detection including the position estimation. After a tree is detected, diameter, height, position and vertical angle of each tree is computed, providing vast information about the forest inventory.

Besides the range measurement, the returned and detected laser intensity can be used for further structural analysis in monitoring of a snow cover [43]. FARO scanner with a 785 nm light wavelength is ideal for this case because of low absorption from snow. In the performed study, dependency of range and intensity

measurements on the snow parameters were investigated. In particular, height of snow cover, size and shape of the snow grains and the overall snow wetness affects the LiDAR data.

In precise farming applications, an airborne laser scanning is typically used. Although airborne technique is able to cover a significant area in a short time, the lack of accuracy and high expenses makes it unsuitable for particular tasks of precise farming such as a growth height estimation or an ear recognition [62]. Using a terrestrial FARO scanner mounted on a movable rack about 3 m high above crops, the precise 3D reconstruction was made, and it was possible to estimate required crop parameters.

4.3 BACKPACK LASER MAPPING SOLUTIONS

In previous sections, handheld and terrestrial laser scanners were introduced. On the one hand, handheld solutions powered by relatively simple 2D LiDAR rangefinders with a limited range are convenient for indoor environments, where high mobility and flexibility is required and the features (corners, distinguishing objects, etc.) are close by. On the other hand, TLS provides long range precise measurements, convenient both for indoor and outdoor environments. However, the mobility is limited and the data acquisition process is quite time-consuming.

Backpack mapping solutions can be considered a golden mean, providing long distance measurements, mobility, flexibility, reasonable accessibility, comfort and – most of all – quick data collection, while preserving a sufficient accuracy for many mapping tasks [54, 66, 84, 93, 52, 104]. A person carrying a mapping backpack is able to climb the stairs, traverse a challenging terrain and plan a convenient trajectory to capture the whole scene. These are significant advantages compared to vehicle or drone based systems. On the contrary, accuracy is still lower compared to TLS systems, and the operator is not able to access very small, low or narrow spaces, what would be possible with a handheld solution. However, one has to realize that these environments are also challenging for handheld scanners as was described in Sec. 4.1.4.

In recent years, multiple backpack solutions have been introduced, varying from research prototypes [84, 54] (see Akhka solution in Fig. 4.5c) to commercial products like Leica Pegasus backpack [56], ROBIN⁵, etc. shown in Fig. 4.5a and Fig. 4.5b. Solutions are usually accompanied by GNSS-IMU aiding for estimation of position and orientation. Moreover, SLAM solutions are prepared for situations when a GNSS signal is disturbed – e.g. indoors, in tunnels or urban canyons.

⁵ <https://www.3dlasermapping.com/>

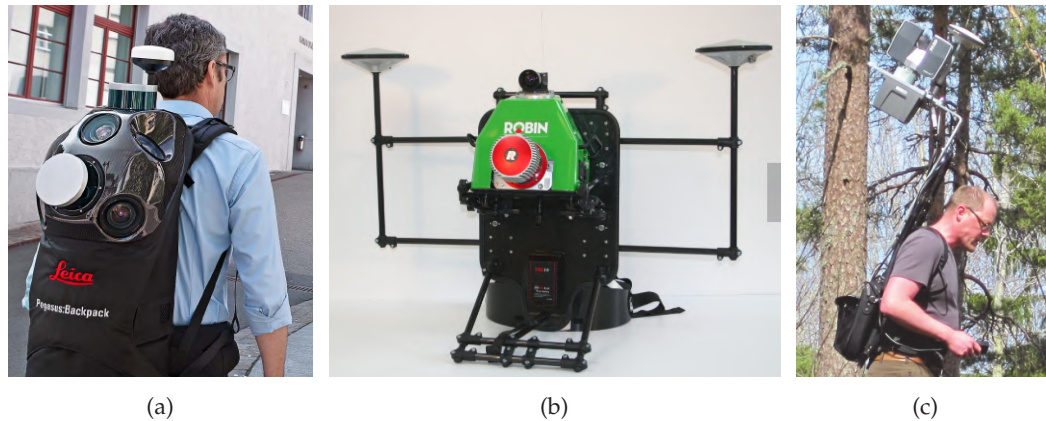


Figure 4.5: Commercial laser backpack mapping solutions Pegasus backpack (a) [56] from Leica and ROBIN (image taken from manufacturer’s website⁵) (b) from 3D laser mapping company. The solution Akhka (c) [84] was developed in Finnish Geospatial Research Institute.

Such 3D mapping backpacks can be used in multiple areas, especially when quick data collection and accuracy for distant ranges are crucial – e.g. in building information modeling (BIM) or disaster management [104]. During construction of new buildings, frequently collected BIM data in environments with difficult accessibility are used for planning, scheduling, and monitoring the changes, milestones and ongoing works. During natural disasters, the data for response management have to be quickly collected in a difficult terrain and from a safe distance.

4.3.1 Leica Pegasus Backpack

Leica Pegasus Backpack [56] combines 3D measurements from two VLP-16 Velodyne LiDARs with data from 5 RGB high resolution cameras with an optional additional lightning, GNSS and IMU sensors. After the compression, the system produces 1 GB of data per each minute. The frame of sensor platform is made of carbon fiber and the total weight of the whole backpack is approximately 12 kg, including batteries enabling 4 hours of operating time.

While a 3D reconstruction from image data only would be too sparse and inaccurate, laser intensity returns are often not sufficient to capture small changes in the texture. The combination of 3D data provided by laser scanners with RGB camera data solves the imperfections of both standalone sensors. Moreover, image data are used to improve the accuracy of estimated position.

Pegasus backpack is able to capture data within 50 m range, what is associated with the maximum range of Velodyne LiDARs. The official documentation claims 5 cm positional accuracy (with GNSS aiding) and 5 – 50 cm for 10 minutes of walking indoors, when only SLAM algorithm is used for 3D data registration. Moreover,

at least 3 loop closures are required in order to achieve this accuracy. There are also known factors influencing the performance of the system in a negative way: small rooms, stairs, uneven pavements, smooth, blank or distant surfaces. However the most significant limitation for a wider deployment is the price of around 150.000 €.

The evaluation of indoor 3D modeling without GNSS support was performed in a medieval bastation [66] consisting of 2 perpendicular underground tunnels, approximately 20 m long, joined into T-section. Terrestrial scanning (Leica C10) provided a ground truth reference, which was built by joining 9 scans from 9 different positions within the bastation. The process of data acquisition took only few minutes but required approximately half an hour for system setup and calibration. Comparing with TLS reference, the modeling by the Pegasus backpack leads to an average error of 4.2 cm with a standard deviation of 0.3 cm which corresponds with the official documentation considering a relatively small area for the mapping.

4.3.2 *ROBIN Backpack*

The solution ROBIN⁵ in Fig. 4.5b by the 3D laser mapping company aims for a higher accuracy by deploying the precise LiDAR scanner RIEGL VUX-1HA with a 3 mm precision and a dual antenna GNSS receiver for precise position and true heading estimation (error below 2 cm for position and 0.03° for heading). Moreover, 12.3 Mpix camera and IMU sensor are used as a supplementary source of data.

Unfortunately, no further information about the precision of resulting data, limitations, nor external evaluations of this product were published to our best knowledge during the time of writing this thesis. The price of this solution is approximately 220.000 € what is probably also one of the factors, why the independent experiment evaluations are missing.

4.3.3 *Akhka Backpack*

In Finnish Geospatial Research Institute, the non-commercial Akhka backpack solution Fig. 4.5c was developed for mobile mapping with improved mobility and flexibility for field analysis [52, 84]. The same hardware solution can be mounted on both the vehicle and the backpack [51] for rugged terrain, which cannot be traversed by wheel platforms.

Mapping system is accompanied with the Faro Focus laser scanner and the Novatel Flexpak6 GNSS-IMU as a positional aiding system. The scanner remains static and it serves as a rangefinder with a 305° field of view with optimal range up to 120 meters. Total weight of the backpack is 21 kg – what is almost twice as much

as for Leica Pegasus Backpack, so it could cause quite a discomfort for the operator carrying the hardware.

The Akhka solution was used in the mapping of multiple outdoor environments – river channel, meteorite impact craters and forest area. In six hours an 8 km river channel was mapped, while the operator actually walked a 22 km trajectory. For the estimation of the system precision, spherical markers were erected and their locations were precisely measured by RTK-GPS equipment. RMSE of their positions in the computed 3D map was 36 mm when compared to these precise reference positions.

Comparison with the tripod TLS scanner in terms of data acquisition speed was made at the impact craters site in Estonian island Saaremaa. The same area of multiple craters was scanned by both the TLS scanner and the Akhka backpack. While the TLS mapping took 2 days and required manual positioning of the scanner into 43 different locations, the whole site was scanned within 90 minutes with the Akhka backpack.

In the evaluation of 2000 m² forest mapping [84], Akhka backpack was evaluated by comparison with a forest model created with a UAV laser scanning by octocopter. The authors of the evaluation assume a good precision of the UAV mapping and they consider its output as a ground truth. The data of Akhka backpack are split into 10-second parts and the average misalignment of 8.7 cm with respect to the UAV model was found. The assumption that the UAV produces a correct model with high precision is quite strong and debatable, but good precision could be obtained by good GNSS reception including RTK corrections over a tree canopy, since the flying altitude was approximately 40 meters during the mapping. During mapping of a forest, the backpack system does not depend on the GNSS positioning, but rather aligns the patches (initially build using IMU data) by the ICP algorithm over the captured terrain. As a preprocessing step, the terrain (ground) is segmented and the rest of data (captured trees, bushes, etc.) is discarded. This can be considered to be a reasonable precaution since these objects are moving (in the wind), and can result in noisy and rather distracting measurements.

4.4 REQUIRED ALGORITHMS AS A PART OF MOBILE LASER SCANNING SYSTEM

The goal of this work is the development of the mobile mapping system using laser scanning. Besides the necessary hardware (LiDAR scanner, GNSS receivers, IMU sensor, ...), the core of this system consists of algorithms for processing of the laser data, especially their correct alignment into the model in a form of a consistent 3D point cloud. Moreover, the post-processing methods are designed

for data enhancement and semantic analysis. They are beneficial in further steps where the 3D model is used in practical tasks described before (see Sec. 2).

Naturally, the algorithms are bound to the properties of the hardware, and therefore to the types of scanners and sensors used. In this work, the Velodyne laser scanners are deployed as a core element of the whole system. The information for a deeper knowledge of this sensor can be found in the Sec. 3.1.2. Velodyne is chosen with respect to the quality and the amount of data it provides, relatively low price (which is moreover continuously dropping), and vast deployment in many other applications (of e.g. localisation, mapping, or self driving). This choice also affects the design of the particular algorithms described in my publications from recent years (2016 – 2018) which are included in the following chapters.

The most significant software of the whole mapping solution – *alignment of laser scans* into the 3D model – is based on the *Collar Line Segments (CLS)* algorithm described in Chapter 5. The design of the method reflects the ring structure and the sparsity of 3D LiDAR data, and it provides SoA precision of the alignment.

The supplementary algorithm for a very *fast odometry estimation*, using *convolutional neural networks*, is described in Chapter 6. It provides real time positional information, especially suitable for an online alignment and a preview of recorded data. The encoding of 3D data into a 2D representation used in this work is also designed for the ring structure of Velodyne measurements and the measurements of 3D LiDARs in general.

The *design of the whole system* is introduced in Chapter 7 including both the software and the hardware details. Necessary solutions for multiple problems are addressed here: interconnections and synchronization of the sensors, data processing and alignment of laser scans based on CLS, global optimization improving the precision of alignment, and the normalization of laser intensities improving visual recognition of the captured objects.

In the last Chapter 8, the *ground segmentation* of 3D LiDAR data by convolutional neural networks is introduced as a useful example of semantic postprocessing. This algorithm splits the 3D data into two categories – ground (the surfaces traversable by moving elements) and obstacles (trees, buildings, people, and other objects). This classification is also necessary for multiple applications like the Digital Terrain Modeling and the forest inventory documentation already presented in the Sec. 2.

Part II

CORE ALGORITHMS FOR ODOMETRY ESTIMATION

These chapters are based on the papers [98, 99].

COLLAR LINE SEGMENTS FOR FAST ODOMETRY ESTIMATION FROM VELODYNE POINT CLOUDS

5.1 ABSTRACT

We present a novel way of odometry estimation from Velodyne LiDAR point cloud scans. The aim of our work is to overcome the most painful issues of Velodyne data – the sparsity and the quantity of data points – in an efficient way, enabling more precise registration. Alignment of the point clouds which yields the final odometry is based on random sampling of the clouds using *Collar Line Segments (CLS)*. The closest line segment pairs are identified in two sets of line segments obtained from two consequent Velodyne scans. From each pair of correspondences, a transformation aligning the matched line segments into a 3D plane is estimated. By this, significant planes (ground, walls, ...) are preserved among aligned point clouds. Evaluation using the KITTI dataset shows that our method outperforms publicly available and commonly used state-of-the-art method GICP for point cloud registration in both accuracy and speed, especially in cases where the scene lacks significant landmarks or in typical urban elements. For such environments, the registration error of our method is reduced by 75% compared to the original GICP error.

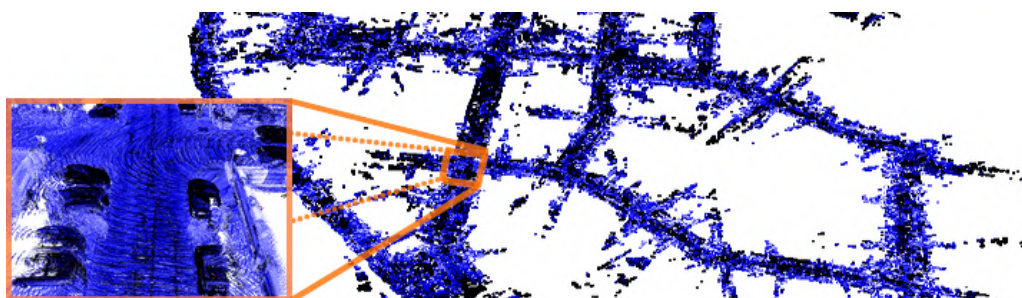


Figure 5.1: The environment map created by merging previously registered Velodyne point clouds.

5.2 INTRODUCTION

Exploration and 3D mapping of the environment surrounding a mobile robot plays a key role in robot's perception systems. Nowadays, the mapping becomes even more interesting as it is an integral part of many systems for semantic querying

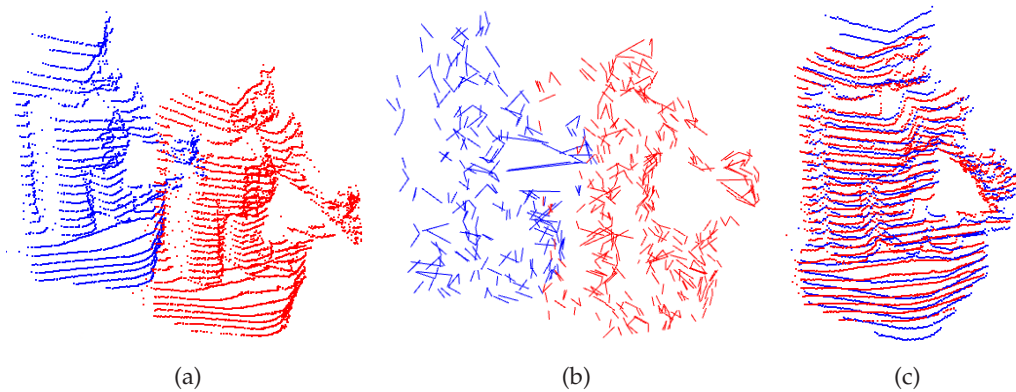


Figure 5.2: An example of the registration process. Two unaligned scans (a), sampled by line segments to produce *line clouds* (b) which are further used to estimate resulting alignment (c).

[67], semantic segmentation of scenes [36], change detection, or monitoring [4]. The source of 3D data ranges from traditional stereo cameras, RGB-D cameras (i.e. cameras enhanced by a depth sensor) extending the 2D data to 2.5D data including the spatial information as well.

Recently, numerous laser sensors – *LiDARs* (Light Detection And Ranging) – have also become popular in robotic systems. Besides the simple range finders providing only information about occupancy in a certain height around the robotic platform, sensors capturing precise 3D information of the environment, covering large horizontal and vertical field of view became available. These sensors enable modeling of the environment by precise and rich maps (Figure 5.1).

Since 2007, the *Velodyne LiDAR* sensor has become a valuable asset of vehicles attending DARPA Urban Challenge¹. This type of sensor captures the full 3D information of the environment around the LiDAR. Currently the most powerful model HDL-64E covers full 360° horizontal field and 26.8° vertical field of view and with up to 15 Hz frame rate captures over 1.3M of points per second. Example point clouds obtained by this sensor can be found in Figure 5.3.

To be used for environment mapping, Velodyne point clouds must be registered (our approach is outlined in Fig. 5.2) and odometry of the mobile platform computed, in order to estimate the pose of the sensor at the time of scanning. Traditional approaches of the point cloud registration like Iterative Closest Point (ICP) [10] or feature based methods (e.g. based on surface normals derived from the neighborhood of a point) fail for such type of data because of its *vertical sparsity* and *ring structure* as shown in Figure 5.3. Since the original ICP approach looks for a transformation by minimizing the distance of the closest points, the unaligned data in Figure 5.3 (left) would be the optimal solution due to large amount of

¹ <http://velodynelidar.com>

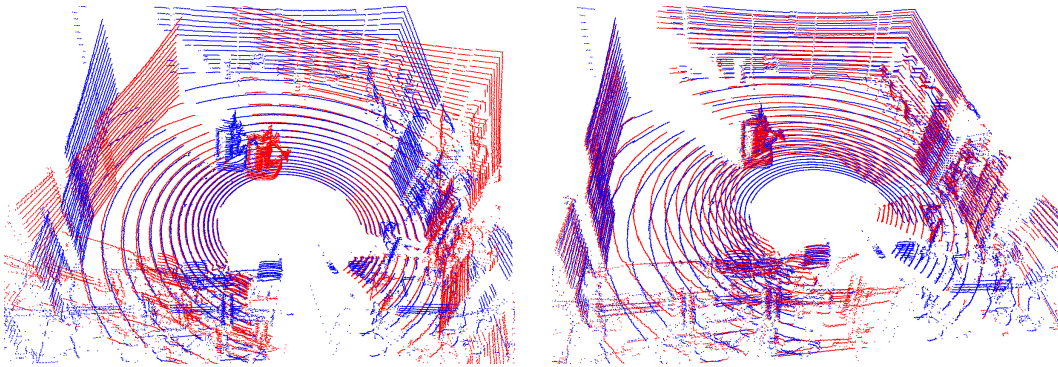


Figure 5.3: Point clouds captured by the Velodyne LiDAR scanner and associated issues. The ring structures fit to each other for unregistered data (left) which disables the convergence of typical ICP approach to proper registration. The data is also sparse (large “gaps” between rings), causing lack of spatial correspondences between scans. See the well registered scans (right) – most of the points on floor in blue scan have no proper correspondence in red scan

points in the floor rings perfectly fitting to each other. Also note in Figure 5.3 (left) that because of data sparsity, a lot of points from the source cloud (blue) miss their spatially corresponding point in the target cloud (red).

This paper presents a novel method of Velodyne point cloud registration in order to estimate odometry of the mobile platform. The main contributions of our work can be summarized in two steps of Velodyne point cloud processing. First, the typical point cloud representation is transformed into a *line cloud* by random generation of *Collar Line Segments (CLS)*. This step overcomes both the quantity and the sparsity of data. Second, we introduce an algorithm for *registration* of this line cloud representation. Our method achieves better results than publicly available state-of-the-art method GICP especially in cases when the scene lacks significant landmarks or typical “Manhattan” urban elements. Also the third contribution is making the implementation of our method and evaluation scripts publicly available².

5.3 RELATED WORK

In recent years, couple of algorithms addressing the point cloud registration problem have been published. Although they are able to register Velodyne scans, the lack of accuracy generally occurs.

Grant et al. [32] introduced a plane detection algorithm for Velodyne scans. Their method is based on the rings analysis and voting in a modified Hough space. For plane matching and computation of the final transformation, existing approach

² https://github.com/robofit/but_velodyne_lib

by Pathak et al. [80] is used. Their method was evaluated in indoor office environment and the error of estimated odometry exceeded 1 m after only ≈ 15 m run. Segmentation of the Velodyne point cloud for the registration was exploited by Douillard et al. [22]. First, the ground plane is removed from the scan by using scan voxelization. Then, the separated clusters of points are used as individual segments. The segments found in the previous step are then matched and a modified version of ICP computes the transformation by a segment-to-segment strategy. This method uses a very coarse voxel grid (20 cm resolution in experiments) which compromises the accuracy.

Accurate and effective registration of sweeping LiDAR scans has been achieved by the LOAM method [107] which was further improved by fusion with data provided by a RGB-D camera [105]. Both methods detect edges and planar points in the LiDAR scans for which a set of nonlinear equations constraining the odometry is generated. The final transformation is the result of a non-linear optimization. So far, these methods achieved the best results in the KITTI evaluation benchmark [28], but the algorithm specifics for processing the Velodyne scans have never been published nor the source codes are publicly available anymore.

Segal et al. [87] introduced a modification of the original ICP algorithm – the Generalized ICP (GICP). They replaced the standard point-to-point matching by a plane-to-plane strategy. The matching is based on covariance matrices of the local surfaces. For Velodyne data used in their evaluation, GICP reaches ± 20 cm accuracy in registration of pairwise scans. This approach also assumes that for each local group of points in the source point cloud, there is a corresponding group in the target cloud. As shown in Figure 5.3, this is not always true in the case of sparse Velodyne data. Our method drops such assumption and approximates the local surfaces in a different way – by randomly generated collar lines. We will demonstrate that this strategy yields better results in terms of average accuracy, speed, and stability for natural and non-urban scenes than GICP.

Another modification of ICP by Pandey et al. [77] benefits from fusion of omnidirectional RGB camera images with the Velodyne LiDAR scans. The prerequisite of this approach is known calibration of these two sensors. Then, the image features can be used for visual bootstrapping of generalized ICP algorithm in order to increase its robustness in case of large distances between scans (> 6 m).

Badino et al. [7] solve visual odometry estimation by using stereo images. Their approach outperformed previously published image based methods on the KITTI benchmark. Instead of the traditional approach they propose a technique that uses the history of the tracked feature points for multi-frame feature integration into a single estimate.

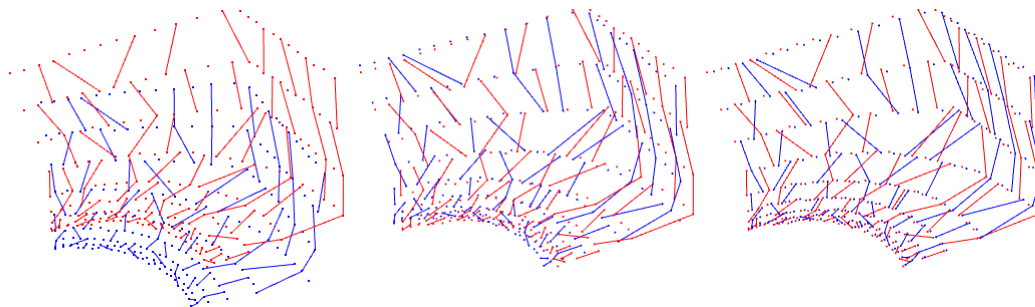


Figure 5.4: Artificially generated Velodyne point cloud of a room corner iteratively registered by our method. Initial pose (left), in progress (middle) and final alignment (right).

This paper proceeds by introducing our novel method of Velodyne data registration using Collar Line Segments, and then introducing its multi-scan modification capable to increase the resulting accuracy.

5.4 VELODYNE POINT CLOUD REGISTRATION

As we showed in the introduction by Figure 5.3, the sparsity and ring structure of Velodyne data are serious issues. In our approach, we overcome these problems by generating a set of collar line segments as shown in Figure 5.4, which naturally fill the “gaps” between rings. They also drag corresponding planes between point clouds (floor, walls, ...) together.

The proposed registration method of Velodyne point clouds consists of two main parts. Both parts are described as a general registration framework together with notes about our implementation used in the experiments. First, the point cloud to be registered is sampled into a set of line segments – a line cloud (Figure 5.2b). Second, the two line clouds are registered and 6 DoF parameters are estimated by a strategy similar to the ICP [10], using line correspondences between the clouds. The steps are described as distinct, but they can be integrated and the sampling can be done on demand from the matching and registration steps.

5.4.1 Sampling by Line Segments

Each 3D point $[x, y, z]$ of the original point cloud is transformed into the *ring coordinates* $P_{r,\alpha} = [r, \alpha]$ where $r \in \{1, \dots, N\}$ is the index of the ring the point belongs to and α is the angle within the ground plane xy :

$$\alpha = \text{atan}(y/x) \quad \alpha \in [0, 360). \quad (5.1)$$

Vertical axis z is not used due to horizontal ring layout of Velodyne LiDAR scans.

For the points of cloud \mathcal{P} in the ring coordinate system, the set of *collar line segments – line cloud* \mathcal{L}_g is generated. Eq. (5.2) describes the *generator* of line segments $l_{r,\alpha,\alpha'} = [P_{r,\alpha}, P_{r+1,\alpha'}]$ between points of consequent rings $r, r+1$:

$$G : P_{r,\alpha} \rightarrow P_{r+1,\alpha'} \cup \{\perp\} \quad P_{r,\alpha}, P_{r+1,\alpha'} \in \mathcal{P} \quad (5.2)$$

This function is required to join the points of similar angle (α is close to α') from subsequent rings so that the generated lines capture local surface properties (\perp indicates that no matching point of interest was found). Line segments are not generated for every point, but the point cloud is randomly sampled. In order to select promising line segments, this line cloud \mathcal{L}_g is further filtered by a *filter function*

$$F : \mathcal{L}_g \rightarrow \mathcal{L}, \quad \text{where } \mathcal{L} \subset \mathcal{L}_g. \quad (5.3)$$

The purpose of this function is to produce an as small as possible set of most descriptive lines.

A practical implementation of functions G and F (Eq. (5.2) and (5.3)) is depicted by Figure 5.5. Segments are generated only within one polar bin (sized φ). This function could alternatively be implemented with higher computational complexity by using a sliding window (rectangular or smooth Gaussian). Within each polar bin, a given number of line segments is randomly generated by G . Filtration (5.3) is implemented as preserving only the shortest of them. Preserving only the shortest lines discards lines formed where the rings cross an object edge as shown in Figure 5.5b. In this case, the single bin where lines were generated is split between multiple planes (object plane and the ground plane).

In our experiments on the KITTI dataset, the best results were achieved when the space was divided into 36 polar bins (per 10°), 20 lines within each polar cell were generated and 5 shortest of them were preserved. Approximately 11k collar line segments are generated for each point-cloud consisting of 64 rings of points (originally 130k of points in total).

5.4.2 Registration of line clouds

Once the Velodyne point clouds are sampled into the set of line segments, these line clouds can be registered by an iterative approach. Alternatively, the original point clouds can be repeatedly resampled.

In the target line cloud \mathcal{L}_t , the matching line segment for each element in source line cloud \mathcal{L}_s is found:

$$M : (l_s, \mathcal{L}_t) \rightarrow l_t \quad l_s \in \mathcal{L}_s, l_t \in \mathcal{L}_t \cup \{\perp\} \quad (5.4)$$

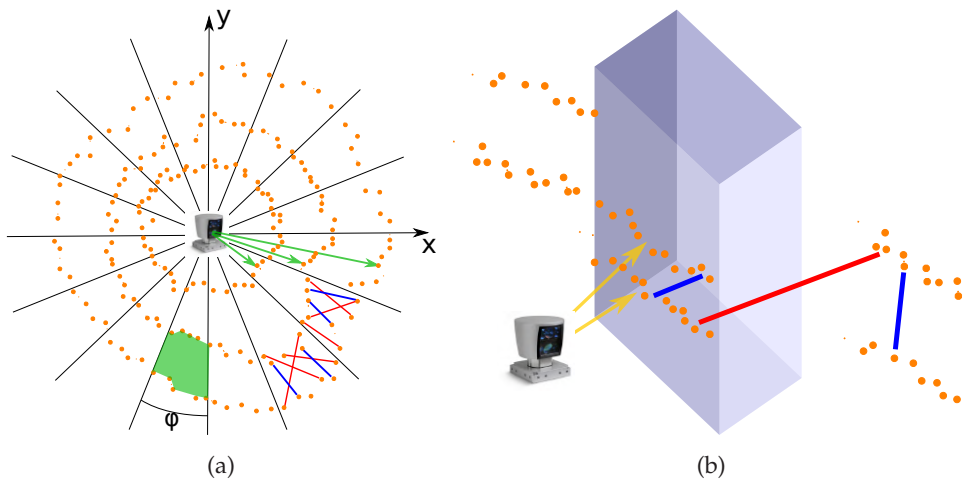


Figure 5.5: Sampling the Velodyne point cloud (orange dots) by CLS (a). The Velodyne casts rays (green arrows) each capturing one “ring” of points. The space around the sensor is divided into a polar grid of bins (one bin is green colored). Within each bin, line segments are generated by randomly generating joints between the points of consequent rings. The shortest ones (blue lines) are preserved, the others (red lines) are discarded. Demonstration of the problem when particular laser scans (i.e. measured “rings” of points) cross an object boundary (b). Preserved shorter line segments are usually generated within real 3D planes (blue lines) and rejected longer segments typically connect different planes (red one).

In our implementation used in the experiments, function M finds the line in \mathcal{L}_t whose center is closest to l_s (euclidean distance), Figure 5.6b. If its distance is above a computed threshold (mean distance), the match is not found (\perp is returned). Finding the closest line is accelerated by kD-tree.

Matching of lines by finding middle points and the building of kD-tree is done only once during the initialization. To eliminate effect of the incorrect matches, those with euclidean distance of line centers bigger than the mean distance of all found matches are discarded. We have also experimented with re-sampling the point cloud by line segments after every few iterations and continuing in registration using the last estimated transformation. This has not brought any more improvement in the registration accuracy.

In order to find the optimal transformation in the same manner as ICP approaches do using SVD (Singular Value Decomposition) [74], the corresponding 3D points P_s, P_t have to be derived for each previously matched pair l_t, l_s :

$$C : (l_s, l_t) \rightarrow (P_s, P_t), \quad (5.5)$$

where points P_s, P_t do not necessarily come from the original point clouds. The computed transformation minimizes the distance of these corresponding points. This process can be perceived that our proposed registration method resamples the original point cloud to a new point cloud in the each iteration of the original ICP

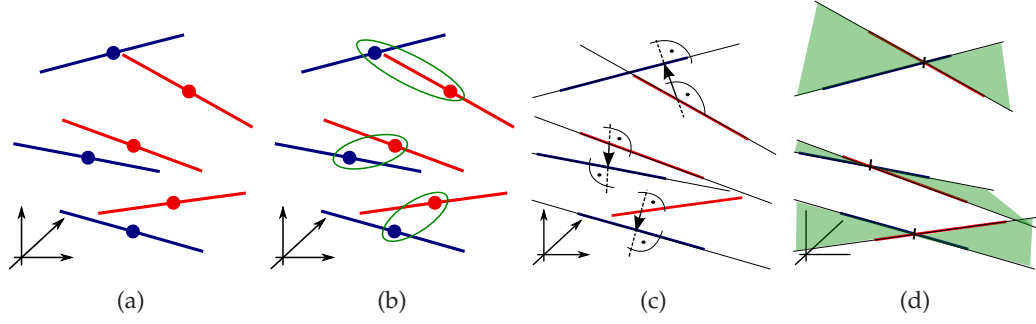


Figure 5.6: Registration of “line clouds” shown on three pairs of matching lines. The middle points of segments are found (a) and used for matching the lines by closest centers (b). The segments are extended into infinite lines and closest points of the matching line pairs are found (arrows in (c)). These correspondences define the transformation which “pushes” the two matched lines into a single (green) plane (d).

algorithm. Resampling is done so that it overcomes the problem of data sparsity capturing the properties of local surface by collar line segments.

The corresponding points (5.5) are found such that the estimated transformation causes matching lines to cross. This simulates fitting the corresponding planes between point clouds as has been previously shown in Figure 5.4. The line segments are extended to infinite lines, and closest points – pair (P_s, P_t) – are found as follows.

