PÁZMÁNY PÉTER CATHOLIC UNIVERSITY

ROSKA TAMÁS DOCTORAL SCHOOL OF
SCIENCES AND TECHNOLOGY

KOVÁCS Lóránt

3D Change Detection and Human Pose Estimation In Lidar Perception

PhD Dissertation

Thesis Supervisor:
Prof. Dr. BENEDEK Csaba DSc

Budapest, 2024

Genius is one percent inspiration
and ninety-nine percent
perspiration.

Thomas Edison
[16]

Acknowledgements

This thesis could not have been done without the support of people around me.

I would like to express my deepest gratitude and appreciation to my PhD supervisor, **Csaba Benedek**. Throughout my doctoral journey, Csaba has been an exceptional mentor, guide, and source of inspiration. His unwavering support, expertise, and dedication have played an instrumental role in shaping my research and personal growth. His insightful guidance and feedback have challenged me to think critically, push the boundaries of knowledge, and strive for excellence in my work. Csaba's ability to ask thought-provoking questions and encourage intellectual exploration has been invaluable in shaping the direction of my research. His commitment to my success extended beyond the academic realm.

I am grateful to my alma maters, *Pázmány Péter Catholic University, Faculty of Information Technology and Bionics (PPKE ITK)*, *Cranfield University, School of Aerospace, Transport and Manufacturing (UK)*, and *Benedictine School of Pannonhalma* for providing me with a solid foundation of knowledge and skills, which have been integral to my academic and personal growth.

I express my sincere gratitude to the present and past leaders of *Roska Tamás Doctoral School of Sciences and Technology*, **Gábor Szederkényi** and **Péter Szolgay**. And also, to the deans of the *Faculty of Information Technology and Bionics at Pázmány Péter Catholic University (PPKE)*, **György Cserey** and **Kristóf Iván**. Their support gave me a solid foundation and background for my PhD student life.

And I must not forget **Mrs. Vida Tivadarné Katinka néni**, who was helpful and patient with all my administrative issues and challenges.

Furthermore, I express my deep appreciation to **Tamás Szirányi** and the *Machine Perception Laboratory (MPLAB)* at the *Institute for Computer Science and Control (HUN-REN SZTAKI)* for providing me with all the research equipment, the sensors, the computers, the servers, and the stimulating research environment as well.

I extend my heartfelt appreciation to my co-authors **Balázs Nagy, Balázs Bódis, Marcell Kégl, Örkény H. Zováthi** for their valuable contributions to my publications. I am grateful to my colleagues and friends **Yahya Ibrahim, Balázs Pálffy, József Kövendi, Zsolt Jankó, László Tizedes, Zoltán Rózsa, Sándor Gazdag, and Marcell Golarits** for their unwavering support and invaluable contributions throughout my journey.

I would like to take a moment to express my heartfelt gratitude to my dear friends and former colleagues: **Dóri, Andris, Domi, Tücsök**. Your unwavering support, love, and companionship have been an invaluable source of strength and joy in my life.

Special thanks to those whom I may have not mentioned here by name, but who supported me directly or indirectly in accomplishing my research.

For financial support, thanks to the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002) and of the Artificial Intelligence National Laboratory (RRF-2.3.1-21-2022-00004) programs. Further support was provided by the TKP2021-NVA-27 and TKP2021-NVA-01 grants and by the OTKA #143274 project of the Hungarian National Research, Development and Innovation NRDI Office.

I would also like to express my gratitude to the reviewers of my dissertation for their insightful comments and suggestions, which have significantly contributed to the refinement and enhancement of my work.

I would like to thank my parents, my grandfather, and my siblings for their unconditional support and for everything they did for me not only during my PhD but also until then.

Last but not least, I would like to thank my family, and most importantly my beloved *Klári*, who provided me with a solid background, and a home where I could always refresh and rest. You always supported me, regardless of the obstacles, and understood that I really wanted to achieve this. Finally, I am thankful to my daughter *Bíbor*, and to my sons *Özséb* and *Donát*, who accepted that in the tough periods, I worked more than they wanted. Their love and understanding were always there, and they were the ones who made me smile and laugh even in the darkest moments.

Abstract

In this thesis, I propose solutions for two research problems in the 3D perception of terrestrial mobile laser scanners.

In the first part of the dissertation, a novel deep neural network-based change detection approach is introduced, which can robustly extract changes between sparse and weakly registered point clouds obtained in a complex street-level environment, tolerating up to 1 m translation and 10° rotation misalignment between the corresponding 3D point cloud frames. In the proposed *ChangeGAN* model, the input point clouds are represented by range images, enabling the use of 2D convolutional neural networks. The result is a pair of binary masks indicating the change regions on each input range image, which can be back-projected to the input point clouds without loss of information. The proposed method utilizes a generative adversarial network-like (GAN) architecture, combining Siamese-style feature extraction, U-net-like multiscale feature usage, and Spatial Transformer Network blocks for optimal transformation estimation. I have evaluated the proposed method on various challenging scenarios, including a new dataset I created, demonstrating its superiority over state-of-the-art change detection methods.

The second part of the thesis focuses on 3D human pose estimation in Lidar point clouds. While Lidar sensors are generally expensive, I demonstrated that with a new and affordable Lidar sensor (*Livox Avia*) featuring a unique Non-Repetitive Circular Scanning (NRCS) pattern, human pose estimation tasks can be solved efficiently despite the sparseness of the point cloud measurements.

My proposed solution needs to implement two challenging steps. The first one concerns foreground-background segmentation of the recorded 3D Lidar point cloud frames. For this reason, I proposed a novel point-level foreground-background separation technique for measurement sequences of an NRCS Lidar sensor mounted in a fixed surveillance position. Here, the main challenge has been efficiently balancing the spatial and temporal resolution of the recorded range data. To address this, a very high-resolution background model of the sensor's Field of View is automatically generated and maintained. For real-time analysis of dynamic objects, a low integration time is used. Consequently, laser reflections from foreground objects provide sparse but geometrically ac-

curate samples of moving objects. These samples are valuable for higher-level shape description, object detection, and pose estimation. I demonstrate the efficiency of this new approach using various realistic NRCS Lidar measurement sequences from my new dataset.

The second step of my proposed Lidar surveillance approach addresses 3D human pose estimation based on purely the NRCS Lidar measurements. I proposed here a novel, vision transformer-based pose estimation method called *LidPose* for real-time 3D human skeleton detection in NRCS Lidar point clouds exploiting my previously introduced foreground segmentation approach. To train and evaluate the *LidPose* method, I created a novel, real-world, multi-modal dataset containing camera images and Lidar point clouds from a Livox Avia sensor, with annotated 2D and 3D human skeleton ground truth. Using this dataset, I demonstrated that the proposed method can efficiently and accurately estimate 3D human poses using only NRCS Lidar point clouds.

Contents

Chapter 1

Introduction

The understanding of the spatial environment has increasing importance in various fields, such as robotics, autonomous vehicles, surveillance, and augmented reality. In recent years, advancements in 3D perception technology have significantly enhanced the understanding of complex environments.

This dissertation covers two research areas of 3D perception using terrestrial mobile laser scanners, specifically focusing on Lidar point clouds.

The first research topic is change detection in Lidar point clouds, described in Chapter 2. Change detection is a crucial technique for various applications, including urban planning, environmental monitoring, monitoring dynamic environments, and infrastructure maintenance. 3D Lidar change detection algorithms analyze sequential point cloud data to identify significant changes over time, such as structural modifications, object movement, or environmental disturbances.

The second research topic is human pose estimation using only Lidar point clouds, described in Chapter 4. Human pose estimation involves detecting and predicting the positions of various body parts. It is relevant in various applications such as human-computer interaction, surveillance, and biomechanical analysis. Human pose estimation is traditionally performed using visual data, recorded with cameras. However, this research investigates the feasibility and advantages of utilizing 3D Lidar data for this purpose.

Together, these topics highlight the potential of Lidar technology in advancing 3D perception capabilities, paving the way for innovative applications and improved methodologies in various fields.

I introduce the two research topics in Sections 1.1 and 1.2, followed by the

general introduction to the Lidar sensor (Section 1.3) and the description of the two types of Lidar sensors used for my research in Sections 1.3.1 and 1.3.2.

1.1 Change detection

Due to the increasing population density, and the rapid development of smart city applications and autonomous vehicle technologies, growing demand is emerging for automatic public infrastructure monitoring and surveillance applications. Detecting possibly dangerous situations caused by e.g., missing traffic signs, and damaged street furniture is crucial. Expensive and time-consuming efforts are required therefore by city management authorities to continuously analyze and compare multi-temporal recordings from large areas to find relevant environmental changes.

From the perspective of machine perception, this task can be formulated as a change detection problem. In video surveillance applications [17, 18], change detection is a standard approach for scene understanding by estimating the background regions and by comparing the incoming frames to this background model. Change detection is also a common task in many remote sensing applications, which require the extraction of the differences between aerial images, point clouds, or other measurement modalities [19, 20]. However, the vast majority of existing approaches assume that the compared image or point cloud frames are precisely registered since either the sensors are motionless, or the accurate position and orientation parameters of the sensors are known at the time of each measurement.

1.2 Human pose estimation

The main task of pose estimation is to localize the anatomical keypoints of the human body. Human pose estimation is an essential task in machine perception and has several real-world applications among others in robotics [21], security and surveillance [22, 23], autonomous driving [24], human-computer interaction [25], sports performance analysis [26], healthcare [27], forensic science [28], entertainment and gaming [29].

The input data of the human pose estimation can be captured by various types of sensors. Human pose estimation is most commonly solved by

camera-based methods [30–34] in the image space. However, such solutions are inherently limited by the camera's incapability to directly measure distance, the high sensitivity of the captured images to various lighting and weather conditions, and the varying visual appearances of real-world objects.

Other sensors, such as motion capture equipment using Inertial Measurement Units (IMU) [35], depth sensors [21,36] and Lidar sensors [37] can provide additional information to the pose estimation methods to increase the prediction accuracy by decreasing depth or occlusion ambiguities [38].

In applications, where privacy is a serious concern, Lidar-based human surveillance can be efficiently applied as the observed people cannot be identified by an observer in the sparse point cloud.

1.2.1 2D Human Pose Estimation

The estimated pose is represented as a "stick figure" for each person or more commonly referred to as the skeleton, shown in Figure 1.1.

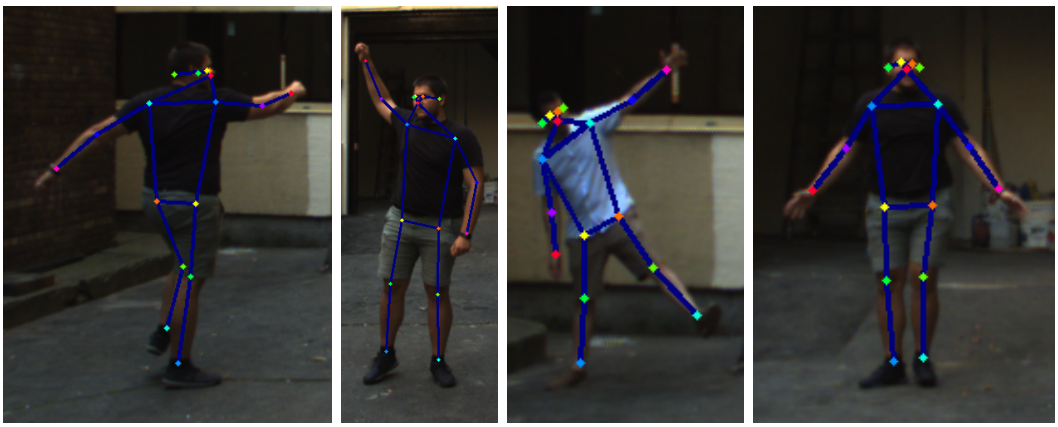


Figure 1.1. Example for 2D pose estimation using camera images by ViTPose [33]. The skeletons are displayed over the input images. The colorful dots represent the joints of the skeleton, and the edges are colored with blue [39].

In single-person 2D human pose estimation, the goal is to localize the joints of a single human subject. If multiple people are present on a single image, a preprocessing step detects the people in the input, and it determines the person's bounding boxes. Each box is treated as a separate input for the pose estimator algorithm, and the results are then stitched back onto the frame.

2D human pose regression approach directly maps the joints of the detected subject to the 2D locations [40], thus giving an end-to-end solution. Regression-based approaches marked the initial step into deep learning-based

human pose estimation methods. After the successful use of cascaded deep neural networks [41], the research interest moved towards convolutional neural networks.

Joint position heatmap or *body part detection*-based methods encode the different body parts and joints in a series of heatmaps. The maps encode the likelihood of the presence of a given joint at a given position. The heatmaps are created as Gaussian distributions, with the mean being the ground truth (GT) location of the body part [42]. With these map-based approaches, a post-processing step is required for the extraction of the precise keypoint locations from the heatmaps [43].

1.2.2 3D Human Pose Estimation

3D human pose estimation predicts the human pose by estimating the 3D position of each joint in the human body. This can be applied in motion capture, augmented reality, and sports analysis [44].

Most 3D methods also depend on monocular images or videos, presenting an ill-posed inverse problem due to the projection and the occlusions in the input data [45]. To resolve these ambiguities, multimodal approaches have been proposed. However, these approaches usually restrict the problem to indoor environments and limit the number of individuals captured in the dataset [46]. Consequently, models trained on such datasets often suffer from poor generalization [34].

Recent results described in [47, 48] introduce the use of off-the-shelf millimeter wave radars for the human pose estimation task. Both approaches use camera-based methods to train and evaluate the radar-based solution. It has to be noted, that the radar's detection density is significantly lower than the density of both the RMB lidar and the NRCS lidar.

1.3 Lidar sensor

Lidar is an active sensor that illuminates the surroundings by emitting laser beams. Distances are measured precisely by processing the received laser reflections from the surfaces. The Lidar sensor works efficiently under different lighting and illumination conditions. However, this robustness decreases in harsh weather conditions: the sensor has weaker performance in fog, snow,

or heavy rain [49]. In dense fog or heavy rain, the water droplets reflect the emitted laser beams by creating false distance measurements from the observed scene. A possible approach for weather-related point cloud denoising is the WeatherNet network, described in [50].

A general Lidar operates by scanning its Field of View (FoV) with one or several near-infrared (NIR) laser beams.

The laser beam is reflected to the scanner from the environment, the returned signal is received by a photodetector. Fast electronics filter the signal and measure the time difference between the transmitted and received signals, which is proportional to the the distance of the reflecting object. The range is estimated from the sensor model based on this calculated time difference. The Lidar outputs 3D point clouds that correspond to the scanned environment and the intensities that correspond to the reflected laser energies [51]. The Lidar's maximum range is limited by the eye-safe transmission power regulations.

The scanning system of a Lidar sensor is responsible for the rapid exploration of the observed space. A few scanning methodologies at different Lidar types are introduced below.

In the *mechanical spinning-type* sensors (rotating multi-beam (RMB) Lidar) the laser beams are steered through a rotating sensor head, having a moving mirror and optics inside. The Lidar I used for my change detection research works following this principle, described in detail in Section 1.3.1.

Another mechanical approach uses *rotation of prisms* to direct the laser beams. The Lidar I used for my Lidar-only human pose estimation research works following this scanning method, described in detail in Section 1.3.2.

The scanning can also be achieved by moving a "mirror" in a chip with elastic and electromagnetic forces in a *Micro-electromechanical system* (MEMS) [52].

Flash Lidars does not have any rotating component [53]. A single emitted laser beam is spread by an optical diffuser to illuminate the whole scene, and the reflections are detected on an array of photodiodes.

1.3.1 Velodyne HDL-64 rotating multi-beam Lidar sensor

The Velodyne HDL-64 sensor (shown in Figure 1.2a) is a high-resolution and high-performance RMB Lidar sensor, that is designed to help the real-time

(a) **(b)**

Figure 1.2. Velodyne HDL-64 rotating multi-beam Lidar sensor and its recorded point cloud in urban environment

perception of autonomous robots and vehicles. It captures high-definition and real-time 3D measurements from its surrounding environment. The sensor has 64 laser beams, determining a 26.9° vertical FoV. Due to the rotating head of the sensor, its horizontal FoV is 360°. The measured data's spatial accuracy is 1-2 cm. Due to the sensor characteristics, the point density quickly decreases with the distance from the sensor. The Velodyne HDL-64 is a pioneer of the RMB Lidars. Recent RMB Lidar sensors are available on the market (e.g., produced by Ouster) having similar characteristics, but their size and consumption have decreased significantly, making the measurements and the research conducted with the Velodyne Lidar in this research still relevant [54].

Ring patterns can be observed in the recorded point clouds, as can be seen in Figure 1.2b, as the laser beams are rotated along the sensor's vertical axis. The sensor continuously streams the 3D measurements, which are collected to point cloud frames, where the term *frame* refers to a single horizontal turnaround of the sensor head.

1.3.2 Livox Avia Lidar sensor with Non-repetitive Circular Scanning

The Livox Avia sensor [55], shown in Figure 1.3, is a lightweight Lidar sensor that has a unique, Non-repetitive Circular Scanning (NRCS) technique. The sensor has six Lidar beams organized in a linear beam array, which is moved and rotated inside the sensor to scan its FoV (horizontal: 70°, vertical: 77°). The sensor has a detection range of up to 320 m if the target object reflects at least 80% of the light and 190 m at 10% object reflectivity. The sensor's distance error (1σ) is less than 2 cm at 20 m. The angular error (1σ) is smaller than 0.05° [56].

Figure 1.3. Livox Avia Lidar sensor

Unlike most RMB Lidars, which boost a repetitive scanning pattern, the Avia does not repeat the exact scanning paths in every frame, but instead, the lasers cover new parts of the FoV. This key difference is both beneficial and implicates some disadvantages. NRCS Lidars cover the complete FoV over time, providing rich spatial information, especially in static scenarios. On the other hand, because the same region is scanned less frequently than by using "regular" RMB Lidars, dynamic objects, such as humans may cause challenges as they induce heavy motion blur in the recorded NRCS point clouds.

Another sensor-specific property of the recorded data is the inhomogeneous point cloud density. More specifically, while the center of the FoV is scanned in every rotation of the pattern, outer regions are sampled less frequently, as demonstrated in Figure 1.4. This inhomogeneous point density distribution makes it difficult to apply existing Lidar point cloud processing approaches on NRCS Lidar point clouds. Note that apart from depth data, the sensor also records the reflection intensity of the laser beams within the range of $0-100\%$ according to the Lambertian reflection model [56].

The sensor continuously records distance measurements with corresponding timestamps following its non-repetitive circular pattern in its FoV. By setting a fixed integration time, the consecutively collected points can be grouped into separate Lidar time frames. The main challenge is to efficiently balance

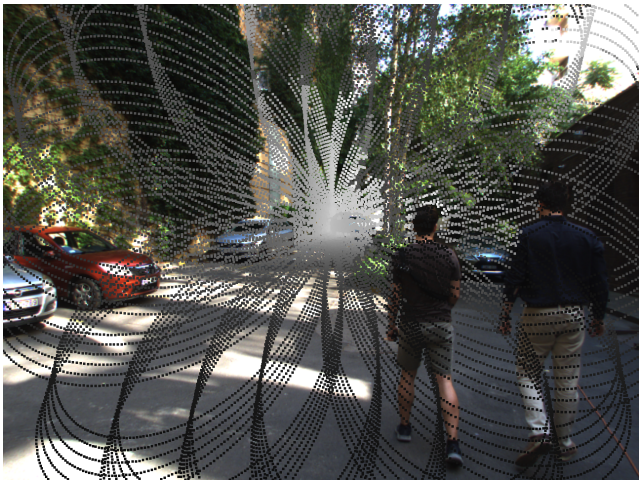


Figure 1.4. NRCS Lidar point cloud with 100 ms integration time represented as a 2D range image overlaid on a sample camera image. The point cloud is colored by the distance: the lighter the point's color, the greater its distance.

between the spatial and the temporal resolution of the recorded range data.

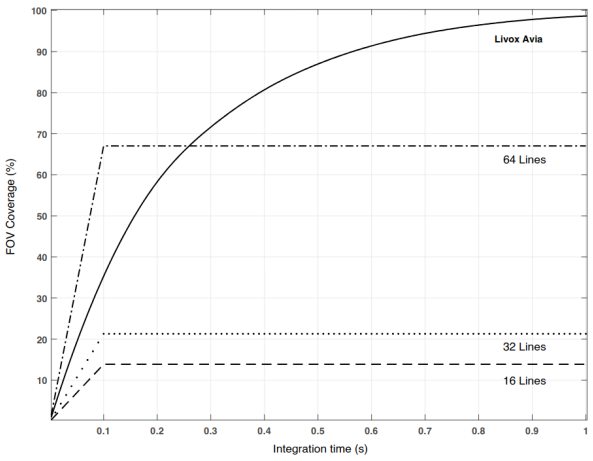


Figure 1.5. Change in FOV coverage over time for the Livox Avia compared to rotating multi-beam sensors with traditional scanning. Source: [56]

While allowing larger integration time, the laser beams cover a higher proportion of the FoV, as shown in Figure 1.5, yielding high spatial measurement resolution of the measurement frame. Figure 1.6 shows how the scanning patterns overlap each other as time passes and create a denser measurement. The object movements of dynamic objects in the observation area induce various artifacts (e.g., blurred pedestrian silhouettes), which do not allow efficient dynamic event analysis. For example, the Livox Avia sensor collects 240000 points within a time window of $1s$, as can be seen in Figure 1.7b. On the other

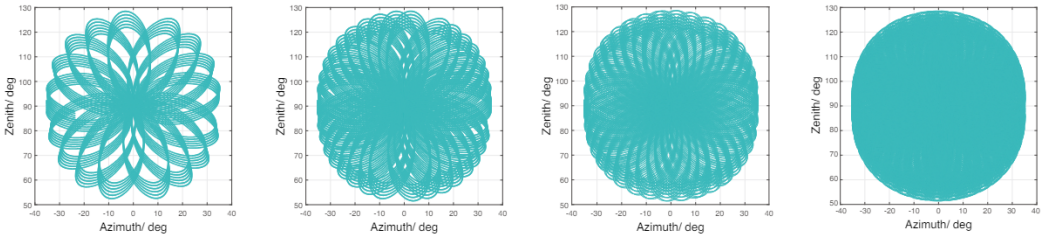
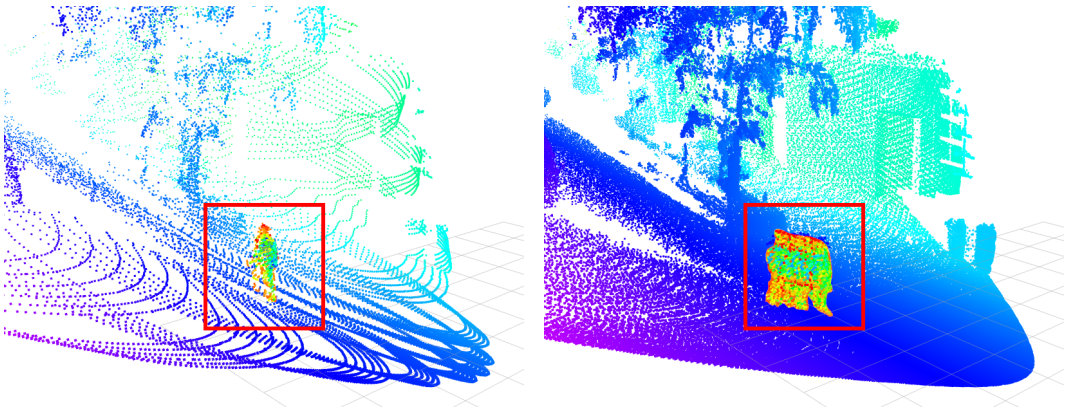


Figure 1.6. Typical point cloud pattern inside the FoV of the Livox Avia Lidar sensor after (from left to right) 0.1 seconds, 0.5 seconds, 1 second, 3 seconds. Source: [56]



(a) NRCS Lidar point cloud with 100 ms integration time

(b) NRCS Lidar point cloud with 1000 ms integration time

Figure 1.7. Point clouds recorded with different integration times using the NRCS Lidar. The increased integration time brings more density, it also introduces motion blur on dynamic objects, as shown with the moving pedestrian marked with the red rectangle. The pedestrian's points are colored with the Lidar's intensity, the background is colored by the Y-axis value.

hand, if the measurements are collected in a narrow time window (e.g., in 100 ms) the resulting point clouds are very sparse, which phenomenon yields a loss of details across the spatial dimension of the FoV: a sample frame of 24000 points is shown in Figure 1.7a.