Assuming the line of the source line cloud l_s and the line of the target cloud l_t is given by 3D point \hat{P}_s and vector \mathbf{u}_s (\hat{P}_t and \mathbf{u}_t for the target line, respectively):

$$l_s : X = \hat{P}_s + \mathbf{u}_s \cdot t_s; \quad t_s \in (-\infty, \infty) \quad (5.6)$$

$$l_t : X = \hat{P}_t + \mathbf{u}_t \cdot t_t; \quad t_t \in (-\infty, \infty), \quad (5.7)$$

the closest points P_s, P_t between these two lines are [76]:

$$P_s = \hat{P}_s + \mathbf{u}_s \cdot t_s^c \quad P_t = \hat{P}_t + \mathbf{u}_t \cdot t_t^c \quad (5.8)$$

$$\text{where} \quad t_s^c = \frac{be - cd}{ac - b^2} \quad t_t^c = \frac{ae - bd}{ac - b^2} \quad (5.9)$$

$$a = \mathbf{u}_s \cdot \mathbf{u}_s \quad b = \mathbf{u}_s \cdot \mathbf{u}_t \quad c = \mathbf{u}_t \cdot \mathbf{u}_t \quad (5.10)$$

$$d = \mathbf{u}_s \cdot \mathbf{w} \quad e = \mathbf{u}_t \cdot \mathbf{w} \quad \mathbf{w} = \hat{P}_s - \hat{P}_t \quad (5.11)$$

Operation $\mathbf{u} \cdot \mathbf{v}$ represents the dot product of two vectors and auxiliary variables a, b, c, d, e are scalars, \mathbf{w} is a vector.

5.4.3 Prediction of Transformation from Previous Frames

Since the LiDAR scanner is commonly mounted on a moving vehicle, the physical constraints like momentum are bound to the vehicle odometry. Thus the previously

computed transformation can be used for prediction and initialization of the next pose estimation.

Traditional solutions of this problem use non-linear predictors like Extended Kalman Filter (EKF) [35]. In our experiments, linear prediction using last N sets of transformation parameters (6DoF) brings significant improvement in prediction of odometry. We have also experimentally compared it with EKF while obtaining almost the same results. So finally, for the sake of speed, we decided to use the simple linear prediction.

Let T_i be the transformation between two consequent scans P_i and P_{i+1} taken by the LiDAR scanner at times i and $(i + 1)$. The transformation is a 6DoF vector $[t_x, t_y, t_z, r_r, r_p, r_y]$. Then, the initial prediction T_{init} of the transformation at time i can be computed as:

$$T_{init} = \frac{2}{N(N+1)} \sum_{j=1}^N (N-j+1) T_{i-j}. \quad (5.12)$$

5.4.4 Processing of Multiple Scans

Badino et al. [7] improved the estimated transformations by multi-frame feature integration. Similarly, in our approach, the history of LiDAR scans and computed transformations are used to improve the odometry precision. New scan P_{i+1} is additionally registered against H previous historical records $P_{i-1}, P_{i-2}, \dots, P_{i-H}$ (Figure 5.7).

In the first experiments, these multiple transformations estimated for each Velodyne scan of data sequence were used to build pose graph further optimized by a nonlinear solver (SLAM++ [39] was used). These experiments resulted in inaccurate results due to sensitivity of the optimizer to noise so we proposed another finally more robust solution described below.

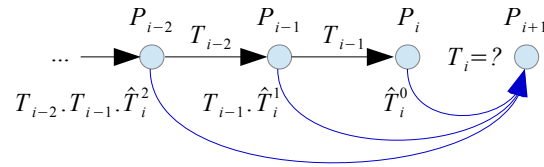


Figure 5.7: New LiDAR scan is registered against multiple previous records $P_{i-1}, P_{i-2}, \dots, P_{i-H}$ and multiple transformations (blue edges) are estimated.

As the previous transformations $T_{i-1}, T_{i-2}, \dots, T_{i-H}$ are known, using their inverse, transformations $\hat{T}_i^1, \dots, \hat{T}_i^H$ (see Figure 5.7) can be derived. Assuming the normal Gaussian distribution of the noise they suffer from, the resulting transfor-

mation T_i can be obtained as a mean of these values. More details can be found in Algorithm 5.1.

Algorithm 5.1: Registration against H previous scans for noise reduction

```

1:  $\hat{T}_i^0 = \text{REGISTER}(P_i, P_{i+1}, T_{\text{init}})$ 
2:  $T_{\text{inv}} = \text{Identity}$ 
3: for  $j = 1$  to  $H$  do
4:    $T_{\text{inv}} = T_{\text{inv}} * \text{INVERT}(T_{i-j})$ 
5:    $S = \{T_{\text{inv}} * p \mid p \in P_{i-j}\}$ 
6:    $\hat{T}_i^j = \text{REGISTER}(S, P_{i+1}, \hat{T}_i^{j-1})$ 
7: end for
8:  $T_i = \frac{1}{H} \sum_{j=0}^H \hat{T}_i^j$ 

```

5.5 EXPERIMENTAL EVALUATION

For evaluation of the odometry estimation, we used the publicly available KITTI odometry dataset [28]. It consists of 22 independent sequences captured during driving in and outside the city of Karlsruhe. The following data sequences are available for each run: stereo gray-scale and color camera images, point clouds captured by Velodyne LiDAR, mutual calibration of sensors, and ground truth data (for the first 11 runs only) obtained by GPS/OXTS. For the odometry estimation, only the Velodyne data was used in our case.

Since our work presents a single standalone component for point cloud registration rather than a complex system, we have chosen GICP method of similar complexity for comparison. Until the other modules (e.g. dynamic objects filter, visual loop closure, fusion with RGB data) are not involved, we do not find KITTI benchmark suitable for objective evaluation.

5.5.1 Evaluation metric

The quality of point cloud registration and the odometry estimation was evaluated by using the first 11 sequences of the KITTI dataset for which the ground truth data are publicly available. Since this data was obtained by a GPS sensor which yields significant imprecision in the vertical position estimation (z axis), only horizontal coordinates (xy plane) are used for the error estimation. The error of a single point cloud registration is then defined as

$$e_i = \sqrt{(t_i.x - g_i.x)^2 + (t_i.y - g_i.y)^2}, \quad (5.13)$$

where t_i is estimated position of i^{th} LiDAR frame with respect to the previous frame $i - 1$ and g_i is the ground truth data. The error of whole sequence (N frames) is defined as

$$e = \frac{1}{N} \sum_{i=1}^N e_i. \quad (5.14)$$

5.5.2 Results: Precision of Registration on the KITTI Dataset

In this section, we compare our method with publicly available state of the art method for point cloud registration GICP [87]. Since our registration process was improved by prediction described in Section 5.4.3, the same prediction was used also for GICP to keep comparison fair. Apart of this, the default parameters of the test application³ were used.

Seq. #	Length [frames]	GICP	CLS	CLS-M
0	4540	0.0315	0.0622	0.0529
1	1100	0.4215	0.0960	0.0685
2	4660	0.3347	0.0858	0.1144
3	800	0.0218	0.0275	0.0239
4	270	0.0497	0.0316	0.0394
5	2760	0.0228	0.0726	0.0413
6	1100	0.0362	0.0327	0.0383
7	1100	0.0132	0.0222	0.0117
8	4070	0.0626	0.1001	0.0643
9	1590	0.0530	0.0688	0.0583
10	1200	0.0177	0.0464	0.0369
weighted avg	2108	0.1153	0.0712	0.0624

Table 5.1: Odometry estimation error for data sequences in the KITTI dataset for which the ground truth data are publicly available. The average error is the weighted average of sequence errors where the weight is the length of sequence. The results referred as CLS-M were obtained while processing multiple frames as described in Section 5.4.4.

The sum of registration error e yielded by GICP and our method can be found in Table 5.1. Comparison of the 3rd and 4th column of Table 5.1 shows that our method preserves stability among different KITTI data sequences captured in different en-

³ http://www.robots.ox.ac.uk/~avsegal/generalized_icp.html

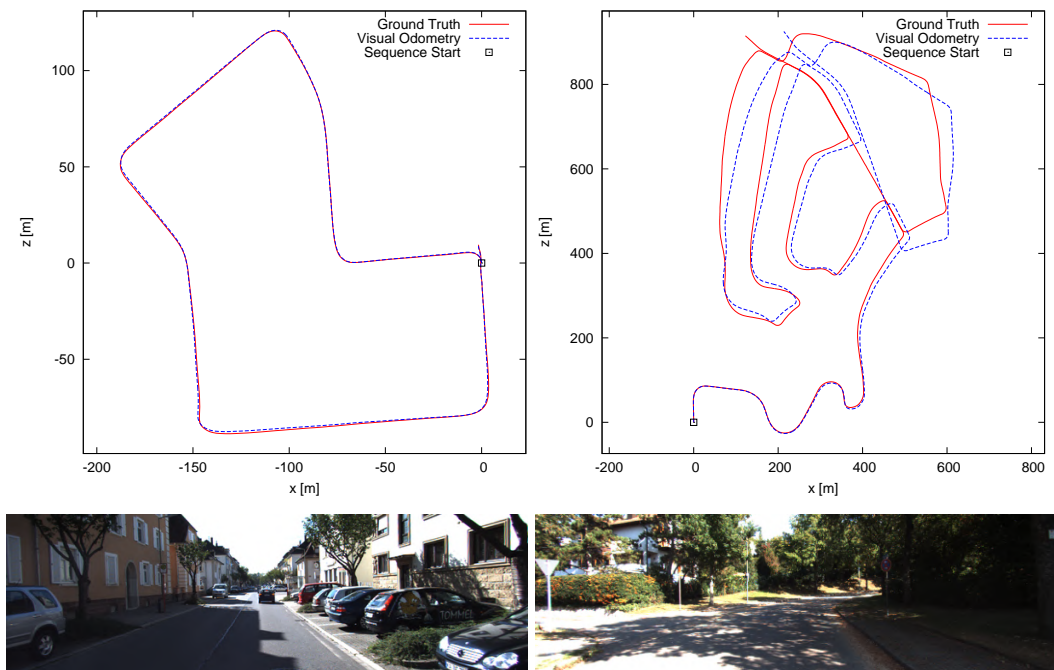


Figure 5.8: The best (left, seq. #7, error 0.0117) and the worst (right, seq. #2, error 0.1144) estimated odometry using our method. Bottom row: Typical images from the sequence (images themselves are not used for processing).

vironments. Our method outperforms GICP especially in challenging sequences of non-Manhattan environment outside of the city – highway (seq. #1) and rural area (seq. #2) – where GICP approach fails. For other sequences, our method reaches comparable results. In average (weighted by sequence length), our method yields better accuracy of the estimated odometry.

The last column of Table 5.1 (CLS-M) contains the results of our method further improved by processing of multiple (i.e. 10) scans as described in Section 5.4.4. Since this modification requires the registration to be repeated multiple times and each single GICP registration is quite time consuming, each scan is registered only with a single predecessor.

The best results were obtained when processing sequence #7 which was recorded in a Manhattan-like urban environment as is shown in Figure 5.8, left. On the other hand, data sequence #2 yields the worst results. It was captured outside the city center and besides the road, it captures mostly natural phenomena (trees, bushes, etc.) as shown in Figure 5.8, right. Compared to our approach, GICP is not able to handle these natural objects in sequence #2 and estimate the vehicle odometry with a reasonable error.

The GICP method is also failing (42cm error) on data sequence #1, which was recorded outside the city on the open highway (Figure 5.9) and the LiDAR captures only the road, without any other significant landmarks. The largest drift appears when the tested car (sensor platform) is overtaken by another vehicle. Our method



Figure 5.9: Images from KITTI data sequence #1 where GICP approach fails but our method preserves accuracy. Images show the challenging situation when over-turning car appear as confusing landmarks on otherwise empty highway.

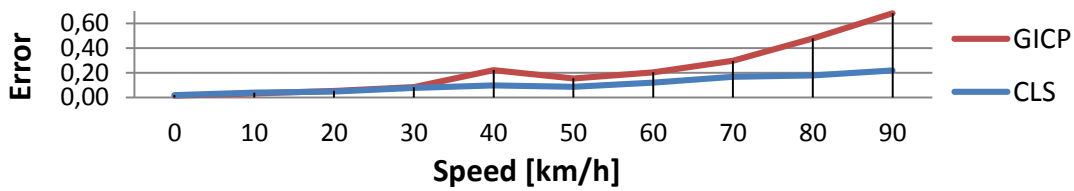


Figure 5.10: Average error for different driving speed. High speed (over 50km/h) was simulated by omitting every 2nd frame.

is able to handle these situations and it estimates odometry of feasible accuracy (error below 7cm for CLS-M).

Evaluation for different driving speeds (Figure 5.10) shows that both methods preserve equal precision for speeds below 30 km/h. For higher speeds (especially over 70 km/h), our method outperforms GICP.

5.5.3 Finding Optimal Parameters

Graphs in Figure 5.11 show the results for different parameters setup of the sampling and registration process. The best setup regarding the registration accuracy, used also for the evaluation above, generates 20 line segments per angular bin, preserves 5 shortest ones of them, uses 3 latest measurements for prediction and 10 registrations against previous scans for noise reduction.

5.5.4 Speed

As shown in Table 5.2, for the registration of a pair of Velodyne scans (no special optimization or parallelization), achieves approximately 10× better frame rate comparing to the publicly available implementation of GICP.

Table 5.2 also shows, that frame rate of our multi-scan modification falls proportionally to the number of previous scans used due to the multiple registrations

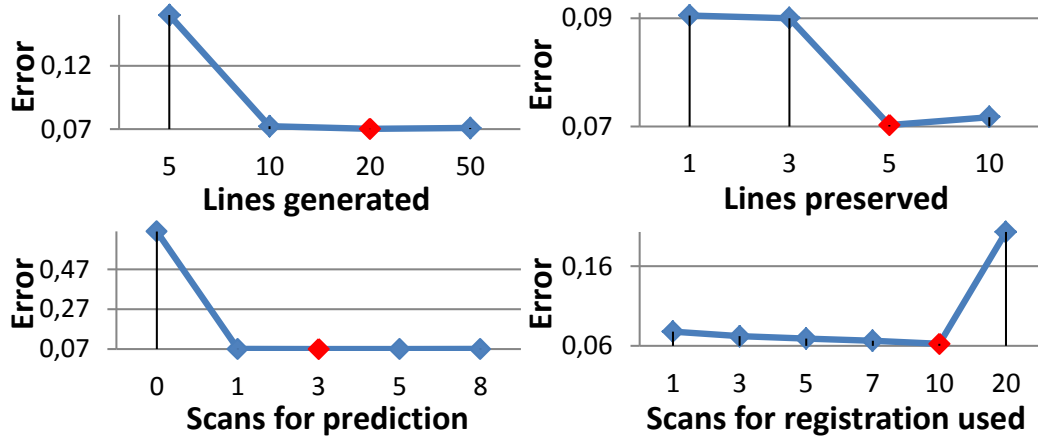


Figure 5.11: Tuning of the registration parameters. Values of the lowest error (5.14) are highlighted in red. Graphs show the error trend based on the number of lines generated and preserved in the each polar bin, the number of previous registrations used for prediction and the number of frames used for multi-scan approach.

	GICP	CLS	CLS-M
Avg. time per frame [s]	25.68	2.36	28.56

Table 5.2: Comparison of time consumption. In CLS-M, the registration was performed against 10 previous scans.

against the previous records. This modification is generally applicable to improve the registration accuracy and so it would also lower the frame rate when applied to the GICP approach.

5.6 CONCLUSION

This paper introduces a novel way of Velodyne point cloud representation using the Collar Line Segments (CLS), the algorithm of “line clouds” registration, and its further improvement by processing multiple preceding scans. These algorithms were used for Velodyne LiDAR scans registration of the KITTI dataset and compared to the state-of-the-art technique Generalized ICP. The new method achieves better results in terms of registration accuracy, especially for challenging situations like natural scenes or lack of relevant landmarks. Considering the time consumption, our approach is approximately $10\times$ faster. Using further proposed improvements, the registration reaches 6 cm weighted average registration error on the KITTI evaluation data sequences.

In the future, visual loop detection, its closure as well as detection and exclusion of disturbing moving objects can be valuable assets in further improvement of the accuracy of the estimated odometry.

CNN FOR IMU ASSISTED ODOMETRY ESTIMATION USING VELODYNE LIDAR

6.1 ABSTRACT

We introduce a novel method for odometry estimation using convolutional neural networks from 3D LiDAR scans. The original sparse data are encoded into 2D matrices for the training of proposed networks and for the prediction. Our networks show significantly better precision in the estimation of translational motion parameters comparing with state of the art method LOAM, while achieving real-time performance. Together with IMU support, high quality odometry estimation and LiDAR data registration is realized. Moreover, we propose alternative CNNs trained for the prediction of rotational motion parameters while achieving results also comparable with state of the art. The proposed method can replace wheel encoders in odometry estimation or supplement missing GPS data, when the GNSS signal absents (e.g. during the indoor mapping). Our solution brings real-time performance and precision which are useful to provide online preview of the mapping results and verification of the map completeness in real time.

6.2 INTRODUCTION

Recently, many solutions for indoor and outdoor *3D mapping* using LiDAR sensors have been introduced, proving that the problem of *odometry estimation* and *point cloud registration* is relevant and solutions are demanded. The Leica¹ company intro-

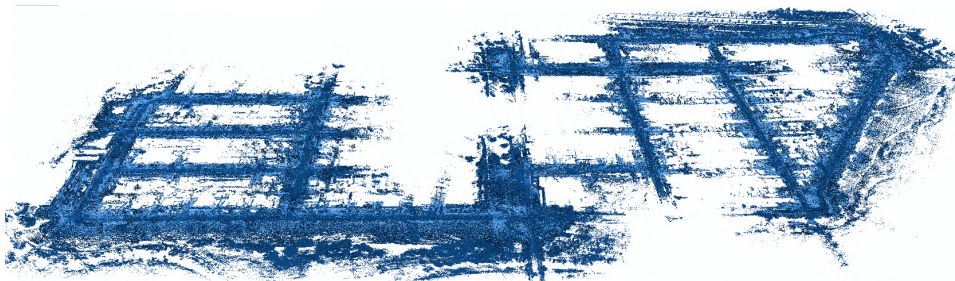


Figure 6.1: Example of LiDAR point clouds registered by CNN estimated odometry. Sequence 08 of KITTI dataset [28] is presented with rotations provided by IMU.

¹<http://leica-geosystems.com>

duced Pegasus backpack equipped with multiple Velodyne LiDARs, RGB cameras, including IMU and GNSS sensors supporting the point cloud alignment. Geoslam² uses simple rangefinder accompanied with IMU unit in their hand-held mapping ZEB products. Companies like LiDARUSA and RIEGL³ build their LiDAR systems primarily targeting outdoor ground and aerial mapping. Such systems require readings from IMU and GNSS sensors in order to align captured point clouds. These requirements restrict the systems to be used for mapping the areas where GNSS is available.

Another common property of these systems is offline processing of the recorded data for building the accurate 3D maps. The operator is unable to verify whether the whole environment (building, park, forest, ...) is correctly captured and whether there are no parts missing. This is a significant disadvantage, since the repetition of the measurement process can be expensive and time demanding. Although the orientation can be estimated online and quite robustly by the IMU unit, *precise position information* requires reliable GPS signal readings including the online corrections (differential GPS, RTK, ...). Since these requirements are not met in many scenarios (indoor scenes, forests, tunnels, mining sites, etc.), the less accurate methods, like odometry estimation from wheel platform encoders, are commonly used.

We propose an alternative solution – a frame to frame *odometry estimation* using *convolutional neural networks* from LiDAR point clouds. Similar deployments of CNNs has already proved to be successful in ground segmentation [96] and also in vehicle detection [57] in sparse LiDAR data.

The main contribution of our work is fast, real-time and precise estimation of positional motion parameters (translation) outperforming the state-of-the-art results. We also propose alternative networks for full 6DoF visual odometry estimation (including rotation) with results comparable to the state of the art. Our deployment of convolutional neural networks for odometry estimation, together with existing methods for object detection [57] or segmentation [96] also illustrates general usability of CNNs for this type of *sparse LiDAR data*.

6.3 RELATED WORK

The published methods for visual odometry estimation can be divided into two groups. The first one consists of direct methods computing the motion parameters in a single step (from image, depth or 3D data). Comparing with the second group of iterative methods, direct methods have a potential of better time performance.

² <https://geoslam.com>

³ <https://www.lidarusa.com>, <http://www.riegl.com>

Unfortunately, to our best knowledge, no direct method for odometry estimation from LiDAR data have been introduced so far.

Since the introduction of notoriously known Iterative Closest Point (ICP) algorithm [16, 9], many modifications of this approach were developed. In all derivatives, two basic steps are iteratively repeated until the termination conditions are met: matching the elements between 2 point clouds (originally the points were used directly) and the estimation of target frame transformation, minimizing the error represented by the distance of matching elements. This approach assumes that there actually exist matching elements in the target cloud for a significant amount of basic elements in the source point cloud. However, such assumption does not often hold for sparse LiDAR data and causes significant inaccuracies.

Grant [32] used planes detected in Velodyne LiDAR data as the basic elements. The planes are identified by analysis of depth gradients within readings from a single laser beam and then by accumulating in a modified Hough space. The detected planes are matched and the optimal transformation is found using previously published method [80]. Their evaluation shows the significant error ($\approx 1\text{m}$ after 15m run) when mapping indoor office environment. Douillard et al. [22] used the ground removal and clustering remaining points into the segments. The transformation estimated from matching the segments is only approximate and it is probably compromised by using quite coarse (20cm) voxel grid.

Generalized ICP (GICP) [88] replaces the standard point-to-point matching by the plane-to-plane strategy. Small local surfaces are estimated and their covariance matrices are used for their matching. When using Velodyne LiDAR data, the authors achieved $\pm 20\text{ cm}$ accuracy in the registration of pairwise scans. In our evaluation [98] using KITTI dataset [28], the method yields average error 11.5cm in the frame-to-frame registration task. The robustness of GICP drops in case of large distance between the scans ($> 6\text{m}$). This was improved by employing visual SIFT features extracted from omnidirectional Ladybug camera [77] and the codebook quantization of extracted features for building sparse histogram and maximization of mutual information [78].

Bose and Zlot [12] are able to build consistent 3D maps of various environments, including challenging natural scenes, deploying visual loop closure over the odometry provided by inaccurate wheel encoders and the orientation by IMU. Their robust place recognition is based on Gestalt keypoint detection and description [14]. Deployment of our CNN in such system would overcome the requirement of the wheel platform and the same approach would be useful for human-carried sensory gears (Pegasus, ZEB, etc.) as mentioned in the introduction.

Our previous work [98] proposed sampling the 3D LiDAR point clouds by *Collar Line Segments (CLS)* to overcome data sparsity. First, the original Velodyne point

cloud is split into polar bins. The line segments are randomly generated within each bin, matched by nearest neighbor search and the resulting transformation fits the matched lines into the common planes. The CLS approach was also evaluated using the KITTI dataset and achieves 7cm error of the pairwise scan registration. Splitting into polar bins is also used in this work for encoding the 3D data to 2D representation (see Sec. 6.4.1).

The top ranks in KITTI Visual odometry benchmark [28] are for last years occupied by LiDAR Odometry and Mapping (LOAM) [107] and Visual LOAM (V-LOAM) [105] methods. Planar and edge points are detected and used to estimate the optimal transformation in two stages: fast scan-to-scan and precise scan-to-map. The map consists of keypoints found in previous LiDAR point clouds. Scan-to-scan registration enables real-time performance and only each n -th frame is actually registered within the map.

The implementation was publicly released under BSD license but withdrawn after being commercialized. The original code is accessible through the documentation⁴ and we used it for evaluation and comparison with our proposed solution. In our experiments, we were able to achieve superior accuracy in the estimation of the translation parameters and comparable results in the estimation of full 6DoF (degrees of freedom) motion parameters including rotation. In V-LOAM [105], the original method was improved by fusion with RGB data from omnidirectional camera and authors also prepared method which fuses LiDAR and RGB-D data [106].

The encoding of 3D LiDAR data into the 2D representation, which can be processed by convolutional neural network (CNN), were previously proposed and used in the ground segmentation [96] and the vehicle detection [57]. We use a similar CNN approach for quite different task of visual odometry estimation. Besides the precision and the real-time performance, our method also contributes as the illustration of general usability of CNNs for sparse LiDAR data. The key difference is the amount and the ordering of input data processed by neural network (described in next chapter and Fig. 6.3). While the previous methods [57, 96] process only a single frame, in order to estimate the transformation parameters precisely we process multiple frames simultaneously.

6.4 METHOD

Our *goal* is the estimation of transformation $\mathbf{T}_n = [t_n^x, t_n^y, t_n^z, r_n^x, r_n^y, r_n^z]$ representing the 6DoF motion of a platform carrying LiDAR sensor, given the current LiDAR frame \mathbf{P}_n and N previous frames $\mathbf{P}_{n-1}, \mathbf{P}_{n-2}, \dots, \mathbf{P}_{n-N}$ in form of point

⁴http://docs.ros.org/indigo/api/loam_velodyne/html/files.html

clouds. This can be written as a mapping Θ from the point cloud domain \mathbb{P} to the domain of motion parameters (6.1) and (6.2). Each element of the point cloud $\mathbf{p} \in \mathbf{P}$ is the vector $\mathbf{p} = [p^x, p^y, p^z, p^r, p^i]$, where $[p^x, p^y, p^z]$ are its coordinates in the 3D space (right, down, front) originating at the sensor position. p^r is the index of the laser beam that captured this point, which is commonly referred as the “ring” index since the Velodyne data resembles the rings of points shown in Fig. 6.2 (top, left). The laser intensity measurement is denoted as p^i .

$$\mathbf{T}_n = \Theta(\mathbf{P}_n, \mathbf{P}_{n-1}, \mathbf{P}_{n-2}, \dots, \mathbf{P}_{n-N}) \quad (6.1)$$

$$\Theta : \mathbb{P}^{N+1} \rightarrow \mathbb{R}^6 \quad (6.2)$$

6.4.1 Data encoding

We represent the mapping Θ by convolutional neural network. Since we use sparse 3D point clouds and convolutional neural networks are commonly designed for dense 1D and 2D data, we adopt previously proposed [57, 96] *encoding* \mathcal{E} (6.3) of 3D LiDAR data to dense matrix $\mathbf{M} \in \mathbb{M}$. These encoded data are used for actual training the neural network implementing the mapping $\tilde{\Theta}$ (6.4, 6.5).

$$\mathbf{M} = \mathcal{E}(\mathbf{P}); \quad \mathcal{E} : \mathbb{P} \rightarrow \mathbb{M} \quad (6.3)$$

$$\mathbf{T}_n = \tilde{\Theta}(\mathcal{E}(\mathbf{P}_n), \mathcal{E}(\mathbf{P}_{n-1}), \dots, \mathcal{E}(\mathbf{P}_{n-N})) \quad (6.4)$$

$$\tilde{\Theta} : \mathbb{M}^{N+1} \rightarrow \mathbb{R}^6 \quad (6.5)$$

Each element $\mathbf{m}_{r,c}$ of the matrix \mathbf{M} encodes points of *the polar bin* $\mathbf{b}_{r,c} \subset \mathbf{P}$ (6.6) as a vector of 3 values: depth and vertical height relative to the sensor position, and the intensity of laser return (6.7). Since the multiple points fall into the same bin, the representative values are computed by averaging. On the other hand, if a polar bin is empty, the missing element of the resulting matrix is interpolated from its neighbourhood using linear interpolation.

$$\mathbf{m}_{r,c} = \varepsilon(\mathbf{b}_{r,c}); \quad \varepsilon : \mathbb{P} \rightarrow \mathbb{R}^3 \quad (6.6)$$

$$\varepsilon(\mathbf{b}_{r,c}) = \frac{\sum_{\mathbf{p} \in \mathbf{b}_{r,c}} [p^y, \|p^x, p^z\|_2, p^i]}{|\mathbf{b}_{r,c}|} \quad (6.7)$$

The indexes r, c denote both the row (r) and the column (c) of the encoded matrix and the polar cone (c) and the ring index (r) in the original point cloud (see Fig. 6.2). Dividing the point cloud into the polar bins follows same strategy as

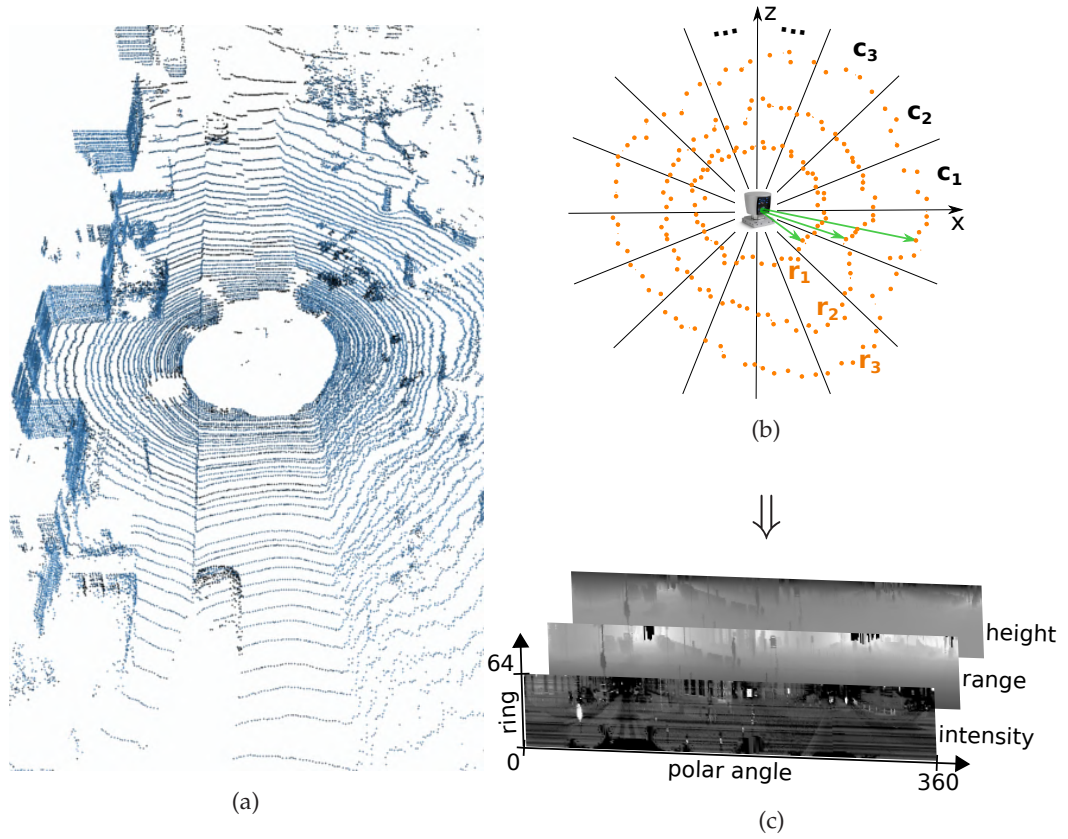


Figure 6.2: Transformation of the sparse Velodyne point cloud (a), (b) into the multi-channel dense matrix (c). Each row represents measurements of a single laser beam (single ring r_1, r_2, r_3, \dots) done during one rotation of the sensor. Each column contains measurements of all 64 laser beams captured within the specific rotational angle interval (polar cone c_1, c_2, c_3, \dots).

described in our previous work [98]. Each polar bin is identified by the polar cone $\varphi(\cdot)$ and the ring index p^r .

$$\mathbf{b}_{r,c} = \{\mathbf{p} \in \mathbf{P} \mid p^r = r \wedge \varphi(\mathbf{p}) = c\} \quad (6.8)$$

$$\varphi(\mathbf{p}) = \left\lfloor \frac{\text{atan}\left(\frac{p^z}{p^x}\right) + 180^\circ}{\frac{360^\circ}{R}} \right\rfloor \quad (6.9)$$

where R is horizontal angular resolution of the polar cones. In our experiments we used the resolution $R = 1^\circ$ (and 0.2° in the classification formulation described below).

6.4.2 From regression to classification

In our preliminary experiments, we trained the network $\tilde{\Theta}$ estimating full 6DoF motion parameters. Unfortunately, such networks provided very inaccurate results. The output parameters consist of two different motion modalities – rotation and translation $\mathbf{T}_n = [\mathbf{R}_n | \mathbf{t}_n]$ – and it is difficult to determine (or weight) the impor-

tance of angular and positional differences in backward computation. So we decided to split the mapping into the estimation of rotation parameters $\tilde{\Theta}_R$ (6.10) and translation $\tilde{\Theta}_t$ (6.11).

$$\mathbf{R}_n = \tilde{\Theta}_R(\mathbf{M}_n, \mathbf{M}_{n-1}, \dots, \mathbf{M}_{n-N}) \quad (6.10)$$

$$\mathbf{t}_n = \tilde{\Theta}_t(\mathbf{M}_n, \mathbf{M}_{n-1}, \dots, \mathbf{M}_{n-N}) \quad (6.11)$$

$$\tilde{\Theta}_R : \mathbb{M}^{N+1} \rightarrow \mathbb{R}^3; \quad \tilde{\Theta}_t : \mathbb{M}^{N+1} \rightarrow \mathbb{R}^3 \quad (6.12)$$

The implementation of $\tilde{\Theta}_R$ and $\tilde{\Theta}_t$ by convolutional neural network is shown in Fig. 6.3. We use *multiple input frames* in order to improve stability and robustness of the method. Such multi-frame approach was also successfully used in our previous work [98] and comes from assumption, that motion parameters are similar within small time window (0.1 – 0.7s in our experiments below).