As demonstrated in [57, 58], this type of NRCS Lidar is suitable for most of the use case scenarios, including traditional mapping and low-speed autonomous driving. The NRCS Lidar sensor can provide measurements for real-time scene analysis, while the sensor is available on the market at affordable prices compared to the other Lidar technologies [59].

Chapter 2

Change detection in coarsely registered point clouds

In this chapter, I introduce a novel change detection approach called *ChangeGAN* [1], [3] for coarsely registered point clouds in complex street-level urban environments.

Mobile and terrestrial Lidar sensors (introduced in Section 1.3) can obtain point cloud streams, providing accurate 3D geometric information in the observed area. Lidar is used in autonomous driving applications supporting the scene understanding process, and it can also be part of the sensor arrays in ADAS systems of recent high-end cars. Since the number of vehicles equipped with Lidar sensors is rapidly increasing on the roads, one can utilize the tremendous amount of collected 3D data for scene analysis and complex street-level change detection. Besides, change detection between the recorded point clouds can improve virtual city reconstruction or Simultaneous Localization and Mapping (SLAM) algorithms [60].

Processing street-level point cloud streams is often a significantly more complex task than performing change detection in airborne images or Lidar scans. From a street-level point of view, one must expect a larger variety of object shapes and appearances, and more occlusion artifacts between the different objects due to smaller sensor-object distances.

Also, the lack of accurate registration between the compared 3D terrestrial measurements may mean a crucial bottleneck for the whole process, for two different reasons: *First*, in a dense urban environment, GPS/GNSS-based accurate self-localization of the measurement platform is often not possible [61].

Second, the differences in viewpoints and density characteristics between the data samples captured from the considered scene segments may make automated point cloud registration algorithms less accurate [61].

In this chapter, a deep neural network-based change detection approach is proposed, which can robustly extract changes between sparse point clouds obtained in a complex street-level environment. As a key feature, the proposed method does not require precise registration of the point cloud pairs. Based on our experiments, it can efficiently handle up to 1 m translation and 10° rotation misalignment between the corresponding 3D point cloud frames.

2.1 Related works

As one of the most fundamental problems in multitemporal sensor data analysis, change detection (introduced in Section 1.1) has had a vast bibliography in the last decade. Besides methods working on remote sensing images, several change detection techniques deal with *terrestrial* measurements, where the sensor is facing towards the horizon and is located on or near the ground. In these tasks optical cameras [62] and rotating multi-beam Lidars [63] are frequently used, solving problems related to surveillance, map construction, or SLAM algorithms [64].

2.1.1 Prior approaches

The related works are categorized based on the applied methodology they use for change detection.

Many approaches are based on *handcrafted features*, such as a set of pixel- and object-level descriptors [65], occupancy grids [66], volumetric features, and point distribution histograms [64], but they all need preliminarily registered inputs. Only a few feature-based techniques deal with compensating small misregistration effects, such as [67], where terrestrial images and point clouds are fused to perform change detection.

Neural network-based change detection techniques can handle in general more robustly the variances originating from viewpoint differences, most frequently using Siamese network architectures [68]. However, prior approaches solely focus here on visual change detection problems in aerial [69] or street-view [62, 70] optical image pairs, and this task is yet to be solved for real

Lidar point cloud-based change detection problems. A new method for detecting structural changes from city images is described in [71]. It creates 3D point clouds using Structure-from-Motion (SfM) from the images and uses a deep-learning-based registration on the 3D clouds.

2.1.2 Registration issues

Most of the methods require that the compared measurements are either recorded from a static platform, or they can be accurately registered into a joint coordinate system by using external navigation sensors, and/or robust image/point cloud matching algorithms. The later registration step is critical for real-world 3D perception problems, since the recorded 3D point clouds often have strongly inhomogeneous density, and the blobs of the scanned street-level objects are sparse and incomplete due to occlusions and the availability of particular scanning directions only. Under such challenging circumstances, conventional point-to-point, patch-to-patch, or point-to-patch correspondence-based registration strategies often fail [72].

Our published paper [1] and submitted patent [3] paper presents the first approach to solving the change detection problem among sparse, coarsely registered terrestrial Lidar point clouds, without needing an explicit fine registration step. Utilizing the STN [73] layer, the proposed model can automatically handle errors of coarse registration.

Our proposed deep learning-based method can extract and combine various low-level and high-level features throughout the convolutional layers, and it can learn semantic similarities between the point clouds, leading to its capability of detecting changes without prior registration. A clear difference between the proposed change detection method and the state-of-the-art is the *adversarial training strategy* which has a regularization effect, especially on limited data. The other main difference is the built-in spatial transformer network yielding the proposed model to be able to learn and handle coarse registration errors.

2.2 Proposed method

Several Lidar devices, such as the RMB sensors introduced in Section 1.3.1 provide high frame rate point cloud streams containing accurate, but relatively sparse 3D geometric information from the environment. These point clouds

can be used for infrastructure monitoring, urban planning [74], and SLAM [60].

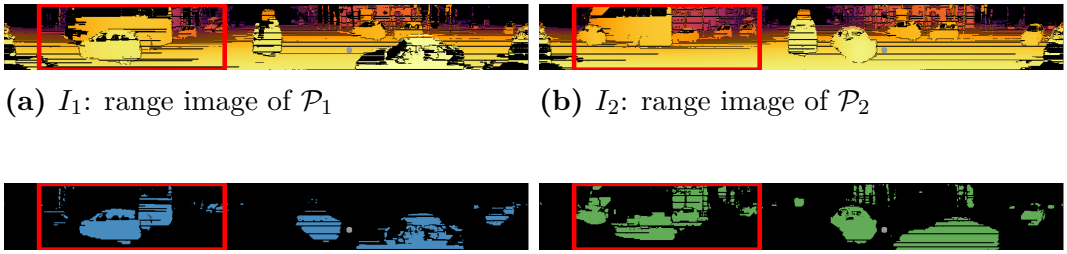
The goal of our proposed method is to extract changes between two coarsely registered and sparse Lidar point clouds, $\mathcal{P}_1$ and $\mathcal{P}_2$. To formally define our change detection task, several considerations should be taken. *First*, both input point clouds may contain various dynamic or static objects, which are not present in the other measurement sample. *Second*, due to the lack of registration, we cannot use a single common voxel grid for marking the locations of changes between the two point clouds.

Instead, using a $\mu(.)$ point labeling process, we separately mark each point $p \in \mathcal{P}_1 \cup \mathcal{P}_2$ as changed ($\mu(p) = \text{ch}$) or unchanged background ($\mu(p) = \text{bg}$), respectively. We label a point $p_1 \in \mathcal{P}_1$ as changed if the surface patch represented by point p_1 in $\mathcal{P}_1$ is not present (changed or occluded) in point cloud $\mathcal{P}_2$ (the label of a point $p_2 \in \mathcal{P}_2$ is similarly defined).

2.2.1 Range image representation

The proposed solution extracts changes between two coarsely registered Lidar point clouds in the range image domain. For example, creating a range image from an RMB Lidar sensor's point stream is straightforward [75] as its laser emitter and receiver sensors are vertically aligned, thus every measured point has a predefined vertical position in the image, while consecutive firings of the laser beams define their horizontal positions. Geometrically, this mapping is equivalent to transforming the representation of the point cloud from the 3D Descartes to a spherical polar coordinate system, where the polar direction and azimuth angles correspond to the horizontal and vertical pixel coordinates, and the distance is encoded in the corresponding pixel's 'intensity' value. Note that range image mapping can also be implemented for other (non-RMB) Lidar technologies, such as Livox sensors. Using appropriate image resolution, the conversion of the point clouds to 2D range images is reversible, without causing information loss. Besides providing a compact data representation, using the range images makes it also possible to adopt 2D convolution operations by the used neural network architectures.

The proposed deep learning approach takes as input two coarsely registered 3D point clouds $\mathcal{P}_1$ and $\mathcal{P}_2$ represented by range images I_1 and I_2, respectively (shown in Figures 2.1a and 2.1b) to identify changes. Our architecture assumes that the images I_1 and I_2 are defined over the same pixel lattice S, and have



(a) I_1: range image of $\mathcal{P}_1$

(b) I_2: range image of $\mathcal{P}_2$

(c) Λ_1: ground truth mask of changes in range image I_1

(d) Λ_2: ground truth mask of changes in range image I_2

Figure 2.1. Input data representation. (a), (b): range images I_1, I_2 from a pair of coarsely registered point clouds $\mathcal{P}_1$ and $\mathcal{P}_2$. (c), (d): binary ground truth change masks Λ_1, Λ_2 for the range images I_1 and I_2, respectively. The *red rectangle* marks the region displayed in 3D in Figure 2.8.

the same spatial *height (h)*, *width (w)* dimensions.

Usually, change detection algorithms working on multitemporal image pairs [62] explicitly define a test and a reference sample, and changes are interpreted from the perspective of the reference data: the resulting change mask marks the image regions which are changed in the test image compared to the reference one. However, this approach cannot be adopted in our case. It is not relevant to assign a single binary change/background label to the pixels of the joint lattice S of the range images, as they may represent different scene locations in the two input point clouds. For this reason, we represent the change map by a two-channel mask image over S, so that to each pixel $s \in S$ we assign two binary labels $\Lambda_1(s)$ and $\Lambda_2(s)$.

Following our change definition used earlier in 3D, for $i \in \{1, 2\}$, $\Lambda_i(s) = \text{ch}$ encodes that the 3D point $p_i \in \mathcal{P}_i$ projected to pixel s should be marked as change in the original 3D point cloud domain of $\mathcal{P}_i$, i.e., $\mu(p_i) = \text{ch}$ (see Figures 2.1c and 2.1d).

Next, our change detection task can be reformulated in the following way: our network extracts similar features from the range images I_1 and I_2, then it searches for the high correlation between the features, and finally, it maps the correlated features to two binary change mask channels Λ_1 and Λ_2, having the same size as the input range images.

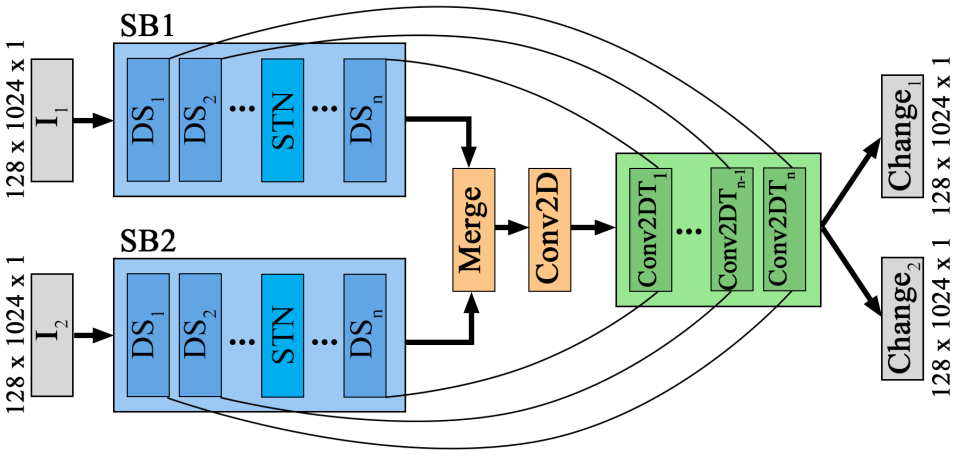


Figure 2.2. Proposed *ChangeGAN* architecture. Notations of components: SB1, SB2: Siamese branches, DS: downsampling, STN: spatial transformer network, Conv2DT: transposed 2D convolution

2.2.2 ChangeGAN architecture

For our purpose, we propose a new generative adversarial neural network-like architecture, more specifically a discriminative method, with an additional adversarial discriminator as a regularizer, called *ChangeGAN*, which is shown in Figure 2.2.

Since the main goal is to find meaningful correspondences between the input range images I_1 and I_2, we have adopted a Siamese style [68] architecture to extract relevant features from the input range image pairs.

The Siamese architecture is designed to share the weight parameters across multiple branches, allowing us to extract similar features from the inputs and decrease memory usage and training time. Each branch of the Siamese network consists of fully convolutional downsampling blocks. The first layer of the downsampling block is a 2D convolutional layer with a stride of 2 which has a 2-factor downsampling effect along the spatial dimensions. This step is followed by using a batch normalization layer, and finally, we activate the output of the downsampling block using a leaky ReLU function. Next, we concatenate the outputs of the Siamese branches for all feature channels, and we apply a 1×1 convolutional layer to aggregate the merged features.

The second part of the proposed model contains a series of transposed convolutional layers to upsample the signal from the lower-dimensional feature space to the original size of the 2D input images. Finally, a 1×1 convolutional layer, activated with a sigmoid function, generates the two binary

change maps Λ_1 and Λ_2. To regularize the network and prevent over-fitting, we use the dropout technique after the first two transposed convolutional layers. To improve the change detection result, we have adapted an idea from U-net [76] by adding higher resolution features from the downsampling blocks to the corresponding transposed convolutional layers.

The branches of the Siamese network can extract similar features from the inputs. In our case, as the point clouds are coarsely registered, the same regions of the input range images might not be correlated with each other. To achieve more accurate feature matching, we have added Spatial Transformer Network blocks [73] for both Siamese branches (see Figure 2.2). STN can learn an optimal affine transformation between the input feature maps to reduce the spatial registration error between the input range images. Furthermore, STN dynamically transforms the inputs, also yielding an advantageous augmentation effect.

2.2.3 Training ChangeGAN

A competitive classifier-discriminator-based adversarial training was implemented for the *ChangeGAN* network, as shown in Figure 2.3.

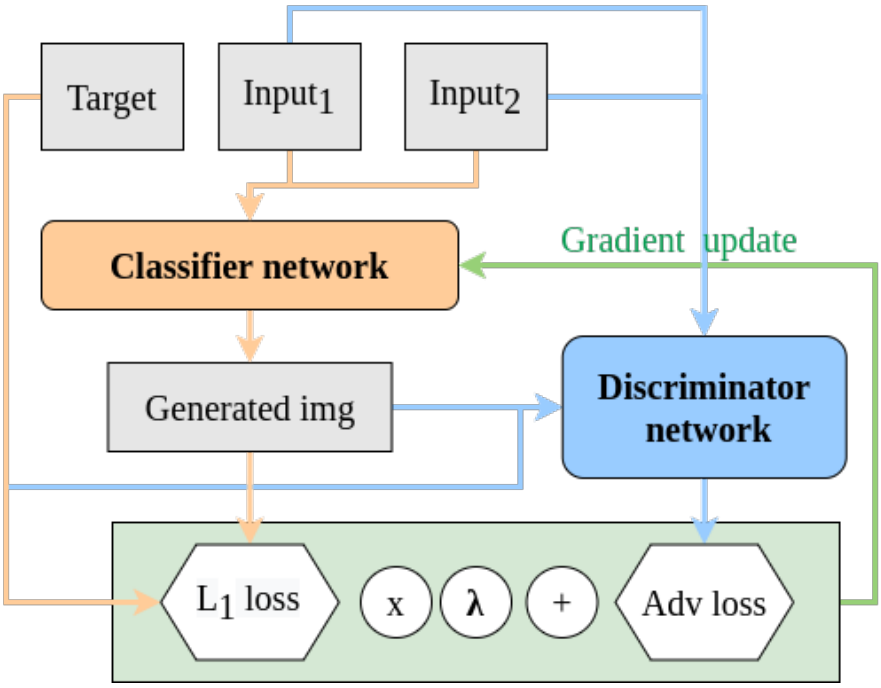


Figure 2.3. Proposed adversarial training strategy of the *ChangeGAN* architecture.

The *classifier* network is responsible for learning and predicting the changes between the range image pairs. In each training epoch, the classifier model is trained on a batch of data. The actual state of the classifier is used to predict validation data, which is fed to the discriminator model.

The *discriminator* network is a fully convolutional network that classifies the output of the classifier network. The discriminator model divides the image into patches and decides for each patch whether the predicted change region is real or fake. During training, the discriminator network forces the classifier model to create better and better change predictions, until the discriminator cannot decide about the genuineness of the prediction.

Figure 2.3 demonstrates the proposed adversarial training strategy. We calculate the L1 Loss (L_{L1}) as the mean absolute error between the generated image and the target image, and we define the Adversarial Loss (L_{Adv}), which is a sigmoid cross-entropy loss of the feature map generated by the discriminator and an array of ones. The final loss function of the method (L) is the weighted combination of the Adversarial Loss and the L1 Loss:

$$L = L_{\text{Adv}} + \lambda * L_{L1}.$$

Based on our experiments, we set $\lambda = 300$.

Both the classifier and the discriminator part of the GAN-like architecture were optimized by the Adam optimizer and the learning rate was set to 10^{-5}. We have trained the model for 300 epochs, which takes almost two days. At each training epoch, we have updated the weights of both the classifier and the discriminator.

We note here, that the *ChangeGAN* method can be trained without the Adversarial Loss (L_{Adv}), relying only on L1 loss. In our preliminary experiments, we followed this simpler approach, which was able to predict some change regions, but the results were notably ambiguous. To increase the generalization ability, we applied the adversarial training strategy in the proposed final model.

2.2.4 Change detection dataset

Considering that the main purpose of the presented *ChangeGAN* method is to extract changes from coarsely registered point clouds, for model training and evaluation we need a large, annotated set of point cloud pairs, collected

in the same area with various spatial offsets and rotation differences.

Following our change definition in Section 2.2, the annotation should accurately mark the point cloud regions of objects or scene segments that appear only in the first frame, or only in the second frame, or which ones are unchanged and thus observable in both frames (see Figures 2.4 and 2.8).

Since the available point cloud benchmark sets cannot be used for this purpose, we have created a new Lidar-based urban dataset called *Change3D*[1]. The measurements were recorded over two days in downtown Budapest using a Velodyne HDL-64 RMB Lidar mounted on a car To our knowledge, this *Change3D* dataset is the largest point cloud dataset for change detection, which contains both registered and coarsely registered point cloud pairs.

2.2.4.1 Ground truth creation approach

Since manual annotation of changes between 3D point clouds is very challenging and time-consuming, we proposed a semi-automatic method using simulated registration errors to create GT for our change detection approach. To ensure the accuracy of the GT, we performed the change labeling for registered point cloud pairs captured from the same sensor position and orientation, then we randomly transformed the reference positions and orientations of the second frames yielding a large set of accurately labeled coarsely registered point cloud pairs. Thereafter, this set has been divided into disjunct training and test sets which could be used to train and quantitatively evaluate the proposed method.

The remaining parts of the collected data including originally unregistered point cloud pairs have been used for qualitative analysis through visual validation (see for example Figure 2.4.) of the model performance.

2.2.4.2 Core data creation for GT annotation

We selected 50 different locations during the test drive when the measurement platform was motionless for a period: it was stopped by traffic lights, crossroads, zebra crossings, parking situations, etc. These locations were taken both from narrow streets from the downtown and wide, large junctions as well. At each location, we took 100 recorded point clouds, and then we randomly selected 400 point cloud pairs among them, obtaining for the 50 locations a

[1]Dataset link: `http://mplab.sztaki.hu/geocomp/Change3D.html`

total number of 20000 point cloud pairs on which the training set was based. The test set is based on 2000 point cloud pairs, which were selected similarly, but in terms of locations and recording time stamps, the test samples were completely separated from the training data.

In these recordings, the differences among the point clouds were only caused by the moving dynamic objects such as vehicles and pedestrians. Alongside the exploitation of real object motion and occlusion effects, some further artificial changes have been synthesized by manually adding and deleting various street furniture elements to selected point cloud scenes. Also, we segmented the point clouds roughly to planes [77] and randomly deleted some selected 2D rectangular segments.

2.2.4.3 Semi-automatic change extraction

Since the above-discussed frame pairs are taken in the same global coordinate system, they can be considered as *registered*. Their GT change annotation could be efficiently created in a semi-automatic way: A high-resolution 3D voxel map was built on a given pair of point clouds. The voxel size defines the resolution of the change annotation. The length of the change annotation cube was set to 0.1 m in all three dimensions, following the voxel size recommendations from [78, 79]. All voxels were marked as changed if 90% of the 3D points in the given voxel belonged to only one of the point clouds. Thereafter, minor observable errors were manually eliminated by a user-friendly point cloud annotation tool. Finally, in both point clouds, all points belonging to *changed voxels* received a $\mu^{\mathrm{GT}}(p) = \mathrm{ch}$ GT labels, while the remaining points were assigned to $\mu^{\mathrm{GT}}(p) = \mathrm{bg}$ labels.

2.2.4.4 Registration offset

To simulate the coarsely registered point cloud pairs requested by our *ChangeGAN* approach, we have applied randomly an up to ± 1 m translation and an up to $\pm 10^{\circ}$ rotation transform around the z-axis for the second frame ($\mathcal{P}_2$) of each point cloud pair, both in the training and test datasets. The $\mu^{\mathrm{GT}}(p)$ GT labels remained attached to the $p \in \mathcal{P}_2$ points and were transformed together with them.

2.2.4.5 Cloud crop and normalization

In the next step, all 3D points were removed from the point clouds, whose horizontal distances from the sensor were larger than 40 m, or whose elevation values were greater than 5 m above the ground level. This step yielded the capability of normalizing the point distances from the sensor between 0 and 1.

2.2.4.6 Range image creation and change map projection

The transformed 3D point clouds were projected to 2D range images I_1, and I_2 as described in Section 2.2.1 (see Figure 2.1). The Lidar's horizontal 360° FoV was mapped to 1024 pixels and the 5 m vertical height of the cropped point cloud was mapped to 128 pixels, yielding that the size of the produced range image is 1024×128.

We note here, that the Lidar sensor used in this experiment has 64 laser emitters yielding that the height of the original range images should be 64. However, to increase the learning capacity of the network we have doubled and interpolated the data among the height dimension since the 2D convolutional layers with a stride of 2 have a 2-factor downsampling effect. Let us observe that the horizons of the range images are at similar positions in the two inputs due to the cropped height of the input point clouds. Besides the range values, the $\mu^{\mathrm{GT}}(p)$ ground truth labels of the points were also projected to the Λ_1^{GT} and Λ_2^{GT} change masks, used for reference during training and evaluation of the proposed network.

2.3 Experiments

We have trained and evaluated the proposed method using the new *Change3D* dataset (see Section 2.2.4), which contains point cloud pairs recorded by a car-mounted RMB Lidar sensor at different times in dense city environments. For a selected coarsely registered point cloud pair, Figure 2.4 shows the changes predicted by the proposed *ChangeGAN* model.

2.3.1 Reference methods

To the best of our knowledge, there are no existing reference methods in the literature that focus on change detection in *coarsely registered* terrestrial

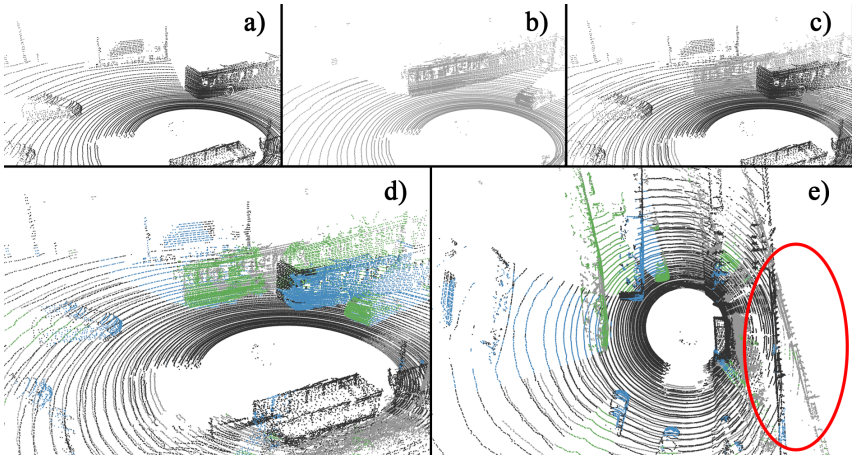


Figure 2.4. Changes detected by *ChangeGAN* for a coarsely registered point cloud pair. (a) and (b) show the two input point clouds, (c) displays the coarsely registered input point clouds in a common coordinate system. (d),(e) present the change detection results: blue and green colored points represent the objects marked as changes in the first- and second point cloud, respectively. The red ellipse draws attention to the global alignment difference between the two coarsely registered point clouds.

point clouds. However, since we reformulated the 3D change detection problem in the 2D range image domain, image-based methods tolerant of registration errors can also be taken into consideration for comparison.

As the *first* baseline, we have chosen the *ChangeNet* method [62], which is a recent approach for visual change detection, being able to detect and localize changes even if the scene has been captured at different lighting, view angle, and seasonal conditions. *ChangeNet* uses a *ResNet* backbone, working with fixed-size input images (224×224). Our created range images could not be given directly to this network, since their resolution (1024×128) and aspect ratio parameters are different. This issue was solved by splitting our range images into eight 128×128 parts, which were upscaled to the image size required by *ChangeNet*. We used the genuine and published implementation of the *ChangeNet* architecture, which was trained using our training data set described in Section 2.2.4.

Our *second* reference method follows a voxel occupancy-based approach [74], where the detection accuracy and the ability to compensate for minor registration errors depend on the chosen voxel resolution. As a core step of the algorithm, [74] applies a registration method between the point cloud pairs. For noise filtering and registration error elimination, a Markov

(a) Ground truth fused change map $\widehat{\Lambda}^{\mathrm{GT}}$ **(b)** *ChangeGAN* output's fused change map $\widehat{\Lambda}$

(c) *ChangeNet* output [62] **(d)** *MRF* output [74]

Figure 2.5. Predicted change masks by the different methods on input data, shown in Figure 2.1. Red rectangles: region displayed in 3D in Figure 2.8.

Random Field (*MRF*) model is adopted, which is defined in the range image domain [74].

The reference methods described above were applied to the task of change detection in 3D point clouds, a use case different from their original purpose. *ChangeNet* was retrained using the depth images for the evaluation.

Comparative results of the proposed method and the reference techniques for the point cloud pair of Figure 2.1 are shown in Figure 2.5, in the range image representation.

Since neither the *ChangeNet* nor the *MRF* methods can distinguish changes by objects of the first and second images, for a direct comparison, we also binarized the output of *ChangeGAN* to get a fused change map $\widehat{\Lambda}$ where $\forall s \in S$: $\widehat{\Lambda}(s) = \max(\Lambda_1(s), \Lambda_2(s))$. The fused GT mask $\widehat{\Lambda}^{\mathrm{GT}}$ was similarly derived.

2.3.2 Quantitative results

We evaluated the proposed *ChangeGAN* method and the two baseline techniques on our new *Change3D* benchmark set. The quantitative performance analysis was performed in the 2D range image domain, using the fused $\widehat{\Lambda}^{\mathrm{GT}}$ mask as a GT reference. To measure the similarity between the binary GT change mask and the binary change masks predicted by the different methods, the mean F1-score, and IoU were calculated alongside pixel-level precision, recall, and accuracy. The used metrics' definition follows standard binary classification metrics [80].

The numerical evaluation results obtained by *MRF-based* [74], *ChangeNet* [62], and the proposed *ChangeGAN* methods over the 2000 range image pairs of the test dataset, are shown in Table 2.1. As Figure 2.6

Table 2.1. Performance comparison of the proposed *ChangeGAN* method to *ChangeNet* [62] and to the *MRF*-based reference approach [74]

	ChangeGAN	ChangeNet	MRF-based
Accuracy	**0.93**	0.78	0.78
Precision	**0.83**	0.43	0.44
Recall	0.71	0.59	**0.88**
F1-score	**0.76**	0.48	0.58
IoU	**0.62**	0.42	0.32
Execution time (s)	0.06	**0.004**	0.51

demonstrates, the *ChangeGAN* method outperforms both reference methods in terms of these performance factors, including the F1-score and IoU values.

The *MRF-based* [74] method is largely confused if the registration errors between the compared point clouds are significantly greater than the used voxel size. Such situations result in large numbers of falsely detected "change pixels", which fact yields on average very low precision result (0.44), although due to several accidental matches, the recall rate might be relatively high (0.88).

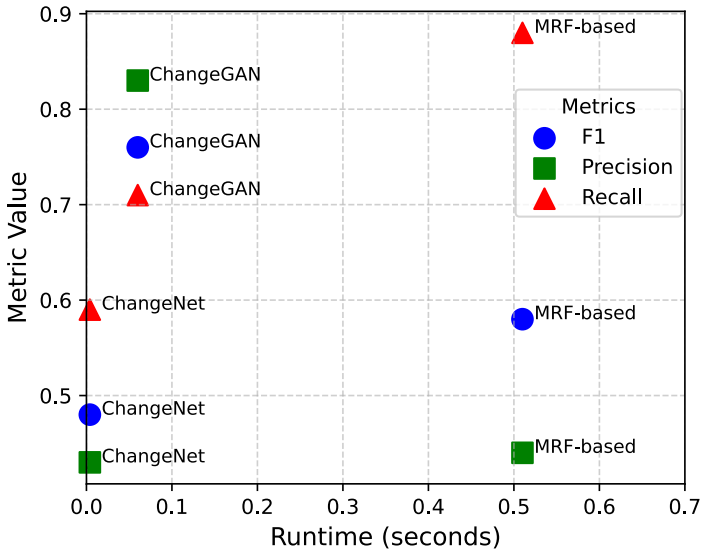


Figure 2.6. Execution time and performance of the proposed *ChangeGAN* method against the *ChangeNet* [62] and the *MRF*-based reference approach [74]

The measured low computational cost means a second strength of the proposed *ChangeGAN* approach, especially versus the *MRF* model, whose execution time is longer by one order of magnitude. Although *ChangeNet* is even

faster than *ChangeGAN*, its performance is significantly weaker compared to the other two methods. Since the adversarial training strategy has a regularization effect [81], and the STN layer can handle coarse registration errors, the proposed *ChangeGAN* model can achieve better generalization ability, and it outperforms the reference models on the independent test set.

2.3.3 Qualitative results

For qualitative analysis, we back-projected the 2D binary change masks to the corresponding 3D point clouds and visually inspected the quality of the proposed change detection approach. During the investigations, we have observed similarly efficient performance for the remaining, originally unregistered point cloud pairs of the *Change3D* dataset, to the point cloud set with simulated registration errors which participated in the quantitative tests of Section 2.3.2.

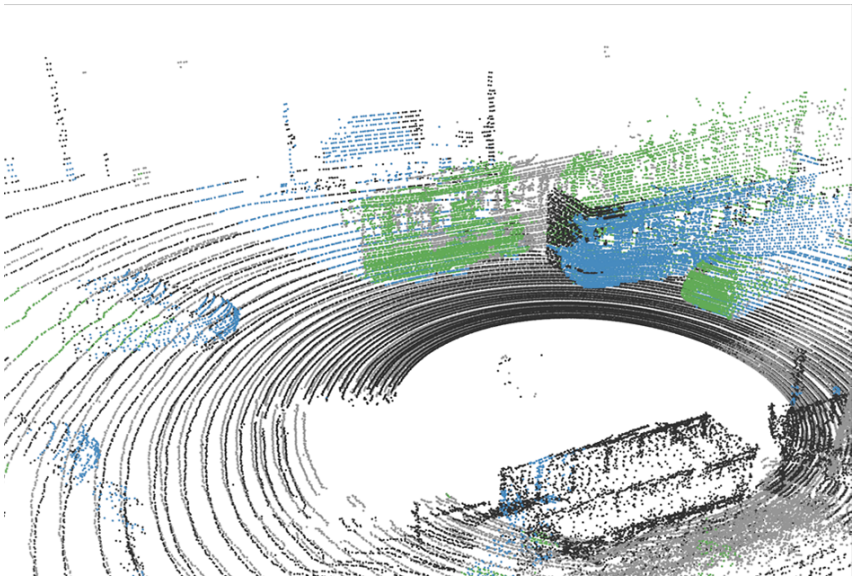
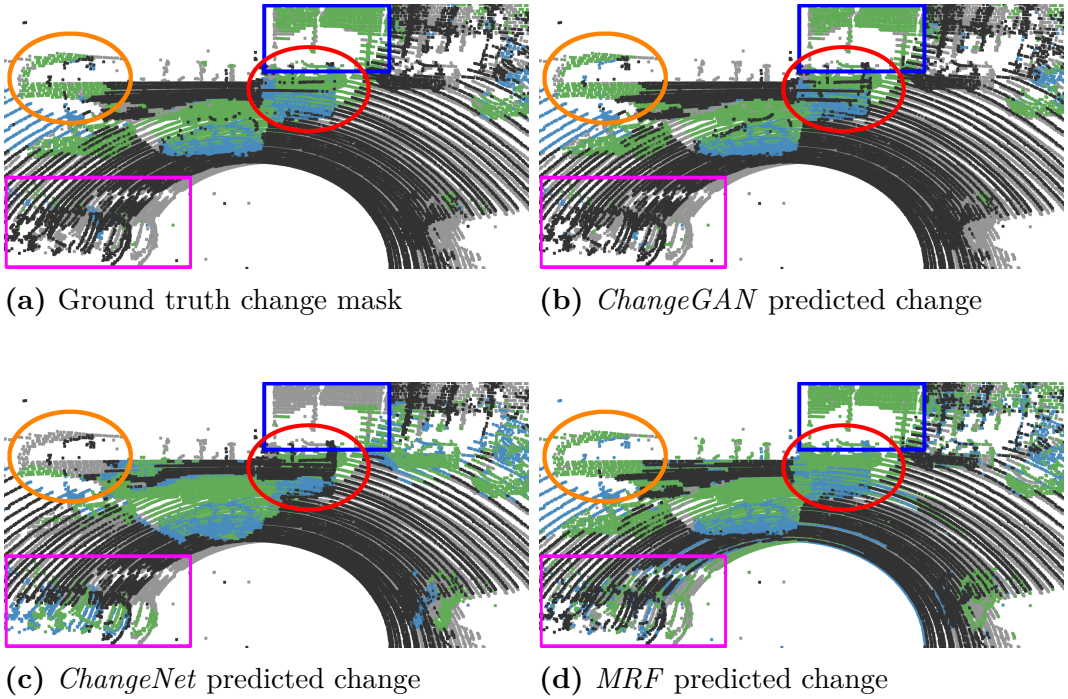


Figure 2.7. Changes detected by *ChangeGAN* for a coarsely registered point cloud pair. Blue and green points represent the changes in the first and second point clouds.

Figure 2.7 contains a busy road scenario, where different moving vehicles appear in the two point clouds. As shown, moving objects, both from the first (blue color) and second (green) frames, are accurately detected despite the large global registration errors between the point clouds (highlighted by a red ellipse). Let us also observe that a change caused by a moving object



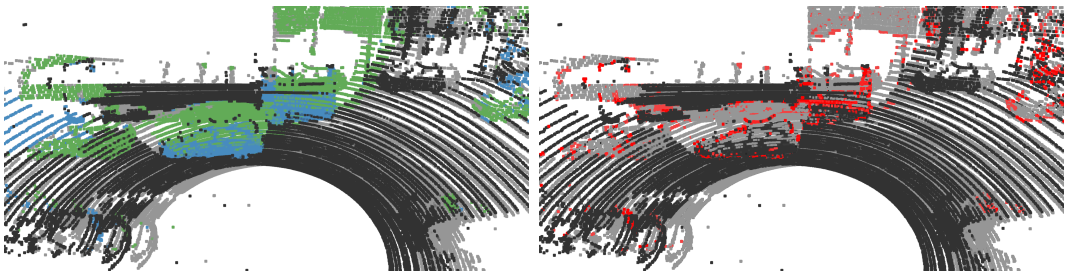
(a) Ground truth change mask **(b)** *ChangeGAN* predicted change

(c) *ChangeNet* predicted change **(d)** *MRF* predicted change

Figure 2.8. Comparative results of the ground truth and the predicted changes by *ChangeGAN* and the reference techniques. Green and blue points mark changed regions in $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Orange and red ellipses mark the detected front and back part of a bus traveling in the upper lane, meanwhile occluded by other cars. The blue square shows a building facade segment, which was occluded in $\mathcal{P}_2$. The magenta boxes highlight false positive changes of the reference methods confused by inaccurate registration.

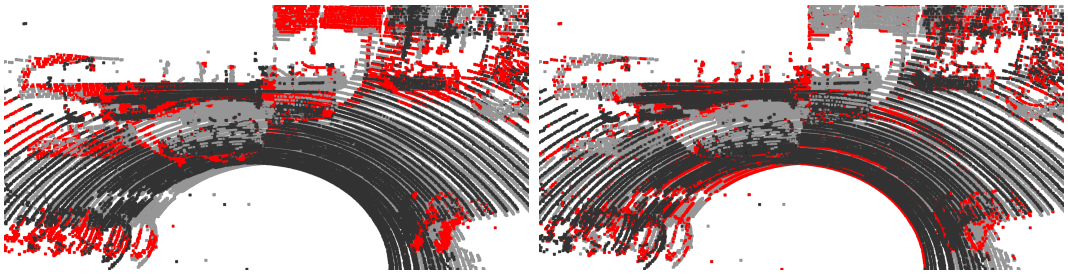
in each frame also implies a changed area in the other frame in its *shadow region*, which does not contain reflections due to occlusion. This phenomenon is a consequence of our change definitions, however, the shadow changes can be filtered out by geometric constraints, if they are not needed for a given application.

Figure 2.8 displays another traffic situation, where the output of the proposed *ChangeGAN* technique can be compared to the manually verified GT and the two reference methods in the 3D point cloud domain. As shown, our results accurately reflect our change concept defined in the paper, while the reference techniques cause multiple missing or false positive change regions. Since a bus traveling in the upper lane was partially occluded by other cars, only its frontal and rear parts could be detected as changes. However, the *ChangeNet* model missed detecting its frontal region and a partially occluded



(a) Change annotation of a point cloud pair

(b) *ChangeGAN* prediction errors



(c) *ChangeNet* prediction errors

(d) *MRF* prediction errors

Figure 2.9. Prediction errors samples of the *ChangeGAN* and the reference methods

facade segment. In addition, both reference methods detected false changes in the bottom left corner of the image, which were caused by the inaccurate registration. Figure 2.9 shows the error of *ChangeGAN* and the reference methods on the same scene as Figure 2.8. It can be seen that *ChangeGAN* has significantly fewer errors than the *ChangeNet* or the *MRF*.

Finally, we note that our method has also successfully performed[2] for frame pairs from the KITTI dataset [82], which were completely independent of our training process.

2.3.4 Robustness analysis

To evaluate the performance dependency of the discussed methods on the translation and orientation differences between the compared point clouds, we generated two specific sample subsets within the new *Change3D* dataset. This experiment was based on 500 (originally registered) point cloud pairs, selected

[2]Sample videos are available here: `https://users.itk.ppke.hu/~kovlo/videos/`

from the 2000 test sample pairs of the dataset.

For translation-dependency analysis, we used an offset domain of $[0.1, 1.0]$ meter, which was discretized using 10 equally spaced bins. For test set generation, we iterated through all the 500 point cloud pairs: For every sample, we chose for each translation bin $0.1 \leq t_i \leq 1.0$ $(i = 1 \dots 10)$ a random rotation value $-10° \leq \alpha_i \leq 10°$, and transformed the second cloud $\mathcal{P}_2$ using (t_i, α_i).

With this process, for each offset bin, we generated 500 coarsely registered point cloud pairs with known registration errors. In total, 10 subsets were created for the 10 offset bins, each one containing 500 samples.

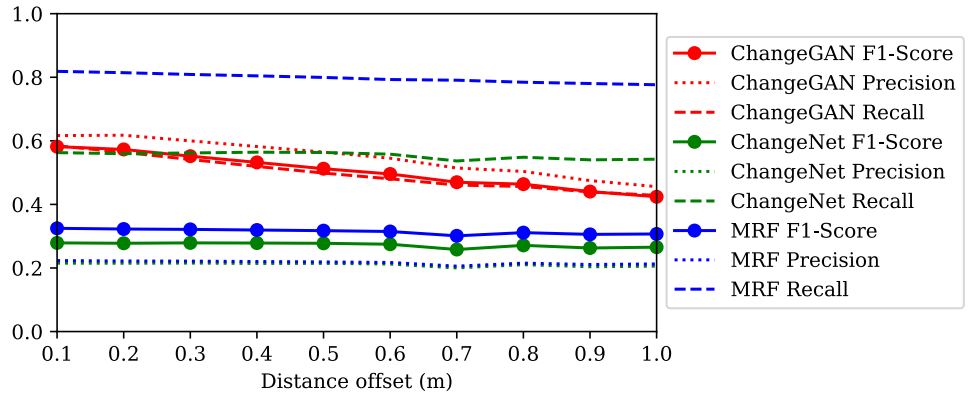


Figure 2.10. Translation dependency of the compared methods' performance (F1-score, Precision, Recall)

Next, we ran our proposed method and the reference techniques on this new set, and we calculated the mean F1-score [75, 83] value for each translation bin i, among samples having an offset parameter t_i. Figure 2.10 displays with solid lines the average F1-scores in a function of various t_i values. The proposed method shows a graceful degradation by increased offsets, and even for a $t_i = 1$ meter offset, the quality of change detection is significantly better than the nearly constant low values provided by the reference approaches.

For measuring the rotation-dependency of the models, we have performed a similar experiment: here we discretized the $-10° \leq \alpha_i \leq 10°$ rotation domain with 10 bins, and within each bin, we generated 500 sample pairs, with random translation values. Finally, we averaged the measured F1-scores within each rotation bin [75, 83]. Results shown in Figure 2.11 with solid lines confirm again the superiority of the proposed method against the tested references.

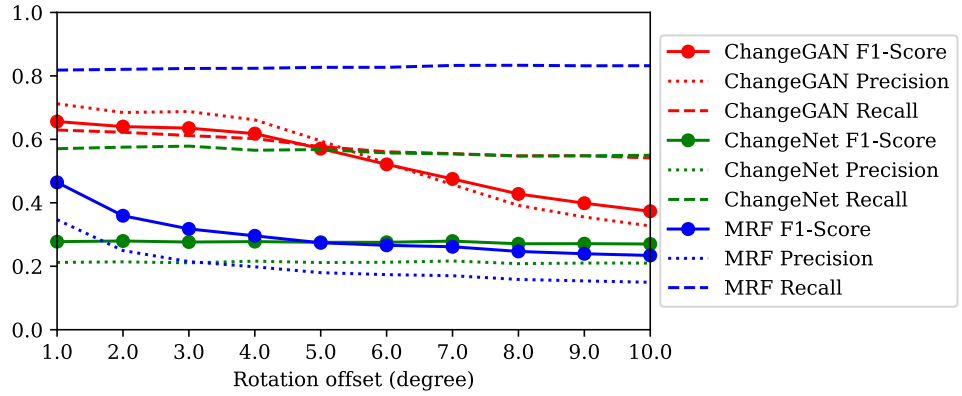


Figure 2.11. Translation rotation dependency of the compared methods' performance (F1-score, Precision, Recall)

Chapter 3

Real-time foreground segmentation in NRCS Lidar point clouds

This chapter presents a new point-level foreground-background separation method by processing measurement sequences of an NRCS Lidar sensor, which is used as a surveillance sensor, mounted in a fixed position.

3.1 Introduction

Accurate and real-time foreground-background separation is a critical task in surveillance applications. As alternative solutions of conventional optical video cameras, range sensors offer significant advantages for scene analysis, since direct geometrical information is provided by them [84]. The use of infrared light-based Time-of-Flight (ToF) cameras [85] or laser-based Light Detection and Ranging (Lidar) sensors [86] enables recording directly measured range images, where we can avoid artifacts of the stereo vision-based depth map calculation.

From the point of view of data analysis, ToF cameras record depth image sequences over a regular 2D pixel lattice, where established image processing approaches, such as morphological filters or Markov Random Fields (MRFs) can be adopted for smooth and observation-consistent segmentation and recognition [87]. However, such cameras can only be reliably used indoors, due to the limitations of current infra-based sensing technologies, and they may have

a narrow FoV, which fact can be a drawback for surveillance and monitoring applications.

A stereo camera system estimates the environment in 3D by detecting and matching features in a pair of corresponding images, followed by triangulation to compute the 3D positions of keypoints. The process involves several critical steps. Accurate depth estimation requires precise camera calibration. Feature extraction and matching demand high-resolution images and feature-rich areas, which makes the calculations computationally intensive. Dynamic scenes with moving objects and varying illumination present additional challenges for stereo-based depth estimation. Furthermore, the overlapping FoV of the cameras restricts the extent of the 3D reconstruction. The baseline, or the distance between the cameras, determines the disparity: the pixel difference between a keypoint in the left and right images. The baseline must be optimized for the task: a smaller baseline is suitable for measuring closer objects (e.g., robotics), while a larger baseline is necessary for distant object measurements (e.g., autonomous vehicles).

By extracting accurate 2D or 3D object silhouettes, one can obtain various sorts of valuable scene information which can be directly exploited in people detection, tracking, biometric recognition, or activity analysis.