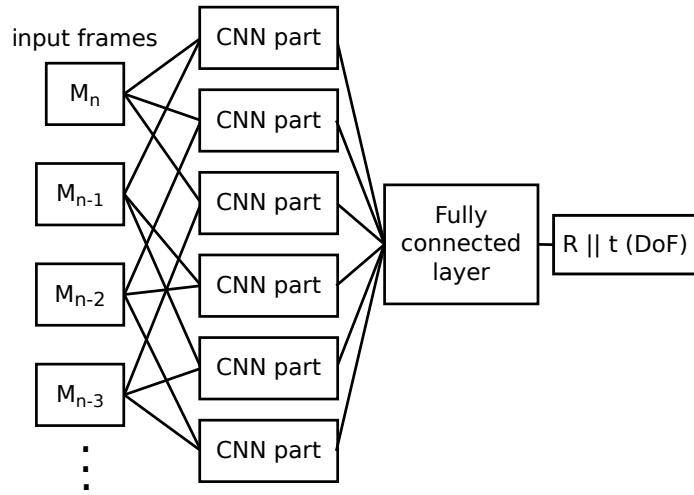


Figure 6.3: Topology of the network implementing $\tilde{\Theta}_R$ and $\tilde{\Theta}_t$. All combinations of current M_n and previous M_{n-1}, M_{n-2}, \dots frames (3 previous frames in this example) are pairwise processed by the same CNN part (see structure in Fig. 6.4) with shared weights. The final estimation of rotation or translation parameters is done in fully connected layer joining the outputs of CNN parts. For training, the euclidean loss was used.

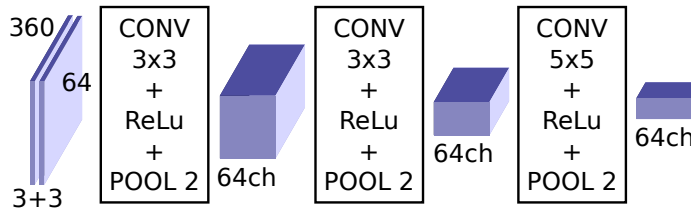


Figure 6.4: Topology of shared CNN component (denoted as “CNN part” in Fig. 6.3) for processing the pairs of encoded LiDAR frames. The topology is quite shallow with small convolutional kernels, ReLu nonlinearities and max pooling after each convolutional layer. The output blob size is $45 \times 8 \times 64$ ($W \times H \times Ch$).

The idea behind proposed topology is the expectation that shared CNN components for pairwise frame processing will estimate the motion map across the input frame space (analogous to the optical flow in image processing). The final estimation of rotation or translation parameters is performed in the fully connected layer joining the outputs of pure CNN components.

Splitting the task of odometry estimation into two separated networks, sharing the same topology and input data, significantly improved the results – especially the precision of translation parameters. However, precision of the predicted rotation was still insufficient. The rotation is represented by Euler angles, but the experiments with quaternions and axis-angles were also performed with no improvement. The original formulations of our goal (6.1) can be considered as solving the *regression task*. However, the space of possible rotations between consequent frames is quite small for reasonable data (distribution of rotations for KITTI dataset can be found in Fig. 6.5). Such small space can be densely sampled and we can reformulate this problem to the *classification task* (6.13, 6.14).

$$R = \arg \max_{i \in \{0, \dots, K-1\}} \Gamma(R_i(\mathbf{M}_n), \mathbf{M}_{n-1}) \quad (6.13)$$

$$\Gamma : \mathbb{M}^2 \rightarrow \mathbb{R} \quad (6.14)$$

where $R_i(\mathbf{M}_n)$ represents rotation R_i of the current LiDAR frame \mathbf{M}_n and $\Gamma(\cdot)$ estimates the probability of R_i to be the correct rotation between the frames \mathbf{M}_n and \mathbf{M}_{n-1} .

Similar approach was previously used in the task of human age estimation [85]. Instead of training the CNN to estimate the age directly, the image of person is classified to be 0, 1, ..., 100 years old.

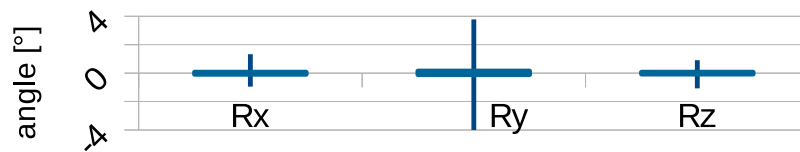


Figure 6.5: Rotations (min-max) around x, y, z axis in training data sequences of KITTI dataset.

The implementation of Γ comparator by a convolutional network can be found in Fig. 6.6. In next sections, this network will be referred as *classification CNN* while the original one will be referred as *regression CNN*. We have also experimented with the classification-like formulation of the problem using the original CNN topology (Fig. 6.3) without sampling and applying the rotations, but this did not bring the improvement.

For the classification network we experienced better results when wider input (horizontal resolution $R = 0.2^\circ$) is provided to the network. This affected also properties of the convolutional component (the CNN part), where wider convolution

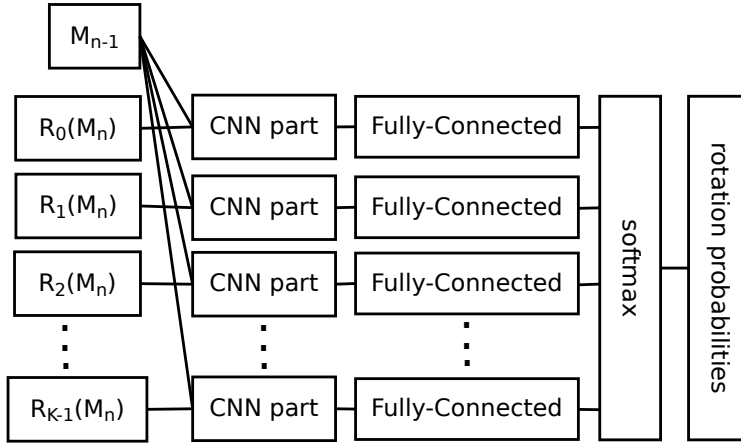


Figure 6.6: Modification of original topology (Fig. 6.3) for precise estimation of rotation parameters. Rotation parameter space (each axis separately) is densely sampled into K rotations R_0, R_1, \dots, R_{K-1} and applied to current frame M_n . CNN component and fully connected layer are trained as comparators Γ with previous frame M_{n-1} estimating probability of given rotation. All CNN parts (structure in Fig. 6.7) and fully connected layers share the weights of the activations.

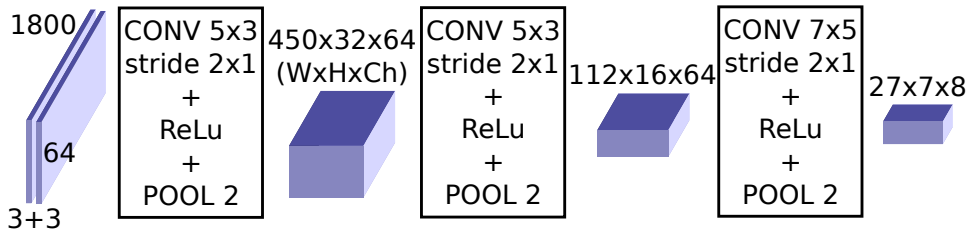


Figure 6.7: Modification of convolutional component for classification network. Wider input (angular resolution $R = 0.2^\circ$) and wider convolution kernels with horizontal stride are used.

kernels are used with horizontal stride (see Fig. 6.7) to reduce the amount of data processed by the fully connected layer.

Although the space of observed rotations is quite small (approximately $\pm 1^\circ$ around x and z axis, and $\pm 4^\circ$ for y axis, see Fig. 6.5), sampling densely (by fraction of degree) this subspace of 3D rotations would result in thousands of possible rotations. Because such amount of rotations would be infeasible to process, we decided to estimate the rotation around each axis separately, so we trained 3 CNNs implementing (6.13) for rotations around x , y and z axis separately. These networks share the same topology (Fig. 6.6).

In the formulation of our classification problem (6.13), the final decision of the best rotation R^* is done by max polling. Since Γ estimates the probability of ro-

tation angle $p(R_i)$ (6.15), assuming the normal distribution we can compute also maximum likelihood solution by weighted average (6.16).

$$p(R_i) = \Gamma(R_i(\mathbf{M}_n), \mathbf{M}_{n-1}) \quad (6.15)$$

$$R^* = \frac{\sum_{i \in S_W} p(R_i) \cdot R_i}{\sum_{i \in S_W} p(R_i)} \quad (6.16)$$

$$S_W = \arg \max_{S=\{i_0, \dots, i_0+W\}} \sum_{i \in S} p(R_i) \quad (6.17)$$

Moreover, this estimation can be done for a window of fixed size W which is limited only for the highest rotation probabilities (6.17). Window of size 1 results in max polling.

6.4.3 Data processing

For training and testing the proposed networks, we used encoded data from Velodyne LiDAR sensor. As we mentioned before, the original raw point clouds consist of x , y and z coordinates, identification of laser beam which captured given point and the value of laser intensity reading. The encoding into 2D representation transforms x and z coordinates (horizontal plane) into the depth information and horizontal angle represented by range channel and the column index respectively in the encoded matrix. The intensity readings and y coordinates are directly mapped into the matrix channels and laser beam index is represented by encoded matrix row index. This means that our encoding (besides the aggregating multiple points into the same polar bin) did not cause any information loss.

Furthermore, we use the same data normalization (6.18) and rescaling as we used in our previous work [96].

$$\bar{h} = \frac{y^i}{H}; \quad \bar{d} = \log(d) \quad (6.18)$$

This applies only to the vertical height h and depth d , since the intensity values are already normalized to interval $(0; 1)$. We set the height normalization constant to $H = 3$, since in the usual scenarios, the Velodyne (model HDL-64E) captures vertical slice approximately 3m high.

In our preliminary experiments, we trained the convolutional networks without this normalization and rescaling (6.18) and we also experimented with using the 3D point coordinates as the channels of CNN input matrices. All these approaches resulted only in worse odometry precision.

6.5 EXPERIMENTS

We implemented the proposed networks using *Caffe*⁵ deep learning framework. For training and testing, data from the KITTI odometry benchmark⁶ were used together with provided development kit for the evaluation and error estimation. The LiDAR data were collected by Velodyne HDL-64E sensor mounted on top of a vehicle together with IMU sensor and GPS localization unit with RTK correction signal providing precise position and orientation measurements [28]. Velodyne spins with frequency 10Hz providing 10 LiDAR scans per second. The dataset consist of 11 data sequences where ground truth is provided. We split these data to training (sequences 00-07) and testing set (sequences 08-10). The rest of the dataset (sequences 11-21) serves for benchmarking purposes only.

The error of estimated odometry is evaluated by the development kit provided with the KITTI benchmark. The data sequences are split into subsequences of 100, 200, . . . , 800 frames (10, 20, . . . , 80 seconds duration). The error e_s of each subsequence is computed as (6.19).

$$e_s = \frac{\|\mathbf{E}_s, \mathbf{C}_s\|_2}{l_s} \quad (6.19)$$

where \mathbf{E}_s is the expected position (from ground truth) and \mathbf{C}_s is the estimated position of the LiDAR where the last frame of subsequence was taken with respect to the initial position (within given subsequence). The difference is divided by the length l_s of the followed trajectory. The final error value is the average of errors e_s across all the subsequences of all the lengths.

First, we trained and evaluated regression networks (topology described in Fig. 6.3) for direct estimation of rotation or translation parameters. The results can be found in Table 6.1. To determine the error of the network predicting translation or rotation motion parameters, the missing rotation or translation parameters respectively were taken from the ground truth data since the evaluation requires all 6DoF parameters.

Evaluation shows that proposed CNNs predict the translation (*CNN-t* in Table 6.1) with high precision – the best results were achieved for network taking the current and $N = 5$ previous frames as the input. The results also show, that all these networks outperform LOAM (error 0.0186, see evaluation in Table 6.3 for more details) in the estimation of translation parameters. On contrary, this method is unable to estimate rotations (*CNN-R* and *CNN-Rt*) with sufficient precision. All networks except the largest one ($N < 7$) are capable of realtime performance with

⁵ caffe.berkeleyvision.org

⁶ www.cvlibs.net/datasets/kitti/eval_odometry.php

N	CNN-t	CNN-R	CNN-Rt	Forward time [s/frame]	
	error	error	error	GPU	CPU
1	0.0184	0.3794	0.3827	0.004	0.065
2	0.0129	0.2752	0.2764	0.013	0.194
3	0.0111	0.2615	0.2617	0.026	0.393
5	0.0103	0.2646	0.2656	0.067	0.987
7	0.0130	0.2534	0.2546	0.125	1.873

Table 6.1: Evaluation of regression networks for different size of input data – N is the number of previous frames. The convolutional networks were used to determine the translation parameters only (column CNN-t), the rotation only (CNN-R) and both the rotation and translation (CNN-Rt) parameters for KITTI sequences 00-08. Error of the estimated odometry together with the processing time of single frame (using CPU only or GPU acceleration) is presented.

Window size W	Odom. error	Window size	Odom. error
1 (max polling)	0.03573	9	0.03704
3	0.03433	11	0.03712
5	0.03504	13	0.03719
7	0.03629	all	0.03719

Table 6.2: The impact of window size on the error of odometry, when the rotation parameters are estimated by classification strategy. Window size $W = 1$ is equivalent to the max pooling, maximal likelihood solution is found also when “all” probabilities are taken into the account without the window restriction.

GPU support (GeForce GTX 770 used) and the smallest one also without any acceleration (running on i5-6500 CPU). Note: Velodyne standard framerate is 10fps.

We also wanted to explore, whether CNNs are capable to predict full 6DoF motion parameters, including rotation angles with sufficient precision. Hence the classification network schema shown in Fig. 6.6 was implemented and trained also using the Caffe framework. The network predicts probabilities for densely sampled rotation angles. We used sampling resolution 0.2° , what is equivalent to the horizontal angular resolution of Velodyne data in the KITTI dataset. Given the statistics from training data shown in Fig. 6.5, we sampled the interval $\pm 1.3^\circ$ of rotations around x and z axis into 13 classes, and the interval $\pm 5.6^\circ$ into 56 classes, including approximately 30% tolerance.

Since the network predicts the probabilities of given rotations, the final estimation of the rotation angle is obtained by max polling (6.13) or by the window ap-

Seq. #	Translation only			Rotation and translation			
	LOAM- full	LOAM- online	CNN- regress.	LOAM- full	LOAM- online	CNN- regress.	CNN- classif.
00	0.0152	0.0193	0.0084	0.0225	0.0516	0.2877	0.0302
01	0.0368	0.0255	0.0079	0.0396	0.0385	0.1492	0.0444
02	0.0383	0.0293	0.0076	0.0461	0.0550	0.2290	0.0342
03	0.0120	0.0117	0.0166	0.0191	0.0294	0.0648	0.0494
04	0.0076	0.0085	0.0089	0.0148	0.0150	0.0757	0.0177
05	0.0092	0.0096	0.0056	0.0184	0.0246	0.1357	0.0235
06	0.0088	0.0130	0.0036	0.0160	0.0335	0.0812	0.0188
07	0.0137	0.0155	0.0077	0.0192	0.0380	0.1308	0.0177
Train avg.	0.0214	0.0197	0.0077	0.0287	0.0433	0.1970	0.0303
08	0.0107	0.0145	0.0096	0.0239	0.0349	0.2716	0.0289
09	0.0368	0.0380	0.0098	0.0322	0.0430	0.2373	0.0494
10	0.0213	0.0196	0.0128	0.0295	0.0399	0.2823	0.0327
Test avg.	0.0186	0.0208	0.0102	0.0268	0.0376	0.2655	0.0343

Table 6.3: Comparison of the odometry estimation precision by the proposed method and LOAM for sequences of the KITTI dataset [28] (sequences 00 – 07 were used for training the CNN, 08 – 10 for testing only). LOAM was tested in the on-line mode (LOAM-online) when the time spent for single frame processing is limited to Velodyne fps (0.1s/frame) and in the full mode (LOAM-full) where each frame is fully registered within the map. Both the regression (CNN-regression) and the classification (CNN-classification) strategies of our method are included. When only translation parameters are estimated, our method outperforms LOAM. On the contrary, LOAM outperforms our CNN odometry when full 6DoF motion parameters are estimated.

proach of maximum likelihood estimation (6.16,6.17). Table 6.2 shows that optimal results are achieved when the window size $W = 3$ is used.

We compared our CNN approach for odometry estimation with the LOAM method [107]. We used the originally published ROS implementation (see link in Sec. 6.3) with a slight modification to enable KITTI Velodyne HDL-64E data processing. In the original package, the input data format of Velodyne VLP-16 is

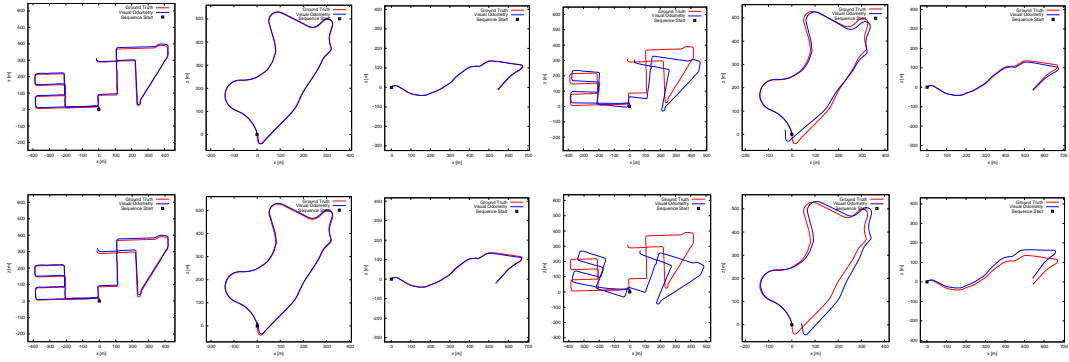


Figure 6.8: The example of LOAM results (top) and our CNNs (bottom row) for KITTI sequences used for testing (08 – 10). When only translation parameters are estimated (first 3 columns), both methods achieves very good precision and the differences from ground truth (red) are barely visible. When all 6DoF motion parameters are estimated (columns 4 – 6), better performance of loam LOAM can be observed.

“hardcoded”. The results of this implementation is labeled in Table 6.3 as *LOAM-online*, since the data are processed online in real time (10fps). This real-time performance is achieved by skipping the full mapping procedure (registration of the current frame against the internal map) for particular input frames.

Comparing with this original online mode of LOAM method, our CNN approach achieves better results in estimation of both translation and rotation motion parameters. However, it is important to mention, that our classification network for the orientation estimation requires 0.27s/frame when using GPU acceleration.

The portion of skipped frames in the LOAM method depends on the input frame rate, size of input data, available computational power and affects the precision of estimated odometry. In our experiments with the KITTI dataset (on the same machine as we used for CNN experiments), 31.7% of input frames is processed by the full mapping procedure.

In order to determine the full potential of the LOAM method, and for fair comparison, we made further modifications of the original implementation, so the mapping procedure runs for each input frame. Results of this method are labeled as *LOAM-full* in Table 6.3 and, in estimation of all 6DoF motion parameters, it outperforms our proposed CNNs. However, the prediction of translation parameters by our regression networks is still significantly more precise and faster. And the average processing time of a single frame by the LOAM-full method is 0.7s. The visualization of estimated transformations can be found in Fig. 6.8.

We have also submitted the results of our networks (i.e. the regression CNN estimating translational parameters only and the classification CNN estimating rotations) to the KITTI benchmark together with the outputs we achieved using the LOAM method in the online and the full mapping mode. The results are similar as

in our experiments – best performing LOAM-full achieves 3.49% and our CNNs 4.59% error. LOAM-online performed worse than in our experiments with error 9.21%. Interestingly, the error of our refactored original implementation of LOAM is more significant than errors reported for the original submission of the LOAM authors. This is probably caused by a fine-tuning of the method for the KITTI dataset which has never been published and authors refused to share both the specification/implementation used and the outputs of their method with us.

6.6 CONCLUSION

This paper introduced novel method of odometry estimation using convolutional neural networks. As the most significant contribution, networks for very fast real-time and precise estimation of translation parameters, beyond the performance of other state of the art methods, were proposed. The precision of proposed CNNs was evaluated using the standard KITTI odometry dataset.

Proposed solution can replace less accurate methods like odometry estimated from wheel platform encoders or GPS based solutions, when GNSS signal is not sufficient or corrections are missing (indoor, forests, etc.). Moreover, with the rotation parameters obtained from the IMU sensor, results of the mapping can be shown in a preview for online verification of the mapping procedure when the data are being collect.

We also introduced two alternative network topologies and training strategies for prediction of orientation angles, enabling complete visual odometry estimation using CNNs in a real time. Our method benefits from existing encoding of sparse LiDAR data for processing by CNNs [96, 57] and contributes as a proof of general usability of such a framework.

In the future work, we are going to deploy our odometry estimation approaches in real-word online 3D LiDAR mapping solutions for both indoor and outdoor environments.

Part III

BACKPACK MOBILE MAPPING SOLUTION

This chapter is based on the paper [100].

INDOOR AND OUTDOOR BACKPACK MAPPING WITH CALIBRATED PAIR OF VELODYNE LIDARS

7.1 ABSTRACT

This paper presents a human-carried mapping backpack based on a pair of Velodyne LiDAR scanners. Our system is a universal solution for both large scale outdoor and smaller indoor environments. It benefits from a combination of two LiDAR scanners, which makes the odometry estimation more precise. The scanners are mounted under different angles, thus a larger space around the backpack is scanned. By fusion with GNSS/INS sub-system, the mapping of featureless environments and the georeferencing of resulting point cloud is possible. By deploying SoA methods for registration and the loop closure optimization, it provides sufficient precision for many applications in BIM (Building Information Modeling), inventory check, construction planning, etc. In our indoor experiments, we evaluated our proposed backpack against ZEB-1 solution, using FARO terrestrial scanner as the reference, yielding similar results in terms of precision, while our system provides higher data density, laser intensity readings, and scalability for large environments.

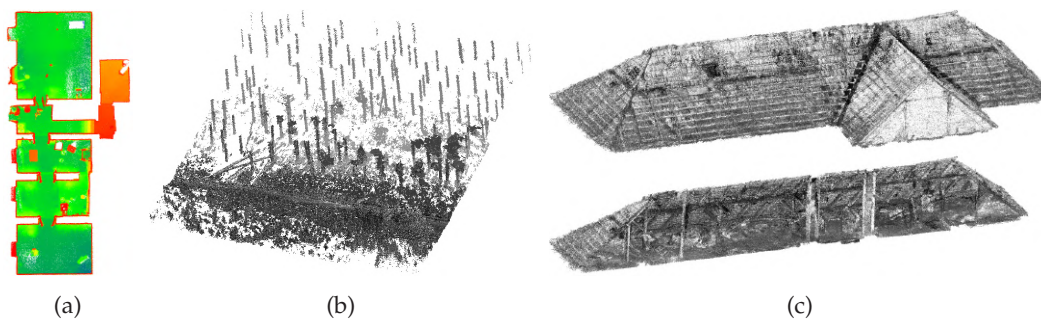


Figure 7.1: The motivation and the results of our work. The reconstruction of indoor environments (a) is beneficial for inspection, inventory checking and automatic floor plans generation. 3D maps of forest environments (b) is useful for quick and precise estimation of the biomass (timber) amount. The other example of 3D LiDAR mapping deployment is preserving cultural heritages or providing models of historical building, e.g., the roof in (c).

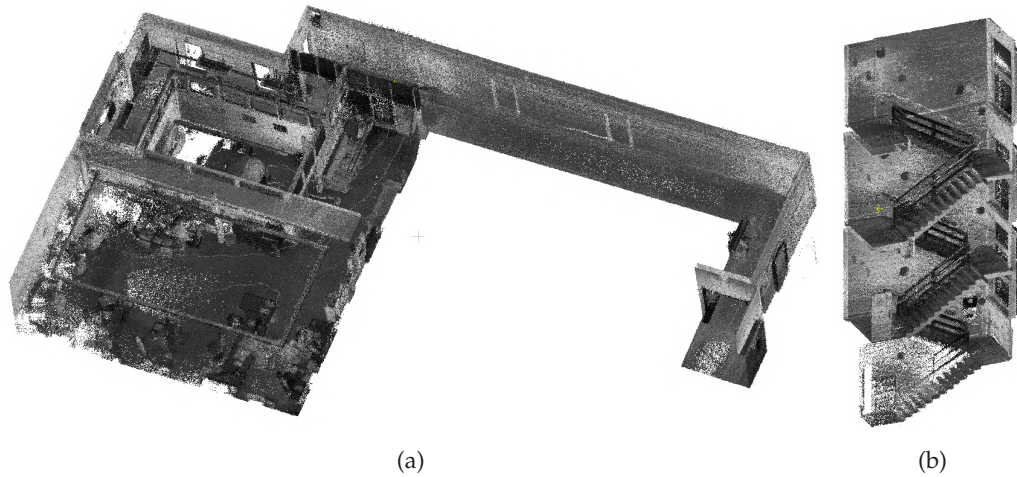


Figure 7.2: The example of resulting models of indoor mapping. The office environment (a) and the staircase (b) were captured by a human carrying our 4RECON backpack. The data acquisition process took 3 and 2 min, respectively.

7.2 INTRODUCTION

In recent years, the LiDAR (Light Detection And Ranging) technology has become very popular in the field of geodesy and related fields, where the availability of 3D models of outdoor or indoor environments can be beneficial: e.g., forestry, architecture, preserving cultural heritage, construction monitoring, etc. The examples of reconstructions from similar practical applications can be found in Fig. 7.1. Using 3D mapping can also be beneficial for time and cost reduction. The same model can be shared among different professionals in different fields of expertise without the need for personal inspection and measuring at a given place individually.

This demand causes a huge interest in developing solutions that would be able to capture the reality and provide reliable 3D reconstructions out of the box. However, there are also other requirements for such a system.

The data acquisition process has to be quick and the planning of fieldwork should be minimized. This requirement discriminates solutions based on static terrestrial lasers (e.g., Leica and Riegl of FARO companies), requiring detailed planning of the data acquisition and manual system set up on a tripod within multiple convenient viewpoints across the scene.

The solution has to be mobile and easy to handle. This naturally leads to the preference of human carried (backpack or handheld) solutions instead of terrestrial or vehicle based solutions, such as NavVis ¹, which, for example, does not support traversing tilted surfaces such as ramps.

¹ <https://www.navvis.com>

However, the necessity for reliability in terms of resulting model precision is in contradiction with these two requirements. Stationary terrestrial LiDAR solutions require time demanding scanning process while providing a great accuracy (in order of millimeters) because of fewer degrees of freedom. Although, for many applications listed above, there is no need for such precision, our goal is the difference between the reality and the resulting 3D model below 5 cm. This value was requested by the experts in the field of geodesy with whom we consulted.

In the practical applications, completeness of the final map should also be guaranteed because it might be difficult to repeat the scanning. The operator has to be aware of the fact that all necessary data of the whole environment were acquired. We fulfilled this requirement by providing a live preview of the collected data.

The resulting model has to be dense enough, so that all important objects such as furniture and other inventory can be recognized and distinguished. This is the typical issue of existing solutions such as ZEB-1, where no LiDAR intensity readings are available. Therefore, our solution relies on Velodyne LiDARs, which provide a huge amount of data and the resulting models are dense (see examples in Fig. 7.2). It also provides the laser intensity readings, which do not depend on the lighting conditions, contrary to camera-aided solutions. Moreover, we propose laser intensity normalization, which increases the recognizability of the objects since the laser intensity readings cannot be considered as the “color” of the object as it depends on the range of measurement, the angle of incidence, and the emitted energy.

Some of the existing solutions are not comfortable enough to use. According to practical experience of the operators, handheld solutions such as ZEB are physically difficult to operate for a longer period of time since the mapping head weighs approximately 0.4–1 kg, and it has to be carried or swept by hand.

The final requirement is an affordable price. We use Velodyne VLP-16 scanners, which are relatively cheap in comparison to the other LiDAR solutions, and a universal IMU (Inertial Measurement Unit) solution, which can be upgraded by a dual antenna and therefore reused in the outdoor environment where GNSS (Global Navigation Satellite System) is available.

The contributions of this paper can be summarized as the proposal of a LiDAR mapping solution with the following characteristics:

- It is capable of both small indoor and large open outdoor environments mapping, georeferencing and sufficient precision in the order of centimeters. These abilities are evaluated using multiple datasets.
- It benefits from a synchronized and calibrated dual LiDAR scanner, which significantly increases field of view. Both scanners are used for both odometry estimation and 3D model reconstruction, which enables scanning of small environments, narrow corridors, staircases, etc.

- It provides the ability to recognize objects in the map due to sufficient point density and our novel intensity normalization for the measurements from an arbitrary range.

We also performed a precise evaluation and comparison of our previously proposed point cloud registration method CLS (Collar Line Segments) with state-of-the-art approach LOAM (LiDAR Odometry and Mapping), which has not yet been published. Moreover, we upgraded our CLS method with automatic overlap estimation for better registration flexibility.

7.3 RELATED WORK

LiDAR based systems for indoor and outdoor mapping are not a brand new tool in the geospatial community. Demand for such solutions drives—among other applications, such as autonomous driving—the development of basic algorithms for LiDAR data processing, point cloud registration, etc., as the essential parts of more complex SLAM (Simultaneous Localization and Mapping) methods (a summary can be found in [72]).

Table 7.1 contains an overview of the existing LiDAR mapping solutions that are related to our work. All such solutions have to solve several typical issues. Besides the construction of hardware mount itself (e.g., a backpack or a drone), the data from multiple sensors have to be synchronized properly, etc. However, the key issue is the software component for odometry estimation—i.e., estimation of the trajectory and the movement of the sensory platform. This is essential for correct alignment of laser measurements into a consistent and precise 3D model. Although there are already numerous methods providing solutions within a certain level of precision for certain types of LiDAR sensors, precise odometry estimation is still an open question.

One of the state-of-the-art methods, performing quite well for both the 3D LiDARs (as Velodyne) and also the 2D rangefinders (as continuously spinning or sweeping Hokuyo LiDAR), is LOAM (LiDAR Odometry And Mapping) [107]. There are also visually [105] or depth enhanced [106] versions where the odometry estimation is supplemented by a RGB camera or a depth sensor (such as Kinect or Asus Xtion), respectively. This whole group can be considered as *feature-based* methods, since, from the original point cloud, only the edge and the plane key points are preserved. These are used for geometrical registration of the current frame within the map and also for building the map itself continuously. Based on the impressive results presented, LOAM method was our first candidate for odometry estimation in our backpack solution. However, our experiments on KITTI Odometry [28] dataset presented later on in Sec. 7.5.1 will show that our previously pub-

Table 7.1: Overview of related LiDAR mobile mapping solutions.








Solution	Sensor (Precision)	Range	System Precision	Price €	Open Method	Properties and Limitations	Intensities
ZEB-1 (2013) ²	 Hokuyo UTM-30LX (3 cm up to 10 m range)	15–20 m (max 30 m under optimal conditions)	up to 3.8 cm indoors [90]	N/A	Proprietary, based on [13, 11]	<ul style="list-style-type: none"> missing (laser) intensity readings no GNSS reference requires visible featuring objects at close distances 	No
ZEB-REVO (2015) ² [30]	 Hokuyo UTM-30LX-F (3 cm up to 10 m range)	15–20 m (max 30 m under optimal conditions) [30]	up to 3.6 cm indoors [20]	34,000	Proprietary, based on [13, 11]	<ul style="list-style-type: none"> missing (laser) intensity readings no GNSS reference requires visible featuring objects at close distances 	No
LiBackpack (2019) ³ [33]	 2 × Velodyne VLP-16 (3 cm)	100 m (Velodyne scanner limitation)	5 cm	60,000	Proprietary	<ul style="list-style-type: none"> intensity readings available GNSS support dual LiDAR (one for odometry only, second for reconstruction) 	Yes
Pegasus (2015) [56]	 2 × Velodyne VLP-16 (3 cm)	50 m usable range	5 cm with GNSS, 5–50 cm without, 4.2 cm in test [66]	150,000	Proprietary	<ul style="list-style-type: none"> intensity readings available GNSS support dual LiDAR (cooperation unknown) 	Yes

Table 7.1: *Cont.*

Solution	Sensor (Precision)	Range	System Precision	Price €	Open Method	Properties and Limitations	Intensities
Viametris bMS3D ⁴ [25]	 2 × Velodyne VLP-16 (3 cm)	100 m (Velodyne scanner limitation)	5 cm under appropriate satellite reception	N/A	Propriet.	<ul style="list-style-type: none"> intensity readings and RGB coloring GNSS support dual LiDAR (cooperation unknown) 	Yes
Robin (2016) ⁵ [3]	 RIEGL VUX-1HA (3 mm)	120/420 m in slow/high frequency mode (for sensor)	up to 3.6 cm at 30 m range (FOG IMU update)	220,000	Propriet.	<ul style="list-style-type: none"> intensity readings dual GNSS (at least weak) GNSS signal required 	Yes
Akhka (2015) [52, 84]	 FARO Focus3D 120S (1 mm)	120 m (sensor range)	8.7 cm in forest environments	N/A	Open [84]	<ul style="list-style-type: none"> intensity readings outdoor only (GNSS required) 	Yes

² <https://geoslam.com/>³ <https://greenvalleyintl.com/>⁴ <https://www.viametris.com/>⁵ <https://www.3dlasermapping.com/>

lished method CLS (Collar Line Segments) [98] outperforms LOAM in terms of accuracy—the error is lowered from 2.9 cm to 1.7 cm per 1 m of elapsed trajectory.

Another solution for odometry estimation, developed and published by Bosse and Zlot [11] in 2009, is designed for continuously spinning 2D LiDAR rangefinder. After three years, this approach was modified and integrated into the prototype of *Zebedee* [13] mobile mapping application which eventually evolved into ZEB products (in Table 7.1) of GeoSLAM company². These products are probably the most related to our solution in terms of pricing (ZEB-REVO including 1 year basic support costs 34,000 €) and therefore also in terms of accessibility to small companies.

Bosse and Zlot [11] proposed a *surfel-based* algorithm *Voxel Sweep Match* which works over the space discretized into a 3D voxel grid. The model of the environment consists of a set of surfels—3D ellipsoids representing the local surface information within the voxel. The internal model is updated and new surfels are added after each “sweep” (the half revolution of the spinning LiDAR) is captured. The algorithm works similarly to the well known ICP (Iterative closest point) [10], but instead of point-to-point matching, the surfels matching in 9D space (including the position, and the orientation of the surfels) is used. Beside these matching constraints of neighboring surfels, another constraints ensuring the smoothness and the continuity of the trajectory are added in the form of linear equations to be solved. After the new continuous trajectory estimation, surfel positions are updated, and the process is repeated until convergence.

This first proposal [11] did not reach good precision and the main contribution is the basis for further development and improvements—especially missing global loop closure is the problem, which has been solved in downstream projects: *Zebedee* [13] solution and probably also in ZEB-1 system. It is likely that ZEB-1 is the evolution of *Zebedee*, since it shares the same ideas and design, but we cannot say this for sure, since it is a closed proprietary solution. Both *Zebedee* and ZEB-1 use Hokuyo 2D LiDAR, and instead of a continuously spinning mount, a flexible spring construction is used to extend the rangefinder into the 3D LiDAR. The spring amplifies low frequency smooth sweeping motions, while it cancels high frequency motions (vibrations and shaking), which are undesirable and difficult to estimate in SLAM solutions. Moreover, an IMU unit was added in order to estimate quick swinging motion and provide additional constraints for optimization.

Regarding the precision of *Zebedee* prototype, the accumulated drift for open loop precision causes approximately 10 cm translation and 2° rotation error per minute. This error is significantly reduced by loop closing in a global optimization. The error of the global solution is not published, since the ground truth for experimental dataset was not available. However, the visualization in Fig. 7.3 still shows

so-called “dual wall” errors: two instances of the same wall in the same model but at different positions. This ambiguity causes significant problems when the model should be used for further processing (by construction engineers, architects, etc.), and it is our goal to avoid this type of error.

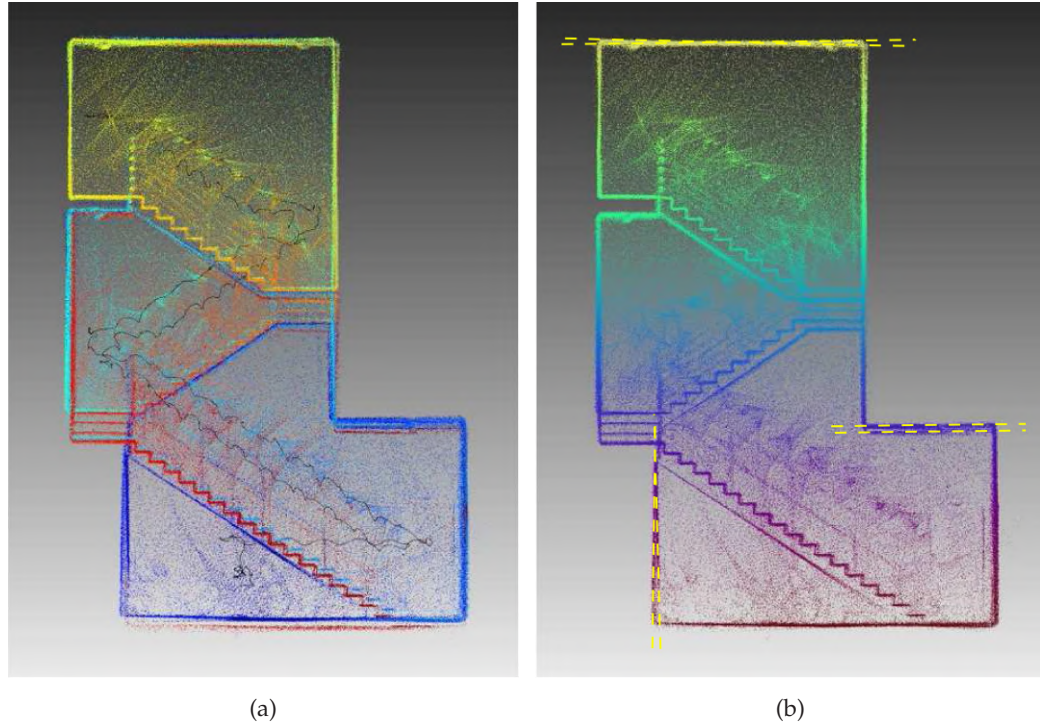


Figure 7.3: “Double walls” error in the reconstruction of Zebedee [13]. The wall and the ceiling appears twice in the reconstruction, causing an ambiguity. In the solution without loop closure (a), the error is quite visible. Double walls are reduced after global loop closure (b), but they are still present (highlighted by yellow dashed lines).

In 2015, the GeoSLAM company released their new alternative version of a handheld LiDAR scanner—ZEB-REVO [29, 30]—where the spring mount was revoked in favor of original continuously spinning design. This update brings better performance in both processing time and accuracy. In addition, the human operator does not have to “whisk” the sensory head in order to correctly capture the whole environment around, as it was required in ZEB-1. However, the weight of the handheld part of the scanner was increased from 0.4 kg (for ZEB-1) to 1 kg, probably due to servo motors and additional electronics. These factors (the necessity to whisk for ZEB-1 and the significant weight for ZEB-REVO) considerably decrease the usage comfort when a larger environment is mapped.

Since the ZEB products are closed and they are using proprietary software, it is not clear how the 3D map is actually built. Fortunately, there are at least several works published, where the quality of the resulting model was evaluated. The evaluation of ZEB-REVO in an underground quarry [20] reported point precision (in

terms of the distance to the best fitting plane for given surface) around 3 cm. In an experiment within a small office environment [64], 22 test planes were selected from the 3D model built by ZEB-REVO. Using the same evaluation, the standard deviation of the point to best fitting-plane distance reached 11 mm. However, these evaluations do not say much about the precision of the whole model and reflect only the local precision. Another work evaluated ZEB-1 [90] by comparison with measurements obtained by a precise terrestrial laser (Leica C10) as the ground truth. For a small indoor environment in Fig. 7.4, the difference in corner-to-corner distances were up to 3.8 cm, and the difference between real and estimated area floor reached 0.4 m². These numbers are consistent with specified positional accuracy between 3–30 cm after 10 min scanning process in user guides [29, 30]. The density of 1000–18,000 points/m² was observed in the point cloud model generated by ZEB-1 which represents an average distance of 0.8–3 cm between the points.

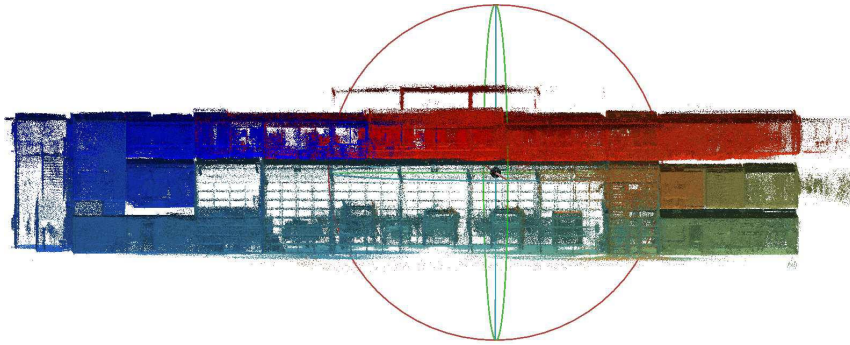


Figure 7.4: Dataset of indoor office environment for evaluation of ZEB-1 scanner [90]. In the experiment, 3.8 cm error of corner-to-corner average distances within the rooms was achieved.

When using Zebedee, ZEB-1 or ZEB-REVO, the user has to follow certain guidelines and also be aware of the limitations of these products [13, 29, 30]. When using Zebedee or ZEB-1 with head mounted on the flexible spring, the user has to keep the sensor in the movement by constant “whisking” or somehow changing the accelerations all the time what could be uncomfortable or inconvenient in many cases. The absence of the swinging motion would degrade the sensor back into an 2D rangefinder and could cause a serious error. In addition, the sensors are sensitive to motions in the scene (people, animals, etc.) and the operator has to preserve the overlap between the current and previous measurements—e.g., by walking backwards when leaving a room or traversing doors, keeping a slow pace, etc., since the sensors observe only the environment in front of the operator.

Moreover, there are certain so called “ill” environments or situations when ZEB solutions are failing—especially featureless and empty spaces, where SLAM solutions are failing in general, and the only solution is the augmentation of the scene

by additional obstacles, boxes, etc. Optimal results can be obtained when the obstacles or featuring objects are within 15–20 m range for outdoor. This is a significant limitation for vast open environments.

Other mobile backpack solutions for the LiDAR mapping can be divided into two groups: fully commercial, such as Leica Pegasus [56], Viametris bMS3D⁶ [25], Robin backpack [3], or GreenValley LiBackpack [33], and research projects, such as Akhka Backpack [84]. Basic properties of these solutions are summarized in Table 7.1. The most significant drawback of these solutions is their high price: 150,000 € for Pegasus, 220,000 € for Robin, and 60,000 € for GreenValley backpack (without GNSS upgrade), which makes them too expensive and inaccessible for small businesses. In comparison, the total cost of HW components in our solution is around 17,500 €. For the whole product (including SW development, support, etc.) we expect the price to increase approximately twofold, which brings us much closer to ZEB scanners. Another disadvantage of these backpack solutions (at least for Leica Pegasus and Robin backpacks) is their high dependency on GNSS, so the quality of mapping drops when the signal of satellites is poor or not available.

Leica, Viametris, 3D Laser Mapping, and GreenValley companies naturally did not publish how their solutions estimate the odometry and the alignment of LiDAR data into 3D model. We know that these systems use GNSS/INS aiding in order to improve the precision. According to the documentation [56], Leica Pegasus is able to achieve up to 5 cm precision after 10 min walk, when GNSS is available and 5–50 cm without GNSS aiding. It uses 2 Velodyne LiDAR scanners as a source of 3D data and an additional set of five high-resolution cameras. Potential problems for small rooms, staircases and featureless environments are reported in the documentation. Independent evaluation has been performed in small (20 m length) underground medieval stronghold [66], where average error of 4.2 cm is reported when the model is compared with terrestrial LiDAR reference. There is not much information published regarding price or precision of Viametris backpack. However, up to 5 cm accuracy is reported when reasonable satellite reception is available [25]. The Robin backpack [3] for outdoor mapping depends on precise GNSS/INS (Inertial Navigation System) with dual antenna, claiming 2 cm positional and 0.03° error. However, the precision of generated models is not specified, and no evaluation papers have been published yet (to our best knowledge). The specification of GreenValley LiBackpack [33] claims ≈ 5 cm relative accuracy of the system.

LiBackpack can be considered as the backpack solution most similar to ours—in terms of price, sensors, and accuracy. However, according to the information given to us by GreenValley company, their solution uses Velodyne scanners separately—

⁶ <https://www.viametris.com/>

one scanner is used for the odometry estimation using SLAM, and the second one for the 3D reconstruction. We find it unfortunate, because the full potential of data is not utilized. In the solution proposed in this paper, both scanners are synchronized and extrinsically calibrated—mutual 6 DoF (Degrees of Freedom) pose is estimated. This makes it possible to use both sensors in both tasks—SLAM and building the 3D model.

Akhka mapping backpack [84, 51, 52, 54] was developed by Finnish Geospatial Research Institute and Aalto University. It deploys Faro Focus LiDAR and depends on the precise Novatel Flexpak6 GNSS-IMU solution. When mapping the environments with wrong GNSS reception, the scans are roughly aligned by IMU within small time windows—segments. Afterwards, ICP is used for registration of these segments. During the experiment in a river channel, RMSE (root mean square error) of 3.6 cm was measured at reference positions. During the mapping of a forest environment, the average misalignment increased to 8.7 cm.

Google released their SLAM software Cartographer [37] for online building 2D floor plans using LiDAR rangefinders. It uses efficient probability 2D occupancy grid (5 cm resolution) as a map representation enabling fast registration and robust loop closure. Google also claims the ability to produce full 3D maps, however, the results reported are not that appealing [75]. As far as we know, no seriously evaluated deployment has been published so far.

The idea behind the well-known KinectFusion [73] project for processing RGB-D data drove the development of a new solution for LiDAR odometry estimation called IMLS-SLAM [19]. Instead of typical scan-to-scan matching and registration, the target LiDAR scan is transformed into implicit surface representation denoted as IMLS surface (Implicit Moving Least Square) originally proposed by [48]. The source frames are registered against these implicit surfaces following the scan-to-model strategy. This work also provides mathematical background for solving such a task as a least-square optimization problem. On average, their method achieved 0.69 cm drift after 1 m of elapsed trajectory.

Droeschel et al. [23] proposed a hierarchical pose graph structure for online mapping and odometry estimation. They split each frame into scan lines (slices of the Velodyne LiDAR frame with 1.33 ms duration), while they also group neighboring frames into local optimization windows. Therefore, there are 3 types of nodes within the graph: map nodes representing local windows on the highest level, scan nodes representing Velodyne LiDAR frames (360° revolution), and the scan line nodes on the lowest level. The surfel based registration is performed only among the frames within the local window (forming edges between map and scan nodes) and among whole local windows (producing edges among map nodes). The global optimization produces a continuous time trajectory, where the transformation is

assigned to each scan line by cubic B-spline interpolation. Therefore, the scans, the pose graph, and the trajectory are iteratively refined. Unfortunately, the paper does not provide the precise evaluation of this method. The visualizations show that the method reduces the thickness of the walls and so-called “double wall” effect in comparison with previous approach without hierarchical structure [24].

We also experimented with a similar hierarchical approach in our SLAM system. The main motivation was to make the process more time-efficient. Eventually, we rejected this idea, since the errors of frame-to-frame registrations, which were introduced into the local map, made the registrations among local maps quite inaccurate.

Mendes et al. [68] decided to run a simple ICP frame-to-frame registration for the stream of LiDAR scans. Instead of registering the consecutive frames together, the current frame is aligned within the local map consisting of last few keyframes. When the overlap (given by the point matching in ICP algorithm) drops under a certain level, the current scan is labeled as a new keyframe and it is added to the local map. The old keyframes (dropped from the local map) are preserved for the loop detection and closure.

Besides the geometrical accuracy of the model, there are also other quality aspects to consider when creating a LiDAR mapping solution, e.g., the point cloud density and especially the ability of so-called “recognizability” of various objects in the map. The consumer of the point cloud model (engineer, architect, geodesist, etc.) has to be able to recognize furniture, surface borders, and in some cases also writings, symbols or the texture of the surface. For this task, the color or at least the intensities have to be correctly introduced into the model. In the previously described solutions of LiDAR mapping, this information is missing (e.g., ZEB-1, basic ZEB-REVO) or introduced by additional RGB camera (e.g., ZEB-REVO [79]). In solutions based on the terrestrial laser scanner or Velodyne LiDARs, the intensity of laser return is used directly to color the points in the model.

Since we want to keep our solution simple and cheap and preserve the invariance to lightning conditions, we decided to “color” our models with LiDAR intensities. However, keeping these raw intensities would cause unwanted artifacts. As was described in previous works [42, 45, 44], the reflectivity of the surface, which we want to capture, is not the only factor affecting these intensity values. The measured intensity depends also on the incidence angle of the laser ray, distance from the sensor (see Fig. 7.5), power of laser transmitter, and, in some cases, also on the atmospheric influences (e.g., fog, dust, and smog). These works addressed the problem providing models and closed form solutions. However, these methods are valid only for large-distance measurements (at least tens of meters) and therefore they are not suitable for typical indoor or smaller outdoor environ-

ments, which we need to address. Hence, we propose a novel probabilistic method for LiDAR intensities normalization which is scalable and capable of processing near-distance measurements.

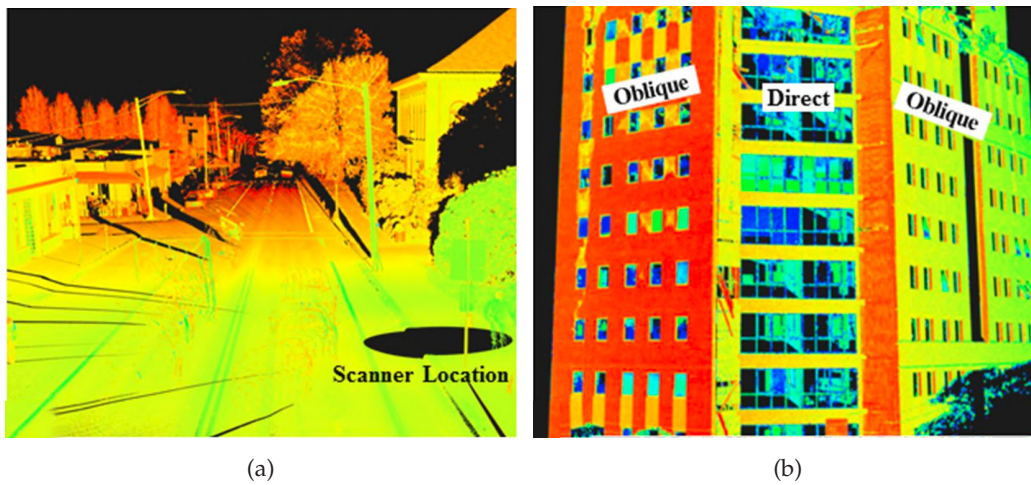


Figure 7.5: The dependency of laser intensity readings (weak readings in red, strong in green) on the measurement range (a) and the angle of incidence (b) [45].

7.4 DESIGN OF THE LASER MAPPING BACKPACK

This section consists of two main parts: First, the hardware design concepts are introduced. Then, the software solutions dealing with calibration, precise odometry estimation, alignment and intensity normalization are presented.

The design of our solution follows the requirements elaborated in Sec. 7.2. They have been carefully formulated and discussed with experts in the field of geodesy and geospatial data processing. Besides the essential goal of reliable 3D reconstruction performed automatically, which is demonstrated in the following section, the proposed solution does the following:

- It fulfils the requirements for precision of the model up to 5 cm. Thanks to the robust loop closure, ambiguities (e.g., “double wall” effects) are avoided.
- The system is comfortable to use and it is as mobile as possible. The backpack weighs 9 kg (plus 1.4 kg for the optional dual antenna extension), and it is easy to carry around various environments including stairs, narrow corridors, rugged terrain, etc.
- The pair of synchronized and calibrated Velodyne LiDARS increases the field of view (FOV) and enables mapping of small rooms, narrow corridors, staircases, etc. (see Fig. 7.6) without the need for special guidelines for scanning process.

- The data acquisition process is fast with verification of data completeness. There are no special guidelines for the scanning process (comparing to the requirements of ZEB) and the operator is required only to visit all places to be captured in a normal pace. Moreover, captured data are visualized online at the mobile device (smartphone, tablet) for operator to see whether everything is captured correctly.
- Since we are using long range Velodyne LiDAR (compared to simple 2D rangefinders such as Hokuyko or Sick) and optional GNSS support, we provide a universal economically convenient solution for both indoor and outdoor use. For such scenarios, where GNSS is available, final reconstruction is georeferenced—the 3D position in the global geographical frame is assigned to every 3D point in the model.
- The final 3D model is dense and colored by the laser intensity, which is further normalized. This helps distinguishing important objects, inventory, larger texts, signs, and some surface texture properties.

7.4.1 Hardware Description

The core of our backpack, in Fig. 7.7, is the pair of Velodyne LiDAR ⁷ scanners VLP-16 (Pucks). Each of them contains 16 laser transmitter–receiver pairs, which are reflected into the environment by a rotating mirror with 10 Hz frequency. This frequency can be decreased or increased up to 20 Hz. However, frequency higher than 10 Hz causes serious undesirable vibration of the sensor, which makes precise odometry estimation impossible. The rotation gives the sensor 360° horizontal FOV with 0.2° horizontal resolution. Vertically, the laser beams are evenly distributed with 2° resolution covering 30° vertical FOV. Each of the scanners weighs 830 g and is considered to be a hybrid solid state LiDAR, since there are no outer moving parts. This type of scanner is able to reach 100 m range with precision around 2 cm. As mentioned above, Velodyne scanners provide also values of intensity readings, which corresponds to the surface reflectivity.

As the aiding sensor, the GNSS/INS (Inertial Navigation System) Advanced Navigation SpatialDual⁸ is deployed. It integrates multiple sensors such as accelerometers, gyroscopes, magnetometer, pressure sensor, and most importantly—the dual-antenna GNSS subsystem providing reliable heading information. With RTK (Real Time Kinematics) or PPK (Post-Processed Kinematics) corrections,

⁷ <https://velodynelidar.com/>

⁸ <https://www.advancednavigation.com/product/spatial-dual>

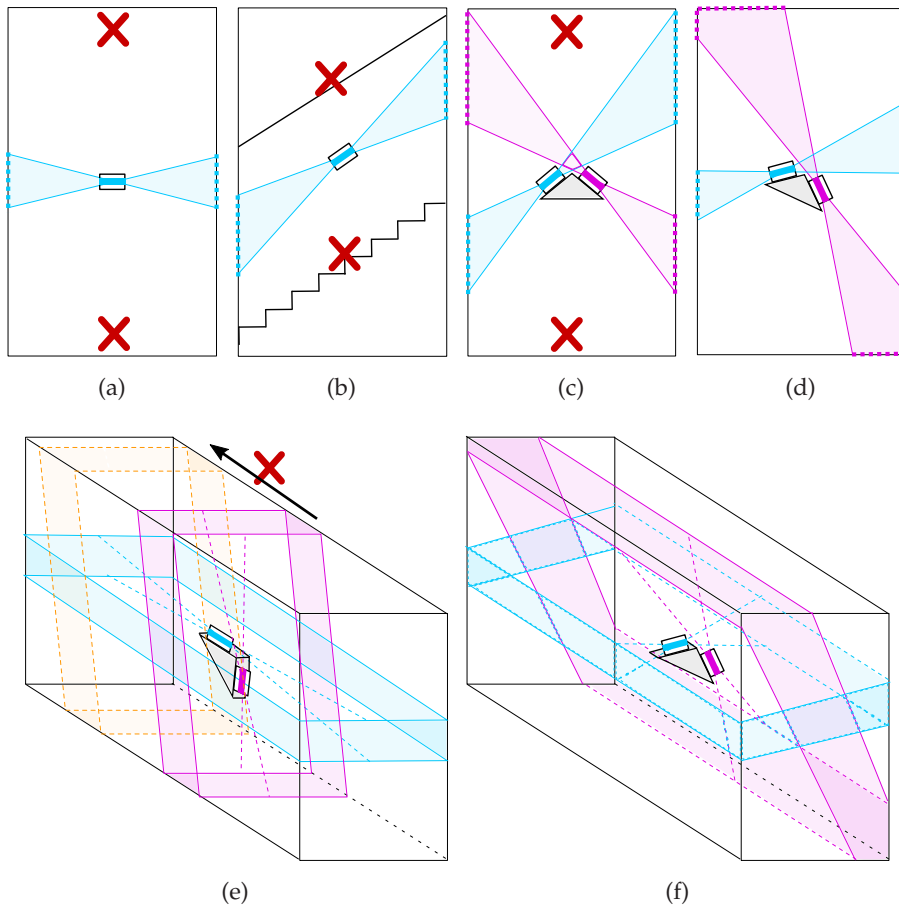


Figure 7.6: Various configurations of LiDAR scanners in worst case scenarios we have encountered in our experiments: narrow corridor (a),(c) and staircase (b). The field of view (30° for Velodyne Puck) is displayed in color. When only single LiDAR (a) was used, the scans did not contain 3D information of the floor or the ceiling (red cross). The situation was not improved when the scanner is tilted because of failing in, e.g., staircases (b). When we added a second LiDAR, our tiled asymmetrical configuration (d) provides better top-bottom and left-right observation than the symmetrical one (c). Moreover, when the LiDARs are aligned in direction of movement (e), there is no overlap between current (violet) and future (yellow) frame, leading to lower accuracy. In our solution (f), the LiDARs are aligned perpendicularly to the walking direction solving all mentioned issues..

the system should provide 8 mm horizontal and 15 mm vertical positional accuracy, and 0.03° and 0.06° orientation precision in terms of roll/pitch and heading angle, respectively. Precise heading information is provided by a dual antenna solution and therefore it is only available outdoors. This limitation also holds for positional data. For indoor scenarios, only roll and pitch angles are reliable and they are relevant for horizontal alignment. The unit weighs 285 g and besides the 6 DoF (six Degrees of Freedom including 3D position and rotation) pose estimation it also provides 1PPS (Pulse Per Second) and NMEA messages for precise

synchronization of both Velodyne LiDAR scanners. The details regarding wiring the components can be found in Fig. 7.8.

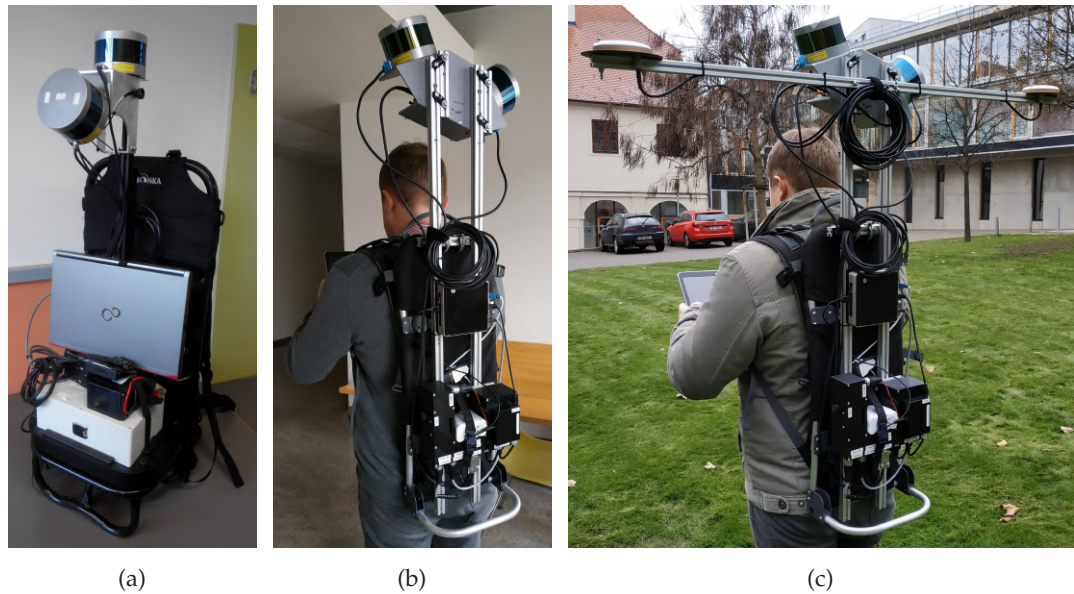


Figure 7.7: The initial (a) and improved (b),(c) prototype of our backpack mapping solution for both indoor (b) and outdoor (c) use. The removable dual GNSS antenna provides precise heading information, aiding for outdoor odometry estimation and also georeferencing of the resulting 3D point cloud model. It should be noted that the position of LiDAR scanners is different in the initial and the later solution. This is elaborated on in the next section.

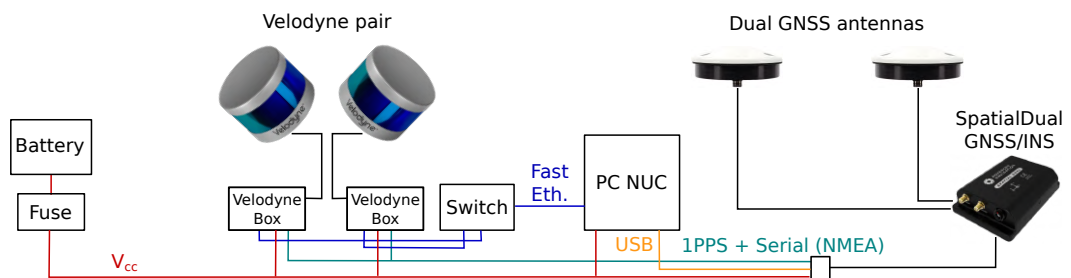


Figure 7.8: Components of the system and the connections. Each Velodyne scanner is connected via a custom wiring “box” requiring power supply (red wires), 1PPS and NMEA synchronization (green) and Fast Ethernet (blue) connection with computer (PC NUC in our case).

The rest of the hardware is responsible for controlling the data acquisition and storing the data (Intel NUC Mini PC), and powering all the components with small Li-Ion battery with capacity 10,400 mAh lasting approximately 2 h.

7.4.2 Dual LiDAR System

During the experiments, we discovered that the limited (30°) horizontal field of view is not an issue for large open spaces. However, when the space is getting smaller and the environment shrinks (e.g., corridors narrower than 2 m), such a field of view causes serious problems, leading to poor accuracy or even total failures of the SLAM system. The worst cases and our solutions are displayed in Fig. 7.6. We experimentally discovered that we need at least two synchronized Velodyne Puck scanners to provide a robust solution that covers both the floor/ceiling and the walls, even in small or narrow rooms.

To achieve good accuracy and to cover the environment, the scanners are mounted perpendicular to the direction of the operator movement—one in horizontal and second in vertical orientation, as displayed in Figures 7.7b,c and 7.6f. All other configurations (e.g., Configuration e.) in our initial prototype in Fig. 7.7a were not able to capture both horizontal and vertical properties of the environment, or did not provide a large coverage necessary for precise pose estimation.

7.4.3 Calibration of the Sensors

To leverage the full potential of using two Velodyne LiDARs, these scanners have to be properly synchronized and calibrated. As mentioned above, the sensors are synchronized via NMEA messages (GPS communication protocol) and 1PPS (Pulse Per Second) signal provided by SpatialDual inertial navigation system. Sufficient intrinsic calibration parameters of LiDAR scanners themselves (corrections) are provided by Velodyne company and processed by the driver (in ROS Velodyne package).

Therefore, the task to solve is the estimation of extrinsic calibration parameters in terms of relative 6DoF pose estimation for both laser scanners C_{V1} , C_{V2} and INS sensor C_I in Fig. 7.9. First, the transformation between the scanners is computed. To do so, two 3D maps of a large indoor space (a large lecture hall in our case) were built by the scanners separately using our previously published method [98]. These two 3D maps are ICP aligned. The resulting 3D geometrical transformation represents mutual position of the sensors $C_{V1}^{-1} * C_{V2}$ and also the alignment of laser data they provide as presented in Fig. 7.10. Since we are interested only in relative transformations between the sensors, the origin can be arbitrarily defined, e.g., as the position of the first Velodyne and $C_{V1} = I$. A single frame point cloud consists of multiple (two in our case) synchronized LiDAR frames and therefore it will be denoted as the *multiframe*.