Prior existing Lidar-based surveillance solutions utilize mainly RMB Lidar sensors [88], introduced in Section 1.3.1. These Lidars provide high frame rate *point cloud videos*, enabling dynamic event analysis in the 3D space. On the other hand, the measurements have low spatial density, which quickly decreases as a function of the distance from the sensor, and the point clouds may exhibit ring patterns typical of the sensor characteristics.

While previous works have shown [17], that RMB Lidar measurements can be used for certain dynamic scene analysis tasks, such as object separation, tracking, and even gait-based biometric person recognition and activity analysis, the constant and low vertical resolution of the measurements that is physically constrained by the number of vertically fixed laser emitters and receivers (typically 32 or 64), means a clear limitation by applying them in a static sensor configuration. Moreover, the RMB Lidar sensors are generally expensive, thus their application is not widespread for surveillance tasks.

An alternative to the RMB Lidars is a new type of Lidar sensor introduced in Section 1.3.2, which implements the unique NRCS technique.

In the proposed approach we generate and maintain a very high-resolution

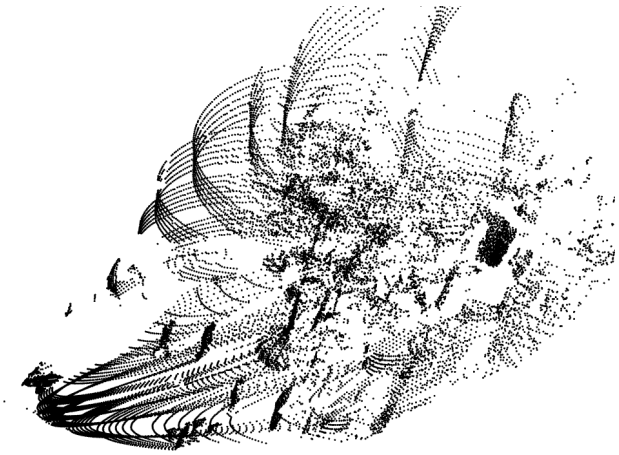


Figure 3.1. Point cloud recording from the Courtyard dataset, recorded using the Livox Avia NRCS Lidar sensor

background model of the scene fully automatically in the range image domain of the sensor's FoV, while for enabling real-time analysis of dynamic objects we use low integration time to extract the consecutive time frames. The measured points are matched to the high-resolution background model components in the closest matching positions. This process ensures that the spatial accuracy of the native measurements is largely maintained, instead of applying a rough spatial down scaling technique. As a result, we can obtain sparse, but geometrically accurate point cloud segments representing the moving objects, which can be used in higher-level scene analysis steps of surveillance systems.

3.2 Proposed Method

The goal of the proposed method is to separate foreground and background regions in Lidar frames extracted with a 100 ms integration window from a measurement sequence of a static NRCS Lidar sensor.

Formally, in a given time frame t, we assign to each point $p \in \mathcal{P}$ a label $\Lambda(p) \in \{\text{fg}, \text{bg}\}$ corresponding to the foreground (fg) or background (bg), respectively.

The sensor's non-repetitive circular scanning approach implies a critical challenge to be handled: the moving laser beams cannot densely cover the whole FoV within the considered data collection window, which results in several sparse/empty regions in the individual Lidar frames. Moreover, we can observe strongly inhomogeneous point density, as shown in Figure 3.1.

Surveillance applications demand real-time solutions. To avoid computationally expensive algorithmic steps in the 3D point cloud domain, and to enable the efficient and robust utilization of the sparse data, we map the problem to the 2D range image domain, by transforming the 3D Euclidean point coordinates into a polar representation.

The proposed method consists of three main steps, as follows:

1. Incoming Lidar measurements are collected within a 100 ms time window for composing the next point cloud frame of the sequence. Thereafter, the distances of the 3D measurement points from the sensor are assigned to corresponding pixels in a high-resolution range image.
2. A local background model is assigned and maintained for each pixel of the range image lattice, following the Mixture of Gaussians (MoG) approach [89] applied for the range values. Considering the sparseness of the captured point clouds, in a given time frame, only the MoG background model components of range image pixels linked to the actual measurement points are updated. The incoming measurement points are classified either as foreground or as background, based on matching the measured range values to the local MoG distributions.
3. False foreground points in dynamic background regions (e.g., by moving vegetation) are filtered out by using an extension of the original MoG approach. To ensure compact shapes for the extracted moving objects, fast spatial filters are adopted for segmentation refinement.

3.2.1 Range image formation

The point cloud's representation is transformed from the 3D Descartes to a spherical polar coordinate system. A 2D pixel lattice (S) is generated by quantizing the horizontal and vertical FoV-s, and each 3D point's distance from the sensor is stored in a pixel determined by the corresponding azimuth and elevation values. The polar direction and azimuth angles correspond to the horizontal and vertical pixel coordinates, and the distance is encoded in the corresponding pixel's 'gray' value. As a result, the upcoming steps of the proposed foreground segmentation method can be developed in the 2D range image domain (I).

Using a narrow timing window, the range image of a certain frame contains several pixels with undefined range values as a consequence of the NRCS

scanning technology. The number of undefined pixels depends on both the timing window and the predefined size of the range image. In our experiments, exploiting the precision parameters of the used Livox Avia sensor, its FoV is mapped onto a 600×660 sized pixel lattice (S), resulting in a 8.5px/° spatial resolution. We also have to consider that the density of the recorded valid range values is decreasing towards the peripheral regions of the range image due to the applied scanning technique: the scanning pattern crosses the optical center of the sensor more frequently than covering the regions of the FoV's perimeter. The sparseness of the range image makes it significantly more difficult to perform, e.g., object-based foreground-background segmentation.

3.2.2 Background model

The scene's estimated background is represented in the 2D range image domain defined in Section 3.2.1.

Our background modeling technique is based on [87], which extends the MoG approach [89] to the range image domain. A fitness term $f_{\text{bg}}(p)$ is assigned to each point $p \in \mathcal{P}$ of the cloud, which measures the quality of the hypothesis that p is a background point. As explained in Section 3.2.1, we map the points to the range image pixels, where we use the predefined and fixed size 2D pixel lattice. For each $s \in S^{\text{bg}}$, we calculate an MoG approximation of the $d(p)$ distance histogram of p points being projected to s. Following the approach of [86], we use 5 components with weight w_s^i, mean μ_s^i, and standard deviation σ_s^i parameters, $i = 1 \dots 5$. Thereafter, the weights are sorted in decreasing order, and the minimal k_s number is determined, which satisfies

$$\sum_{i=1}^{k_s} w_s^i > T_{\text{bg}}, \tag{3.1}$$

where we used $T_{\text{bg}} = 0.89$ based on the work in [90].

We consider the components with the k_s largest weights as the background components. Then, denoting by $\eta()$ a Gaussian density function, and by Π^{bg} the projection transform onto S^{bg}, the $f_{\text{bg}}(p)$ background evidence term is obtained as:

$$f_{\text{bg}}(p) = \sum_{i=1}^{k_s} w_s^i \cdot \eta\left(d(p), \mu_s^i, \sigma_s^i\right), \text{ where } s = \Pi^{\text{bg}}(p). \tag{3.2}$$

The Gaussian mixture parameters are calculated and refreshed, as follows. 3D point's depth, $d(p)_t$, is compared with the existing 5 Gaussian distributions until a match is identified. If none of the 5 distributions match the current point's depth value at the proper pixel, the least probable Gaussian component is replaced with a new distribution with the current depth value as its mean, a high variance, and a low prior weight.

The prior weights at time t are calculated as follows:

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}), \tag{3.3}$$

where α is the learning rate and $M_{k,t} = 1$ for the model which matched as a background and $M_{k,t} = 0$ for the remaining models. After this approximation, the weights are renormalized. [89]

The μ and σ parameters for the unmatched components are not changed. The parameters of the matching component are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho \cdot d(p)_t \tag{3.4}$$

$$\sigma^2 = (1 - \rho)\sigma^2_{t-1} + \rho(d(p)_t - \mu_t)^T(d(p)_t - \mu_t) \tag{3.5}$$

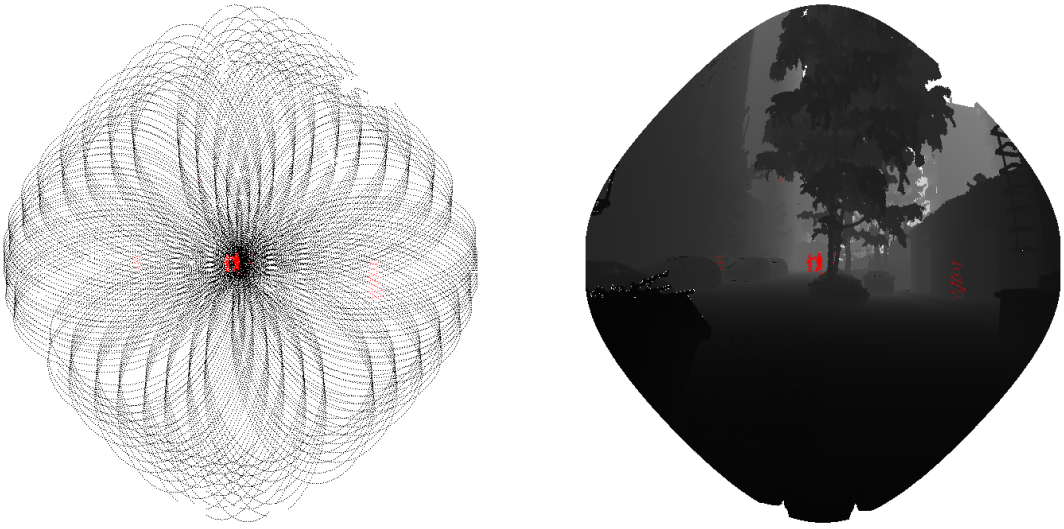
where

$$\rho = \alpha\eta(d(p)_t|\mu_k, \sigma_k) \tag{3.6}$$

is the learning factor for adapting the Gaussian component parameters [89]. By thresholding $f_{\text{bg}}(p)$, we can get a dense foreground/background labeling of the point cloud [86, 89].

As the incoming points from the consecutive sparse NRCS Lidar frames are processed one after another, each pixel of the high-resolution background range image lattice becomes covered by valid range measurement several times, thus the associated MoG distribution can learn the appropriate parameters. The used background model is adaptive; thus it automatically updates itself when the background scene changes: for example, a static object is relocated, or a parking car departs. Besides updating the high-resolution background map, the method also classifies the incoming frame's points, whether they belong to the foreground or the background classes.

Although the MoG technique is regarded as a highly robust approach for optical video processing, as demonstrated in Figure 3.2b, the above-described foreground-background classification process is notably noisy for NRCS Lidar-



(a) Detected foreground (red) in a single time frame of the NRCS Lidar image sequence

(b) Detected foreground region (red) displayed over the generated high-resolution background range image

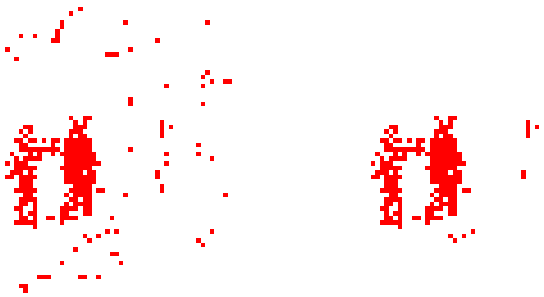
Figure 3.2. Foreground detection results (red) in the *City Center* scene, displayed in 3D point cloud representation.

based range image sequences, especially in scenarios recorded in large outdoor environments. Various sources of noise are present, including oscillations and small movements in the background (tree leaves, branches), whose regions are often classified falsely as foreground. Although by fine-tuning the parameters of the algorithm, the negative effects of oscillations can be decreased, usually these artifacts cannot be eliminated in acceptable quality. As a consequence, to reliably eliminate the oscillation artifacts, further noise filtering steps are needed, as described in the next subsection.

As for the speed of adaption, the initialization period of the method in a new scene needs about $50 - 100$ time frames, to obtain an efficient initial background range value for each pixel of the high-resolution background map. Additional $100 - 300$ frames are required to let the background model's MoG distribution parameters converge, exploiting the repetitive sensor measurements from the observed background scene.

3.2.3 Foreground noise filtering

In this section, we propose filtering steps applied to the MoG-based segmentation output, to obtain a smoothly uniform and observation-consistent



(a) Foreground detection without the spatial filtering adjustment

(b) Foreground detection result with the proposed method

Figure 3.3. Foreground detection results in the central area of Figure 3.2a, displayed in range image representation.

segmentation of the point cloud sequence recorded by the NRCS Lidar.

Vibrations of objects (e.g., tree leaves, branches) in the background regions are usually composed of relatively small, but frequent movements. The vibrating objects' edge points often oscillate between neighboring pixels of the range image lattice, causing challenges for the original MoG approach.

As the background oscillations are often quasi-periodic, by observing the pixels of these oscillating areas, the two Gaussian components with the highest weight can be used. Thus, based on the thresholding rule of Equation (3.1), these regions are marked as background.

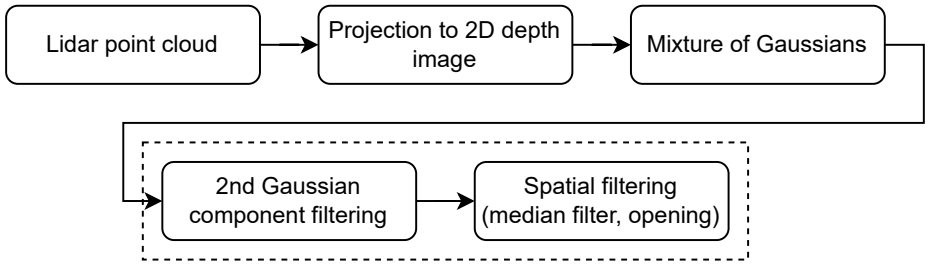


Figure 3.4. Steps of the proposed method for foreground-background separation in NRCS Lidar point clouds

However, there are regions in the observed scene, where real foreground objects (persons, cars, etc.) are frequently observable. The general distance of a true foreground point is stored in the second component of the background model, which still has to be detected as foreground. To avoid false filtering of the real foreground points, we apply an additional condition: if the deviation

of the Gaussian component with the highest weight is saliently small (which indicates a compact background surface), it is added to the background model.

Since the above-described MoG-based method works independently on each pixel of the range image, noise may result in many standalone false foreground pixels surrounded by background regions. These artifacts can be removed by applying spatial filtering in the 2D range image domain using median filter and morphological opening. Figure 3.4 shows the steps of the proposed method for foreground-background separation in NRCS Lidar point clouds.

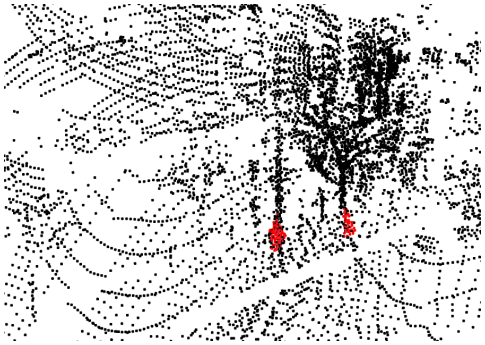
As a result, the number of false positive foreground pixels can be significantly decreased as shown in Figure 3.3, and we can obtain compact connected object shapes, as shown in Figure 3.5.

3.3 Dataset collection

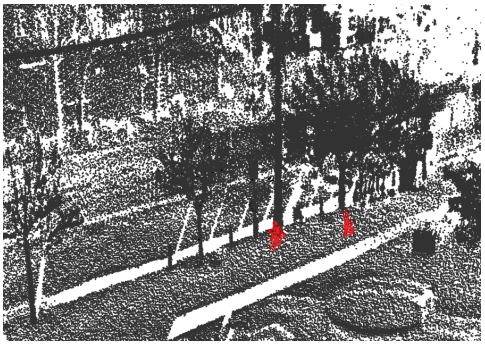
For the development and evaluation of the proposed method, two measurement sequences were recorded by a tripod-mounted Livox Avia sensor in two different, outdoor locations.

In the *Courtyard* scene, five people were walking in a narrow inner courtyard surrounded by large building facades, while canopies of trees and bushes were waving in the background due to the wind. The observed courtyard is 15 m wide, and its width is parallel to the NRCS Lidar's front plane, while the length of the observed area is 40 m. This measurement setup was suitable for the 70° horizontal FoV of the Livox sensor. The sensor was placed horizontally, looking towards the horizon. Five to seven walking pedestrians formed the foreground regions of the scene, while the background consisted of parking cars, walls, trees, ground areas, etc. This setup utilized the benefits of the NRCS Livox sensor, as the foreground regions appeared close to the center of the sensor's FoV, resulting in better spatial resolution than in the peripheral FoV regions.

The *City Center* sequence was recorded in a busy scene in downtown Budapest, containing several moving vehicles and pedestrians. The selected square and junction were observed from a higher location, where the sensor was placed looking towards the ground. The foreground regions of this scene include various types of moving objects, including pedestrians, cars, trams, cyclists, etc. In this experiment, the observed area was in an open space, thus the observed distances were also limited by the sensor's reflection detection ca-



(a) Detected foreground (red) in a single time frame of the NRCS Lidar sequence



(b) Detected foreground regions (red) displayed over the generated high-resolution background point cloud

Figure 3.5. Foreground detection results (red) in the *City Center* scene, displayed in 3D point cloud representation.

pabilities, not only by the static field objects such as buildings/vegetation. As the observed area was farther from the sensor than in the *Courtyard* scene, the *City Center* sequence has sparser data. Because of the sparser measurements, we observed here a slightly longer initialization period of the high-resolution background model.

3.4 Results and discussion

The method was tested and evaluated using the *Courtyard* and *City Center* Livox Lidar measurements (see Section 3.3).

A demonstrating example for foreground classification on a sparse sample frame from the *Courtyard* sequence and the generated dense background model are displayed in Figure 3.2 in the range image representation.

A sample result from the *City Center* dataset is displayed in Figure 3.5 in point cloud representation. Here both the foreground and background objects were at larger distances, resulting in even sparser Lidar point cloud frames.

3.4.1 Quantitative Results

Numerical evaluation of the algorithm's performance was conducted by comparing the detection results to GT segmentation, which was manually generated for selected keyframes of both the *Courtyard* and the *City Center* Lidar measurement sequences. More specifically, we considered 25 s long

	MoG only method	MoG + Filtering
Precision	0.67	**0.72**
Recall	0.80	**0.83**
F1 Score	0.72	**0.77**
IoU	0.57	**0.62**

Table 3.1. Result of the quantitative evaluation of the method on the annotated Courtyard and City Center datasets

measurement segments in both scenes and manually annotated every 5th point cloud (i.e., the annotation frame rate was 2 frames per second (FPS)) via a 3D annotation tool, separating the foreground and background regions.

The quantitative performance analysis was performed by the comparison of each point's label after the assignment of the 3D corresponding points of the GT and the output clouds. To measure the similarity between the binary annotation of the GT point cloud, and the binary classification of each point in the result point cloud, the mean F1-score and Intersection over Union (IoU) were calculated alongside precision and recall. The used metrics' definition follows the standard binary classification metrics [91].

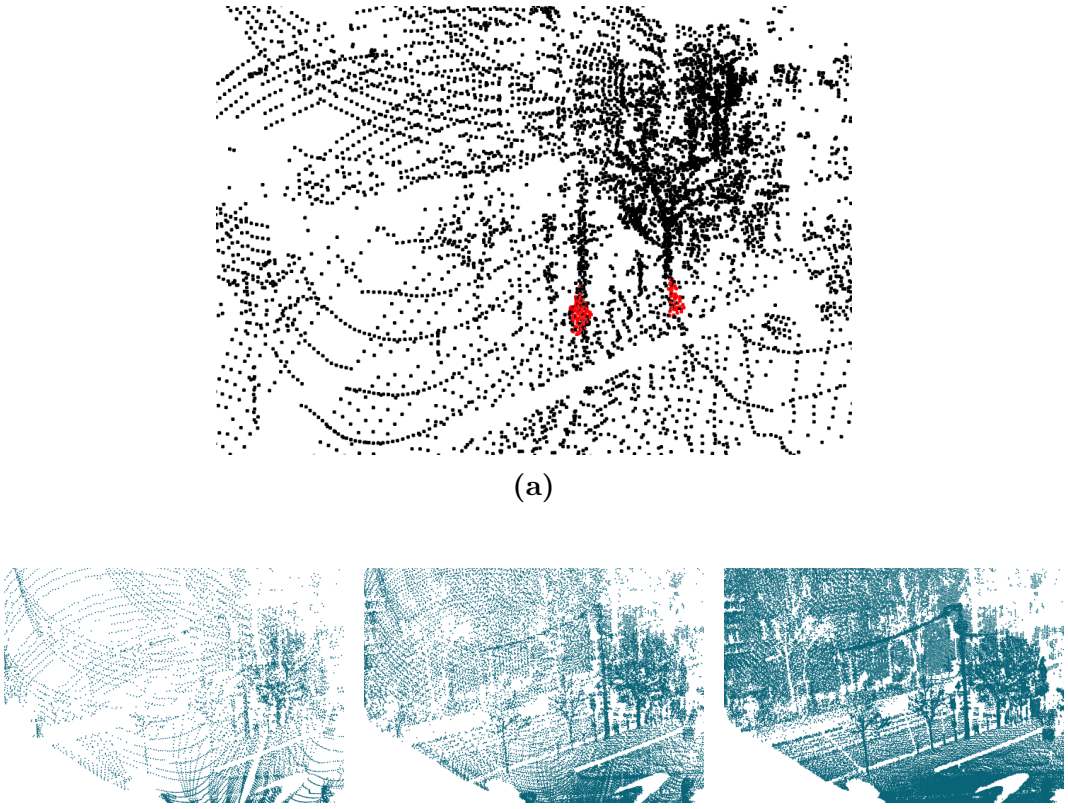
The results of the quantitative evaluation are listed in Table 3.1. These initial results are satisfying considering our low-level classification approach, which observation can also be confirmed by qualitative experiments. The average running speed of the method was 80 ms for each point cloud on a PC with an i7-7500UK CPU @2.7 GHz with 16 GB RAM.

3.4.2 Qualitative Results

For qualitative analysis, we constructed first a dense 3D point cloud from the 2D high-resolution background model.

Then, the moving objects detected in the consecutive Lidar frames (Figure 3.5a) can be displayed with the background's dense point cloud in the same coordinate system, which can provide a useful visualization effect for the operators of a surveillance system (Figure 3.5b).

We demonstrate the development phases of the dense background model by the adopted MoG approach in Figure 3.6. As time elapses, the sensor's non-repetitive scanning pattern covers more and more regions of its FoV, resulting

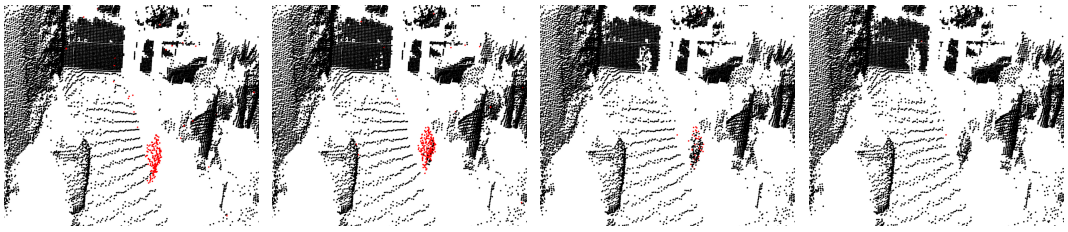


(a) **(b)** **(c)** **(d)**

Figure 3.6. Evolution of the high-resolution background model in the City Center dataset

in a step-by-step evolution of the background point cloud. By the end of the initialization process, all undefined regions disappear, and all pixels in the FoV receive a valid range value. Once the high-resolution background model is built, it is updated continuously during the surveillance process.

During the experiments, we also tested the adaptivity of the background model, by investigating the transition of different scene regions from foreground to background classes and vice versa. Figure 3.7 displays consecutive point cloud frames, where a walking pedestrian stopped for a certain time, and its point cloud was built into the background model. It is also noteworthy, that when the pedestrian resumed walking, a rapid "revival" could be observed, as the range values were temporarily stored in the second-strongest Gaussian components of the concerning pixels.



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

Figure 3.7. Transition of a region from the foreground (red) to the background (black), while a pedestrian stopped and stood in place for $5s$.

Chapter 4

Human pose estimation using only NRCS Lidar data

4.1 Introduction

The main task of this pose estimation is to localize the anatomical keypoints of the human body in three-dimensional space.

In this chapter, we demonstrate the efficiency of using the Livox Avia Lidar sensor introduced in Section 1.3.2 for the human pose estimation described in Section 1.2. We propose a visual transformer-based [92] neural network to detect and fit human skeleton models solely based on the NRCS Lidar data.

4.1.1 Related works

For 3D human pose estimation [24, 93] use semi-supervised learning approaches, where the 2D annotations are lifted to the 3D space and the methods use the fusion of camera images and Lidar point clouds.

Aside from camera-based methods, the human pose estimation task has also been addressed by processing Lidar measurements. The Lidar-based human pose estimation faces several challenges, including sparse data representation, limited FoV and limited spatial resolution. The sparseness of the point clouds emerges from the limited number of laser beams in the sensors. The Lidar's limited FoV is caused by the placement of the laser array and the scanning method. Upon proposing a Lidar-based solution these issues must be addressed.

In [94] the authors proposed a method for 3D human pose and shape estimation from a point cloud sequence. Although that method can regress the 3D mesh of a human body, it does not make predictions about the underlying human skeleton. Similarly, *LiveHPS* proposed in [95] estimates the human pose and shape using a point cloud sequence, recorded with an RMB Lidar. Although this method extracts point-wise features and predicts the human body joint positions, it uses the IMU sensor's data alongside the Lidar point clouds for the pose detection, similarly to the *LIP* method described in [96]. Dense depth images can be used to estimate human pose, as shown in [97], using a deep graph convolutional neural network-based network [98]. The input of this method is a point cloud, derived from the 2D depth images recorded with a depth camera. That method relies on the denseness of the point cloud, which does not make it suitable to process sparse point clouds recorded with an NRCS Lidar sensor.

The *LPFormer* method [99] works on point clouds recorded with RMB Lidars, and it is developed and tested on the Waymo Open Dataset [100]. However, that technique exploits particular measurement modalities apart from the 3D point coordinates, namely the intensity, elongation, and the timestamp associated with each Lidar point, which requirements give limitations for using the LPFormer method with different Lidar types, including the NRCS Lidar sensors.

Vision transformers made significant progress and successes recently in several computer vision tasks [92, 101], such as object detection [102], image generation [103–105], but also in pose estimation [33, 99, 106]. A notable approach for camera-based human pose estimation is ViTPose [33], a vision transformer-based human pose estimator. The method yields state-of-the-art results while running in real-time on camera images. Given the attractive properties of ViTPose [33] and the fact that transformers [101] handle sparse data better than the mostly convolution-based skeleton estimation methods [107–109], we propose here a modified ViTPose architecture to process the sparse Lidar input data for 3D human pose estimation, expecting that the transformer-based [101] approach can handle the sparse Lidar input data more efficiently than the mostly convolution-based skeleton detection methods [107–109].

4.2 Proposed Method

The goal of the proposed method [1], [3] is to detect human poses (introduced in Section 1.2) in Lidar frames, recorded by an NRCS Lidar sensor.

The sensor's non-repetitive circular scanning pattern presents a significant challenge: The scanning laser beams are unable to densely cover the entire FoV of the sensor within a data collection window. This limitation leads to numerous sparse and even empty regions within the individual Lidar frames, particularly near the edges of the sensor's FoV. Additionally, there is a noticeable inhomogeneous point density, as illustrated in Figure 1.7.

The human pose estimation task can be applied in surveillance applications, which demand real-time solutions. To address this need, our approach involves transforming the representation of the NRCS Lidar point cloud from 3D Cartesian coordinates to a spherical polar coordinate system, similarly to our previous works described in Chapters 2 and 3 and in publications [1] and [5]. We generate a 2D pixel grid by discretizing the horizontal and vertical FoV-s, where each 3D point's distance from the sensor is mapped to a pixel determined by corresponding azimuth and elevation values. The polar direction and azimuth angles correspond to the horizontal and vertical pixel coordinates, while the distance is encoded as the intensity value of the respective pixel. This process allows the subsequent steps of our proposed Lidar-only 3D human pose estimation method to be developed within the domain of 2D range images.

Depending on the timing window of data collection, as illustrated in Figure 1.7, the range image of a specific Lidar frame may contain numerous pixels with undefined range values due to the NRCS scanning pattern. The number of these undefined pixels depends on both the measurement integration time and the predefined dimensions of the range image. For this method we used the range image representation of the Lidar point cloud, as described in Section 3.2.1.

The proposed method is based on the state-of-the-art ViTPose [33] human pose estimation method, working on camera images, based on a Vision Transformer (ViT) architecture [92], which was trained on the COCO dataset [110].

4.2.1 ViTPose

ViTPose is a deep learning-based method for human skeleton estimation, that can achieve real-time performance and outstanding estimation accuracy [33]. ViTPose works on images containing a single person with a tight crop. It has three main parts: network backbone, network head, and joint position reconstruction.

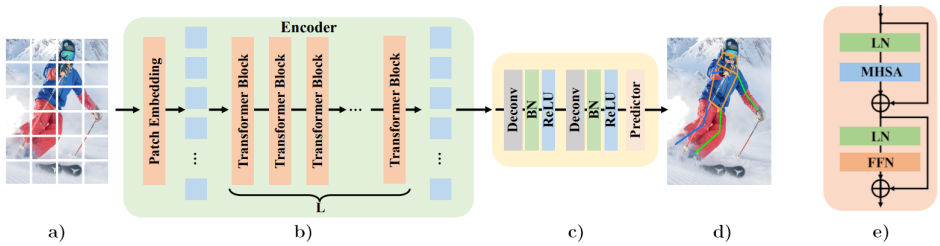


Figure 4.1. The structure of the ViTPose method [33].
a) Input image, split into patches.
b) transformer-encoder
c) classical human pose estimation decoder
d) output pose overlaid on the input image
e) transformer block from the encoder

The network's backbone is a plain and non-hierarchical vision transformer, as shown in Figure 4.1. Its input is a camera image, cropped around the human subject. The backbone embeds the input data into tokens using patch embedding and downsampling. These embedded tokens are fed to several transformer layers. Each of these layers consists of a *Multi-Head Self-Attention (MHSA)* layer and a *Feed-Forward Network (FFN)*. The output of the transformer layer is processed by a decoder. ViTPose's head is the decoder network, which processes the transformer blocks' output in the feature space. It employs direct upsampling with bilinear interpolation, which is followed by a *Rectified Linear Unit (ReLU)* and a 3×3 convolution. The output of the network head is a set of heatmaps, one heatmap for each joint in a down-scaled and uniformed feature space. The heatmap encodes the likelihood of the presence of a joint at each pixel position. Thus, the maxima of each heatmap correspond to the estimated joint locations. The third part of the method retrieves the final keypoint predictions from the heatmaps predicted by the network head and transforms the keypoint locations back to the original input image domain.

4.2.2 LidPose

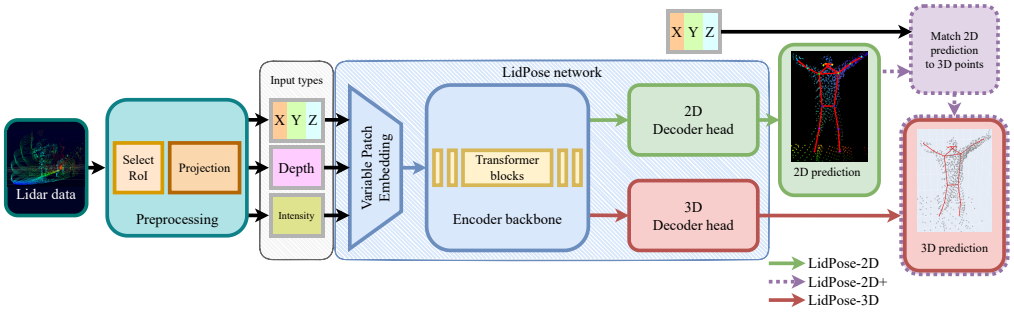


Figure 4.2. *LidPose* end-to-end solution:
Lidar data: full Lidar point cloud. Select ROI: selects the 3D points in the vicinity of the observed human. Projection stores the 3D point cloud in a 2D array. Input types: 3D XYZ coordinates (`XYZ`), Depth (`D`) and Intensity (`I`). *LidPose* network: Both *LidPose–2D* and *LidPose–3D* use our patch embedding module and the encoder backbone, visible in blue. *LidPose–2D* and *LidPose–3D* use the corresponding Decoder head and *LidPose–2D+* is calculated from the 2D prediction and the input point cloud.

The proposed *LidPose* method is an end-to-end solution, which solves the human detection and pose estimation task using only NRCS Lidar measurements, in a surveillance scenario, where the sensor is mounted in a fixed position. The *LidPose* method's workflow is shown in Figure 4.2.

First, the moving objects are separated from the static scene regions in the NRCS Lidar measurement sequence, by applying a foreground-background segmentation technique that is based on the MoG approach adopted in the range image domain, as described in Chapter 3 and in [5]. The incoming measurement points are then classified as either foreground or background by matching the measured range values to the local MoG distributions.

Second, the foreground point regions are segmented to separate individual moving objects, and the footprint positions of the detected pedestrian candidates are estimated. Here a 2D lattice is fitted to the ground plane, and the foreground regions are projected to the ground. At each cell in the ground lattice, the number of the projected foreground points is counted, which is used to extract each foot position, as described in [75]. The result of this step is a set of bounding boxes for the detected people, which can be represented both in the 3D space and in the 2D range image domain. As shown in [75], due to the exploitation of direct range measurements the separation of partially occluded

pedestrians is highly accurate, however in a large crowd the efficiency of the approach can be deteriorated.

In the next step, the NRCS Lidar point cloud and the range image are cropped with the determined bounding boxes. The cropped regions correspond to Lidar measurement segments containing points either from a person or from the ground under their feet.

To jointly represent the different available measurement modalities, we propose a new 2D data structure that can be derived from the raw Lidar measurements straightforwardly and can be efficiently used to train and test our proposed *LidPose* model. More specifically, we construct from the input point cloud a five-channel image over the Lidar sensor's 2D range image lattice, where two channels directly contain the depth and intensity values of the Lidar measurements, while the remaining three layers represent the `X,Y,Z` coordinates of the associated Lidar points in the 3D world coordinate system.

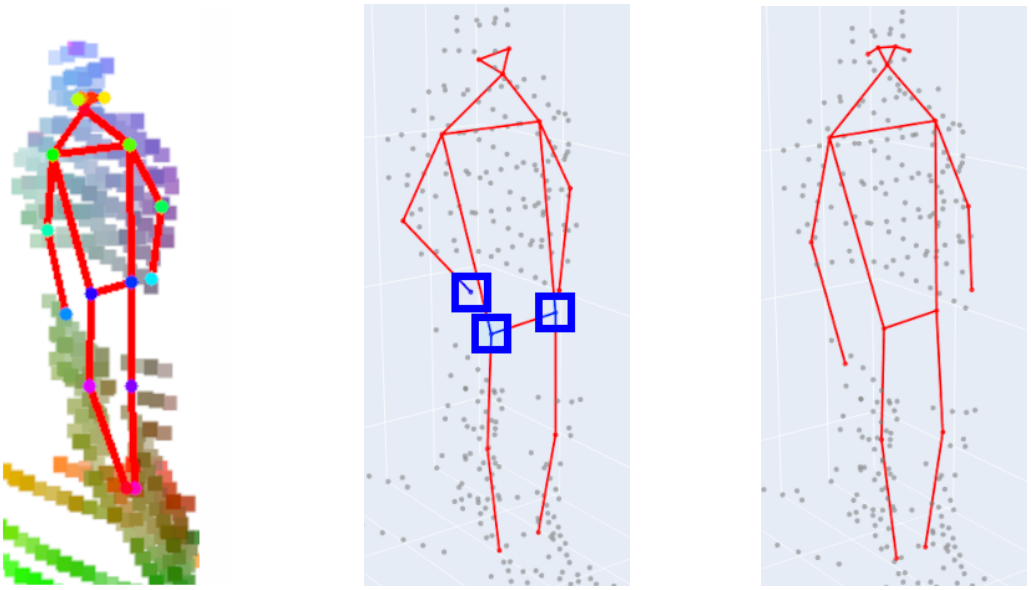
Note that in our model, the pose estimator part of the method is independent of the sensor placement. While in this paper we demonstrate the application purely in a static Lidar sensor setup, we should mention that with an appropriate segmentation method for a given scene, the *LidPose* pose estimation step could also be adapted to various - even moving - sensor configurations.

To comprehensively explore and analyze the potential of using NRCS Lidar data for the human pose estimation task, we introduce and evaluate three alternative model variants:

- *LidPose–2D* predicts the human poses in the 2D domain, i.e. it detects the projections of the joints (i.e. skeleton keypoints) onto the pixel lattice of the range images, as shown in Figure 4.3a. While this approach can lead to robust 2D pose detection, it does not predict the depth information of the joint positions.

- *LidPose–2D+* extends the result of the LidPose–2D prediction to 3D for those joints, where valid values exist in the range image representation of the Lidar point cloud, as shown in Figure 4.3b. This serves as the baseline of the 3D prediction, with a limitation that due to the sparsity of the Lidar range measurements, some joints will not be associated with valid depth values (marked by blue boxes in Figure 4.3b).

- *LidPose–3D* is the extended version of LidPose–2D+, where depth values



(a) (b) (c)

Figure 4.3. Predicted human poses of the *LidPose* variants, overlaid on the input data.
(a) LidPose–2D: 2D predicted skeleton (red) over the 2D Lidar point cloud representation (colored based on 3D coordinate value).
(b) LidPose–2D+: 2D predicted skeleton (red) is extended to the 3D space using the Lidar points (gray) where they are available. Points where Lidar measurement is not available are highlighted in blue.
(c) LidPose–3D: 3D predicted skeleton (red) over the Lidar point cloud (gray).

are estimated for all joints based on a training step. This approach predicts the 3D human poses in the world coordinate system from the sparse input Lidar point cloud, as shown in Figure 4.3c.

The ViTPose [33] network structure was used as a starting point in the research and development of the proposed *LidPose* methods' pose estimation networks. My main contributions to the proposed *LidPose* method:

- A new patch embedding implementation was applied to the network backbone to handle efficiently and dynamically the different input channel counts.
- The number of transformer blocks used in the *LidPose* backbone is increased to enhance the network's generalization capabilities by having more parameters.

- The output of the LidPose–3D network has been modified by extending its dimensions to include joint depths alongside the 2D predictions.

As Figure 4.2 demonstrates, the *LidPose* network structure can deal with different input and output configurations, depending on the considered channels of the above-defined five-layer image structure. The optimal channel configuration is a hyperparameter of the method, that can be selected upon experimental evaluation, as described in detail in Section 4.4. In our experiments, we tested the *LidPose* networks with the following five input data configurations:

- Lidar depth only (`D`)
- 3D real world coordinates (`XYZ`)
- 3D + Lidar depth (`XYZ+D`)
- 3D + Lidar intensity (`XYZ+I`)
- 3D + depth + intensity. (`XYZ+D+I`)

For the training and testing of the proposed method, a new dataset was introduced, comprising an NRCS Lidar point cloud segment and the co-registered human pose GT information for each sample object. The dataset is described in detail in Section 4.3. The three model variants introduced above are detailed in the following subsections.

4.2.2.1 *LidPose–2D*

For pose estimation in the 2D domain, the *LidPose–2D* network was created based on ViTPose [33] architecture. The patch embedding module of the ViTPose backbone was changed to handle custom input dimensions for the different channel configurations (`XYZ, D, I,` and their combinations).

This newly designed network architecture was trained end-to-end from an uninitialized state, with five separate networks trained for the input combinations listed above. For these methods predicting 2D joint positions, the training losses were calculated in the joint-heatmap domain. An example of the *LidPose–2D* prediction can be seen in Figure 4.3a.

4.2.2.2 *LidPose–2D+*

In this model variant, called *LidPose–2D+*, the 2D predictions created by *LidPose–2D* configuration are straightforwardly extended to the 3D space.

Each predicted 2D joint is checked, and if a valid depth measurement exists around the joint's pixel location in the Lidar range image, the 3D position of a given joint is calculated from its 2D pixel position and the directly measured depth value. This transfer from the 2D space to the 3D space implies a simple baseline method for 3D pose prediction models. However, the *LidPose–2D+* approach has a serious limitation originating from the inherent sparseness of the NRCS Lidar point cloud. 2D joints, whose positions are in regions with missing depth measurements in the 2D range image, cannot be extended to 3D. An example of the *LidPose–2D+* prediction is shown in Figure 4.3b, highlighting three joints that cannot be assigned to range measurements.

4.2.2.3 *LidPose–3D*

The limitations of *LidPose–2D+* can be eliminated by a new network, called *LidPose–3D* that aims to predict the depth of each detected joint, apart from its pixel position in the range image lattice. Similarly to the *LidPose–2D* variants described above, this network structure can handle inputs with different configurations of the `XYZ, D, I,` channels.

The *LidPose–3D* network's output is constructed with the extension of ViTPose [33] to predict depth values for the joints alongside their 2D coordinates. The normalized depth predictions are performed on a single channel depth image, in the same down-scaled image space (64×48), where the joint heatmaps are predicted. An example of the *LidPose–3D* prediction can be seen in Figure 4.3c.

4.2.3 *LidPose* training

The training input data is a 2D array with a given number of channels - depending on the training configuration (combinations of `XYZ, D, I`). For the different channel configurations, different patch embedding modules were defined to adopt the variable numbers of parameters in the input, as shown in Figure 4.2. For training and evaluation of the network, we also need the GT pose data, which we assume is available at this point. (Details of GT

generation will be presented in Section 4.3.)

Regarding the loss function of the *LidPose–2D* network, we followed the ViTPose [33] approach by using *Mean Squared Error (MSE)* among the predicted and the GT heatmaps:

$$L_{\text{LidPose–2D}} := L_{\text{joint2D}} = MSE(\text{HM}_{\text{pred}}, \text{HM}_{\text{GT}}), \tag{4.1}$$

where HM_{pred} and HM_{GT} are the predicted joint heatmap and the GT joint heatmap, respectively.

For the *LidPose–3D* network, the training loss is composed of two components: one responsible for the joints' 2D prediction accuracy (L_{joint2D}), the other reflecting the depth estimation accuracy (L_{depth}). The total training loss is a weighted sum of the position and depth losses:

$$L_{\text{LidPose–3D}} = W_{\text{joint2D}} \cdot L_{\text{joint2D}} + W_{\text{depth}} \cdot L_{\text{depth}} \tag{4.2}$$

For calculating the 2D joint position loss term L_{joint2D}, Equation (4.1) was used again. Regarding the depth loss L_{depth}, we tested three different formulas: *L1 loss*, *L2 loss* and *Structural Similarity Index Measure (SSIM)* [111]. Based on our evaluations and considering training runtime, the *SSIM* was selected for the depth loss measure in the proposed *LidPose–3D* network. Following a grid search optimization, the weighting coefficients in the loss function were set as $W_{\text{joint2D}} = 10$ and $W_{\text{depth}} = 1$.

4.3 Dataset for Lidar-only 3D human pose estimation

For the development and evaluation of the proposed *LidPose* method, we created a new dataset, since we have not found any public benchmark sets containing NRCS Lidar measurements with human pose GT.

GT annotation proved to be a challenging process since the visual interpretation of sparse 3D Lidar point clouds is difficult for human observers, and the inhomogeneous NRCS pattern makes this task even harder. For facilitating GT generation and the analysis of the results, in our experimental configuration, a camera was mounted near the NRCS Lidar sensor to record optical images as well, besides the point clouds. The camera images were only used

for creating the GT information for human pose estimation, and for helping the visual evaluation of the results of *LidPose*. During annotation, the operator used the camera images to mark, validate, and verify the skeleton joint positions.

During the dataset collection, the NRCS Lidar (Livox Avia [55]) and the *RGB* camera were mounted together on a standing platform, and the measurement sequences were recorded in two outdoor and one indoor location, where persons were walking in the sensors' FoV.

4.3.1 Spatio-temporal registration of Lidar and camera data

Since our experimental configuration uses both camera and Lidar data for creating the GT human poses and validating the results, the spatial transformation parameters between the two sensors' coordinate systems need to be determined by a calibration process.

The camera's extrinsic and intrinsic parameters were calibrated using OpenCV [112, 113] libraries and a Livox-specific, targetless calibration method [114]. The camera images were undistorted using the calibration distortion coefficients to remove lens distortion and provide rectified images for the dataset. Thereafter, the camera images and the Lidar range images were transformed into a common coordinate system.

To establish the spatial correspondence among the camera and Lidar sensors, the requirement of time synchronization of the data recording arose. The camera and the Lidar data were properly timestamped following the synchronization process described in the IEEE 1588 standard [115], using the *Precision Time Protocol daemon (PTPd)* [116], running on the data collector computer.

This enabled time-synchronous processing of both the camera and the Lidar sensor data with a precision of 1 ms. The camera and the Lidar data were recorded with different, sensor-specific data acquisition rates, at 30 Hz on the camera and at 10 Hz in the case of the Lidar. The corresponding image-point cloud pairs were created by selecting the camera image with the smallest time difference for each recorded Lidar point cloud. In other words, the data collection was adjusted to the Lidar's slower frame rate.

4.3.2 Human pose ground truth

Although the proposed *LidPose* method performs human pose estimation from solely NRCS Lidar point clouds, in the GT generation phase we also took advantage of the co-registered camera images that were recorded in parallel with the Lidar measurements.

4.3.2.1 2D human pose ground truth

The GT generation has been implemented in a semi-automatic way, exploiting established camera-based person detection and pose-fitting techniques. In the first step, in each data sample, the YOLOv8 [117] was run to detect the persons in the camera images. The detected persons' bounding boxes with sizes smaller than ViTPose's native input resolution (192×256) were discarded. The bounding box of a detected person was used to crop the person's region both on the camera image and in the Lidar data in the 2D range image domain.