To be able to use data provided by the INS system, an extrinsic calibration \mathbf{C}_1 between the laser scanners and the INS sensor needs to be estimated. All sensors are fixed on the custom made aluminum mount and therefore the translation parameters can be found in the blueprints of the mount or can be measured with millimeter precision. However, mutual rotation has to be estimated more precisely, because just a fraction of degree misalignment would cause serious errors for long range laser measurements.

We found that the rotation parameters as the transformation between the floor normal vector \vec{n}_i in the point cloud data and the gravity vector \vec{g}_i provided by the INS sensor, since these vectors should be aligned. Points of the floor are selected manually and the normal of the best fitting plane is computed. This can be performed in arbitrary software for visualization and processing of the point clouds—CloudCompare⁹ in our case. We performed multiple measurements for different inclines of the backpack in the indoor corridor with a perfectly straight floor. The final rotation \mathbf{R}_{C_1} between the Velodynes and INS sensor was estimated by SVD (Singular value Decomposition) [74] (Equation (7.2)) of covariance matrix \mathbf{A} of these 3D vector pairs (Equation (7.1)) (floor normal and the gravity). Multiplication with matrix \mathbf{E} (Equation (7.5)) solves the ambiguity between right/left hand rotation—we always compute right-hand representation. Equations (7.1)–(7.5) are based on the work [74].

$$\mathbf{A} = \sum_i \vec{n}_i^T \cdot \vec{g}_i \quad (7.1)$$

$$\mathbf{U}\Sigma\mathbf{V}^* = \mathbf{A} \quad (7.2)$$

$$e = \begin{cases} 1, & \text{if } |\mathbf{V}\mathbf{U}^T| \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (7.3)$$

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e \end{bmatrix} \quad (7.4)$$

$$\mathbf{R}_{C_1} = \mathbf{V}\mathbf{E}\mathbf{U} \quad (7.5)$$

⁹<https://www.danielgm.net/cc/>

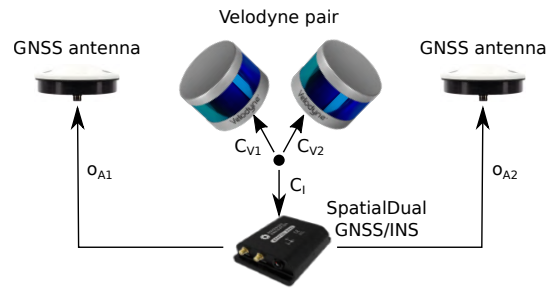


Figure 7.9: Extrinsic calibration required in our system. The mutual positions between the Velodyne scanners and the GNSS/INS unit are computed. The offsets $\vec{o}_{A1}, \vec{o}_{A2}$ of the antennas are tape measured.

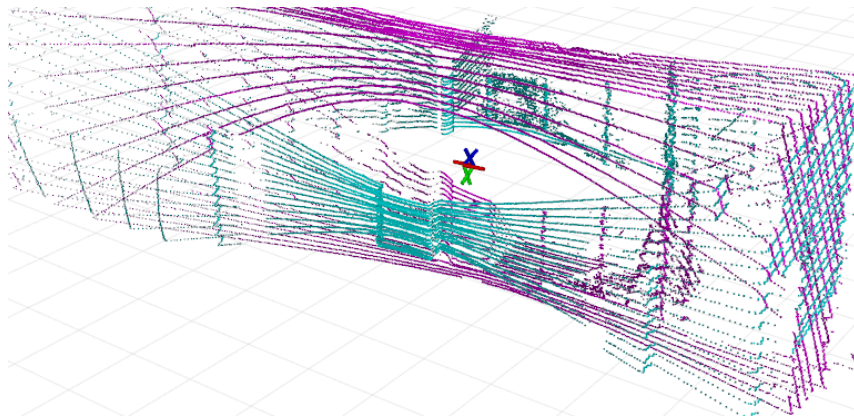


Figure 7.10: Two Velodyne LiDAR frames aligned into the single *multiframe*. This data association requires time synchronization and precise extrinsic calibration of laser scanners.

7.4.4 Point Cloud Registration

The core element of the software part is the alignment of the point cloud data into a 3D map of the environment. There are multiple state-of-the-art approaches for point cloud registration and odometry estimation, including our previously published approach Collar Line Segments [98]. We compared our approach with LOAM [107] algorithm, using the implementation available. The results of this experiment are presented in Table 7.2, which shows the superior accuracy of our method, thus CLS was a natural choice for our mapping backpack solution.

The basic idea of the CLS method is to overcome the data sparsity of 3D LiDAR scanner (e.g., Velodyne) by sampling the data by line segments. The points captured by individual laser beams form so called “ring” structures displayed in Fig. 7.11a. There is a large empty space between these rings and while moving, same places of the scene are not repeatedly scanned, valid matches are missing and the closest point approaches (e.g., ICP) are not applicable. By using CLS, the space

between the rings is also covered and correct matching of structures in the LiDAR frames is enabled.

The environment in the field of view is represented by the set of CLS line segments. They are randomly generated between the neighboring ring points within the azimuthal bin as described in Fig. 7.11a. Since we are using two LiDAR scanners, collar line segments are generated for the scans of each sensor individually. Using the transformation established by extrinsic calibration described in Sec. 7.4.3, line segments are transformed and joined into the single set for each multiframe.

After the sampling is done, matching of the closest line segments is performed. The line segments are extended into the infinite lines, and the closest points between matching lines are used for direct estimation of translation. SVD [74] is used again for estimation of rotation parameters in the same manner, as described in Sec. 7.4.3. This description is only a brief introduction to the CLS method and more information can be found in our previous publication [98].

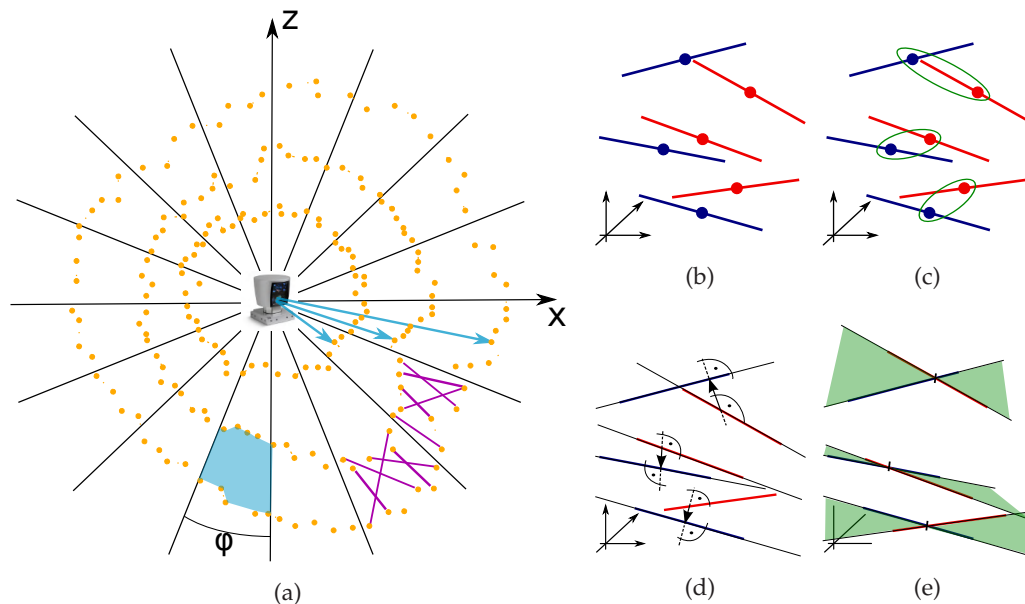


Figure 7.11: The sampling of Velodyne point cloud by the Collar Line Segments (CLS) (a). The segments (purple) are randomly generated within the polar bin (blue polygon) of azimuthal resolution ϕ . The registration process (b–e) transforms the line segments of the target point cloud (red lines) to fit the lines of the source cloud (blue). First, the lines are matched by Euclidean distance of mid-points (c); then, the segments are extended into infinite lines and the vectors between closest points are found (d); and, finally, they are used to estimate the transformation that fits the matching lines into common planes (green in (e)).

7.4.5 Overlap Estimation

This work provides a novel solution for automatic estimation of the core parameter of the CLS approach. Before the transformation is estimated, invalid matches must be discarded. In our previous work, this was done by a simple distance thresholding, or by keeping a certain portion of matches (e.g., 50%). However, using a constant threshold or portion value is not flexible enough. It can cause significant registration misalignments, when invalid matches are used, or insufficient convergence when the valid matches are ignored.

Assuming that an initial coarse alignment is known, we are able to estimate *the overlap* between these frames and use this value as the portion of matches to keep (e.g., for 30% overlap, 30% of best matches are kept). This solution adapts to the specific situation of each pair of LiDAR frames to be registered and leads to a significantly better precision.

The overlap value (Fig. 7.12a) is effectively estimated by *spherical z-buffer* structure [97] in Fig. 7.12b. First, the target cloud is transformed into the source cloud coordinate frame and the $[x, y, z]$ coordinates of all the points are transformed to spherical coordinates ϕ, θ, r (polar angle, elevation angle, and range). Each spherical bin of the z-buffer is assigned with minimal range value from the source point cloud. The minimal value is chosen since unwanted reflections sometimes cause invalid long range measurements and therefore there is the best chance that the minimum range measurement is valid. Then, all the points of target point cloud (also transformed to spherical coordinates) with range below the value in z-buffer (including certain tolerance) are considered to be overlapping points and the ratio to all the points is considered to be the *overlap* value. More formally, if the point p with range p_r within the spherical bin i fulfills the requirement

$$p_r < r_{\min}^i \cdot t_r + t_a, \quad (7.6)$$

it is considered to be a part of the overlap. Value r_{\min}^i denotes the minimal range value stored within the spherical bin. Absolute t_a and relative t_r tolerance values represent the acceptable translation and rotation error. Especially the error of rotation causes larger displacements for larger ranges. Equation (7.6) follows our *error model*, where the error e is the distance between precise point coordinates p (which are unknown) and known erroneous coordinates p^e which can be approximately estimated as:

$$e = |p - p^e| = p_r^e \cdot \text{tg}(e_r) + e_t, \quad (7.7)$$

where e_r represents rotation, e_t is the translation error, and p_r^e is the range of the erroneous point (see also Fig. 7.13). In our experiments, we used the tolerance values $t_r = 0.1$ and $t_a = 0.3$ for the overlap estimation. This allows rotation error e_r

approximately 5° and translation error 30 cm for the initial coarse transformation between the scans.

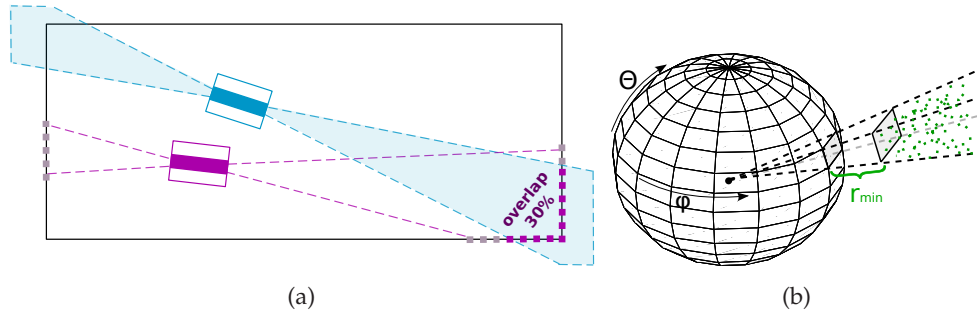


Figure 7.12: The overlap (a) between the source (blue) and the target (purple) LiDAR frame. In this case, approximately 30% of source points are within the view volume of target frame. The view volume can be effectively represented by *spherical z-buffer* (b) where range information (minimum in this case) or the information regarding empty space within the spherical grid is stored.

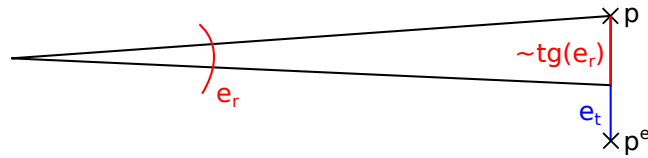


Figure 7.13: The error of measurement (Euclidean distance between points p and p^e) can be split into rotation e_r and translation e_t part. The impact of rotation error $2 \cdot \text{tg}(e_r/2)$ can be simplified to $\text{tg}(e_r)$ due to near linear properties of tangent function for small angles.

7.4.6 Rolling Shutter Corrections

As mentioned in the description of Velodyne sensor, spinning frequency is approximately 10 Hz which leads to 100 ms duration of a single LiDAR scan acquisition. This is a relatively long time when significant movement is assumed. Large translation in the case of fast vehicles or possible fast rotations in case of human carrier can cause distortions in LiDAR frame displayed in Fig. 7.14. We denote this effect as *rolling shutter* because it resembles rolling shutter distortion of optical sensors.

This means that the LiDAR data cannot only be rigidly transformed, but a continuous transformation needs to be applied or at least approximated. The single Velodyne Puck frame consists of approximately 75 packets, each carrying a slice of the frame. Slices are evenly distributed in both time and space. Thus, for each i th frame, we compute the relative transformation $\mathbf{T}_{i \rightarrow j}$ that occurred during the

acquisition of the current frame using the global position \mathbf{P}_i of the current frame and the pose \mathbf{P}_{i+1} of the next one as:

$$\mathbf{T}_{i \rightarrow j} = \mathbf{P}_i^{-1} \cdot \mathbf{P}_{i+1}. \quad (7.8)$$

The correction for each slice is estimated by interpolation of this transformation. The translation parts are interpolated linearly and, for the rotations, Spherical Linear Interpolation (SLERP) [89] over quaternion representation is used. For the first slice, zero transformation is estimated and the last one is transformed by $\mathbf{T}_{i \rightarrow j}$.

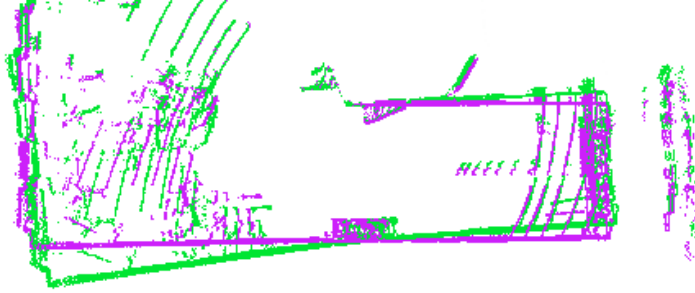


Figure 7.14: Example of a LiDAR frame distorted by the rolling shutter effect when the operator with mapping backpack was turning around (green) and the corrected frame (purple). This is the top view and the distortion is mostly visible on the “bent” green wall at the bottom of this picture.

7.4.7 Pose Graph Construction and Optimization

The proposed CLS method for point cloud alignment can only provide consecutive frame-to-frame registration. However, since each registration is burdened by a small error, after some time, the accumulated error (*drift*) is no longer acceptable. To reduce this drift and also to close loops of revisited places, we propose an iterative process of *progressive pose graph* construction and optimization. The key idea of this algorithm is progressive refinement of odometry estimation from local precision within small time window to global precision across the whole model. This iterative method is described in Fig. 7.15 and more formally in Algorithm 1.

First, only consecutive frames (within neighborhood of size 1) are registered, and then the neighborhood is gradually enlarged (size d in Algorithm 1, step 1) until it covers all N frames. CLS registration is performed for each pair (i th and j th frame) within the current neighborhood where a significant overlap is found and then efficient pose graph optimization using SLAM++ framework [40] is performed. Modulo operator in Step 3 reflects the fact that we assume a circular trajectory. This assumption of beginning and ending the data acquisition process at the same place is common also for other similar solutions (ZEB-1, ZEB-REVO, etc.)

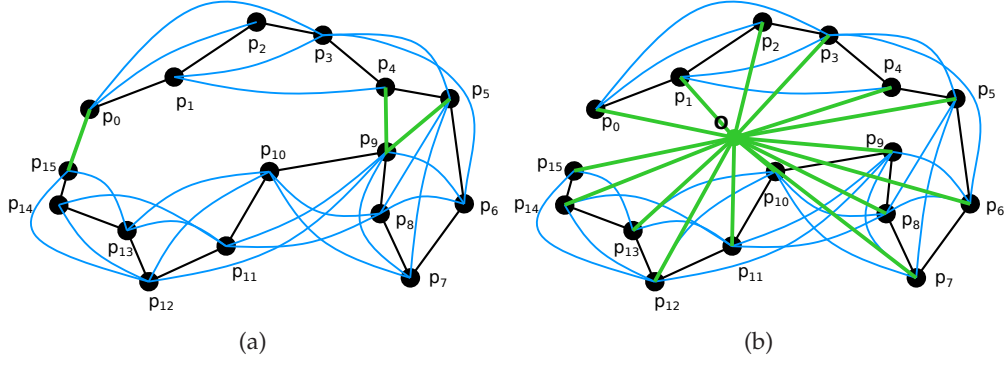


Figure 7.15: Pose graph as the output of point cloud registration and the input of SLAM optimization. The goal is to estimate 6DoF poses $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$ of graph nodes (vertices) p_1, p_2, \dots, p_{15} in the trajectory. The edges represent the transformations between LiDAR frames for given nodes estimated by point cloud registration. Black edges represent transformations between consequent frames, blue edges are for transformations within a certain neighborhood (maximum distance of three frames in this example) and the green edges (in (a)) represent visual loops of revisited places detected by a significant overlap between the given frames. When GNSS subsystem is available (b), additional visual loops are introduced as transformations from the origin \mathbf{O} of some local geodetic (orthogonal NED) coordinate frame.

[30]. It helps the system to identify at least one visual loop that guarantees reasonable results from the global SLAM-based optimization.

Before a pair of frames is registered, the presence of overlap larger than t_o is verified (Line 5 in Algorithm 1) in order to preserve the registration stability. We used minimal 0.5 overlap in our experiments. This also plays the role of visual loop detection every time a place is revisited.

Moreover, after the CLS registration is performed, we verify the result of registration (Line 8) using the error model described in Equation (7.7). As the reference range value, we take the median range of the source point cloud. In our experiments, we used tolerance values $t_r = 0.01$ and $t_a = 0.05$ representing tolerance of approximately 0.5° in rotation and 5 cm in positional error.

For outdoor mapping, the absolute position and orientation are provided by the GNSS/INS subsystem with PPK (Post Processed Kinematics) corrections. While the global error of these poses is small, relative frame-to-frame error is much larger when compared to the accuracy of pure SLAM solution. Therefore, we combine our SLAM (in the same way as described above) with additional edges in the pose graph representing the global position in some geodetic frame, as shown in Fig. 7.15b.

Algorithm 7.1: Progressive refinement of 6DoF poses $\{\mathbf{P}_i\}_{i=1}^N$ for sequence of frames $\{\mathbf{f}_i\}_{i=1}^N$ by optimizing pose graph \mathbf{G} .

```

1: for  $d = 2$  to  $\frac{N}{2}$  do
2:   for  $i = 1$  to  $N$  do
3:      $j := (i + d) \bmod N$ 
4:      $\mathbf{T}_{i \rightarrow j} := \mathbf{P}_j^{-1} \cdot \mathbf{P}_i$ 
5:      $o_{ij} := \text{OVERLAP}(\mathbf{f}_i, \mathbf{f}_j, \mathbf{T}_{i \rightarrow j})$ 
6:     if  $o_{ij} > t_o$  then
7:        $\mathbf{T}_{i \rightarrow j}, e := \text{CLSREGISTRATION}(\mathbf{f}_i, \mathbf{f}_j, \mathbf{T}_{i \rightarrow j}, o_{ij})$ 
8:       if  $e \leq \text{MEDIANRANGE}(\mathbf{f}_i) \cdot t_r + t_a$  then
9:          $\mathbf{G} := \mathbf{G} \cup \{\text{EDGE}(i, j, \mathbf{T}_{i \rightarrow j})\}$ 
10:      end if
11:    end if
12:  end for
13:   $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N = \text{OPTIMIZE}(\mathbf{G})$ 
14: end for
15: return  $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$ 

```

7.4.8 Pose Graph Verification

After the registration is performed, a new edge is added into the pose graph only if the registration error is below a certain threshold modeled by Equation (7.7) (Line 8 of Algorithm 1). However, this simple rejection is not robust enough—some registrations are falsely rejected or accepted. After all overlapping frames are registered, additional verification is performed for all edges.

Expected transformation \mathbf{T}_{ij}^e is computed (Equation (7.9)) using alternative path $\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_{K-1}, \mathbf{T}_K$, as described in Fig. 7.16. The L2 norm of positional difference between expected transformation \mathbf{T}_{ij}^e and the transformation \mathbf{T}_{ij} found by registration (Equations (7.10)–(7.12)) is considered as the error value related to this edge. Note that the positional difference is also affected by the difference in rotation and therefore it is included in this error.

$$\mathbf{T}_{ij}^e = \mathbf{T}_1 \cdot \mathbf{T}_2 \cdot \dots \cdot \mathbf{T}_{K-1} \cdot \mathbf{T}_K \quad (7.9)$$

$$\Delta_{ij} = \mathbf{T}_{ij}^{-1} \cdot \mathbf{T}_{ij}^e \quad (7.10)$$

$$\Delta_{ij} = [\mathbf{R}_{ij} | \mathbf{t}_{ij}] \quad (7.11)$$

$$e_{ij} = \|\mathbf{t}_{ij}\|_2 \quad (7.12)$$

For each edge, all alternative paths up to a certain length are found and their errors are estimated. We use paths of length up to 3 as a tradeoff between the time complexity and robustness. An edge is rejected when the median of these error

values is below accepted threshold (10 cm in our experiments). This cannot be considered as target error of our reconstruction since the pose graph optimization process further decreases the cumulative error. The whole process is repeated until there is no edge to reject.

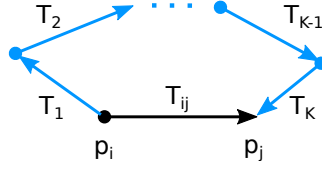


Figure 7.16: Verification of edge (p_i, p_j) representing transformation T_{ij} is performed by comparison with transformation $T_1 \cdot T_2 \dots T_K$ of alternative path (blue) between i th and j th node.

7.4.9 Horizontal Alignment of the Indoor Map

While, for outdoor environment, the model is georeferenced and aligned with NED geodetic coordinate frame (north, east, and down), there is no such possibility when mapping indoors since the GNSS signal is not available. However, practical indoor applications of our 3D mapping solution require at least horizontal alignment—the alignment of gravity vector with Z-axis and the alignment of straight floors/ceilings with XY-plane in resulting 3D model as Fig. 7.17 shows.

This alignment is possible, since roll and pitch angles are provided by IMU (using measurements by accelerometers and gyroscopes) and extrinsic calibration of Velodyne sensors to the IMU frame C_1 estimated as described in Sec. 7.4.3. The simplest solution would be to use these roll and pitch angles directly to align the LiDAR scans individually and deploy the SLAM only to estimate the remaining parameters (heading and translation). Unfortunately, this is not possible because the accuracy of roll and pitch angles is not sufficient—error in order of degrees happens during the motion. Since our goal is to reduce the cost of our solution, we did not want to use additional expensive hardware. We rather propose an alternative approach to estimate horizontal alignment from these noisy measurements.

We can leverage the fact that there are multiple (thousands) of roll/pitch measurements and only a single transformation for horizontal alignment needs to be computed. First, we are able to split each transformation (for each LiDAR frame) estimated by SLAM into the rotation and the translation

$$\mathbf{P}_{\text{SLAM}} = [\mathbf{R}_{\text{SLAM}} | \mathbf{t}_{\text{SLAM}}]. \quad (7.13)$$

Our partial goal is to estimate horizontal alignment \mathbf{A}_h fulfilling Equation (7.14). The transformation of point cloud data X by SLAM rotations \mathbf{R}_{SLAM} and horizontal alignment \mathbf{A}_h is the same, as the transformations of these data by IMU

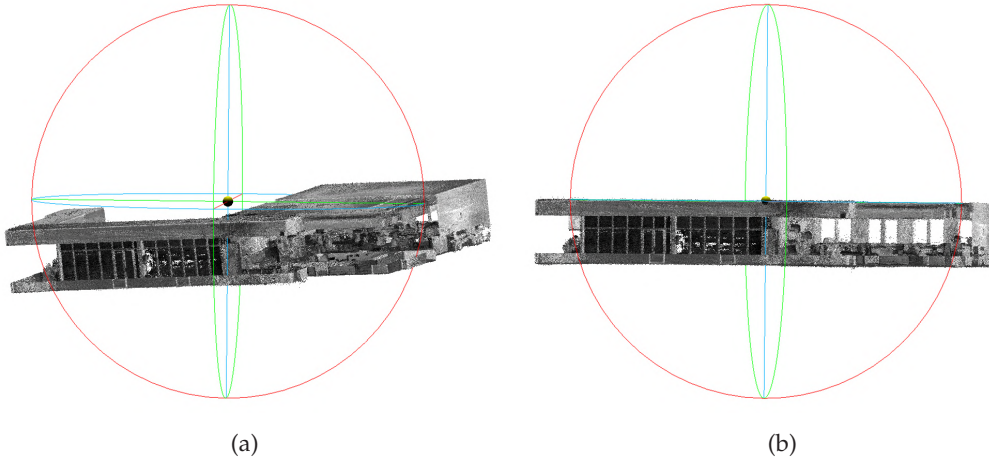


Figure 7.17: The reconstruction built by our SLAM solution before (a) and after (b) the alignment of horizontal planes (floor, ceiling, etc.) with XY plane (blue circle).

measured rotation \mathbf{R}_{IMU} (including the calibration \mathbf{C}_I). In addition, each rotation (SLAM or IMU provided) can be split into roll \mathbf{R}^R , pitch \mathbf{R}^P and heading \mathbf{R}^H (Equation (7.15)). Since the IMU sensor is not able to provide accurate heading information indoors, we supplement the heading $\mathbf{R}_{\text{SLAM}}^H$ estimated by SLAM.

$$\mathbf{R}_{\text{IMU}} \cdot \mathbf{C}_I \cdot \mathbf{X} = \mathbf{A}_h \cdot \mathbf{R}_{\text{SLAM}} \cdot \mathbf{X} \quad (7.14)$$

$$\mathbf{R}_{\text{SLAM}}^H \cdot \mathbf{R}_{\text{IMU}}^P \cdot \mathbf{R}_{\text{IMU}}^R \cdot \mathbf{C}_I = \mathbf{A}_h \cdot \mathbf{R}_{\text{SLAM}} \quad (7.15)$$

$$\mathbf{A}_h = \mathbf{R}_{\text{SLAM}}^H \cdot \mathbf{R}_{\text{IMU}}^P \cdot \mathbf{R}_{\text{IMU}}^R \cdot \mathbf{C}_I \cdot \mathbf{R}_{\text{SLAM}}^{-1} \quad (7.16)$$

Using Equation (7.16), we are able to estimate the (noisy and inaccurate) horizontal alignment \mathbf{A}_h for each pair of SLAM and IMU provided rotations of the same timestamp. During the mapping, there are usually thousands of these pairs (10 pairs per second) which are synchronized. The precise horizontal alignment is then computed by averaging the quaternions [65] representing noisy partial alignments \mathbf{A}_h .

7.4.10 Intensities Normalization

Another quality we would like to introduce into the 3D model is the approximate surface “color” information to improve the ability of visual recognition of various objects (inventory, signs, etc.). To avoid additional HW, and preserve invariance to illumination conditions, we use the laser return intensity. However, these intensity values cannot be directly considered as surface reflectivity, since they are affected by various additional factors such as angle of incidence, range of the measurement or gain of the particular laser beam. These factors were reported by previous works [42, 45, 44] and also confirmed by our experiments in Fig. 7.18.

Previously published works propose various closed-form solutions of intensity normalization for long range measurements (over 10 m) [42, 45, 44]. However, this is not applicable for smaller indoor environments and therefore we propose an alternative solution. If the normalized intensity represents only the surface reflectivity, there should be no dependency on other factors and probability distribution of the intensities should be the same for different laser beams, angles of incidence, or ranges.

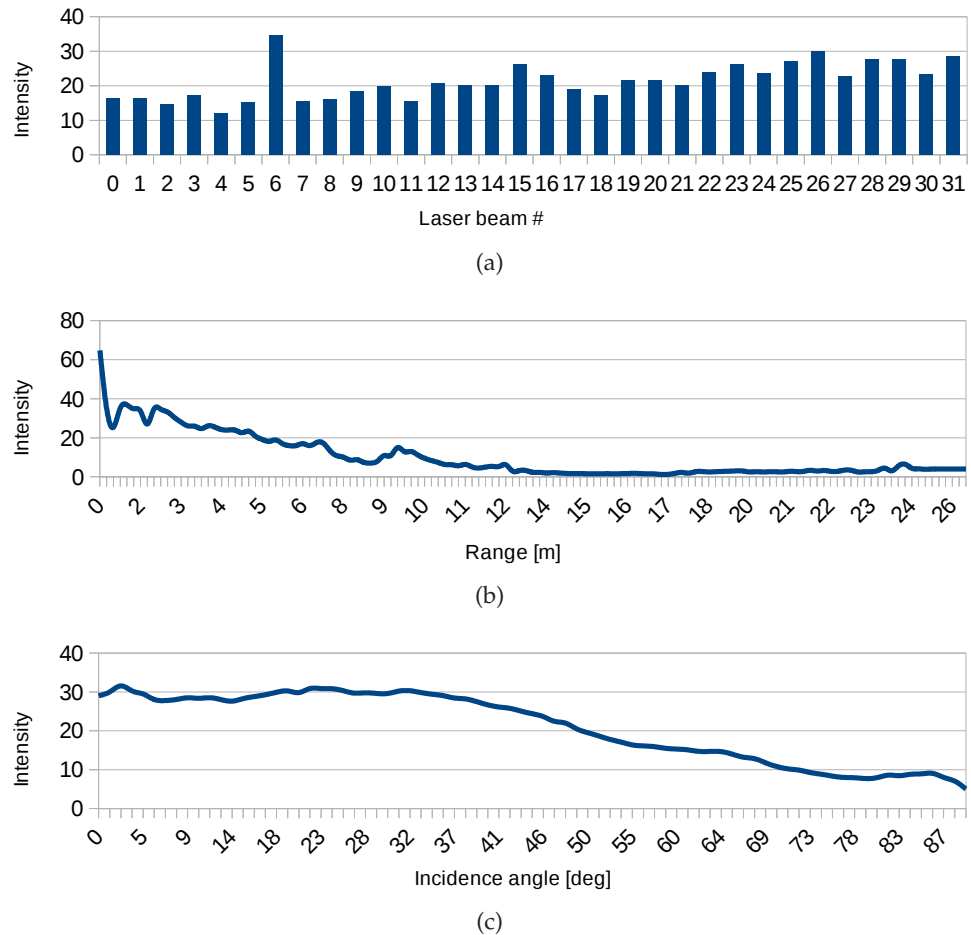


Figure 7.18: The dependency of laser return intensity on: the source beam (a); range of the measurement (b); and the angle of incidence (c). We are using 2 LiDAR scanners with 16 laser beams per each scanner, 32 beams in total.