In the second step, the initial pose estimation was created on the cropped camera images by the state-of-the-art 2D human pose estimator ViTPose [33] network with its *huge* configuration. This network configuration, where the network backbone had 32 transformer blocks, was selected based on its superior results in comparison to the smaller network variants. The trained model *ViTPose-huge* was obtained from the ViTPose [33] implementation from the repository at [118].

In the third step, the camera images were used to manually check, validate, filter, and fine-tune each 2D human pose, resulting in the 2D GT of human poses.

Since the Lidar range images and the camera images were co-registered (both in time and space), the filtered camera-based pose models can be directly used as GT of the 2D human poses in the Lidar's range image domain. The skeleton parameters in the 2D GT are stored in COCO-Pose [110] data format, which represents a given human pose with 17 keypoints, facilitating detailed pose estimation (see in Figure 4.3).

4.3.2.2 3D human pose ground truth

The 3D human pose GT is created by the extension of the 2D human skeleton dataset, so that we attempt to assign to each joint a depth value, based on the depth measurements of the Lidar sensor around the joint's 2D position.

The challenge of this 2D-to-3D point assignment task arises from the sparseness of the measured NRCS Lidar range image, which implies that some 2D joints cannot be assigned to genuine Lidar depth measurements on the considered Lidar frames. In these cases, we applied spatio-temporal interpolation, i.e. we interpolated the depth values of joints without direct range measurements from the depth values of other nearby joints, and nearby frames. When a given 2D joint's pair in 3D did not have its corresponding 3D pair, the estimation of the joint depth was done in the following order:

1. Finding valid depth measurements in the 2D depth image with a 9×9 kernel and using the point with the minimum value as the estimated depth.

2. Calculating the mean depth value of the previous and the next Lidar frames, if both contained valid depth measurements.

3. Mix of steps 1 and 2: Calculating the mean depth value of the previous and the next Lidar frames in a 9×9 kernel, and if both contained valid depth measurements, the point with the minimum distance is used as the estimate of the given point's depth.

4.3.3 Transforming the point cloud to the five-channel range image representation

As described in Section 4.2, the *LidPose* method requires that the 3D Lidar point cloud is transformed to a spherical polar coordinate system, using a 2D pixel lattice generated by quantizing the horizontal and vertical FoV-s. The 3D world coordinates of the Lidar points are stored in the 2D range image domain in different image channels.

As mentioned in Section 4.2.2, five different 2D data layers are created for each Lidar point cloud. The first layer is the depth map, where values are the distances of the Lidar points from the camera center. The second layer is the intensity map, where the values are the reflection intensity of the Lidar points. The remaining three layers store the coordinates of the Lidar points in the 3D space $(XYZ)_{3D}$ at the calculated (u, v) range image locations.

4.3.4 Dataset parameters

Independent recordings were made for the training, test, and validation datasets, where several moving pedestrians were observable in the sensors' FoV. One to three persons were walking at the same time following arbitrary directions in the observed field, meanwhile, they occasionally stopped during the movement, and some of them did gymnastic exercise-like activities. In parallel with the data capturing, the MoG-based foreground-background segmentation method [5] was run on the Lidar data, and the binary classification of the 3D points was stored for each frame alongside the camera and Lidar measurements.

In total, our created new dataset contains 9500 skeletons, and 161000 joints. The dataset was split into the independent training, validation, and test sets, having 5500, 490, and 3400 skeletons, respectively, as shown in Table 4.1.

The training set consists of two sequences, both containing three individuals moving in a narrow courtyard. The validation set comprises two sequences which are recorded in a wide courtyard containing two individuals. The test set consists of three further sequences: The first one is recorded indoors, in a large room with a single observed individual. The second test sequence is captured on a wide courtyard with two subjects, and the third one is recorded in the same location with a single individual.

To support the deeper analysis and understanding of the structure and properties of our new dataset, we created the following graphical demonstrations. Figure 4.4 demonstrates the number of joints at a given depth X_{3D} from the Lidar sensor. Figure 4.5 shows the number of joints in a given direction in

Dataset	Count		Location Mean, STD (m)		
	Joint	Skeleton	X	Y	Z
Train	94248	5544	$7.34(\pm 2.27)$	$0.13(\pm 1.29)$	$-0.54(\pm 0.54)$
Validation	8364	492	$7.59(\pm 2.26)$	$-0.05(\pm 2.22)$	$-0.50(\pm 0.52)$
Test	59228	3484	$6.86(\pm 2.28)$	$-0.25(\pm 1.55)$	$-0.55(\pm 0.52)$
Total	161840	9520			
Average			$7.26(\pm 2.27)$	$-0.06(\pm 1.69)$	$-0.53(\pm 0.53)$

Table 4.1. Overview of the distributions of the *LidPose* dataset over its Train, Validation, and Test splits.

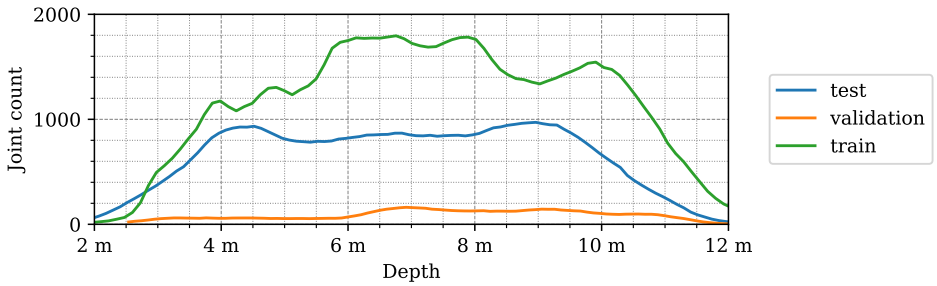


Figure 4.4. Distribution of the joints in the *LidPose dataset*, based on the depth coordinate (X) of the 3D joints.

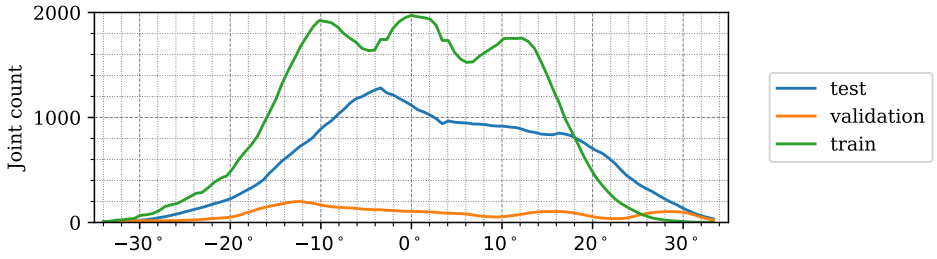


Figure 4.5. Distribution of the joints recorded in the *LidPose dataset*, based on the local emergence angle of the Lidar sensor

the Lidar FoV for the different datasets. It can be seen that the majority of the joint positions were recorded in the central, 40° wide region of the Lidar FoV.

Figure 4.6 presents the number of human poses displayed on the ground $(XY)_{3D}$ plane from a bird's eye view. It demonstrates that as the observed people were crossing the sensor FoV, the central regions registered more skeletons than the regions near the FoV edge.

Figure 4.7 shows the number of joints in the 2D camera image plane (u, v) in the pixel regions overlaid on a sample camera image. As the majority of the joints are recorded from the human torso, the regions above the ground with 1 m registered more keypoints than the lower, ankle- and knee regions.

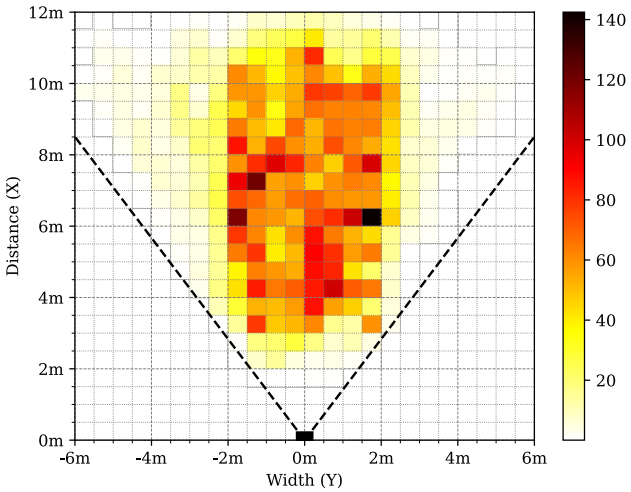
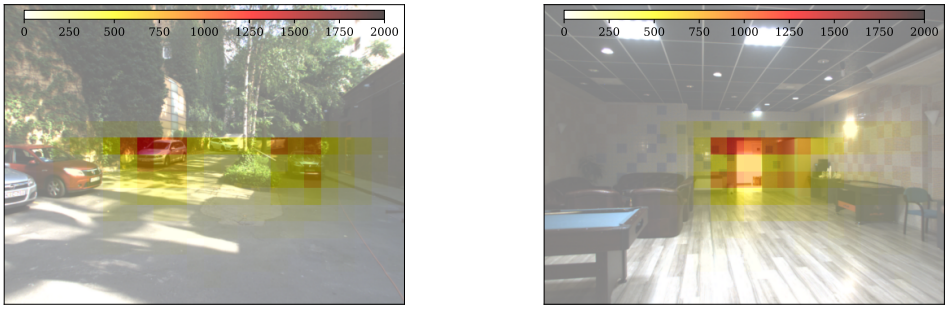


Figure 4.6. Distribution of joint positions in the *LidPose dataset*, displayed on the ground plane $(X, Y)_{3D}$ from the bird's-eye view.



(a) Distribution of 2D joint coordinate positions in the **outdoor test dataset**, overlaid on a sample camera image.

(b) Distribution of 2D joint coordinate positions in the **indoor test dataset**, overlaid on a sample camera image.

Figure 4.7. Distribution of 2D joint coordinate positions in the test dataset overlaid on a sample camera image.

4.4 Results and discussion

The proposed *LidPose* networks were trained to estimate human poses both in 2D and 3D. For *LidPose–2D*, 5 model variants were trained with different patch-embedding blocks on the corresponding input data configurations (`D`, `XYZ`, `XYZ+D`, `XYZ+I`, `XYZ+D+I`), as listed in Table 4.2,

Regarding *LidPose–3D*, we trained 12 model variants. On one hand, for each input configuration (`XYZ`, `XYZ+D`, `XYZ+I`, `XYZ+D+I`), the network was trained with different patch-embedding blocks. On the other hand, each configuration was trained with three different depth prediction losses: *L1, L2, and SSIM*. The trained models with their input and training loss are listed in Table A.1.

4.4.1 Metrics

The following metrics were calculated to compare the *LidPose* models. The visibility of a predicted joint j in a skeleton i is represented by $v_{(i,j)} \in [0, 1]$, indicating whether there is GT data for it. Thus, let N be the total number of visible joints in each dataset:

$$N := \sum_{i,j} v_{(i,j)}.$$

Additionally, let Y and $\hat{Y}$ be the GT and predicted coordinates of the keypoints, respectively.

Average Distance Error (ADE) measures the average Euclidean distance between the predicted pose and the GT pose across all skeleton joints, providing a measure of overall pose estimation accuracy. In the 2D case, normalization is applied based on the skeleton height to eliminate the varying skeleton sizes in the 2D image space. ADE, as defined in Equation (4.3). The lower the value, the better the performance.

$$\mathrm{ADE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i,j} v_{(i,j)} \left\| Y_{(i,j)} - \hat{Y}_{(i,j)} \right\|_2 \tag{4.3}$$

Mean Per-Joint Position Error (MPJPE) [119] measures the position errors of different joint types, as defined in Equation (4.4). MPJPE is

similar to the ADE metric, however, it can highlight the performance differences between different body parts, and regions.

$$\text{MPJPE}(Y, \hat{Y}, J) = \frac{1}{\sum_j^J \sum_i v_{(i,j)}} \sum_j^J \sum_i v_{(i,j)} \left\| Y_{(i,j)} - \hat{Y}_{(i,j)} \right\|_2, \tag{4.4}$$

where J is a subset of all joints.

Percentage of Correct Keypoints (PCK) [120] shows the percentage of joints in the estimated pose that fall within a certain threshold distance from their corresponding GT keypoints. In the 2D space, the distance threshold is set in pixels, while in the 3D space, it is set in meters. This measure defined in Equation (4.5) assesses the accuracy of joint localization at different levels of precision, the higher the value, the better the prediction.

$$\text{PCK}(Y, \hat{Y}, \alpha) = \frac{1}{N} \sum_{i,j} \delta_{(i,j)}(\alpha), \tag{4.5}$$

where α is the error threshold and δ is an indicator function:

$$\delta_{(i,j)}(\alpha) := \begin{cases} 1 & \text{if } \|Y_{(i,j)} - \hat{Y}_{(i,j)}\|_2 \leq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

the PCK curve can be constructed by sweeping the distance threshold.

The *Area Under Curve (AUC)* value of a PCK curve is a good generalizing metric for human pose estimation tasks [121]. PCK evaluates the performance of an examined human pose estimation method based on a single threshold, PCK-AUC on the other hand uses a series of thresholds, providing a more comprehensive assessment of the method's performance. This also reduces the sensitivity of the results to the choice of the parameter.

Limb Angle Error (LAE) calculates the mean angular difference between the orientations of corresponding limbs (arms, legs) in the predicted skeleton and the GT skeleton, as defined in Equation (4.7). It assesses the accuracy of orientation estimation both in the 2D and 3D space.

$$\text{LAE}(Y, \hat{Y}, L) = \frac{1}{\sum_i v_i^L} \sum_i |\operatorname{angle}(Y_i, L) - \operatorname{angle}(\hat{Y}_i, L)|, \tag{4.7}$$

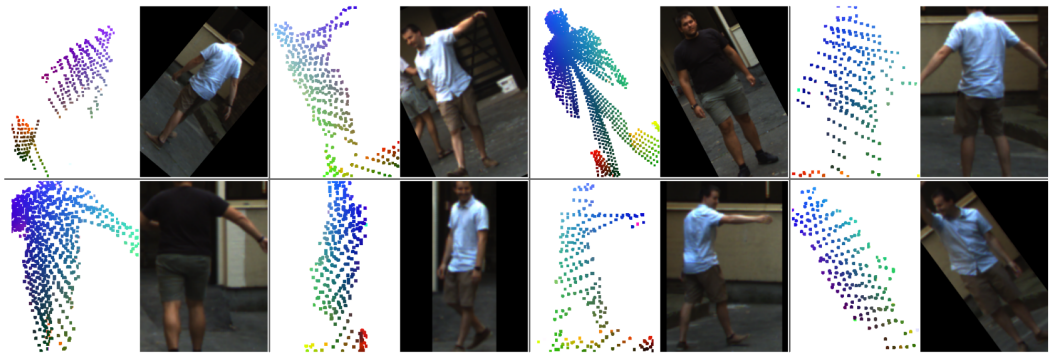


Figure 4.8. Example training batch of input data with the randomly applied augmentations (horizontal mirroring, scaling, rotation, half body transform). The camera images are shown for visual reference only.

where L is a subset of joints that has three elements that are connected by the skeleton edges, and $v_i^L \in [0, 1]$ indicates whether the whole limb is present in the prediction and GT for a given skeleton. *angle()* calculates the angle of the skeleton edges at the middle joint of the limb.

Limb Length Error (LLE) was calculated on skeleton limbs (arms, legs) to measure how the network predicts their total length, as defined in Equation (4.8). This measure does not penalize if the elbow or the knee is not predicted accurately until the total limb length is estimated correctly.

$$\text{LLE}(Y, \hat{Y}, L) = \frac{1}{\sum_i v_i^L} \sum_{l}^{L} \sum_{i} ||Y_{(i,l)}| - |\hat{Y}_{(i,l)}||, \tag{4.8}$$

where L and v_i^L notations are the same as in Equation (4.7).

4.4.2 Experiment parameters

During the training of the *LidPose* models, data augmentation was applied both to the five-channel 2D input arrays and the GT skeletons. Vertical mirroring, scaling, and rotation transforms were added to each data sample randomly to enhance model robustness and estimation efficiency. To enhance the network's robustness on partial skeletons, *half-body transform* was applied randomly during the training process, where either the upper body or the lower body of a skeleton was selected and cropped, as in [33]. Figure 4.8 shows a batch of input data with the randomly applied augmentations mentioned above.

Table 4.2. LidPose–2D network results on different input types with position loss. The meaning of the *Input* values: `D`: Lidar distance; `XYZ`: point 3D coordinates; `I`: Lidar intensity; Percentage of Correct Keypoints (*PCK*) was calculated with the error being at most 10 pixels. The *AUC-PCK* was calculated on the $[0, 30]$ pixel interval as shown in Figure 4.9.

Model	Input	ADE↓	PCK↑	AUC-PCK↑	LAE↓	LLE↓
2D–1	`D`	18.0726	0.4316	0.5360	13.7856	9.7695
2D–2	`XYZ`	14.4013	0.4960	0.5952	12.6956	9.5330
2D–3	`XYZ+D`	14.6881	0.4966	0.5926	12.7078	**9.4509**
2D–4	`XYZ+I`	**13.2473**	**0.5278**	**0.6166**	**12.5251**	10.6579
2D–5	`XYZ+D+I`	13.8399	0.5122	0.6049	12.6762	11.1547

During the training of *LidPose*, AdamW was used with weight decay coefficient $\lambda = 0.1$ and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The maximum learning rate was set to $\gamma = 5 \cdot 10^{-4}$, this was reached after 3 batches with a ramp-up. Learning rate decay was used to decrease the learning rate exponentially by a factor of 0.1 between epochs $20 - 30$, $30 - 35$, and $35 - 100$.

The proposed *LidPose* runs at 52 FPS on the prerecorded dataset in offline processing on singleton batches. In the end-to-end application of the proposed pipeline, the frame rate of the method is determined by the NRCS Lidar's sampling rate (10 FPS).

4.4.3 *LidPose–2D* evaluation

The evaluation results based on the metrics described in Section 4.4.1 are shown in Tables 4.2 and 4.3. The test results show that Model *2D–4* outperforms the other model variants with almost all the metrics for the 2D human skeleton estimation task. This best model variant corresponds to the `XYZ+I` channel configuration, i.e. it uses the 3D point coordinate values and the Lidar reflection intensity.

From Table 4.2 it can be seen that the depth-only (`D`) method (*2D–1*) has weak performance, as the network does not have enough information to estimate the 2D skeleton positions accurately. If the input of the *LidPose–2D* network is the real world 3D point coordinate data in three input channels (`XYZ`) (*2D–2*), the ADE and the LAE scores decrease significantly, showing more accurate estimations. This means, that the use of the 3D coordinates

Table 4.3. Mean Per-Joint Position Error (MPJPE) values (↓) of the LidPose–2D network for different joints.

Model	head	shoulders	elbows	wrists	hips	knees	ankles	ADE↓
2D–1	12.4838	17.5230	25.2190	27.6912	13.6437	16.6865	21.6442	18.0726
2D–2	11.3505	12.8925	17.3831	19.7888	**10.9468**	**13.7076**	19.3161	14.4013
2D–3	11.2303	13.1998	18.1070	20.5023	11.2541	14.0968	19.6134	14.6881
2D–4	**10.0393**	**11.1264**	**14.5071**	**17.0304**	11.5661	13.9160	19.3576	**13.2473**
2D–5	10.1436	11.6997	15.8984	18.6925	12.7537	14.1663	**19.0698**	13.8399

(`XYZ`) instead of the Lidar depth values (`D`) increases the network's generalization capability. The combination of the two formers, i.e. the depth values (`D`) and the 3D joint coordinates (`XYZ`) used as the input, model variant *2D–3* achieves the lowest LLE score. If the previous variant is extended by the Lidar intensity (*2D–5*), the network does not outperform the *2D–4* network variant, as the former achieves 13.84 ADE, while the latter scores 13.2 ADE. This shows, that adding the depth features which can be calculated from the 3D coordinates does not enhance the network's performance.

Table 4.3 lists the MPJPE values for the different LidPose–2D model variants. It can be seen that the torso joints (head, shoulders, hips) have lower MPJPE scores than the limb-related joints. This can be explained by the smaller size of those parts and thus the fewer or no measurements in the sparse Lidar point cloud at those locations. An example of this can be seen in the left leg of the person in Figure 4.11f.

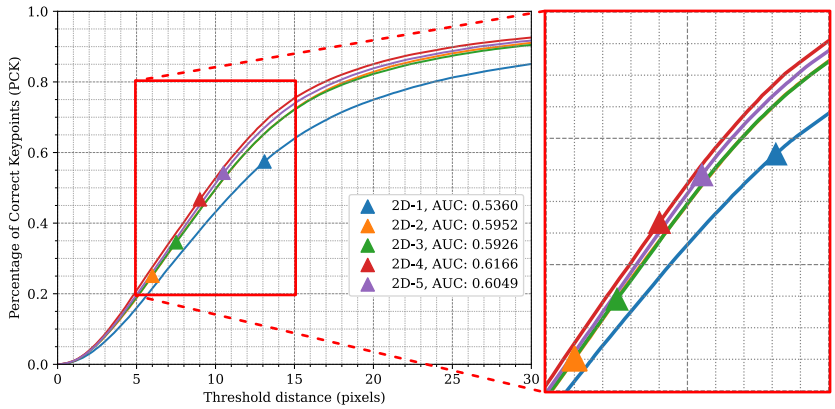


Figure 4.9. *LidPose–2D*: Percentage of Correct Keypoints for the different 2D networks with different joint-correspondence threshold acceptance values. Model *2D–4*, which has been trained on 3D coordinates + Lidar intensity, has the best PCK curve.

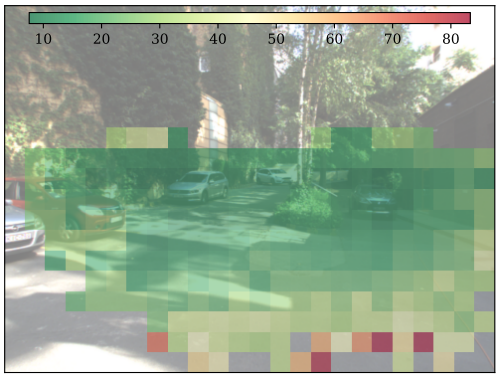


Figure 4.10. 2D Average Distance Error of the selected *2D–4* model, overlaid on a sample camera image.

Figure 4.9 shows the PCK values of each 2D model for different threshold values. The AUC-s of these PCK graphs were calculated (also shown in Table 4.2), where the *Model 2D–4* has the highest score.

The ADE of the selected model was evaluated in different 2D image regions, as shown in Figure 4.10. From this figure it can be seen that as the 2D estimation positions in this 2D camera image space are getting closer to the edge of the Lidar FoV, the ADE value increases above 50 pixels, meanwhile, in the central regions, the ADE score is below 20 pixels. This behavior is the consequence of the inhomogeneous nature of the NRCS Lidar point cloud, where the point cloud sparseness increases with the distance from the sensor's optical center.