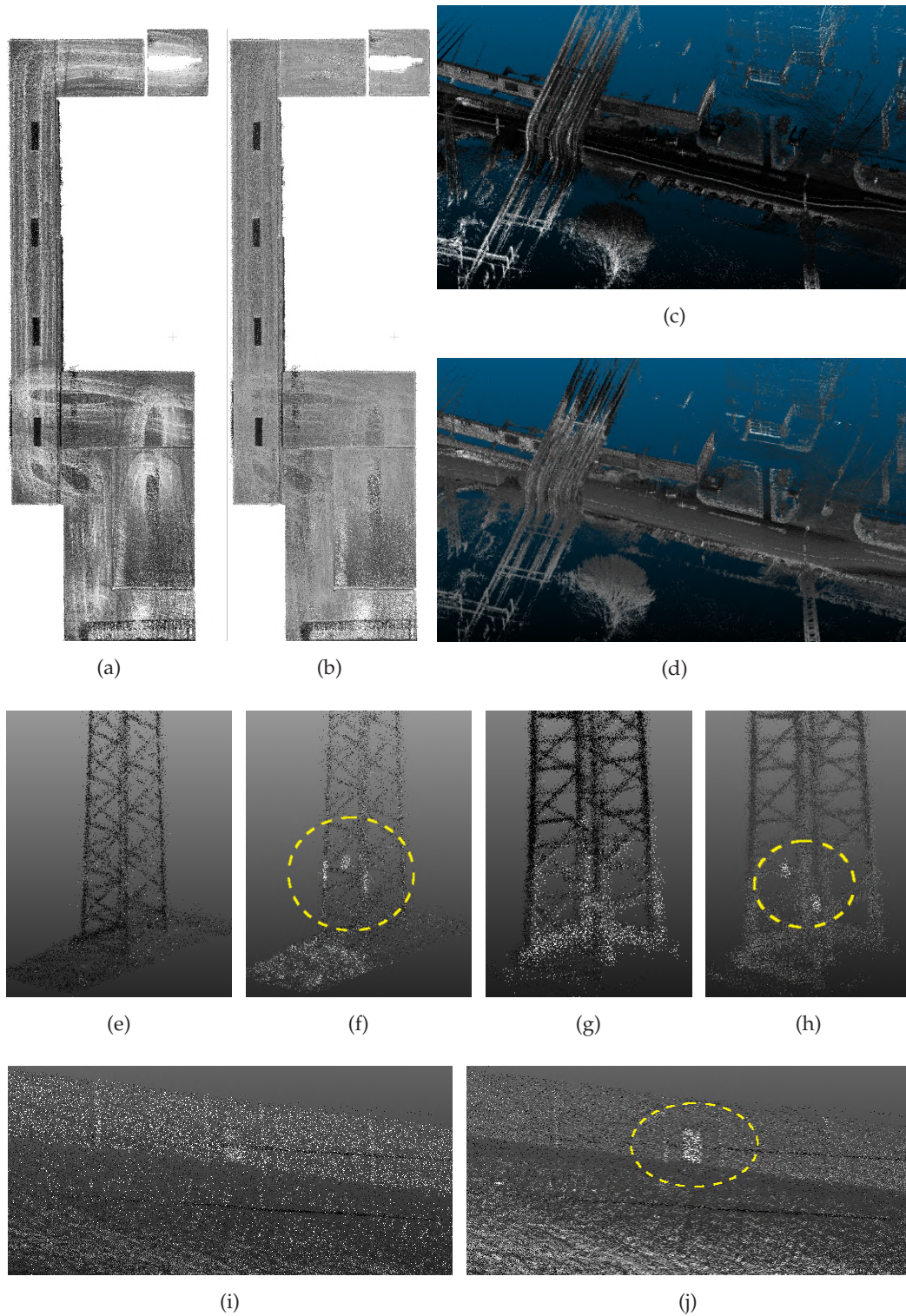


Figure 7.19: Results of 3D reconstruction without (a), (c), (e), (g), (i) and with (b), (d), (f), (h), (j) the normalization of laser intensities. One can observe more consistent intensities for solid color ceiling (b) reducing the artifacts of trajectory, while preserving the contrast with ceiling lights. Besides the consistency, normalization of intensities reduces the noise (d). The most significant improvement is the visibility of important objects e.g., markers at the electrical towers (f), (h) or emergency exit doors (j) at the highway wall. All these objects can not be found in the original point clouds (e), (g), (i).

Therefore, we discretize the space of ranges and angles with some small resolution (e.g., 20 cm and 1°, respectively) and we distribute all the points of the point cloud model into a 3D grid based on the source beam ID (already discrete), the angle of incidence and the range. Our goal is to achieve that the intensity probability distribution will be the same for each bin of points. Assuming normal distribution of surface reflectivities (“colors”), the same target distribution $\mathcal{N}(\mu, \sigma^2)$ will be achieved within each bin by a simple transformation:

$$\mathcal{N}(\mu, \sigma^2) = \mathcal{N}(\mu_i, \sigma_i^2) \cdot \frac{\sigma}{\sigma_i} + (\mu - \mu_i), \quad (7.17)$$

where $\mathcal{N}(\mu_i, \sigma_i^2)$ is the original distribution of laser intensities within i th bin.

There are no ground truth data to perform any objective evaluation of our proposed method for intensity normalization. We are only able to compare the results of 3D reconstruction with and without the normalization. Examples of results can be found in Fig. 7.19.

7.5 EXPERIMENTS

This section presents mapping results of our system in various scenes and scenarios—outdoor environments where GNSS is available, indoor scenes with GNSS denied, small rooms, staircases, and a narrow corridor. A usable and precise solution must avoid so called “double walls” described in Fig. 7.3, which are a typical issue in 3D reconstructions causing ambiguity. Unfortunately, evaluation of such duplicities cannot be performed automatically, thus the operator (a certified geodesist) verified the reconstructions for us by inspecting multiple slices across the model. Moreover, the data density and point coloring by the intensity readings are required for better visual recognition of various objects in the environment. All the raw data collected by our backpack solution, and also the 3D reconstructions used in this evaluation, are publicly available¹⁰.

Regarding the precision, our goal is to achieve 5 cm relative precision (e.g., distance of the point from ground truth) denoted as e_r . For outdoor environments, there are also constraints for absolute error e_a in global geodetic frame. The average of this absolute error is required to be below 14 cm for position in horizontal plane and 12 cm for height estimation. However, the constraints for maximal error are set to double of these values—up to 28 cm for horizontal and 24 cm vertical error. These values were obtained through consultation with experts in the field of geodesy and follow the requirements for creating the building models, outdoor vector maps, inventory check, etc. Global error constraints are applicable only

¹⁰<http://www.fit.vutbr.cz/~ivelas/files/4RECON-dataset.zip>

outdoors, where some global positioning system is available. To sum up, in this section, we show that our solution provides:

- sufficient relative precision e_r under 5 cm;
- global absolute error e_a within the limits described above;
- data density and coloring by normalized intensities for visual inspection; and
- data consistency without ambiguity (no dual walls effects).

7.5.1 Comparison of Point Cloud Registration Methods

We compared our previously published CLS method [98] with different modes (online and offline) of state-of-the-art method LOAM [107] using the data of KITTI Odometry Suite [28] providing both the Velodyne LiDAR data and ground truth poses. The error metrics used in this evaluation are defined by the KITTI dataset itself. The data sequences are split into subsequences of 100, 200, ..., 800 frames (of 10, 20, ..., 80 s duration). The error e_s of each subsequence is computed as:

$$e_s = \frac{\|\mathbf{E}_s - \mathbf{C}_s\|_2}{l_s}, \quad (7.18)$$

(provided by [28]) where \mathbf{E}_s is the expected position (from the ground truth) and \mathbf{C}_s is the estimated position of the LiDAR where the last frame of subsequence was taken with respect to the initial position (within given subsequence). The difference is divided by the length l_s of the followed trajectory. The final error value is the average of errors e_s across all the subsequences of all the lengths.

		Error e_s (7.18)			
Sequence	Length	LOAM Online	LOAM Offline	CLS Single	CLS Multi-Frame
0	4540	0.052	0.022	0.022	0.018
1	1100	0.038	0.040	0.042	0.029
2	4660	0.055	0.046	0.024	0.022
3	800	0.029	0.019	0.018	0.015
4	270	0.015	0.015	0.017	0.017
5	2760	0.025	0.018	0.017	0.012
6	1100	0.033	0.016	0.009	0.008
7	1100	0.038	0.019	0.011	0.007
8	4070	0.035	0.024	0.020	0.015
9	1590	0.043	0.032	0.020	0.018
Weighted average	2108	0.043	0.029	0.022	0.017

Table 7.2: Comparison of visual odometry error for SoA method LOAM and our CLS method. The experiments were performed on KITTI Odometry dataset [28]. For CLS, frame to frame (single) or frame to multiple (10) neighboring frames (multi-frame) registrations without any loop closures were performed. In LOAM experiments, both the original online version (providing real time performance) and offline version (with full procedure for each frame omitting approximations) was used. In all data sequences, except the short sequence No. 4 where the car drives only forward without any turns, our multi frame approach outperformed the LOAM solution.

The experiment is summarized in Table 7.2 and it leads to the conclusion that our CLS approach outperforms LOAM with approximately 1 cm lower drift per 1 m of trajectory elapsed. For clarification, LOAM can run in two different modes. In the online mode (10 fps), mapping is skipped for a certain number of frames, which are only roughly aligned. In the offline mode, which is approximately $3\times$ slower, every frame undergoes the full mapping procedure.

The precision of our method was estimated for frame-to-frame approach, where only consequent frames were registered, and also for the scenario, where each frame is registered with all other frames within a small neighborhood (10 neigh-

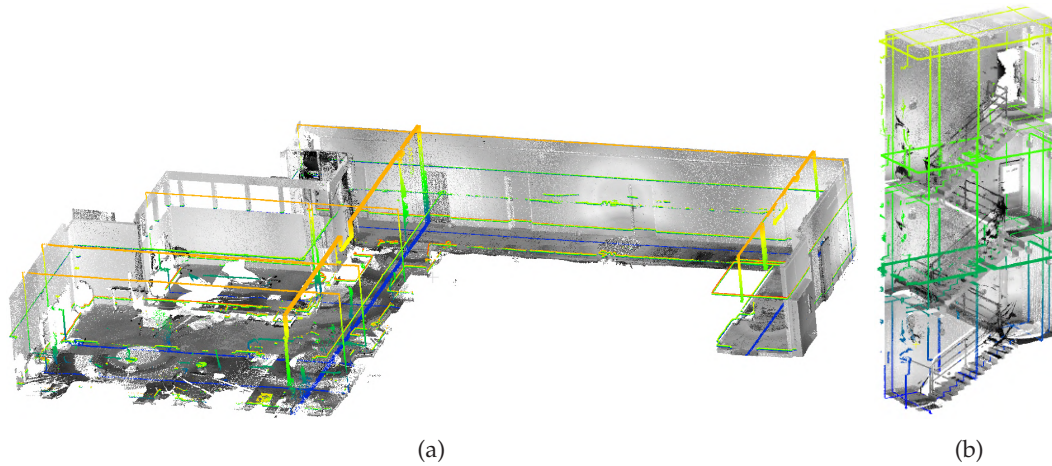


Figure 7.20: Experimental environments Office (a) and Staircase (b), and the highlighted slices that were used for precision evaluation.

boring frames used in this experiment). In this experimental multi-frame approach, the final pose is estimated by simple averaging.

In our previous publication [98], the superior performance of CLS over GICP method (Generalized ICP) [88] was presented, too. All these evaluations led to the choice of CLS for the LiDAR frames registration in our 4RECON backpack solution.

7.5.2 Indoor Experiments

For indoor evaluation of our system, we chose two different environments—the office and staircase in Fig. 7.20—where our partner company has already performed 3D mapping using different laser scanners and generously provided the accurate output models to us. The reconstructions from static *FARO* scanner achieving very high accuracy (in order of millimeters) were used as the ground truth. The same strategy has been already used for evaluation of other mapping systems [66, 90, 64]. For the office environment only, also the 3D reconstruction created by *ZEB-1* solution was provided to us. This allowed us to compare our solution in terms of accuracy, data density, model usability and completeness.

To evaluate the relative error, all the models of the same environment provided by different scanners (*FARO*, *ZEB-1*, and our solution 4RECON) were aligned using ICP. As displayed in Fig. 7.20, several reference slices (8 slices per model, 16 slices in total) were created for the evaluation of precision. Within each slice, the average error (in Table 7.3) was estimated as the average distance of the 3D points to the ground truth model created by the *FARO* scanner. Our solution achieved approximately 1.5 cm relative error on average, which is only slightly worse result than 1.1 cm error for *ZEB-1* that is burdened by the multiple limitations listed

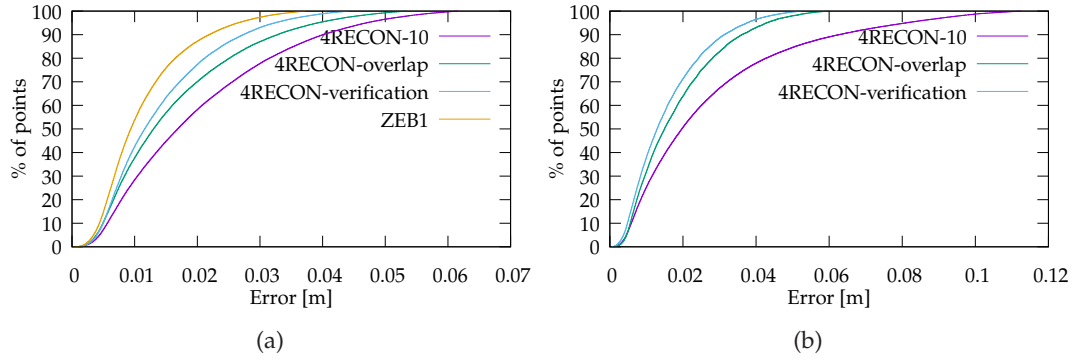


Figure 7.21: Error e_r distribution (the amount of the points within certain error) for our system $4RECON$ and ZEB-1 product. The experiments were performed for all test slices in Fig. 7.20 on Office (a) and Staircase (b) dataset. Note that the model built by ZEB-1 was not available and therefore the evaluation is missing.

below in this section. Moreover, we provide information about the distribution of displacement relative error in Fig. 7.21. The error was estimated for ZEB-1 and different modes of our system:

- in $4RECON-10$, the registrations were performed only within small neighborhood of 10 nearest frames (1 s time window) and reflects the impact of accumulation error;
- for $4RECON-overlap$, the registrations were performed for all overlapping frames as described in Sec. 7.4.7 reducing the accumulation error by loop closures at every possible location; and
- pose graph verification (see Sec. 7.4.8) was deployed in $4RECON-verification$, yielding the best results with good precision and no ambiguities.

Both ZEB-1 and our solution including pose graph verification achieved sufficient accuracy below 5 cm. Moreover, the precision of 2 cm was fulfilled for more than 70% of data. Slightly better precision of ZEB-1 solution was achieved thanks to the Hokuyo sensor with $4\times$ higher scanning frequency while preserving much lower vibrations compared with Velodyne LiDAR.

Dataset	Slice #	4RECON- 10	4RECON- Overlap	4RECON- Verification	ZEB-1
Office	1	2.50	1.71	1.49	1.44
	2	1.97	1.47	1.31	1.06
	3	1.70	1.75	1.55	1.22
	4	1.82	1.54	1.31	1.22
	5	1.93	1.63	1.53	1.44
	6	2.13	1.49	1.47	1.29
	7	2.09	1.68	1.37	0.97
	8	2.07	1.36	1.37	1.31
	Average e_r (cm)	2.01	1.62	1.41	1.14
Staircase	1	3.23	2.11	1.81	-
	2	3.99	1.87	1.60	-
	3	2.63	1.65	1.61	-
	4	2.74	1.71	1.53	-
	5	2.42	1.68	1.50	-
	6	2.98	2.67	1.67	-
	7	1.76	1.75	1.29	-
	8	1.82	1.67	1.56	-
	Average e_r (cm)	2.74	1.82	1.57	-

Table 7.3: Relative error e_r of our method and ZEB-1 product within selected slices visualized in Fig. 7.20. Presented values are average displacements (cm) of the points comparing with the ground truth point cloud obtained by FARO static scanner. The results are missing for ZEB-1 and Staircase dataset since there was no reconstruction using this scanner available.

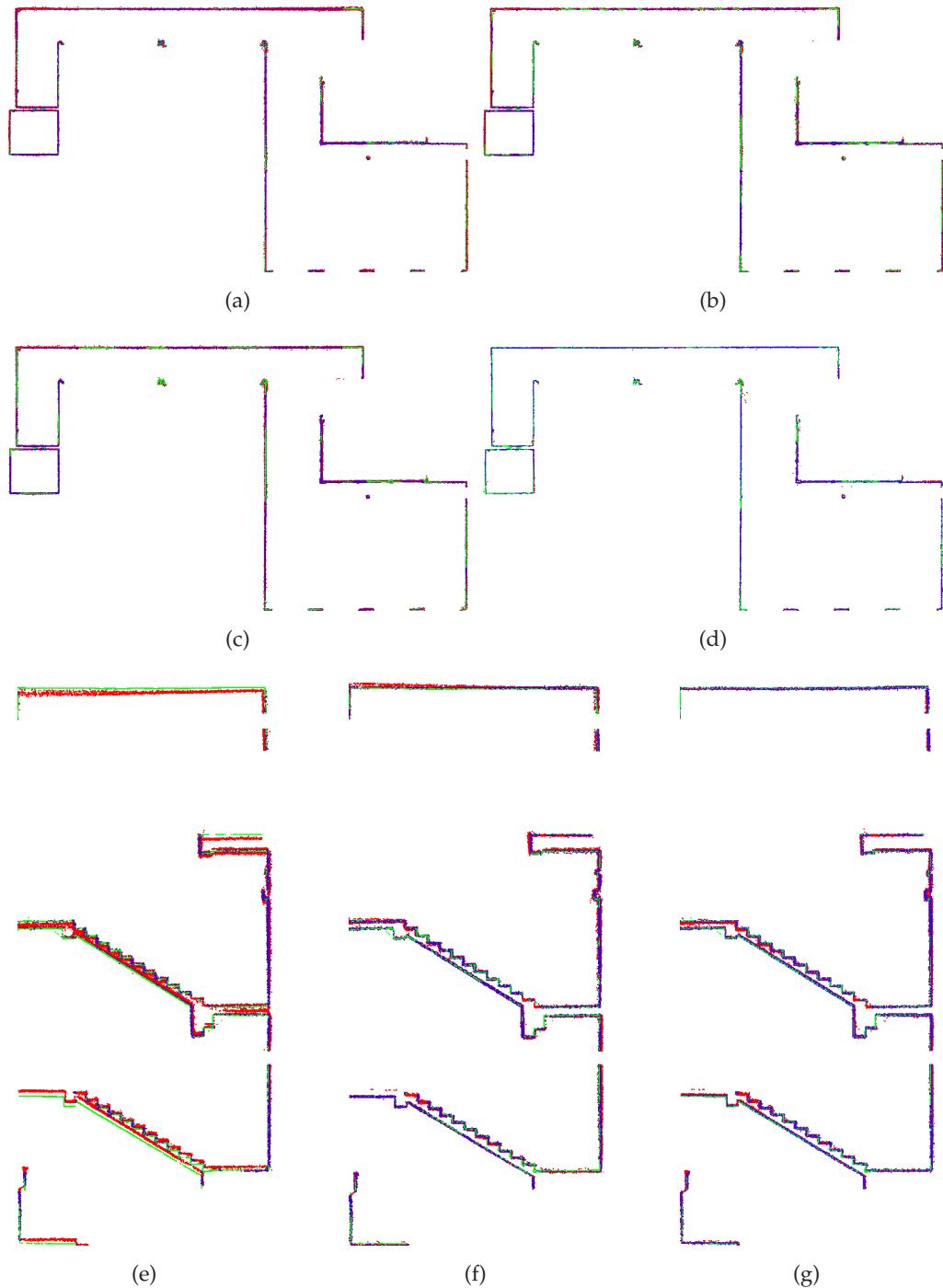


Figure 7.22: Color coded errors within the horizontal reference slice of the Office dataset (a)–(d) and vertical slice in Staircase dataset (e)–(g). Blue color represents zero error, red color stands for 10 cm error and higher. The ground truth FARO data are displayed in green. The results are provided for 4RECON-10 (a,e), 4RECON-overlap (b,f), 4RECON-verification (c,g), and ZEB-1 (d). For Office dataset, there are no ambiguities (double walls) even without visual loop detection while both loop closure and pose graph verification is necessary for more challenging Staircase dataset to discard such errors. Moreover, one can observe that ZEB-1 solution yields lower noise reconstruction thanks to the less noisy Hokuyo LiDAR.

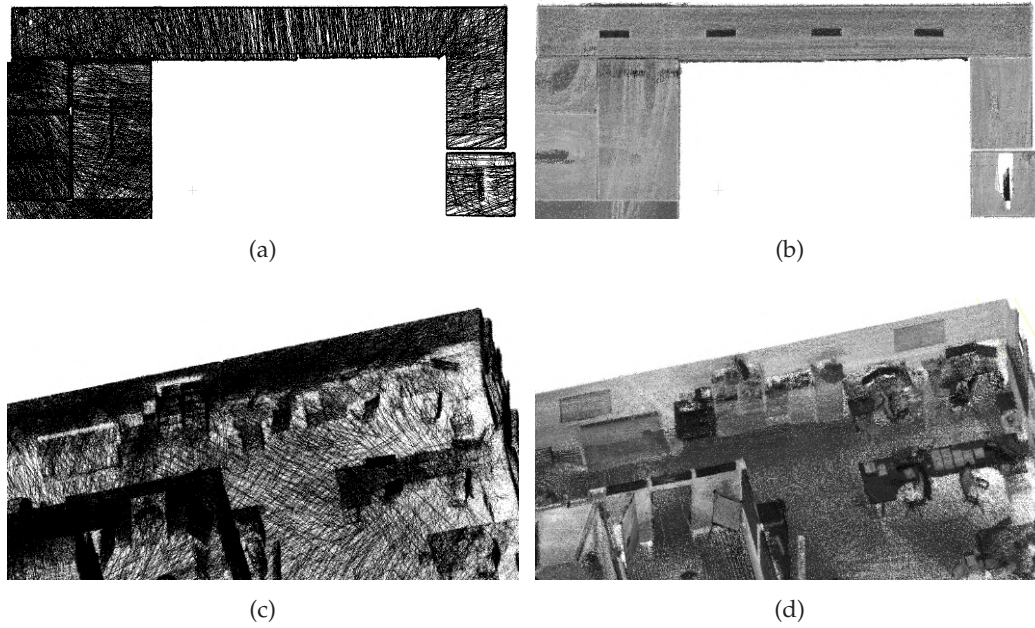


Figure 7.23: The comparison of data density provided by ZEB-1 (a,c) and our (b,d) solution. Since the ZEB-1 solution is based on the Hokuyo scanner, the laser intensity readings are missing and data density is much lower compared with our solution. Multiple objects which can be distinguished in our reconstruction (lamps on the ceiling in the top, furniture and other equipment in the bottom image) are not visible in the ZEB-1 model.

Fig. 7.22 also shows the precision within representative slices—horizontal slice for Office dataset and vertical slice across model of Staircase. These slices demonstrate the noise within data coming from different sensors—Hokuyo LiDAR for ZEB-1 solution and Velodyne for our 4RECON system—and also the precision for different modes of operation. For Staircase dataset, the necessity of pose graph optimization is also demonstrated.

Our evaluations show that the precision of our 4RECON backpack is comparable to the solution ZEB-1 while fulfilling basic requirement for relative error below 5 cm. Note that the error values are also comparable (and in some cases better) to the precisions of other solutions in Table 7.1. In our solution, higher noise can be observed comparing with ZEB-1. This corresponds with higher error values and it is the main reason for little lower accuracies.

However, it is important to point out two most significant advantages of our solution comparing with ZEB solutions. First, our solution is *usable in vast open spaces* with fewer and more distant featuring objects, as is demonstrated in the next sections. In indoor environments featuring objects at distances significantly larger than 15–20 m [29], ZEB solutions based on the Hokuyo sensor fail.

Second, our Velodyne-based solution is able to provide much higher *data density, map completeness and visibility of objects* in the scene. We chose two large surfaces

(the ceiling and the side wall in Fig. 7.23) with 230 m² in total area. Models of these surfaces created by ZEB-1 solution achieved average data density 0.9 points per cm² (2.2 million points in total). Models created by our 4RECON backpack consist of more than 23 million points, achieving much higher data density—10.1 points per cm². Better visibility of objects in Fig. 7.23 is achieved thanks to the laser intensity readings provided by Velodyne sensor and employing our normalization process as described in the Sec. 7.4.10. This might appear to be only a “cosmetic” property, but the visibility of the construction elements, equipment, furniture, etc. in the scene is important for usability in real applications—e.g., an operator needs to distinguish between the window and the blackboard.

7.5.3 Outdoor Experiments

Our system is a universal solution—both for indoor scenes, where the usability was proven by the previous section, and for outdoor scenes, including vast open ones. We tested and evaluated our system during a real task—high voltage lines mapping and measurement. The area of interest, including the details of some important objects, is visualized in Fig. 7.24. The main goal of this mission was position estimation of electric pylons (including footprint of the base, total height and the positions of the wire grips) and the heights and the hangings of the wires. Fig. 7.24 shows that these details can be recognized in the 3D model. The usability of our 3D reconstructions was also confirmed by the geodetic company we asked for manual data inspection and evaluation.

In the same way as during the indoor mapping, the ambiguities in multiple instances of objects disqualifies the reconstructions to be used in practical geodetic measurements. Such error in comparison with the desired result of the reconstruction is shown in Fig. 7.25. Multiple instances of the same object, blurred and noisy results were successfully avoided by our solution (see Figures 7.24 and 7.25).

Since our solution integrates precise GNSS/INS module for outdoor scenarios, the model is *georeferenced*—the coordinates of all the points are bound in some global geodetic frame.

To verify the absolute positional accuracy of our model, we performed precise measurements on so-called survey markers. This is commonly used technique to verify the precision of resulting maps (including 3D maps). Precise positions of the survey markers are estimated using specialized geodetic GNSS system, which is placed statically on the survey point for several seconds, until the position converged. The precision up to 2 cm is achieved using RTK (Real Time Kinematics) which are received online via internet connection.

Survey markers (Fig. 7.26a) are highlighted using high-reflective sprays. Thanks to the coloring of point cloud by laser intensities, these markers are also visible in the reconstructions as can be seen in Fig. 7.26b.

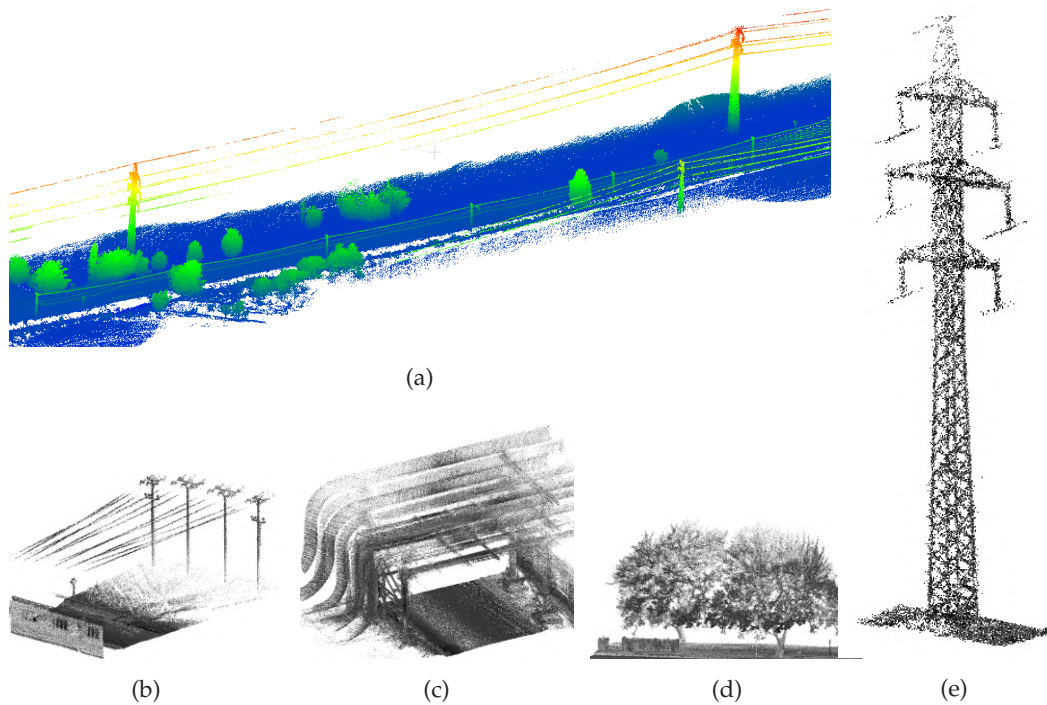


Figure 7.24: The example of 3D reconstruction of open field with high voltage electrical lines (a). The model is height-colored for better visibility. The estimation of positions and height of the lines (b), towers (e), etc. was the main goal of this mapping task. The other elements (c,d) in the scene are shown for demonstration of the reconstruction quality.



Figure 7.25: Example of ambiguities caused by reconstruction errors (a), which disqualifies the model to be used for practical measurements. We obtained such results when we used only poses provided by GNSS/INS subsystem without any refinements by SLAM or point cloud registration. Our solution (including SLAM) provides valid reconstructions (b), where both towers and wires (in this case) can be distinguished.

The evaluation in Table 7.4 shows that our 3D mapping for 0.5 km test track fulfills the requirements for absolute error, as described at the beginning of this section—average error below 14 cm for position in horizontal plane and 12 cm for

height estimation and maximal error up to 28 cm and 24 cm, respectively (double values of expected average error).

Thanks to the ability of point cloud coloring by laser intensities, it is possible to also run such evaluation for the validation of each 3D model, which should be used in real application. This is also an important quality, since there are requirements for double measurements in geodesy to ensure that the accuracy is sufficient.

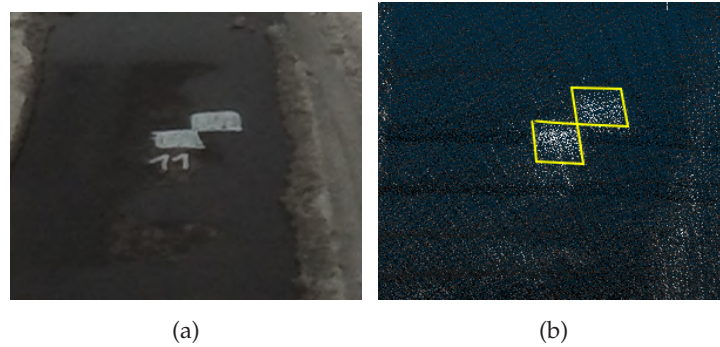


Figure 7.26: Geodetic survey markers painted on the road (a) is also visible in the point cloud (b) thanks to the coloring by laser intensities.

Table 7.4: Errors measured (cm) on geodetic survey marker points at the beginning and at the end of survey track. The distance between the control points is 523 m.

Ref. Point	dX	dY	Horizontal Error	dZ (Vertical)	Total Error e_a
1	-5.9	-1.2	6.0	-15.2	16.3
2	-5.6	0.5	5.6	-4.7	7.3

7.5.4 Comparison of Single and Dual Velodyne Solution

Finally, we compared the robustness of our dual LiDAR solution over the system with single LiDAR only. We computed reconstructions of the Office environment using our solution with two synchronized and calibrated LiDARs (one aligned vertically and second horizontally) in Figure 7.27a,b and also using only single LiDAR—horizontally (Figure 7.27c,d) or vertically aligned (Figure 7.27e,f).

Our evaluation shows that the dual LiDAR solution provides a valid reconstruction. However, the solution with horizontal LiDAR only is not able to provide vertically correct alignment (Figure 7.27d), and vice versa, the solution with vertical LiDAR is horizontally misaligned (Figure 7.27e).

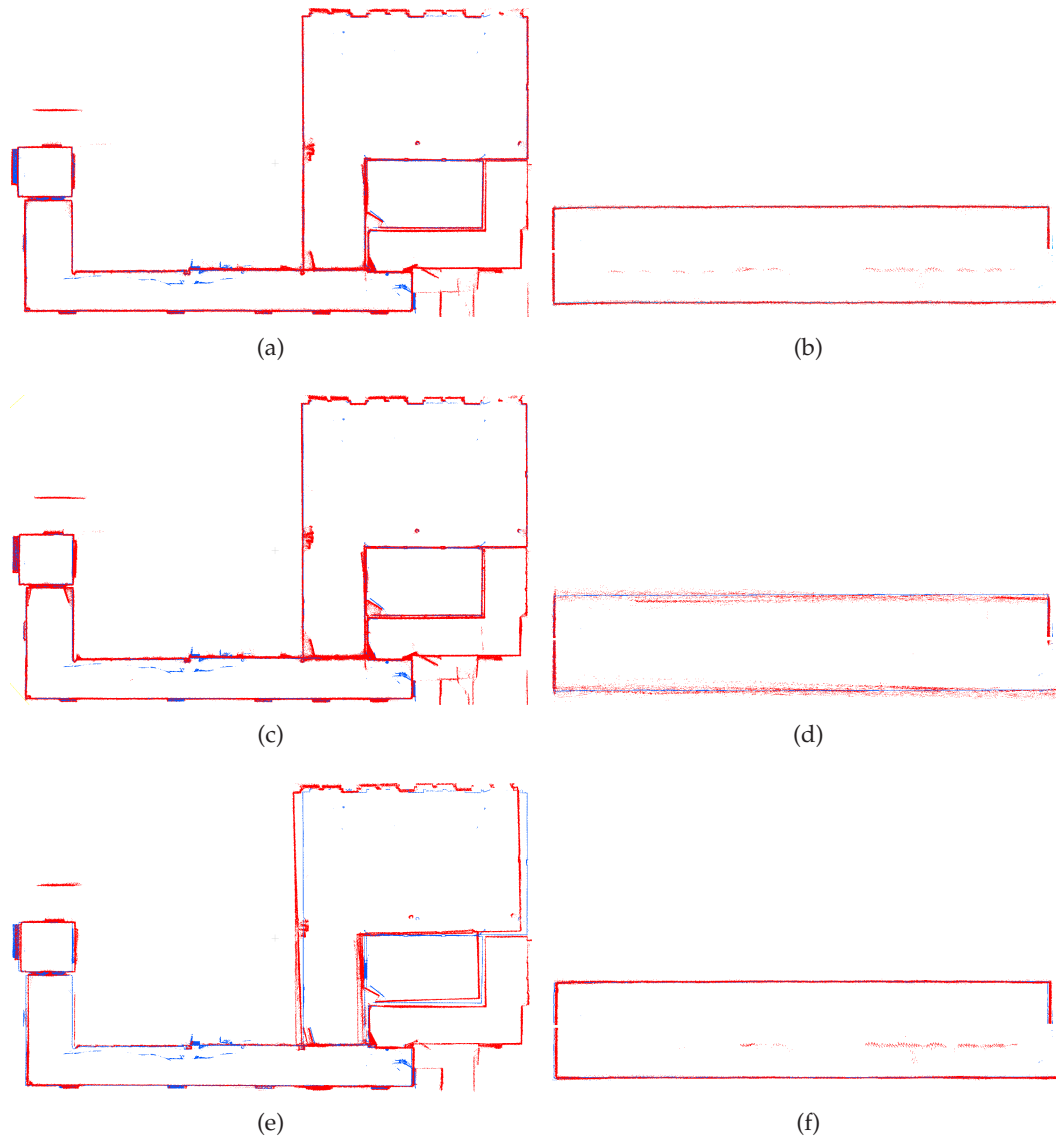


Figure 7.27: Comparison of reconstructions provided by dual LiDAR system—floor plan top view (a) and side view of the corridor (b)—with the reconstruction built using only single horizontally (c,d) or vertically (e,f) positioned Velodyne LiDAR. The reconstructions are red colored with ground truth displayed in blue.