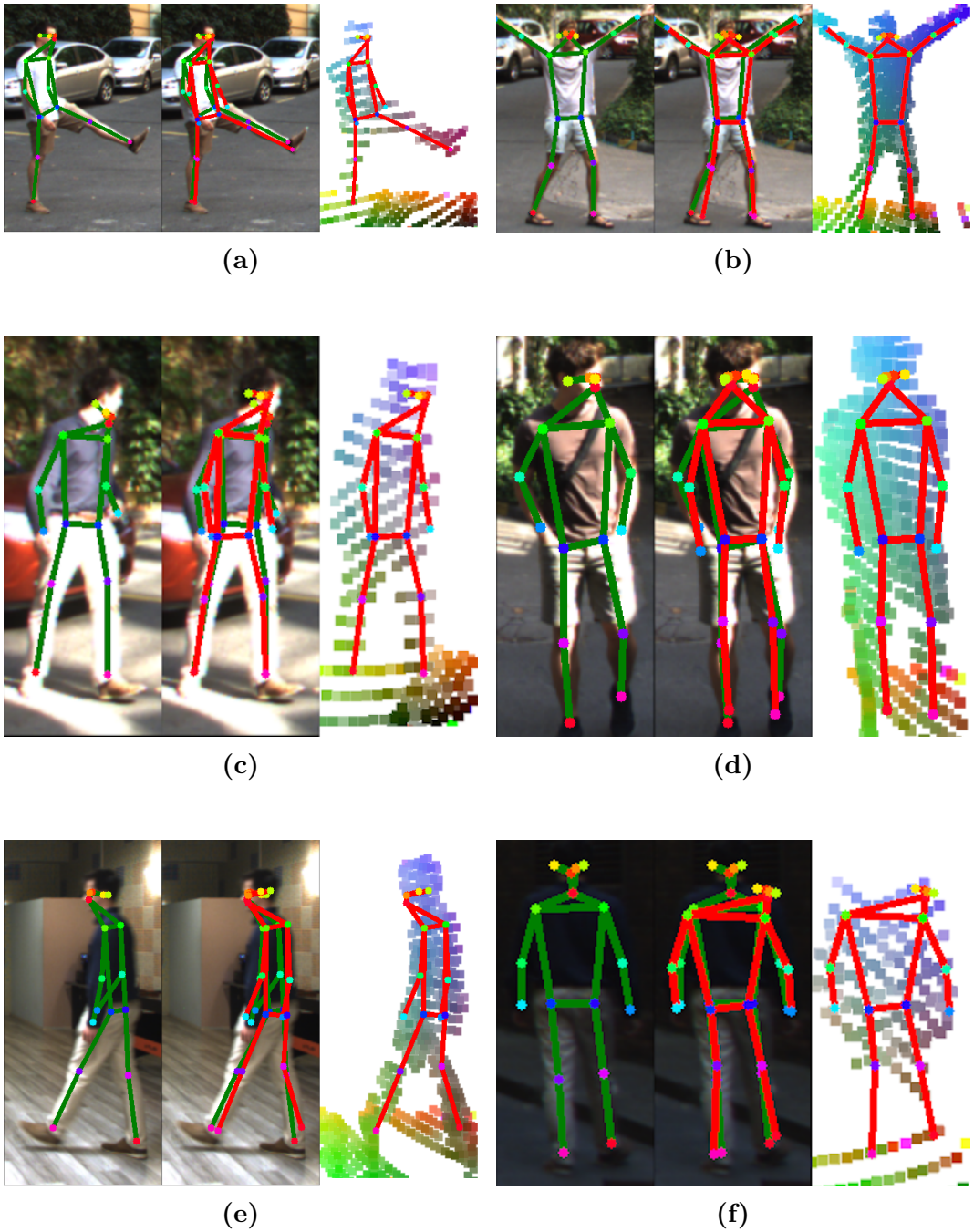


Figure 4.11. *LidPose–2D* predictions are shown in red, overlaid on the input Lidar point cloud (*right*). The GT is shown in green, drawn over the corresponding camera frame (*left*). The prediction in red and the GT in green are shown together in the input Lidar point cloud (*middle*).

Example pose estimations are shown in Figure 4.11, where the GT is shown in the camera image, and the Lidar-based 2D skeleton prediction is displayed on the sparse point cloud. Figure 4.11b and 4.11d show skeletons, where

the human was at 5 m distance to the Lidar resulting in less sparse point clouds. On the contrary, Figure 4.11a, 4.11c and 4.11e show skeletons at 10 m distance, having much less Lidar points in the frame. It can be observed that the skeleton estimation accuracy is high, as the predicted and the GT are very close. Figure 4.11f shows an example, where the prediction makes a mistake on the person's head as there are no recorded 3D points from that region at that given frame.

4.4.4 *LidPose–3D* evaluation

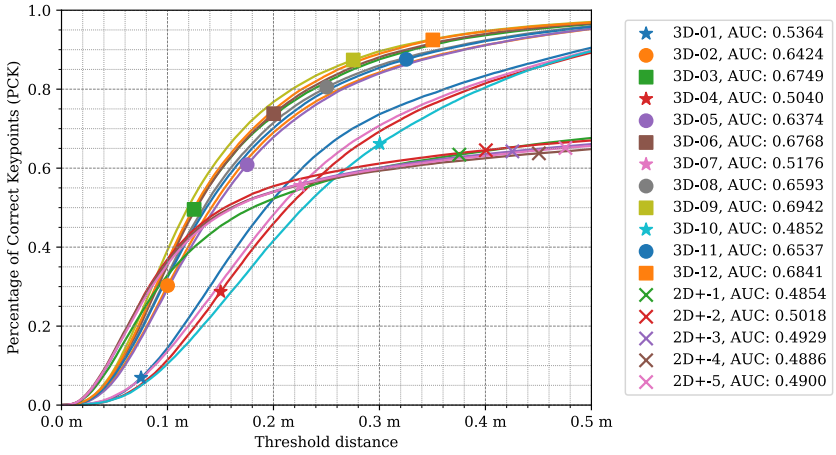


Figure 4.12. *LidPose–3D*: Percentage of Correct Keypoints in the 3D space for the different 3D (and 2D+) networks with different joint-correspondence threshold distance acceptance values. *Model 3D-9*, which has been trained on 3D coordinates + Lidar intensity with SSIM-based depth loss, has the best PCK curve.

The *LidPose–3D* networks predict the 2D joint positions in the same manner as LidPose–2D, and the depth values for each joint. From the predicted 2D position and the depth values the 3D joint positions are calculated. The results are evaluated using various 3D metrics in the 3D space as described in Section 4.4.1. The baseline of the 3D evaluation is the *LidPose–2D+*, described in Section 4.2.2.2. Tables A.1 and A.2, and Figure A.1 show the results for both LidPose–3D and LidPose–2D+ models. As we can see from these results, the predictions of the *LidPose–3D* models are considerably better overall.

Upon assessing the PCK values of the 3D models in Figure 4.12, the models can be grouped based on their PCK curve shape. The first group consists of models, that did not learn the depth properly during training. Namely, *3D–01, 3D–04, 3D–07* and *3D–10* have failed to learn depth estimation. Their

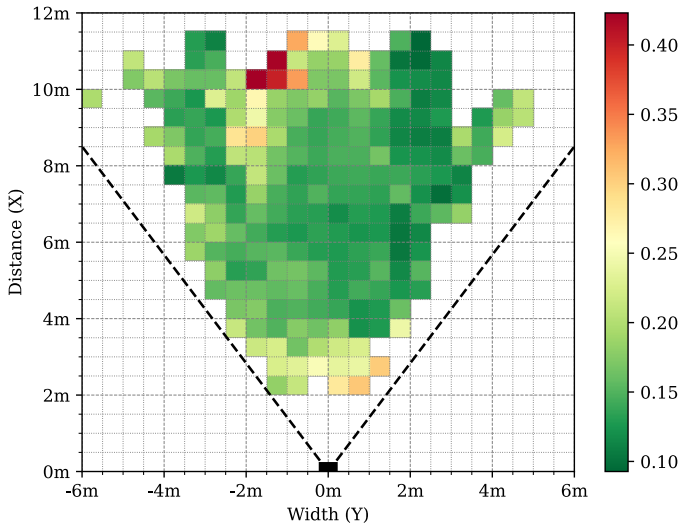


Figure 4.13. Distribution of Average Distance Error of the predicted joints in bird's eye view, using the selected *3D–09* model. Only cells with more than 24 annotated joints are shown.

common attribute is that they were using L1 loss to penalize the depth error during the learning process.

The second group contains the projected 2D models (LidPose–2D+). These models all perform very similarly to each other, while distinctly from the other two groups. They serve as a baseline for the proposed method. Their performance is equal to or better than the 3D models if the threshold is set between $0 - 0.1$ m, as they have significantly more correct predictions than in a larger distance. This is due to the assembly of the 3D predictions from existing 3D points at the predicted 2D joints' positions. These characteristics highlight, that while this approach works well with sparse but homogeneous Lidar measurements, as shown in [93], it fails on point clouds recorded with NRCS Lidar.

Lastly, the third group is the rest of the 3D models, which use L2 loss and SSIM as the depth criterion. As can be seen, these models correctly estimate the human poses, and the trend is similar to the 2D models in Figure 4.9. Notably, while the shape of these curves is similar, models with the SSIM-based depth loss outperform the models trained with L2 loss. *Model 3D–09* outperforms all other configurations.

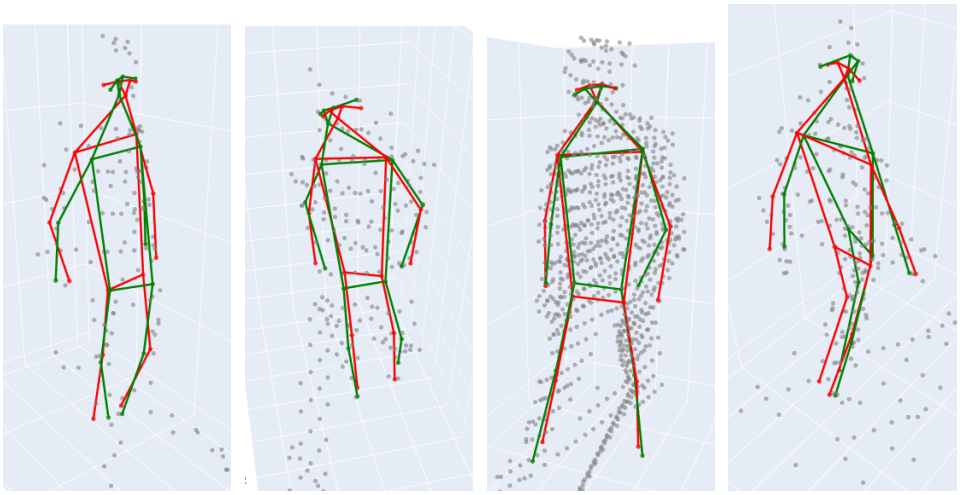
The best 3D network, *3D–09* was evaluated with the ADE metrics on the ground plane on the test dataset to show the spatial dependency of the pose

estimation performance at different regions, as shown in Figure 4.13. Although the maximum ADE is 0.5 m, most of the cells of the ground grid have less than 0.3 m average error rates.

Table A.2 shows the MPJPE results for the 3D methods. It can be seen that the projected 2D+ models (LidPose–2D+) are outperformed with all the LidPose–3D networks. The 2D models listed in Table 4.2 were projected to 3D prediction using the inhomogeneous sparse Lidar data. This was done by using nearby 3D data where it was available for the back-projected 2D predictions. However, due to the characteristics of the NRCS Lidar sensor, this approach has its limitations. Figures 4.12 and A.1, and Table A.1 also show, that *LidPose–3D* outperforms the extended *LidPose–2D+* networks.

In Figure 4.14, 3D human pose samples are shown from different viewing angles. By inspecting Figures 4.14a and 4.14b, it can be seen that there is a correlation between the density of the points and the accuracy of the network. This angle- and distance dependency can also be observed in Figures 4.10 and 4.13.

The experiments in this section have shown that the proposed *LidPose* methods are capable of the efficient and accurate estimation of the human poses. Our obtained results provide strong evidence, that the NRCS Lidar sensor is suitable for solving the Lidar-only 2D and 3D human pose estimation tasks.



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

Figure 4.14. *LidPose3D* predicted skeletons using the 3D–09 configuration. Red skeleton: 3D prediction. Green skeleton: GT. Gray points: NRCS Lidar points.

Chapter 5

Conclusions of the thesis

This thesis deals with three different research problems that raise important challenges for experts in machine perception.

The change detection task among coarsely registered Lidar point clouds was solved using state-of-the-art neural network components. The custom dataset created for this task has been recorded with an RMB Lidar sensor. The solution for the point cloud change detection is created by using an adversarial training strategy, where two neural networks compete against each other to achieve more accurate change detection results. It has been shown by a detailed example, that the proposed method is capable of efficiently detecting changes among coarsely registered Lidar point clouds.

The second research problem was raised by experimenting with a new type of Lidar sensor, having an unusual non-repetitive circular scanning. The foreground-background classification task of this specific Lidar point cloud could not be solved by existing methods, which step was required by other higher-level perception tasks. In the dissertation, a method was proposed for this research problem. It has been shown in real measurements and in a working demonstration, that the proposed method can classify Lidar points both from recordings and live sensor data.

The Lidar sensor with non-repetitive circular scanning was used for perceiving the third research problem. The human pose estimation task using only the NRCS Lidar point cloud task was solved by defining a vision-transformer-based neural network. It was shown by several examples, that the proposed solution is capable of estimating the human poses using only NRCS Lidar data.

5.1 New Scientific Results

1. Thesis: I proposed a novel change detection approach for coarsely registered RMB Lidar point clouds in complex, street-level urban environments. The input point clouds are represented by range images, the result of the method is a pair of binary masks showing the change regions on each input range image, which can be back-projected to the input point clouds without loss of information. I have evaluated the proposed method in various challenging scenarios, and I have shown its superiority against state-of-the-art change detection methods.

The method, called *ChangeGAN* was published in a journal paper [1], and it was submitted to a patent application [3].

In the initial phase of this research, in conference paper [6] a method was described for multi-object detection in urban scenes utilizing 3D background maps and tracking. It uses a dense 3D city map to increase the accuracy of object detection on a sparse point cloud from a Lidar sensor. This method can extend the camera-based machine perception of a road vehicle, described in [7]. For the evaluation of the results considering the object trajectories, a track-to-track evaluation method can be used [8].

The need to solve the point-based detection of changed regions due to object displacements between initially unmatched (coarsely registered) pairs of point clouds can be emphasized with practical cases, where reliable registration and therefore the change detection cannot be achieved with currently available methods. I introduced a novel problem formulation: I described the differences among a coarsely registered pair of point clouds without exactly matching the available input point cloud measurements.

As a key feature, the proposed method does not require precise registration of the point cloud pairs. Based on my experiments, the proposed method is more efficient than existing solutions, and it can efficiently handle up to 1 m translation and 10° rotation misalignment between the corresponding 3D point cloud frames.

1.1. Subthesis: I have defined a deep neural network structure, capable of learning and robustly extracting changes between coarsely registered 3D sparse point clouds obtained in a complex street-level environment. For the training of this neural network, I proposed a semi-automatic method to create a change detection dataset with coarsely registered point cloud pairs using simulated registration errors.

The proposed deep learning approach takes as input two coarsely registered 3D point clouds recorded with an RMB Lidar sensor $\mathcal{P}_1$ and $\mathcal{P}_2$ represented by range images I_1 and I_2, respectively (shown in Figures 2.1a and 2.1b). The proposed architecture assumes that the images I_1 and I_2 are defined over the same pixel lattice and have the same spatial dimensions.

I have adopted a Siamese style [68] architecture to extract relevant features from the input range image pairs. The Siamese architecture is designed to share the weight parameters across multiple branches, allowing us to extract similar features from the inputs and to decrease the memory usage and training time. Each branch of the Siamese network consists of fully convolutional downsampling blocks. This step is followed by using a batch normalization layer, and finally, the output of the downsampling block is activated using a leaky ReLU function. Next, the outputs of the Siamese branches are concatenated for all feature channels, and a 1×1 convolutional layer is applied to aggregate the merged features.

The second part of the proposed model contains a series of transposed convolutional layers to upsample the signal from the lower-dimensional feature space to the original size of the 2D input images. Finally, a 1×1 convolutional layer, activated with a sigmoid function, generates the two binary change maps Λ_1 and Λ_2.

To regularize the network and prevent over-fitting, dropout technique is used after the first two transposed convolutional layers. To improve the change detection result an idea was adopted from U-net [76] by adding higher resolution features from the downsampling blocks to the corresponding transposed convolutional layers.

To achieve more accurate feature matching, Spatial Transformer Network blocks [73] were added for both Siamese branches. STN can learn an optimal affine transformation between the input feature maps to reduce the spatial

registration error between the input range images. Furthermore, STN dynamically transforms the inputs, also yielding an advantageous augmentation effect.

For the training of the *ChangeGAN* neural network I have created a new Lidar-based urban dataset called *Change3D*[1]. The measurements were recorded over two days in downtown Budapest using a Velodyne HDL-64 RMB Lidar mounted on a car.

The manual annotation of point cloud differences is very challenging, even if the point clouds originate from the same coordinate system. To ensure the accuracy of the GT, I performed the change labeling for registered point cloud pairs captured from the same sensor position and orientation, then the reference positions and orientations of the second frames were randomly transformed yielding a large set of accurately labeled coarsely registered point cloud pairs.

The training database contains 20000 point cloud pairs from 50 locations, while the test set was composed of 2000 point cloud pairs from completely different measurement locations.

In summary, I have created a new dataset suitable for training and evaluating new change detection methods where accurate registration of the compared point clouds is not required.

The proposed architecture outperforms the state-of-the-art methods on the created Change3D dataset [1].

1.2. Subthesis: I have proposed a novel, competitive classifier - discriminator-based adversarial training method for the change detection task on a coarsely registered pair of 3D point clouds.

The *classifier* network is responsible for learning and predicting the changes between the range image pairs. In each training epoch, the classifier model is trained on a batch of data. The actual state of the classifier is used to predict validation data, which is fed to the discriminator model.

The *discriminator* network is a fully convolutional network that classifies the output of the classifier network. The discriminator model divides the image into patches and decides for each patch whether the predicted change region is real or fake. During training, the discriminator network forces the classifier

[1]Dataset link: `http://mplab.sztaki.hu/geocomp/Change3D.html`

model to create better and better change predictions, until it cannot decide about the genuineness of the prediction.

Figure 2.3 demonstrates the proposed adversarial training strategy. I calculate the L1 Loss (L_{L1}) as the mean absolute error between the generated image and the target image, and I define the Adversarial (Adv) Loss (L_{Adv}), which is a sigmoid cross-entropy loss of the feature map generated by the discriminator and an array of ones. The final loss function of the method (L) is the weighted combination of the Adversarial Loss and the L1 Loss: $L = L_{\text{Adv}} + \lambda * L_{L1}$.

2. Thesis: I proposed a novel, end-to-end method for real-time foreground-background segmentation and human pose estimation, solely based on point cloud measurements of a Non-repetitive Circular Scanning Lidar sensor.

The method was published in a journal [2] and a conference paper [5]. I introduced a modified ViTPose [33] approach, which is adapted to the 3D point clouds and can efficiently handle the sparsity and the unusual rosetta-like scanning pattern of the NRCS Lidars. The proposed method's first step utilizes a foreground-background segmentation technique [5] for the NRCS Lidar sensor to select foreground points. In the next step, the *LidPose* human pose estimator network estimates the human pose in the filtered NRCS Lidar point cloud segments.

The proposed method is a complete and end-to-end approach to human pose estimation from raw NRCS Lidar measurement sequences, captured by a static sensor for surveillance scenarios.

To evaluate the method, I have created a novel, real-world, multi-modal dataset, containing camera images and Lidar point clouds from a Livox Avia sensor with annotated 2D and 3D human skeleton GT.

Figure 4.11 shows the predictions of the proposed *LidPose* method in 2D.

2.1. Subthesis: I proposed a point-level foreground-background segmentation technique for NRCS Lidar point cloud sequences recorded in a static sensor configuration. I proved that the proposed method can handle the sparsity of the NRCS Lidar measurements in a surveillance scenario. I created a database for the testing and evaluation of the proposed approach and demonstrated its efficiency [5].

To solve the point-wise foreground-background segmentation task, it is required to efficiently balance between the spatial and the temporal resolution of the recorded NRCS Lidar data, shown in Figure 1.7. For this reason, I create and maintain a very high-resolution background model of the sensor's FoV using a MoG-based method [5], displayed in Figure 3.5b. On the other hand, to enable real-time analysis of dynamic objects, I use low integration time to extract the consecutive Lidar frames. As a result, the laser reflections from foreground objects reflect sparse, but geometrically accurate samples of the silhouettes (shown in Figure 3.5a) providing valuable input for higher-level shape description, object detection, and pose estimation, as described in Subthesis 2.3. I demonstrated the efficiency of the new approach in different realistic NRCS Lidar measurement sequences.

2.2. Subthesis: I have proposed a semi-automatic method to create a human pose dataset with camera images and NRCS Lidar measurements.

GT annotation of Lidar point clouds is a challenging process, since the visual interpretation of sparse 3D Lidar point clouds is difficult for human observers, and the inhomogeneous NRCS pattern makes this task even harder. In the experiments, a camera was mounted near the NRCS Lidar sensor to record optical images as well, besides the point clouds, as shown in Figure 1.4. The camera images were only used for creating the GT information for human pose estimation, and for helping the visual evaluation of the results of *LidPose*.

GT generation has been implemented in a semi-automatic way, exploiting established camera-based person detection and pose-fitting techniques.

1. In each data sample, the YOLOv8 [117] was run to detect the persons in the camera images.
2. The initial pose estimation was created on the cropped camera images

by the state-of-the-art 2D human pose estimator ViTPose [33] network.
3. The camera images were used to manually check, validate, filter, and fine-tune each 2D human pose, resulting in the 2D GT of human poses.
4. The filtered camera-based human pose model was directly used as the GT of the 2D human poses in the co-registered Lidar's range image domain.
5. The 3D human pose GT is created by the extension of the 2D human skeleton dataset, so I attempted to assign to each joint a depth value, based on the depth measurements of the Lidar sensor around the joint's 2D position.
6. Spatio-temporal interpolation was applied on joints without direct range measurements from the depth values of other nearby joints, and nearby frames.

In total, the created new dataset contains 9500 skeletons, and 161000 joints. The dataset was split into independent training, validation, and test sets, having 5500, 490, and 3400 skeletons.

In summary, I have created a new dataset suitable for training and evaluating a new human pose estimation method that uses only NRCS Lidar point cloud as an input. To prove the usability of the dataset, I have proposed a vision transformer-based neural network to perform human pose estimation, the details of which are described in Subthesis 2.3.

2.3. Subthesis: I proposed a novel, visual transformer-based method for real-time human pose estimation from inhomogeneous and sparse Lidar point clouds recorded with an NRCS Lidar sensor.

First, the moving objects are separated from the static scene regions in the NRCS Lidar point clouds, as described in Subthesis 2.1 and in [5].

In the next step, the NRCS Lidar point cloud and the range image are cropped with the foreground regions' bounding boxes.

To jointly represent the different available measurement modalities, I proposed a new 2D data structure that can be derived from the raw Lidar measurements straightforwardly and can be efficiently used to train and test our proposed *LidPose* model. I generate a five-channel image from the input point cloud, mapped onto the Lidar sensor's 2D range image lattice. Two channels store the depth and intensity values of the Lidar measurements, while the remaining three channels encode the `X,Y,Z` coordinates of the corresponding

points in the 3D world coordinate system.