7.6 DISCUSSION

When we look on our 4RECON mapping backpack in the context of the other available solutions (see overview in Table 7.1), we can summarize its advantages and disadvantages.

Comparing to the ZEB products, our backpack achieves much higher data density, better visibility of the objects in the resulting model, higher comfort of data acquisition, and, most importantly, usability also in the outdoor featureless open spaces, including the option of georeferencing the reconstructed point map. How-

ever, we must admit that ZEB scanners achieve better accuracy and lower noise in the models of indoor environments.

In terms of universality of the usage, our solution also outperforms Robin and Akhka backpacks, which require GNSS readings and therefore indoor scanning is not possible. For outdoor tasks, Robin achieves better precision than our 4RECON backpack, but it is also important to point out the very high price of the Robin solution.

Laser mapping backpacks Pegasus, Viametris bMS3D and LiBackpack can be considered as the most similar solutions to our work. All these systems claim precision up to 5 cm, which is also the accuracy of 4RECON (according to the evaluation in Fig. 7.20). The advantages of these solutions are more professional design and the presence of additional RGB cameras (for Pegasus and Viametris backpacks). The integration of panoramic RGB camera into our backpack is the plan for future work. Our solution on the other side provides open SLAM method in comparison with the proprietary solutions deployed in these backpacks, and also potentially much lower price.

7.7 CONCLUSIONS

This paper presents a dual LiDAR system for mobile mapping. Our solution can be easily carried as a backpack together with a reliable dual antenna GNSS/INS system. This leads to the universality of its usage. In small or narrow indoor environments with many obstacles, two LiDAR sensors increase the field of view. On the other side, in open outdoor spaces with lack of features, the reliable positional subsystem keeps the result accurate.

Thanks to the type of LiDARs used, our solution also brings multiple other beneficial properties: data density, map completeness and coloring by laser intensities normalized by our novel algorithm. The intensities enables better visual recognition of the elements in the scene as well as the visibility of geodetic survey markers for checking the model validation.

The proposed solution was evaluated in both indoor and outdoor scenarios. During the mapping of the office or staircase environment, our solution fulfilled the requirement of error below 5 cm and achieved a similar precision as solution ZEB-1. The average error in terms of the points displacements is approximately 1.5 cm. For outdoor experiments, our reconstruction met the requirements for absolute precision with 11.8 cm average error in the global geodetic frame. This proves higher universality of our mapping backpack compared to the previous ZEB-1 solution. In all our experiments, data consistency was preserved and unambiguous models were built.

Part IV

SEMANTIC GROUND SEGMENTATION

This chapter is based on the paper [96].

CNN FOR VERY FAST GROUND SEGMENTATION IN VELODYNE LIDAR DATA

8.1 ABSTRACT

This paper presents a novel method for *ground segmentation in Velodyne* point clouds. We propose an encoding of sparse 3D data from the Velodyne sensor suitable for training a *convolutional neural network (CNN)*. This general purpose approach is used for segmentation of the sparse point cloud into ground and non-ground points. The LiDAR data are represented as a multi-channel 2D signal where the horizontal axis corresponds to the rotation angle and the vertical axis represents channels – laser beams. Multiple topologies of relatively shallow CNNs (i.e. 3-5 convolutional layers) are trained and evaluated, using a manually annotated dataset we prepared. The results show significant improvement of performance over the state-of-the-art method by Zhang et al. in terms of *speed* and also minor improvements in terms of accuracy.

8.2 INTRODUCTION

Recent development in exploration and 3D mapping of the environment surrounding a mobile robot aims at techniques which capture semantic information besides the simple geometrical properties. The analysis of scene dynamics was successfully used in the task of object detection (pedestrians, cars, bicycles, ...) [101], and by filtering out moving objects, 3D maps capturing only static parts of the environment can be built [41]. Such maps are useful for the localization or the motion planning where measurements of moving objects are undesirable and introduce motion artifacts into the map. Successful methods for the *detection and tracking of moving objects (DATMO)* assume that the way, in which sensors are used, causes that only the objects (static or dynamic) are captured [92], or that the ground can be detected (see Fig. 8.1) and filtered out in the preprocessing stage [58, 17, 102, 6, 69]. For these purposes, we intend to reliably and efficiently *segment the data to ground/non-ground* parts. We consider the ground to be every surface traversable by commonly moving objects (pedestrians, cars, bikes, etc.).

In these DATMO systems, the ground detection is typically based on primitive features with low discriminative capabilities. The state of the art technique for

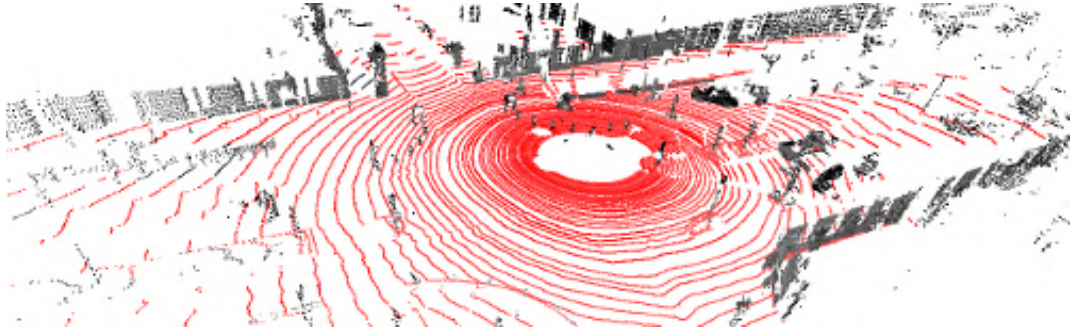


Figure 8.1: Expected segmentation of the Velodyne LiDAR point cloud into sets of ground (red) and non-ground (grey) points.

robust ground segmentation by Zhang et al. [108] achieves good results in terms of accuracy by building a Markov Random Field (MRF) and inference using the Loopy belief propagation. Unfortunately, the robustness of this method is achieved by compromising its time efficiency (over 2 minutes per frame).

The *Velodyne* sensor – nowadays common source of *LiDAR* (Light Detection And Ranging) data – captures the full 3D information about environment, in contrast to simple range finders, providing information about occupancy in a certain height around the robotic platform only. Currently, the most powerful model HDL-64E covers full 360° horizontal field and 26.8° vertical field of view, and with up to 15 Hz frame rate, captures over 1.3M of points per second. This sensor scans the surrounding area by 64 rotating laser beams while each beam produces one *ring* of 3D points (red circles in Fig. 8.1).

Since the breakthrough in machine learning after introduction of AlexNet [49], the attractiveness of *Convolutional Neural Networks (CNNs)* has grown rapidly and this model was successfully used for many computer vision tasks including image classification, object detection, face recognition, semantic segmentation [61], etc. In this work, we deployed convolutional neural networks for the task of *ground segmentation* in sparse Velodyne point cloud data. We designed multiple networks with shallow topologies (3-5 convolutional layers) fulfilling the requirements for robustness and accuracy. We trained and evaluated them by using a hand-annotated dataset.

The main contributions of this work are the following:

- we show that the *sparse 3D LiDAR data can be encoded into a multi-channel 2D signal* (analogous to HHA encoded range images [34] or LiDAR data encoding in the vehicle detection task [57]) and processed by convolution neural network;
- *new approach to ground segmentation in Velodyne point clouds using CNN* which outperforms current state of the art in accuracy and time performance.

Besides this, we developed a *semi-automatic ground annotation* tool and we annotated a part of the KITTI tracking dataset. Source code of the annotation tool and LiDAR point clouds preprocessing methods, design and configuration of the trained convolutional networks, as well as annotated ground truth data are publicly available¹.

8.3 RELATED WORK

As mentioned above, we define the ground as a surface traversable by commonly moving objects. A similar definition has been already used for an outdoor robot [47]. The traversability estimation was performed using geometric features (extracted from stereo-vision) and texture features (from RGB images). By clustering, the labels are assigned to parts of the surrounding environment. Compared to our approach, this method requires explicit feature specification and different type of input data – stereo RGB vision, IMU, and motor current sensor.

Convolutional networks were deployed for learning rich descriptors of RGB-D data [34] useful for per-pixel object detection. The input of networks encodes horizontal disparity (equivalent to the range), height and normals angle. Our work proposes a similar type of encoding suitable for processing the sparse LiDAR data. Since the normals can not be robustly estimated in these data, the angles are not used.

Many DATMO (detection and tracking of moving objects) methods segment and filter out the ground measurements from LiDAR data in a preprocessing stage [58, 17, 6, 69, 82, 102, 81]. These approaches usually rely on primitive features with low discriminative capabilities like mean or variance of measured height in a certain small area, or changes in the elevation between the rings in Velodyne data.

More traditional DATMO methods operate over data from simple laser rangefinders [92], assuming the measurements provided by LiDAR positioned approximately parallel to the ground surface, capturing only the upright (moving and/or static) objects and not the ground. Over such data, the occupancy grid can be built and detection of movement is performed by particle filtering.

When data from multiple laser sensors including Velodyne 3D LiDAR are fused [69], building the occupancy grid starts to be an issue, since the sensors cover a significantly larger area including the ground. The ground measurements must be recognized and filtered out in order to build a valid occupancy grid representing free space, the space occupied by obstacles, and currently unobserved areas. For the sake of effectivity, authors [69] selected a computationally inexpensive approach where all measurements within a certain height range are considered to

¹ https://github.com/robofit/but_velodyne_cnn

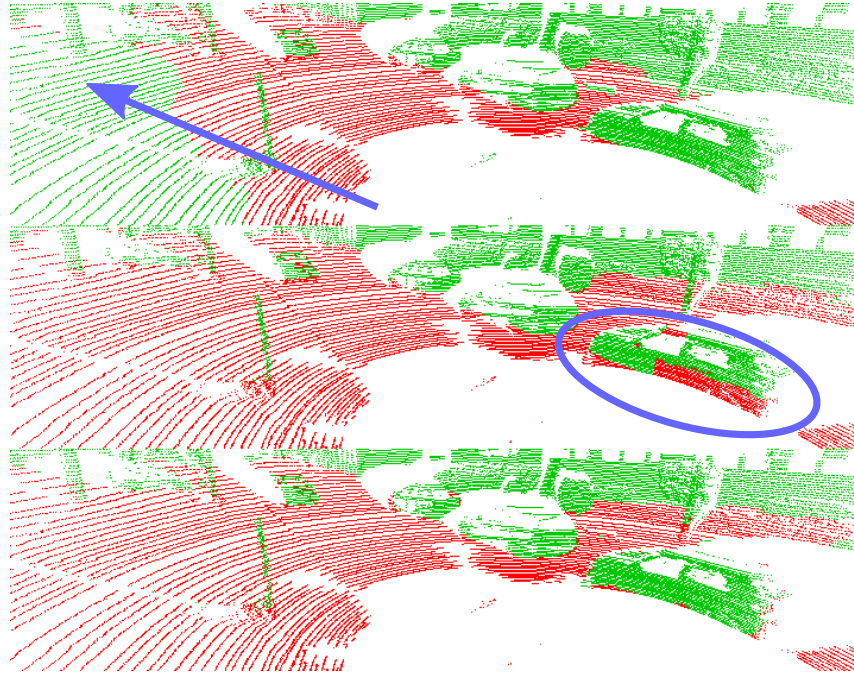


Figure 8.2: Different results of ground segmentation methods. **top:** Simple height thresholding can not deal with terrain elevation; **middle:** Loopy belief propagation [108] produces incorrect results when objects are close to the sensor; **bottom:** our method.

be ground. Besides the sensitivity to selection of optimal thresholds, the robustness/repeatability of such approach is far from the optimal (see Fig. 8.2).

The motion detection generalized to motion field estimation [58] in a polar grid benefits from the large area covered by the Velodyne LiDAR scanner. The preprocessing step, same as in the previously mentioned work – i.e. the ground detection and filtering – is performed as well. Using the simple thresholding, this method shares the same disadvantages. The areas (polar grid cells), fulfilling at least one of the following conditions, are considered to be ground: the average height fits an exactly defined range, the standard deviation of the height is below a certain threshold, or the difference between the minimal and the maximal height inside the cell is below another threshold. A very similar approach with only small modifications was used by Asvadi et al. [6] in a DATMO system operating over a regular orthogonal grid. The area within one grid cell is considered to be ground if both the mean height and the standard deviation of the heights fit below a predefined threshold.

Other approaches analyse changes in the elevation in order to segment the ground in Velodyne LiDAR scans [82, 102, 81]: each vertical slice consisting of all points captured at exactly the same moment by all laser rays, is analysed separately. Three points A, B, C from adjacent rings form two vectors \vec{AB} and \vec{BC} . If the dot product of these (normalized) vectors is above a certain threshold, a significant change of elevation – the breakpoint – is found. Such breakpoints form the

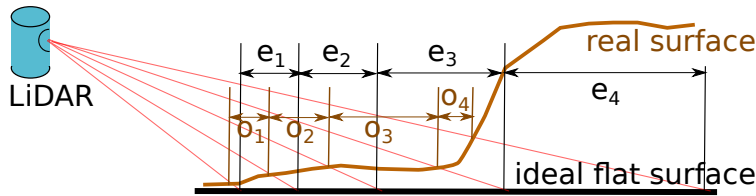


Figure 8.3: Ground detection by comparison of expected range difference e_i with observed difference o_i . Since $e_4 - o_4 > th$, the border between the obstacle and the ground is found [17].

border between the ground (points between the sensor and the breakpoint) and an obstacle (points behind the breakpoint). Besides the lack of robustness, this approach does not allow to reason about the space behind the first obstacle where the ground can be observed again.

Analysis of ranges differences between two adjacent Velodyne rings (Fig. 8.3) was also used for the ground segmentation [17] in LiDAR data. On the ideal flat horizontal surface, the expected range difference e_i between two adjacent rings can be computed, assuming the height and the vertical angle of each laser beam is known. This range difference decreases with increasing elevation of the surface. At the ideal vertical obstacle, this difference becomes zero.

Besides the previously mentioned DATMO methods, the ground detection and filtering plays important role in point cloud registration by scan segments matching [22]. However, in a preprocessing step, the ground points are also detected by thresholding the mean and the variance of vertical height within the cells of voxel grid [21].

The lack of accuracy and robustness in previously mentioned methods, mostly caused by the fixed thresholding of simple features with low discriminative power, was overcome by the inference in *Markov Random Field (MRF)* [108]. Although the introduced 3D volumetric grid is built by estimation of a slope in each vertical slice in a similar way to 2D occupancy grids, the final segmentation to ground/obstacle is not made directly. At first, based on the slope detected, the points are categorized as unknown, probably ground, probably obstacle, and probably obstacle borders. This categorization implies the initial cost assigned to each volumetric element of the regular 3D polar grid. The key improvement is done by Loopy Belief Propagation inference in order to estimate ground height within a certain region. All measurements within this region with a smaller height are considered to be the ground points. The rest is classified as non-ground. Unfortunately, the robustness of this method is achieved by compromising its time efficiency. In our experiments with the original MATLAB implementation, kindly provided by the authors, the processing of single Velodyne HDL-64E frame takes approximately 145s.

The key improvement achieved by our method is the reduction of time complexity of the ground segmentation process to the fraction of the time required by Loopy Belief Propagation [108], while slightly better results in terms of accuracy were achieved as well. Processing of a single Velodyne frame by our *Lo5+deconv* network takes 140 ms on average, using only CPU. By using GPU (GeForce GTX 770), the processing time is further reduced to 7 ms per frame.

Simultaneously with our work, the Baidu research team [57] proposed a similar encoding of sparse LiDAR data into 2D matrices for the vehicles detection by convolutional neural networks. Our encoding differs from their work in polar bin aggregation of LiDAR points (described in Sec. 8.4.1) to improve the stability of prediction.

8.4 PROPOSED GROUND SEGMENTATION METHOD

The goal of our method is to assign a *binary label ground/non-ground* (8.1) to each 3D point $\mathbf{p} \in \mathbb{P}$ measured by the LiDAR sensor. The point cloud elements \mathbf{p} are represented by *3D coordinates* originating at the LiDAR sensor position, accompanied by the laser *intensity* reading and the *ring ID* identifying the source laser beam which was used to measure the point $\mathbf{p} = [p_x, p_y, p_z, p_i, p_r]$. Since we do not assign the ground label to each LiDAR point separately, we solve the assignment (8.2) of binary labels to all the points jointly.

$$g : \mathbb{P} \rightarrow \{0, 1\} \quad (8.1)$$

$$G : \mathbb{P} \rightarrow \{0, 1\}^{|\mathbb{P}|}, \quad \mathbb{P} \in \mathbb{P} \quad (8.2)$$

8.4.1 Encoding Sparse 3D Data Into a Dense 2D Matrix

In order to process the Velodyne LiDAR data by a convolutional neural network, we *encode* the original *sparse point cloud* \mathbb{P} into a multi-channel *dense matrix* \mathbf{M} . The original 3D data are treated as a 2D signal in the domain of the ring (the ID of the source laser beam) and the horizontal angle, as illustrated in Fig. 8.4. The size of the resulting matrix \mathbf{M} depends on the number of rings in the LiDAR frame (i.e. number of laser beams used) and the sampling rate R of the horizontal angle. In our experiments, we used Velodyne LiDAR HDL-64E with 64 rays and resolution $R = 1^\circ$.

At first, the point cloud is aggregated into the polar bins $\mathbf{b}_{r,c}$ (8.5) analogous to our previous work [98]. All the points assigned to the same bin share the same ring ID r (points captured by the same laser beam) and fit into the same polar cone $c = \varphi(\mathbf{p})$ (8.6), computed according to the horizontal angle of the point. Each polar

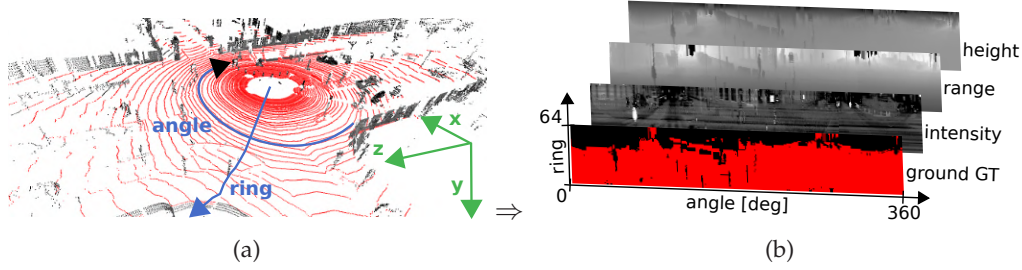


Figure 8.4: Transformation of the sparse Velodyne point cloud (a) into the multi-channel dense matrix (b). Each row represents measurements of a single laser beam done during one rotation of the sensor. Each column contains measurements of all 64 laser beams captured at a specific rotational angle at the same time.

bin is encoded into the element $\mathbf{m}_{r,c}$ of matrix \mathbf{M} in its r -th row and c -th column (8.3). Since multiple points fall into the same bin (the horizontal representation of our encoding is coarser than original Velodyne resolution), a single representative of the bin is found as the average (8.4). Moreover, since the horizontal index in the matrix \mathbf{M} encodes the rotational angle in the 3D horizontal XZ plane, we can reduce the number of channels by replacing XZ coordinates p_x, p_z by depth (or range) value $d = \|p_x, p_z\|_2$ without the loss of any information.

$$\mathbf{m}_{r,c} = \varepsilon(\mathbf{b}_{r,c}) \quad (8.3)$$

$$\varepsilon(\mathbf{b}_{r,c}) = \frac{\sum_{\mathbf{p} \in \mathbf{b}_{r,c}} [p_y, \|p_x, p_z\|_2, p_i]}{|\mathbf{b}_{r,c}|} \quad (8.4)$$

$$\mathbf{b}_{r,c} = \{\mathbf{p} \in \mathbf{P} \mid p_r = r \wedge \varphi(\mathbf{p}) = c\} \quad (8.5)$$

$$\varphi(\mathbf{p}) = \left\lfloor \frac{\text{atan}\left(\frac{p_z}{p_x}\right) + 180^\circ}{\frac{360^\circ}{R}} \right\rfloor \quad (8.6)$$

In case of empty bins (e.g. no measurement exists in this area due to the sensor limits), the value in the matrix \mathbf{M} is linearly interpolated from the neighbourhood.

8.4.2 Training Dataset

The most serious issue in development of the proposed system was the lack of training data, especially missing annotations of ground data in the Velodyne scans. The development of KITTI Semantic Segmentation dataset² is still in progress and only small subsets are available at the moment. The only annotations relevant to our task were created by Richard Zhang [109] in his work on semantic segmentation of urban scenes. However, Zhang used the LiDAR point clouds as a supplementary data only, and annotations were made for RGB camera images

² http://www.cvlibs.net/datasets/kitti/eval_semantics.php

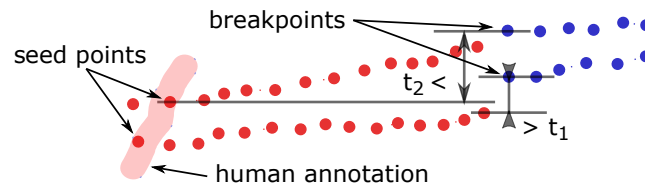


Figure 8.5: Flooding the human annotation from seed points along the ring. The ground points are red. When the breakpoint is found (first of blue not-ground points), the flooding is stopped.

in the first place. These annotations were probably back-projected into the LiDAR frames and spread across consequent frames which caused serious inaccuracies in the ground annotations and made these data unsuitable for our training and testing.

Therefore we prepared a *semiautomatic tool for ground annotation* in 3D Velodyne data³. Using a pen-like drawing tool, the user highlights certain ground points as ground seed points \mathbf{p}^s . From these points, the annotation *automatically floods* along the ring until a breakpoint \mathbf{p}^b is found (see Fig. 8.5). The breakpoint is defined as the first point, where the height difference with respect to the previous point $|\mathbf{p}_y^b - \mathbf{p}_y^{i-1}| > t_1$, or with respect to the seed point $|\mathbf{p}_y^b - \mathbf{p}_y^s| > t_2$, is above a respective threshold. When annotating the dataset, we found the values $t_1 = 3$ cm and $t_2 = 7$ cm work best as they save annotator's time and they reduce manual changes.

Using this tool, we prepared *accurate annotations of the ground* in 3D LiDAR data for a subset of KITTI Tracking Dataset – the same data as was annotated by [109] in RGB images. The subset consists of 8 data sequences taken from moving vehicle in different urban and suburban environments. In total, there are 252 frames captured in 1 s interval. We randomly split those frames into training and evaluation set in 70 : 30 ratio.

Since the amount of available annotated data is quite small, we prepared automatic artificial annotations for the rest of the KITTI Tracking Dataset (19 k frames) by thresholding simple features, like the mean and the variance of height, and the distance and the elevation differences between rings, as used in the previous works [58, 17, 6, 69, 82, 102, 81]. These artificial annotations are used for CNNs pretraining. The resulting parameters are used as initial weights of convolutional kernels for further training on more precise human annotations.

We also tried to use data augmentation and generate artificial 3D LiDAR frames automatically. Unfortunately, this approach proved to be infeasible, since the available 3D models are not detailed enough, lack fine surface details, and substitute

³https://github.com/robofit/but_velodyne_cnn/tree/master/ground-annotator

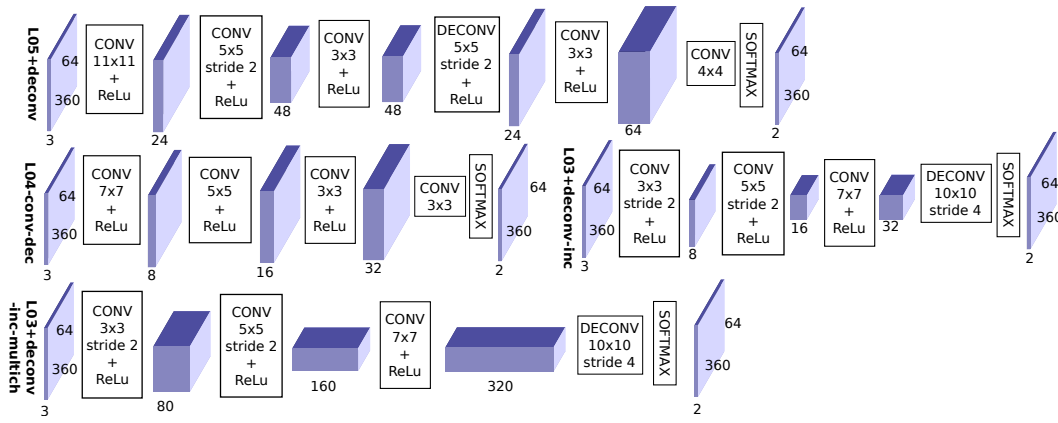


Figure 8.6: Topology of the four proposed CNNs including dimensions of intermediate data blobs (blue blocks) and the number of channels below each blob. *Lo5+deconv* consists of 5 convolutional layers plus single deconvolution to restore the original frame width and height. *Lo4-conv-dec* process the input frame by 4 convolutional layers with decreasing size (7, 5, 3, 3) of convolution kernel. In *Lo3+deconv-inc*, 3 convolutional layers with increasing kernel size are used. Deconvolution is used to restore original frame size in both this topology and in *Lo3+deconv-inc-multich* where the number of output channels are significantly larger comparing with other networks. Note: if the stride parameter N is set in (de-)convolutional layers, the width and height of the output blob is (larger or) smaller N -times.

this structure information (trees, bushes, curbs, etc.) by texturing flat surfaces (so called billboarding).

8.4.3 Topology and Training of the Proposed Networks

Because of the small amount of annotated training data, we used *shallow CNN architectures* only. All the networks are fully convolutional. They consist of convolution and deconvolution layers with ReLU non-linearities. Gradient descent is used as the optimization method for the training. The most interesting and successful topologies we experimented with are presented in Fig. 8.6.

The multi-channel matrix \mathbf{M} , obtained by the encoding described in 8.4.1, is the input of all proposed networks. The probability of being a ground point $p_g = p(g(\mathbf{p}) = 1)$ is estimated for each pixel of this matrix. Therefore, the output of all networks has the same size as the input matrices except the number of channels. The output channels represent probabilities p_g and $1 - p_g$ since the softmax is applied.

Presented architectures (Fig. 8.6) differ in the type and number of layers used, dimension of convolutional kernels, and in the number of channels within each layer. Deconvolutional layers (previously also used in semantic segmentation [61])

were used in 3 of 4 presented topologies, including the best topology *Lo5+deconv*, which performs best in our experiments. In topologies *Lo5+deconv* and *Lo4-conv-dec*, the size of the convolutional layers is decreasing, when compared to the other two topologies. The effect of a significantly larger number of intermediate output channels is evaluated for topology *Lo3+deconv-inc-multich*.

The input of the CNN, which is prepared as described in Sec. 8.4.1 Eq. (8.3-8.6), is normalized and rescaled (8.7). This applies only to the depth d and the height p_y channels, since the intensity values of Velodyne sensor are already normalized to the interval $(0; 1)$. In our experiments, the normalization constant is set to $H = 3$, since in usual scenarios, the Velodyne HDL-64E captures a vertical slice approximately 3m high.

$$\overline{p_y} = \frac{p_y}{H}, \quad \overline{d} = \log(d) \quad (8.7)$$

We applied this logarithmic rescaling for the depth channel to get approximately the same range differences between consequent rings for flat surfaces, both close and far from the sensor. The rescaling should suppress differences between the rings, due to varying distance from the sensor, and highlight those differences caused by the structure of observed scene – i.e. the obstacles (illustrated in Fig. 8.3). In the similar manner, the horizontal disparity was previously used as an input of a convolutional network, instead of using range value directly [34], what finally results in a normalization similar to ours.

8.5 EXPERIMENTS

The proposed convolutional networks were implemented, trained and evaluated using *Caffe*⁴ deep learning framework. The human annotated dataset and the automatically annotated dataset were both used for training and pre-training of the proposed networks. We compared the results of our CNNs with the results of the robust state-of-the-art method [108] (using the original MATLAB implementation shared by the authors). It is necessary to mention one limitation of the Zhang’s method. Because dimensions of the polar grid need to be set, the maximal range from the sensor is limited. In the experiments we used the 60m limit by default (and the 30m limit in the time performance test). In order to make fair evaluation, we computed the accuracy of our method for both the maximal range set to 60m (same conditions as for [108]) and the unlimited range (to illustrate behavior for more distant measurements). Also, since the Zhang’s method has no parameter for tuning false positives to false negatives ratio, only a single precision/recall value can be computed instead of the whole PR curve.

⁴<http://caffe.berkeleyvision.org/>

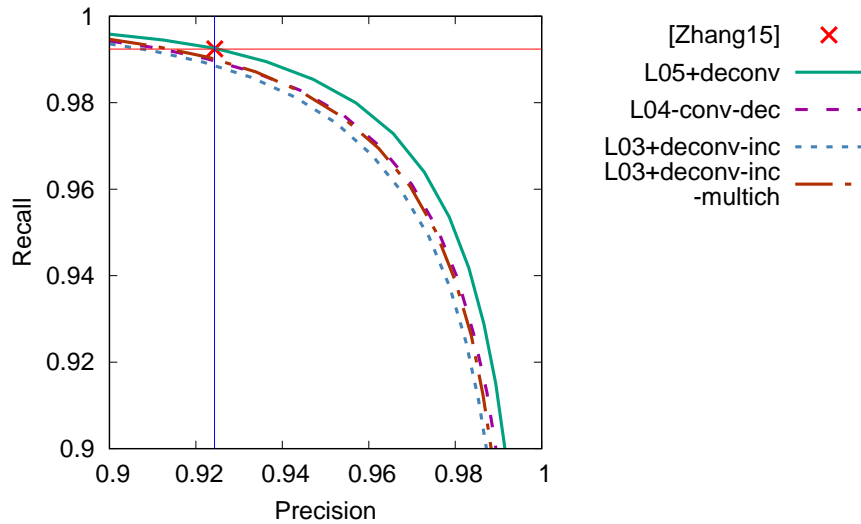


Figure 8.7: The accuracy of the proposed networks and the reference method [108] for comparison. See Table 8.1 for numerical results.

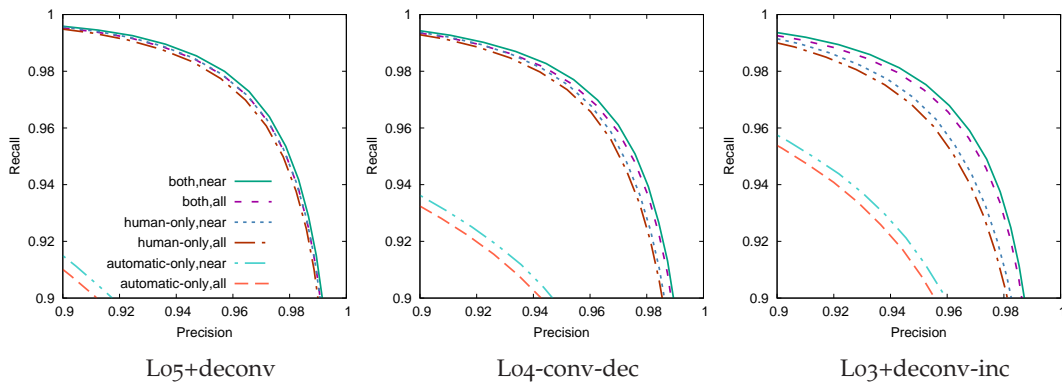


Figure 8.8: Comparison of the different CNN topologies. The networks were trained using the human-made annotations (label *human-only*), artificial annotations (*automatic-only*) and both datasets for initialization and training (label *both*). All LiDAR points were processed, or only the points within the 60m range (label *near*) were taken into the account.

Fig. 8.7 shows the comparison of different networks with the reference method [108]. The results are also summarized in Table 8.1 by means of the average precision and F-score as the metrics of accuracy. All networks were pre-trained using the automatically annotated data, trained and evaluated using the human annotated data and only the points within the range of 60m were taken into account.

The results (Fig. 8.7 and Table 8.1) show that the accuracy is quite similar for different network topologies. Better accuracy is achieved with the networks where the size of convolution kernels decreases (*L05+deconv* and *L04-conv-dec* CNNs) and also with larger networks. The accuracy of *L05+deconv* network is also slightly higher compared to the reference method [108]. Preserving the same recall we were able to achieve 0.5% better precision and vice versa: 0.1% higher recall while

	AP	Precision* recall=.992	Recall* prec=.924	Best F-score
[108]	-	0.924	0.992	0.957
Lo5+deconv	0.996	0.929	0.993	0.969
Lo4-conv-dec	0.995	0.914	0.990	0.966
Lo3+deconv-inc	0.994	0.910	0.989	0.964
Lo3+deconv-inc-multich	0.995	0.916	0.990	0.966

Table 8.1: Average precision (area under the PR curve), precision, recall and the best F-score of the proposed networks compared to [108]. *The precision (and the recall) was estimated for points where the recall (and precision respectively) is the same as the results of [108] (also displayed in Fig. 8.7 by red and blue line). The best F-score is taken as the highest value of harmonical average of precision and recall within the whole PR curve.

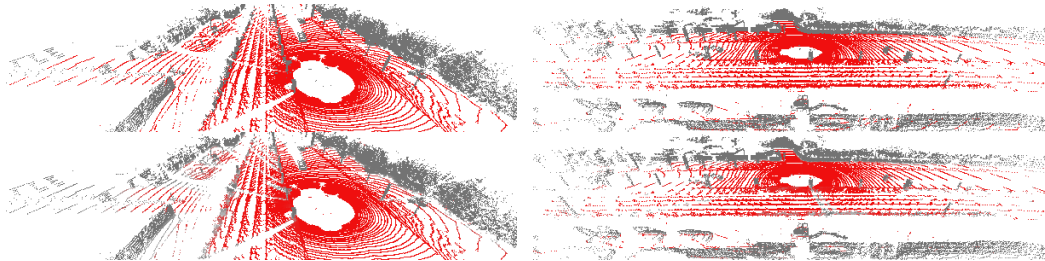


Figure 8.9: Ground segmentations (outputs of CNN *Lo5+deconv*, bottom) for different LiDAR scans compared with human-made annotations (up). The results are near ideal but small differences are still visible under closer inspection.

preserving the same precision. Also, since our method enables balancing FP:FN ratio, we were able to find an optimal operating point yielding better F-score.

In Fig. 8.8, the precision-recall curves of different network topologies trained and evaluated in different ways are shown. We compared CNNs which were trained either by using the human-made annotations only (label *human-only*), or just by automatically annotated dataset (*automatic-only*), or by using both datasets together (label *both*). Moreover, we evaluated the accuracy of the situation in which all points are considered (label *all*), or when the maximal range is limited to 60m (label *near*) as used also by Zhang [108]. The examples of CNN outputs can be found in Fig. 8.9.

The results depicted in Fig. 8.8 show that cases in which reasoning about the ground was made only within the certain range (label *near*) yield better results.

	CPU only [ms]	with GPU [ms]
L05+deconv	139	7.0
L04-conv-dec	90	3.2
L03+deconv-inc	8	1.2
L03+deconv-inc-multich	355	6.9

Table 8.2: Performance comparison of the proposed networks in terms of speed. The average processing time per single Velodyne LiDAR HDL-64E frame is presented. The mini-batches of size 4 were used (i.e. 4 frames were processed in parallel).

This is expected, since the density of measurements in farther areas is much lower. Also, the CNNs trained with human annotated datasets behave more accurately than CNNs trained on artificial data (evaluation is always made using the human annotations). An interesting fact is that this gap is less significant for networks with smaller architectures (e.g. L_03 compared to L_05). This is probably caused by higher generalization which compromises discriminative power when learned on real annotations.

Table 8.2 shows the average processing time of proposed networks using CPU implementation (Intel i5-6500) and using GPU acceleration (GeForce GTX 770) on a standard desktop computer. These numbers indicate the usability of the networks for certain mobile robot platforms. $L_03+deconv-inc$ requires low CPU consumption and therefore it is suitable also for small robots with low computational power. On the contrary, the $L_05+deconv$ topology is suitable for platforms where GPU acceleration is available because of the superior accuracy.

As was said before, the main advantage of our method is superior time performance when compared to the method of Zhang et al. [108]. In our experiments, when using the Zhang’s MATLAB implementation, the processing time of Velodyne HDL-64E LiDAR frame was 145 sec and consumed 11 GB of memory on average (note: no memory swapping which would compromise the performance happened during the experiments). Also, when we decreased the maximal range (and also the size of the internal 3D polar grid) to 30m, the processing time dropped to 75sec per frame and the memory consumption to approximately one half. However, this is still really far from real-time performance.

8.6 CONCLUSION

We presented a real time and robust ground segmentation method of Velodyne LiDAR data which outperforms the current state-of-the art methods in both the accuracy and speed. Our results show that the sparse LiDAR data can be encoded into its dense 2D representation and effectively processed by CNN. Our method improved the precision of state of the art [108] (by 0.5%) and significantly improved speed of the ground segmentation process from minutes to 140 ms using CPU and 7 ms with GPU acceleration.

In this paper we demonstrate that CNN approach is suitable for simpler task of ground segmentation where the results are near ideal. In the follow-up work, we want to explore the potential of this approach in more challenging semantic segmentation or move detection and also in quite different tasks of visual odometry estimation or point cloud registration.

As a secondary outcome of our work, we created the dataset with ground annotated and made it publicly available along with the annotation tool. Such data can be used to design, train, and evaluate other ground segmentation approaches.

Part V

SUMMARY

CONCLUSION

Mobile laser mapping plays an important role in a field of surveying, geodesy, construction, and even in planning of road maintenance. Compared to traditional approaches of geodetic survey, laser scanning is able to provide richer information about surrounding environment and – more importantly – it enables time efficient mapping of the large areas.

This work introduces a way of development of a whole mobile mapping solution from smaller pieces of puzzle – novel algorithms for the odometry estimation and the point cloud registration. Moreover, the ground segmentation algorithm presented in the last chapter represents a source of semantic information, which can be automatically incorporated into the 3D data.

Compared to traditional approaches, the point cloud registration algorithms, described in the second part of this thesis, are designed especially for large and sparse data of 3D LiDAR. The CLS (Collar Line Segments) algorithm overcame data sparsity by a random sampling of the point cloud by line segments. The evaluation proved the robustness and the accuracy with the odometry estimation error 1.7% (of elapsed trajectory length) for a standard KITTI dataset.

On the other hand, odometry estimation using convolution neural networks represents a faster alternative, when only translation motion parameters are required. While providing an online performance with GPU support, a 1,2% relative error of translation was achieved on the same KITTI dataset. This method is convenient especially in the situations, when an online preview of the odometry estimation is necessary and the rotation motion parameters are available from another source (e.g. from IMU sensor).

A significant contribution of this work is the design and the realization of the mobile backpack laser solution, where the CLS algorithm plays the key role as the frontend of SLAM (Simultaneous Localisation And Mapping). The most significant contribution of this solution is the design with a synchronized and calibrated pair of Velodyne 3D LiDARS accompanied with a dual antenna GNSS/INS solution. This combination provides universality for both small indoor and also large open outdoor areas. The evaluation showed, that the basic requirements – relative error below 5 cm and the average of absolute error of georeferencing under 14 cm – was fulfilled. Moreover, this solution provides high data density and normalized laser intensities in the resulting 3D model, which enables better recognition of important objects during the inspection and manual post-processing.

9.1 FUTURE WORK

From a scientific point of view, there are multiple areas of future work. The first one is the improvement of the accuracy. Our current development aims at deeper integration of the IMU sensor for the correction of a rolling shutter distortion, which is currently solved by a simple linear interpolation of estimated motion during the LiDAR rotation.

Another interesting task is the integration of a spherical camera for both the creation of panoramic virtual tour around the environment, and also for colouring the point cloud. The colouring task would require a precise synchronization and an extrinsic calibration of this omnidirectional RGB camera.

Third goal is the support for the arbitrary mono-antenna geodetic GNSS solution instead of currently deployed dual antenna solution. This improvement would make our backpack potentially cheaper and more accessible, since such GNSS solutions are even owned by smaller geodetic companies.

The last – and lets say a non-scientific – future goal is the commercialization of this backpack mapping solution, which demonstrated its potential in the evaluations presented in this work.

BIBLIOGRAPHY

- [1] Velodyne's HDL-64E: A high definition lidar sensor for 3D applications. Technical report, Velodyne Acoustics Inc., Morgan Hill, California, October 2007. URL <http://velodynelidar.com/downloads.html>.
- [2] LiDAR: Driving the future of autonomous navigation. Technical report, Frost & Sullivan, Mountain View, California, February 2016. URL <http://velodynelidar.com/downloads.html>.
- [3] *Robin datasheet*. 3D laser mapping, 2017. Available at <https://www.3dlasermapping.com/wp-content/uploads/2017/09/ROBIN-Datasheet-front-and-reverse-WEB.pdf>.
- [4] R. Ambrus, N. Bore, et al. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ Int. Conference on*, 2014. doi: 10.1109/IROS.2014.6942806.
- [5] Karen Anderson, Steven Hancock, Mathias Disney, and Kevin J. Gaston. Is waveform worth it? a comparison of LiDAR approaches for vegetation and landscape characterization. *Remote Sensing in Ecology and Conservation*, 2(1): 5–15, 2016. ISSN 2056-3485. doi: 10.1002/rse2.8.
- [6] A. Asvadi, P. Peixoto, and U. Nunes. Detection and tracking of moving objects using 2.5D motion grids. In *IEEE Int. Conference on Intelligent Transportation Systems*, pages 788–793, Sept 2015. doi: 10.1109/ITSC.2015.133.
- [7] Hernan Badino, Akihiro Yamamoto, and Takeo Kanade. Visual odometry by multi-frame feature integration. In *International Workshop on Computer Vision for Autonomous Driving (ICCV)*, December 2013.
- [8] N. Barbour and G. Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*, 1(4):332–339, Dec 2001. ISSN 1530-437X. doi: 10.1109/7361.983473.
- [9] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. ISSN 0162-8828. doi: 10.1109/34.121791.
- [10] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. ISSN 0162-8828. doi: 10.1109/34.121791.

- [11] M. Bosse and R. Zlot. Continuous 3D scan-matching with a spinning 2d laser. In *2009 IEEE International Conference on Robotics and Automation*, pages 4312–4319, May 2009. doi: 10.1109/ROBOT.2009.5152851.
- [12] M. Bosse and R. Zlot. Place recognition using keypoint voting in large 3D LiDAR datasets. In *2013 IEEE International Conference on Robotics and Automation*, pages 2677–2684, May 2013. doi: 10.1109/ICRA.2013.6630945.
- [13] M. Bosse, R. Zlot, and P. Flick. Zebedee: Design of a spring-mounted 3D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119, Oct 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2200990.
- [14] Michael Bosse and Robert Zlot. Keypoint design and evaluation for place recognition in 2D LiDAR maps. *Robotics and Autonomous Systems*, 57(12): 1211 – 1224, 2009. ISSN 0921-8890. Inside Data Association.
- [15] Stuart Cadge. Welcome to the ZEB REVolution. *GEOmedia*, 20(3), 2016.
- [16] Yang Chen and Gerard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10:145–155, 1992. ISSN 0262-8856. doi: 10.1016/0262-8856(92)90066-C.
- [17] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer. Multi-target tracking using a 3D-lidar sensor for autonomous vehicles. In *IEEE Int, Conference on Intelligent Transportation Systems*, pages 881–886, Oct 2013. doi: 10.1109/ITSC.2013.6728343.
- [18] Mathieu Dassot, Thierry Constant, and Meriem Fournier. The use of terrestrial LiDAR technology in forest science: application fields, benefits and challenges. *Annals of Forest Science*, 68(5):959–974, Aug 2011. ISSN 1297-966X. doi: 10.1007/s13595-011-0102-2.
- [19] J. Deschaud. IMLS-SLAM: Scan-to-model matching based on 3D data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2480–2485, May 2018. doi: 10.1109/ICRA.2018.8460653.
- [20] Thomas JB Dewez, Emmanuelle Plat, Marie Degas, Thomas Richard, Pierre Pannet, Ysoline Thuon, Baptiste Meire, Jean-Marc Watelet, Laurent Cauvin, Joël Lucas, et al. Handheld mobile laser scanners ZEB-1 and ZEB-REVO to map an underground quarry and its above-ground surroundings. In *2nd Virtual Geosciences Conference: VGC 2016*, 2016.
- [21] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3D lidar point clouds. In *2011 IEEE ICRA*, pages 2798–2805, May 2011. doi: 10.1109/ICRA.2011.5979818.

- [22] B. Douillard, A. Quadros, et al. Scan segments matching for pairwise 3D alignment. In *Robotics and Automation (ICRA), 2012 IEEE Int. Conference on*, pages 3033–3040, May 2012. doi: 10.1109/ICRA.2012.6224788.
- [23] D. Droeschel and S. Behnke. Efficient continuous-time SLAM for 3D LiDAR-based online mapping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, May 2018. doi: 10.1109/ICRA.2018.8461000.
- [24] David Droeschel, Max Schwarz, and Sven Behnke. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robotics and Autonomous Systems*, 88(C):104–115, February 2017. ISSN 0921-8890. doi: 10.1016/j.robot.2016.10.017.
- [25] Parul Dubey. New bMS3D-360: The first backpack mobile scanning system including panoramic camera. *Informed Infrastructure*, March 2018.
- [26] BERND Eissfeller, GERALD Ameres, VICTORIA Kropp, and DANIEL Sanroma. Performance of GPS, GLONASS and galileo. In *Photogrammetric Week*, volume 7, pages 185–199, 2007.
- [27] Karolina D. Fieber, Ian J. Davenport, James M. Ferryman, Robert J. Gurney, Jeffrey P. Walker, and Jorg M. Hacker. Analysis of full-waveform LiDAR data for classification of an orange orchard scene. *ISPRS Journal of Photogrammetry and Remote Sensing*, 82:63 – 82, 2013. ISSN 0924-2716.
- [28] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. doi: 10.1177/0278364913491297.
- [29] *The ZEB-REVO Brochure*. GeoSLAM Ltd., apr 2018. Available at https://geoslam.com/wp-content/uploads/2018/04/GeoSLAM-ZEB-REVO-Solution_v9.pdf?x97867.
- [30] *The ZEB-REVO Solution*. GeoSLAM Ltd., 2018. Available at <http://download.geoslam.com/docs/zeb-revo-rt/ZEB-REVO%20RT%20User%20Guide%20V1-0-1.pdf>, version 9.
- [31] D. Girardeau-Montaut, Michel Roux, Raphaël Marc, and Guillaume Thibault. Change detection on points cloud data acquired with a ground laser scanner, 2005.
- [32] W.S. Grant, R.C. Voorhies, and L. Itti. Finding planes in LiDAR point clouds for real-time registration. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conference on*, pages 4347–4354, Nov 2013. doi: 10.1109/IROS.2013.6696980.

- [33] *LiBackpack DG50, Mobile Handheld 3D Mapping System*. GreenValley International, apr 2019. Available at <https://greenvalleyintl.com/wp-content/uploads/2019/04/LiBackpack-DG50.pdf>.
- [34] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, and Jitendra Malik. Learning rich features from RGB-D images for object detection and segmentation. In *European Conference on Computer Vision, Zurich, Switzerland, 2014*. ISBN 978-3-319-10584-0. doi: 10.1007/978-3-319-10584-0_23.
- [35] S. Haykin. *Kalman Filtering and Neural Networks*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley, 2004. ISBN 9780471464211.
- [36] A. Hermans, G. Floros, and B. Leibe. Dense 3D semantic mapping of indoor scenes from RGB-D images. In *Robotics and Automation (ICRA), 2014 IEEE Int. Conference on*, pages 2631–2638, May 2014. doi: 10.1109/ICRA.2014.6907236.
- [37] W. Hess, D. Kohler, H. Rapp, and D. Andor. Real-time loop closure in 2D LiDAR SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, May 2016. doi: 10.1109/ICRA.2016.7487258.
- [38] M. Holopainen, M. Vastaranta, V. Kankare, H. Hyyppä, M. Vaaja, J. Hyyppä, X. Liang, P. Litkey, X. Yu, H. Kaartinen, A. Kukko, S. Kaasalainen, and A. Jaakkola. The use of ALS, TLS and VLS measurements in mapping and monitoring urban trees. In *2011 Joint Urban Remote Sensing Event*, pages 29–32, April 2011. doi: 10.1109/JURSE.2011.5764711.
- [39] S. Viorela Ila, Lukas Polok, Marek Solony, Pavel Zemcik, and Pavel Smrz. Fast covariance recovery in incremental nonlinear least square solvers. In *Proceedings of IEEE ICRA*. IEEE Computer Society, 2015. ISBN 978-1-4799-6922-7.
- [40] Viorela Ila, Lukas Polok, Marek Solony, and Pavel Svoboda. SLAM++ – a highly efficient and temporally scalable incremental SLAM framework. *The International Journal of Robotics Research*, 36(2):210–230, 2017. doi: 10.1177/0278364917691110.
- [41] C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux. Static-map and dynamic object reconstruction in outdoor scenes using 3D motion segmentation. *IEEE Robotics and Automation Letters*, 1(1):324–331, Jan 2016. ISSN 2377-3766. doi: 10.1109/LRA.2016.2517207.
- [42] Boris Jutzi and H Gross. Normalization of LiDAR intensity data based on range and surface incidence angle. *ISPRS Journal of Photogrammetry and Remote Sensing*, 38, 01 2009.

- [43] Sanna Kaasalainen, Harri Kaartinen, and Antero Kukko. Snow cover change detection with laser scanning range and brightness measurements. *7*, 01 2008.
- [44] Sanna Kaasalainen, Anttoni Jaakkola, Mikko Kaasalainen, Anssi Krooks, and Antero Kukko. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. *Remote Sensing*, 3(10): 2207–2221, 2011. ISSN 2072-4292. doi: 10.3390/rs3102207.
- [45] Alireza G. Kashani, Michael J. Olsen, Christopher E. Parrish, and Nicholas Wilson. A review of LiDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration. *Sensors*, 15(11):28099–28128, 2015. ISSN 1424-8220. doi: 10.3390/s151128099.
- [46] Alireza G. Kashani, Michael J. Olsen, Christopher E. Parrish, and Nicholas Wilson. A review of LIDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration. *Sensors*, 15(11):28099–28128, 2015. ISSN 1424-8220. doi: 10.3390/s151128099.
- [47] Dongshin Kim, Jie Sun, Sang Min Oh, J. M. Rehg, and A. F. Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 518–525, May 2006. doi: 10.1109/ROBOT.2006.1641763.
- [48] Ravikrishna Kolluri. Provably good moving least squares. *ACM Transactions on Algorithms*, 4(2):18:1–18:25. ISSN 1549-6325. doi: 10.1145/1361192.1361195.
- [49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
- [50] Antero Kukko. Mobile laser scanning – system development, performance and applications, 2013. URL <http://urn.fi/URN:ISBN:978-951-711-307-6>.
- [51] Antero Kukko, Harri Kaartinen, Juha Hyypä, and Yuwei Chen. Multiplatform mobile laser scanning: Usability and performance. *Sensors*, 12(9):11712–11733, 2012. ISSN 1424-8220. doi: 10.3390/s120911712.
- [52] Antero Kukko, Harri Kaartinen, and M Zanetti. Backpack personal laser scanning system for grain-scale topographic mapping. In *Proceedings of 46th Lunar and Planetary Science Conference*, volume 2407, 2015.
- [53] Richard B Langley. RTK GPS. *GPS World*, 9(9):70–76, 1998.

- [54] Helge A. Lauterbach, Dorit Borrmann, Robin Heß, Daniel Eck, Klaus Schilling, and Andreas Nüchter. Evaluation of a backpack-mounted 3D mobile scanning system. *Remote Sensing*, 7(10):13753–13781, 2015. ISSN 2072-4292. doi: 10.3390/rs71013753.
- [55] Michael A. Lefsky, Warren B. Cohen, Geoffrey G. Parker, and David J. Harding. LiDAR remote sensing for ecosystem studies. *BioScience*, 52(1):19–30, 2002. doi: 10.1641/0006-3568(2002)052[0019:LRSFES]2.0.CO;2.
- [56] *Leica Pegasus: Backpack, Mobile reality capture*. Leica Geosystems AG, mar 2017. Available at https://www.gefos-leica.cz/data/original/skenery/mobilni-mapovani/backpack/leica_pegasusbackpack_ds.pdf, version 03.17.
- [57] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D LiDAR using fully convolutional network. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016. doi: 10.15607/RSS.2016.XII.042.
- [58] Qingquan Li, Liang Zhang, Qingzhou Mao, Qin Zou, Pin Zhang, Shaojun Feng, and Washington Ochieng. Motion field estimation for a dynamic scene using a 3D LiDAR. *Sensors*, 14(9):16672–16691, 2014. ISSN 1424-8220.
- [59] Xingxing Li, Maorong Ge, Xiaolei Dai, Xiaodong Ren, Mathias Fritsche, Jens Wickert, and Harald Schuh. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and galileo. *Journal of Geodesy*, 89(6):607–635, Jun 2015. ISSN 1432-1394. doi: 10.1007/s00190-015-0802-8.
- [60] Xingxing Li, Xiaohong Zhang, Xiaodong Ren, Mathias Fritsche, Jens Wickert, and Harald Schuh. Precise positioning with current multi-constellation global navigation satellite systems: GPS, GLONASS, galileo and BeiDou. *Scientific reports*, 5:8328, 2015.
- [61] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE CVPR*, 2015. doi: 10.1109/CVPR.2015.7298965.
- [62] Juho Lumme, Mika Karjalainen, Harri Kaartinen, Antero Kukko, Juha Hyyppä, Hannu Hyyppä, Anttoni Jaakkola, and Jouko Kleemola. Terrestrial laser scanning of agricultural crops. Technical report, 2008.
- [63] H. G. Maas, A. Bienert, S. Scheller, and E. Keane. Automatic forest inventory parameter determination from terrestrial laser scanner data. *International Journal of Remote Sensing*, 29(5):1579–1593, March 2008. ISSN 0143-1161. doi: 10.1080/01431160701736406.

- [64] Mehdi Maboudi, Dávid Bánhidi, and Markus Gerke. Evaluation of indoor mobile mapping systems. In *GFal Workshop 3D-NordOst 2017 (20th Application-oriented Workshop on Measuring, Modeling, Processing and Analysis of 3D Data)*, 12 2017.
- [65] F. Landis Markley, Yang Cheng, John Lucas Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, Jul 2007. ISSN 0731-5090. doi: 10.2514/1.28949.
- [66] A Masiero, F Fissore, A Guarnieri, M Piragnolo, and A Vettore. Comparison of low cost photogrammetric survey with TLS and leica pegasus backpack 3D modelss. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:147, 2017. doi: 10.5194/isprs-archives-XLII-2-W8-147-2017.
- [67] J. Mason and B. Marthi. An object-based semantic world model for long-term change detection and semantic querying. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conference on*, 2012. doi: 10.1109/IROS.2012.6385729.
- [68] E. Mendes, P. Koch, and S. Lacroix. Icp-based pose-graph slam. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 195–200, Oct 2016. doi: 10.1109/SSRR.2016.7784298.
- [69] Christoph Mertz, Luis E. Navarro-Serment, MacLachlan, et al. Moving object detection with laser scanners. *J. Field Robot.*, 30(1):17–43, January 2013. ISSN 1556-4959. doi: 10.1002/rob.21430.
- [70] Fabien Napolitano. Fiber-optic gyroscopes key technological advantages, 2010.
- [71] *What is LiDAR?* National Ocean Service, Jun 2018. URL <https://oceanservice.noaa.gov/facts/lidar.html>. Accessed: 2019-09-06.
- [72] Yoshua Nava. Visual-LiDAR SLAM with loop closure. Master’s thesis, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden, 2018.
- [73] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011. doi: 10.1109/ISMAR.2011.6092378.
- [74] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM with approximate data association. In *Advanced Robotics, 2005. ICAR. Proceedings.*,

- 12th Int. Conference on*, pages 242–249, July 2005. doi: 10.1109/ICAR.2005.1507419.
- [75] A. Nuchter, M. Bleier, J. Schauer, and P. Janotta. Continuous-Time SLAM—improving google’s cartographer 3D mapping. In *Latest Developments in Reality-Based 3D Surveying and Modelling*, pages 53–73. MDPI, Basel, Switzerland, January 2018. doi: 10.3390/books978-3-03842-685-1/4.
- [76] J. Olive. *Maths: A Student’s Survival Guide: A Self-Help Workbook for Science and Engineering Students*. Cambridge Uni. Press, 2003. ISBN 9780521017077.
- [77] G. Pandey, J.R. McBride, S. Savarese, and R.M. Eustice. Visually bootstrapped generalized ICP. In *Robotics and Automation (ICRA), 2011 IEEE Int. Conference on*, pages 2660–2667, May 2011. doi: 10.1109/ICRA.2011.5980322.
- [78] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice. Toward mutual information based automatic registration of 3D point clouds. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2698–2704, Oct 2012. doi: 10.1109/IROS.2012.6386053.
- [79] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan. Elastic LiDAR fusion: Dense map-centric continuous-time SLAM. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1206–1213, May 2018. doi: 10.1109/ICRA.2018.8462915.
- [80] K. Pathak, A. Birk, et al. Fast registration based on noisy planes with unknown correspondences for 3D mapping. *Robotics, IEEE Transactions on*, 26(3):424–441, June 2010. ISSN 1552-3098. doi: 10.1109/TRO.2010.2042989.
- [81] Anna Petrovskaya and Sebastian Thrun. Efficient techniques for dynamic vehicle detection. In *Experimental Robotics, The Eleventh International Symposium, ISER 2008, July 13-16, Athens, Greece*, pages 79–91, 2008. doi: 10.1007/978-3-642-00196-3_10.
- [82] Anna Petrovskaya and Sebastian Thrun. Model based vehicle tracking for autonomous driving in urban environments. In *Robotics: Science and Systems IV, Eidgenossische Technische Hochschule Zurich, Zurich, Switzerland, June 25-28, 2008*, 2008. doi: 10.15607/RSS.2008.IV.023.
- [83] ND Quadros, PA Collier, and CS Fraser. Integration of bathymetric and topographic LiDAR: a preliminary investigation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:1299–1304, 2008.

- [84] Petri Rönholm, Xinlian Liang, Antero Kukko, Anttoni Jaakkola, and Juha Hyypä. Quality analysis and correction of mobile backpack laser scanning data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:41, 2016. doi: 10.5194/isprsannals-III-1-41-2016.
- [85] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.
- [86] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *2009 IEEE International Conference on Robotics and Automation*, pages 4293–4299, May 2009. doi: 10.1109/ROBOT.2009.5152255.
- [87] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.
- [88] Aleksandr Segal, Dirk Hähnel, and Sebastian Thrun. Generalized-ICP. In Jeff Trinkle, Yoky Matsuoka, and José A. Castellanos, editors, *Robotics: Science and Systems*. The MIT Press, 2009. ISBN 978-0-262-51463-7. doi: 10.15607/RSS.2009.V.021.
- [89] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '85*, pages 245–254, New York, NY, USA, 1985. ACM. ISBN 0-89791-166-0. doi: 10.1145/325334.325242.
- [90] Beril Sirmacek, Yueqian Shen, Roderik Lindenbergh, Sisi Zlatanova, and Abdoulaye Diakite. Comparison of ZEB-1 and leica C10 indoor laser scanning point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:143, 2016. doi: 10.5194/isprs-annals-III-1-143-2016.
- [91] SBG Systems. *Ellipse AHRS and INS, High Performance, Miniature Inertial Sensors, User Manual*.
- [92] G. Tanzmeister, J. Thomas, D. Wollherr, and M. Buss. Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation. In *2014 IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 6090–6095, May 2014. doi: 10.1109/ICRA.2014.6907756.
- [93] G Tucci, V Bonora, A Conti, and L Fiorini. Digital workflow for the acquisition and elaboration of 3d data in a monumental complex: The fortress of saint john the baptist in florence. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42, 2017.

- [94] Martin van Leeuwen and Maarten Nieuwenhuis. Retrieval of forest structural parameters using LiDAR remote sensing. *European Journal of Forest Research*, 129(4):749–770, Jul 2010. ISSN 1612-4677. doi: 10.1007/s10342-010-0381-4.
- [95] Claudio Vanneschi, Matthew Eyre, Mirko Francioni, and John Coggan. The use of remote sensing techniques for monitoring and characterization of slope instability. *Procedia Engineering*, 191:150 – 157, 2017. ISSN 1877-7058. ISRM European Rock Mechanics Symposium EUROCK 2017.
- [96] M. Velas, M. Spanel, M. Hradis, and A. Herout. CNN for very fast ground segmentation in velodyne LiDAR data. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 97–103, April 2018. doi: 10.1109/ICARSC.2018.8374167.
- [97] Martin Velas, Thomas Faulhammer, Michal Spanel, Michael Zillich, and Markus Vincze. Improving multi-view object recognition by detecting changes in point clouds. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, Dec 2016. doi: 10.1109/SSCI.2016.7850045.
- [98] Martin Velas, Michal Spanel, and Adam Herout. Collar line segments for fast odometry estimation from velodyne point clouds. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4486–4495, May 2016. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487648.
- [99] Martin Velas, Michal Spanel, Michal Hradis, and Adam Herout. CNN for IMU assisted odometry estimation using velodyne LiDAR. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 71–77, April 2018. doi: 10.1109/ICARSC.2018.8374163.
- [100] Martin Velas, Michal Spanel, Tomas Sleziak, Jiri Habrovec, and Adam Herout. Indoor and outdoor backpack mapping with calibrated pair of velodyne lidars. *Sensors*, 2019(1):1–34, 2019. ISSN 1424-8220. doi: 10.3390/s19183944.
- [101] Dominic Zeng Wang, I. Posner, and P. Newman. What could move? finding cars, pedestrians and bicyclists in 3D laser data. In *2012 IEEE Int. Conference on Robotics and Automation*, pages 4038–4044, May 2012. doi: 10.1109/ICRA.2012.6224734.
- [102] N. Wojke and M. Häselich. Moving vehicle detection and tracking in unstructured environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 3082–3087, May 2012. doi: 10.1109/ICRA.2012.6224636.
- [103] Oliver J. Woodman. An introduction to inertial navigation. Technical report, Computer Laboratory, University of Cambridge, 2007.

- [104] Stuart Woods. Laser scanning on the go. *GIM International*, pages 29–31, 2016.
- [105] J. Zhang and S. Singh. Visual-LiDAR odometry and mapping: low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181, May 2015. doi: 10.1109/ICRA.2015.7139486.
- [106] J. Zhang, M. Kaess, and S. Singh. Real-time depth enhanced monocular odometry. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4973–4980, Sept 2014. doi: 10.1109/IROS.2014.6943269.
- [107] Ji Zhang and Sanjiv Singh. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference (RSS 2014)*, 2014. ISBN 978-0-9923747-0-9. doi: 10.15607/RSS.2014.X.007.
- [108] M. Zhang, D. D. Morris, and R. Fu. Ground segmentation based on loopy belief propagation for sparse 3D point clouds. In *2015 International Conference on 3D Vision*, pages 615–622, Oct 2015. doi: 10.1109/3DV.2015.76.
- [109] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor. Sensor fusion for semantic segmentation of urban scenes. In *IEEE Int. Conference on Robotics and Automation (ICRA)*, pages 1850–1857, May 2015. doi: 10.1109/ICRA.2015.7139439.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". It is available for L^AT_EX via CTAN as `classicthesis`.

Final Version as of January 7, 2020 (`classicthesis` version 1.0).