The ViTPose [33] network structure was used as a starting point in the research and development of the proposed *LidPose* methods' pose estimation networks. My main contributions to the proposed *LidPose* method:

- A new patch embedding implementation was applied to the network backbone to handle efficiently and dynamically the different input channel counts.
- The number of transformer blocks used in the *LidPose* backbone is increased to enhance the network's generalization capabilities by having more parameters.
- The output of the LidPose–3D configuration has been modified as well by extending the predictions' dimension to be able to predict the joint depths alongside the 2D predictions.

The obtained results published in [1] confirm, that the proposed method can detect human skeletons in sparse and inhomogeneous NRCS Lidar point clouds. The results of the 3D human pose estimation using the proposed *LidPose* method are shown in Figures 4.11 and 4.14.

The approach gives accurate human pose estimation results in real-time in the 3D world coordinate system of the scene, which can be used in higher-level scene analysis steps of surveillance systems.

5.2 Application and dissemination of the results

5.2.1 ChangeGAN

The proposed *ChangeGAN* [1], [3] can robustly extract changes between sparse point clouds obtained in a complex street-level environment. As a key feature, the proposed method does not require precise registration of the point cloud pairs. Based on my experiments, it can efficiently handle up to 1 m translation and 10° rotation misalignment between the corresponding 3D point cloud frames. This makes the proposed method suitable for real-world applications, where the precise registration of the point clouds is not feasible due to the complexity of the environment or the limitations of the sensors. The method can be applied in automatic public infrastructure monitoring, where detecting

possibly dangerous situations caused by e.g., missing traffic signs, and damaged street furniture is crucial. Expensive and time-consuming efforts can be reduced in city management offices by applying this method to automatically and continuously analyze and compare multi-temporal recordings from large areas to find relevant environmental changes.

5.2.2 LidPose

In the *LidPose* paper [2] I gave evidence, that the Livox Avia [55] NRCS Lidar can be widely adopted in real-life scenarios due to its low price, can be used for solving complex human pose estimation tasks, while the process highly respects the observed people's privacy as the people are barely recognizable by human observers from the recorded sparse point clouds.

The change detection accuracy can be increased by applying a novel depth image completion technique, which eliminates the uneven sparseness of the NRCS Lidar data, as described in a submitted patent application [4].

5.2.3 Publications and dissemination

The research results were published mainly in prestigious journals and conferences, as cited in the theses.

On top of those I presented my research progress at the biannual *Conference of the Hungarian Association for Image Analysis and Pattern Recognition (KÉPAF)* [9–11] and in the *PhD proceedings, annual issues of the Doctoral School, Faculty of Information Technology and Bionics* [12–15].

I demonstrated my results among others at the *Researcher's Night*[2], and at various events organized by the *Artificial Intelligence National Laboratory (MILAB)* and *National Lab for Autonomous Systems (ARNL)*, including the AI & Aut Expo 2023[3].

[2] https://sztaki.hun-ren.hu/kutatok-ejszakaja-2022#xr

[3] https://www.facebook.com/photo/?fbid=8779797418761582&set=pcb.8780186178722706

5.3 Computational resources

5.3.1 ChangeGAN

For the training and evaluation of the *ChangeGAN* method [1], [3] a PC was used with an i8-8700K CPU @3.7 GHz with 12 threads, 32 GB RAM, and a GeForce GTX 1080Ti. This setup was sufficient for the training of the proposed method, and the training time was reasonable. The proposed *ChangeGAN* runs at 16 FPS on the prerecorded dataset in offline processing on singleton batches. This makes the method usable for real-world applications, as the method's computational requirements are limited, and the required hardware is widely available nowadays.

5.3.2 LidPose

For both training and inference of the LidPose [2], two types of computers were used: a set of desktop computers having 12/16 CPU threads, 32 GB RAM and 11 GB vRAM in Nvidia GeForce 1080Ti GPU, and a cloud computer instance in HUN-REN Cloud [122] with 8 vCPU cores, 32 GB RAM, and 16 GB vRAM in an Nvidia Tesla V100 GPU cluster. The training was run with a batch size of 48, and one step took 5 seconds on both types of computers.

The proposed *LidPose* runs at 52 FPS on the prerecorded dataset in offline processing on singleton batches. In the end-to-end application of the proposed pipeline, the frame rate of the method is determined by the NRCS Lidar's sampling rate (10 FPS).

Bibliography

Journal publications of the thesis

[1] B. Nagy, **L. Kovács**, and C. Benedek, "ChangeGAN: A deep network for change detection in coarsely registered point clouds," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8277–8284, 2021, **IF: 4.3, Scimago Q1/D1**. (Cited on pages 10, 12, 44, 69, 71, 75, and 77.)

[2] **L. Kovács**, B. M. Bódis, and C. Benedek, "LidPose: Real-time 3d human pose estimation in sparse lidar point clouds with non-repetitive circular scanning pattern," *Sensors*, vol. 24, no. 11, 2024, **IF: 3.9, Scimago Q1**. (Cited on pages 72, 76, and 77.)

Patents related to the thesis

[3] B. Nagy, **L. Kovács**, C. Benedek, T. Szirányi, Ö. Zováthi, and L. Tizedes, "Training method for training a change detection system, training set generating method therefor, and change detection system," WO Patent application, WO/2023/007198, International Filing Date: 08.07.2022, Priority data: P2100280, 27.07.2021, HU. (Cited on pages 10, 12, 44, 69, 75, and 77.)

[4] Ö. Zováthi, Z. Rózsa, B. Pálffy, Z. Jankó, C. Benedek, T. Szirányi, **L. Kovács**, and M. Kégl, "Methods for spatial and temporal densification of Lidar measurements," Patent application, Priority data: P2300075, 01.03.2023, HU. (Cited on page 76.)

Conference publications of the thesis

[5] **L. Kovács**, M. Kégl, and C. Benedek, "Real-time foreground segmentation for surveillance applications in nrcs lidar sequences," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B1-2022, pp. 45–51, 2022. (Cited on pages 44, 46, 55, 72, 73, and 74.)

[6] Ö. Zováthi, **L. Kovács**, B. Nagy, and C. Benedek, "Multi-object detection in urban scenes utilizing 3d background maps and tracking," in *2019 International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO)*, 2019, pp. 231–236. (Cited on page 69.)

[7] A. Horvath, I. Horvath, A. Kiss, D. Huszar, A. Palffy, **L. Kovács**, D. Babicz, B. Farkas, G. Majoros, and C. Rekeczky, "Cellular vision based adas applications," in *CNNA 2016; 15th International Workshop on Cellular Nanoscale Networks and their Applications*, 2016. (Cited on page 69.)

[8] **L. Kovács**, L. Lindenmaier, H. Nemeth, V. Tihanyi, and A. Zarandy, "Performance evaluation of a track to track sensor fusion algorithm," in *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications*, 2018. (Cited on page 69.)

Other publications of the author

[9] **L. Kovács**, B. M. Bódis, and C. Benedek, "LidPose: Real-time 3d human pose estimation in sparse lidar point clouds with non-repetitive circular scanning pattern," in *15th Conference of the Hungarian Association for Image Analysis and Pattern Recognition*, Hévíz, 2025. (Cited on page 76.)

[10] **L. Kovács**, M. Kégl, and C. Benedek, "Real-time foreground segmentation for surveillance applications in sequences from a non-repetitive circular scanning lidar," in *14th Conference of the Hungarian Association for Image Analysis and Pattern Recognition*, Gyula, 2023. (Cited on page 76.)

[11] **L. Kovács**, B. Nagy, and C. Benedek, "Demonstration of changegan: change detection in unregistered point clouds using neural networks," in

13th Conference of the Hungarian Association for Image Analysis and Pattern Recognition, Budapest, 2021. (Cited on page 76.)

[12] **L. Kovács**, "Change detection in unregistered 3d point clouds," in *PhD proceedings, annual issues of the Doctoral School, Faculty of Information Technology and Bionics*, vol. 16, 2021, p. 67. (Cited on page 76.)

[13] **L. Kovács**, "Change detection in lidar point clouds," in *PhD proceedings, annual issues of the Doctoral School, Faculty of Information Technology and Bionics*, vol. 15, 2020, p. 68. (Cited on page 76.)

[14] **L. Kovács**, "Challenges in track to track sensor fusion using neural networks," in *PhD proceedings, annual issues of the Doctoral School, Faculty of Information Technology and Bionics*, vol. 14, 2019, p. 60. (Cited on page 76.)

[15] **L. Kovács**, "Challenges in sensor fusion," in *PhD proceedings, annual issues of the Doctoral School, Faculty of Information Technology and Bionics*, vol. 13, 2018, p. 55. (Cited on page 76.)

Bibliography

[16] F. Dyer and T. Martin, *Edison: His Life and Inventions*, ser. Edison: His Life and Inventions. Harper & Brothers, 1910, no. v. 2. [Online]. Available: https://books.google.hu/books?id=B7A4AAAAMAAJ&q=perspiration (Cited on page i.)

[17] C. Benedek, B. Gálai, B. Nagy, and Z. Jankó, "Lidar-based gait analysis and activity recognition in a 4d surveillance system," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 28, no. 1, pp. 101–113, 2018. (Cited on pages 2 and 30.)

[18] F. Oberti, L. Marcenaro, and C. S. Regazzoni, "Real-time change detection methods for video-surveillance systems with mobile camera," in *European Signal Processing Conference*, 2002, pp. 1–4. (Cited on page 2.)

[19] C. Benedek, X. Descombes, and J. Zerubia, "Building development monitoring in multitemporal remotely sensed image pairs with stochastic

birth-death dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 33–50, 2012. (Cited on page 2.)

[20] S. Ji, Y. Shen, M. Lu, and Y. Zhang, "Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples," *Remote Sensing*, vol. 11, no. 11, 2019. (Cited on page 2.)

[21] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1986–1992. (Cited on pages 2 and 3.)

[22] M. Cormier, A. Clepe, A. Specker, and J. Beyerer, "Where are we with human pose estimation in real-world surveillance?" in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2022, pp. 591–601. (Cited on page 2.)

[23] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004. (Cited on page 2.)

[24] A. Zanfir, M. Zanfir, A. Gorban, J. Ji, Y. Zhou, D. Anguelov, and C. Sminchisescu, "Hum3dil: Semi-supervised multi-modal 3d human pose estimation for autonomous driving," in *Proceedings of The 6th Conference on Robot Learning*, vol. 205, 2022, pp. 1114–1124. (Cited on pages 2 and 42.)

[25] N. Rossol, I. Cheng, and A. Basu, "A multisensor technique for gesture recognition through intelligent skeletal pose analysis," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 350–359, 2016. (Cited on page 2.)

[26] P. Sharma, B. B. Shah, and C. Prakash, "A pilot study on human pose estimation for sports analysis," in *Pattern Recognition and Data Analysis with Applications*, D. Gupta, R. S. Goswami, S. Banerjee, M. Tanveer, and R. B. Pachori, Eds. Singapore: Springer Nature Singapore, 2022, pp. 533–544. (Cited on page 2.)

[27] J. Chua, L.-Y. Ong, and M.-C. Leow, "Telehealth using posenet-based system for in-home rehabilitation," *Future Internet*, vol. 13, no. 7, 2021. (Cited on page 2.)

[28] E. V. Rabosh, N. S. Balbekin, A. M. Timoshenkova, T. V. Shlykova, and N. V. Petrov, "Analog-to-digital conversion of information archived in display holograms: Ii. photogrammetric digitization," *J. Opt. Soc. Am. A*, vol. 40, no. 4, pp. B57–B64, 2023. (Cited on page 2.)

[29] T. Nguyen, T. Qui, K. Xu, A. Cheok, S. Teo, Z. Zhou, A. Mallawaarachchi, S. Lee, W. Liu, H. Teo, L. Thang, Y. Li, and H. Kato, "Real-time 3d human capture system for mixed-reality art and entertainment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp. 706–721, 2005. (Cited on page 2.)

[30] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Real-time multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 01, pp. 172–186, 2021. (Cited on page 3.)

[31] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7157–7173, 2023. (Cited on page 3.)

[32] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, and W. Yang, "RTMO: Towards high-performance one-stage real-time multi-person pose estimation," 2023. (Cited on page 3.)

[33] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "Vitpose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 38 571–38 584. (Cited on pages 3, 43, 44, 45, 48, 49, 50, 51, 53, 60, 72, 74, 75, 98, and 100.)

[34] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, 2023. (Cited on pages 3 and 4.)

[35] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. (Cited on page 3.)

[36] T. Zhi, C. Lassner, T. Tung, C. Stoll, S. G. Narasimhan, and M. Vo, "Texmesh: Reconstructing detailed human texture and geometry from rgb-d video," 2020. (Cited on page 3.)

[37] K. Wang, J. Xie, G. Zhang, L. Liu, and J. Yang, "Sequential 3d human pose and shape estimation from point clouds," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7273–7282. (Cited on page 3.)

[38] M. Hassan, V. Choutas, D. Tzionas, and M. J. Black, "Resolving 3d human pose ambiguities with 3d scene constraints," 2019. (Cited on page 3.)

[39] B. M. Bódis, "3d pose estimation using sparse depth data," 2024, master's thesis at Pázmány Péter Catholic University. (Cited on pages 3 and 98.)

[40] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," 2016. (Cited on page 3.)

[41] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660. (Cited on page 4.)

[42] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," 2015. (Cited on page 4.)

[43] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. (Cited on page 4.)

[44] G. M. Difini, M. G. Martins, and J. L. V. Barbosa, "Human pose estimation for training assistance: a systematic literature review," in *Proceedings of the Brazilian Symposium on Multimedia and the Web*, ser. WebMedia '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 189–196. (Cited on page 4.)

[45] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt, "Probabilistic monocular 3d human pose estimation with normalizing flows," 2021. (Cited on page 4.)

[46] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social interaction capture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. (Cited on page 4.)

[47] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang, "Capturing human pose using mmwave radar," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6. (Cited on page 4.)

[48] S.-P. Lee, N. P. Kini, W.-H. Peng, C.-W. Ma, and J.-N. Hwang, "HuPR: A Benchmark for Human Pose Estimation Using Millimeter Wave Radar," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 5704–5713. (Cited on page 4.)

[49] C. Benedek, A. Majdik, B. Nagy, Z. Rozsa, and T. Sziranyi, "Positioning and perception in lidar point clouds," *Digital Signal Processing*, vol. 119, p. 103193, 2021. (Cited on page 5.)

[50] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "Cnn-based lidar point cloud de-noising in adverse weather," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2514–2521, 2020. (Cited on page 5.)

[51] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020. (Cited on page 5.)

[52] H. W. Yoo, N. Druml, D. Brunner, C. Schwarzl, T. Thurner, M. Hennecke, and G. Schitter, "Mems-based lidar for autonomous driving," *e & i Elektrotechnik und Informationstechnik*, vol. 135, no. 6, pp. 408–415, Oct 2018. (Cited on page 5.)

[53] F. Amzajerdian, V. E. Roback, A. Bulyshev, P. F. Brewster, and G. D. Hines, "Imaging flash lidar for autonomous safe landing and spacecraft proximity operation," 2016. (Cited on page 5.)

[54] A. Palffy, E. Pool, S. Baratam, J. Kooij, and D. Gavrila, "Multi-class road user detection with 3+1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, pp. 1–1, 2022. (Cited on page 6.)

[55] "Livox avia specifications," https://www.livoxtech.com/avia/specs, accessed: 2024-03-11. (Cited on pages 7, 52, and 76.)

[56] "Livox avia user manual," https://www.livoxtech.com/avia/downloads, accessed: 2024-03-11. (Cited on pages 7, 8, 9, and 98.)

[57] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3126–3131. (Cited on page 9.)

[58] Y. Wang, Y. Lou, Y. Zhang, W. Song, F. Huang, and Z. Tu, "A robust framework for simultaneous localization and mapping with multiple non-repetitive scanning lidars," *Remote Sensing*, vol. 13, no. 10, 2021. (Cited on page 9.)

[59] C. L. Glennie and P. J. Hartzell, "Accuracy assessment and calibration of low-cost autonomous sensors," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B1-2020, pp. 371–376, 2020. (Cited on page 9.)

[60] C.-C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Int. Conf. on Robotics and Automation (ICRA)*, vol. 3, 2002, pp. 2918–2924. (Cited on pages 10 and 13.)

[61] B. Nagy and C. Benedek, "Real-time point cloud alignment for vehicle localization in a high resolution 3d map," in *ECCV 2018 Workshops, LNCS*, 2019, pp. 226–239. (Cited on pages 10 and 11.)

[62] A. Varghese, J. Gubbi, A. Ramaswamy, and P. Balamuralidhar, "Changenet: A deep learning architecture for visual change detection," in *ECCV 2018 Workshops, LNCS*, 2019, pp. 129–145. (Cited on pages 11, 14, 21, 22, 23, 99, and 103.)

[63] Y. Wang, Q. Chen, Q. Zhu, L. Liu, C. Li, and D. Zheng, "A survey of mobile laser scanning applications and key techniques over urban areas," *Remote Sensing*, vol. 11, no. 13, pp. 1–20, 2019. (Cited on page 11.)

[64] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 166–178, 2016. (Cited on page 11.)

[65] P. Xiao, X. Zhang, D. Wang, M. Yuan, X. Feng, and M. Kelly, "Change detection of built-up land: A framework of combining pixel-based detection and object-based recognition," *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 402–414, 2016. (Cited on page 11.)

[66] W. Xiao, B. Vallet, M. Brédif, and N. Paparoditis, "Street environment change detection from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 107, pp. 38–49, 2015. (Cited on page 11.)

[67] R. Qin and A. Gruen, "3D change detection at street level using mobile laser scanning point clouds and terrestrial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, pp. 23–35, 2014. (Cited on page 11.)

[68] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Sackinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, p. 25, 1993. (Cited on pages 11, 15, and 70.)

[69] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change Detection Based on Deep Siamese Convolutional Network for Optical Aerial

Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1845–1849, 2017. (Cited on page 11.)

[70] E. Guo, X. Fu, J. Zhu, M. Deng, Y. Liu, Q. Zhu, and H. Li, "Learning to measure change: Fully convolutional siamese metric networks for scene change detection," 2018. (Cited on page 11.)

[71] Z. J. Yew and G. H. Lee, "City-scale scene change detection using point clouds," in *International Conference on Robotics and Automation (ICRA)*, 2021. (Cited on page 12.)

[72] R. Qin, J. Tian, and P. Reinartz, "3D change detection–Approaches and applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 122, pp. 41–56, 2016. (Cited on page 12.)

[73] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *Advances in Neural Information Processing Systems (NIPS)*, 2015. (Cited on pages 12, 16, and 70.)

[74] B. Gálai and C. Benedek, "Change detection in urban streets by a real time Lidar scanner and MLS reference data," in *Int. Conf. Image Analysis and Recognition, LNCS*, 2017, pp. 210–220. (Cited on pages 13, 21, 22, 23, 99, and 103.)

[75] C. Benedek, "3d people surveillance on range data sequences of a rotating lidar," *Pattern Recognition Letters*, vol. 50, pp. 149–158, 2014. (Cited on pages 13, 27, and 46.)

[76] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Medical Image Computing and Comp.-Ass. Intervention*, 2015, pp. 234–241. (Cited on pages 16 and 70.)

[77] A. Börcs, B. Nagy, and C. Benedek, "Fast 3-D urban object detection on streaming point clouds," in *ECCV 2015 Workshops, LNCS*, 2015, pp. 628–639. (Cited on page 19.)

[78] B. Nagy and C. Benedek, "On-the-fly camera and lidar calibration," *Remote Sensing*, vol. 12, no. 7, 2020. (Cited on page 19.)

[79] B. Nagy and C. Benedek, "3d cnn-based semantic labeling approach for mobile laser scanning data," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10 034–10 045, 2019. (Cited on page 19.)

[80] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, 1978. (Cited on page 22.)

[81] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," in *NIPS 2016 Workshops on Adversarial Training*, 2016. (Cited on page 24.)

[82] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231 – 1237, 2013. (Cited on page 26.)

[83] J. Schauer and A. Nüchter, "Removing non-static objects from 3d laser scan data," *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 15–38, 2018. (Cited on page 27.)

[84] A. Börcs, B. Nagy, and C. Benedek, "Instant object detection in lidar point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 7, pp. 992–996, 2017. (Cited on page 29.)

[85] I. Schiller and R. Koch, "Improved video segmentation by adaptive combination of depth keying and Mixture-of-Gaussians," in *Proc. Scandinavian Conference on Image Analysis, Ystad, Sweden*, ser. LNCS, vol. 6688. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 59–68. (Cited on page 29.)

[86] R. Kaestner, N. Engelhard, R. Triebel, and R.Siegwart, "A Bayesian approach to learning 3D representations of dynamic environments," in *Proc. International Symposium on Experimental Robotics (ISER)*. Berlin: Springer Press, 2010. (Cited on pages 29, 33, and 34.)

[87] C. Benedek, D. Molnár, and T. Szirányi, "A dynamic MRF model for foreground detection on range data sequences of rotating multi-beam lidar," in *Advances in Depth Image Analysis and Applications*. Springer Berlin Heidelberg, 2013, pp. 87–96. (Cited on pages 29 and 33.)

[88] Y. Alkhalili, M. Luthra, A. Rizk, and B. Koldehofe, "3-d urban objects detection and classification from point clouds," in *13th ACM International Conference on Distributed and Event-Based Systems*, ser. DEBS '19, New York, NY, USA, 2019, p. 209–213. (Cited on page 30.)

[89] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 747–757, 2000. (Cited on pages 32, 33, and 34.)

[90] C. Benedek and T. Sziranyi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 608–621, 2008. (Cited on page 33.)

[91] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, 1978. (Cited on page 39.)

[92] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. (Cited on pages 42, 43, and 44.)

[93] J. Zheng, X. Shi, A. Gorban, J. Mao, Y. Song, C. R. Qi, T. Liu, V. Chari, A. Cornman, Y. Zhou, C. Li, and D. Anguelov, "Multi-modal 3d human pose estimation with 2d weak supervision in autonomous driving," 2021. (Cited on pages 42 and 66.)

[94] K. Wang, J. Xie, G. Zhang, L. Liu, and J. Yang, "Sequential 3d human pose and shape estimation from point clouds," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7273–7282. (Cited on page 43.)

[95] Y. Ren, X. Han, C. Zhao, J. Wang, L. Xu, J. Yu, and Y. Ma, "Livehps: Lidar-based scene-level human pose and shape estimation in free environment," 2024. (Cited on page 43.)

[96] Y. Ren, C. Zhao, Y. He, P. Cong, H. Liang, J. Yu, L. Xu, and Y. Ma, "Lidar-aid inertial poser: Large-scale human motion capture by sparse inertial and lidar sensors," *IEEE Transactions on Visualization and*

Computer Graphics, vol. 29, no. 5, pp. 2337–2347, 2023. (Cited on page 43.)

[97] Y. Zhou, H. Dong, and A. E. Saddik, "Learning to estimate 3d human pose from point cloud," *IEEE Sensors Journal*, vol. 20, no. 20, pp. 12 334–12 342, 2020. (Cited on page 43.)

[98] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. (Cited on page 43.)

[99] D. Ye, Y. Xie, W. Chen, Z. Zhou, L. Ge, and H. Foroosh, "Lpformer: Lidar pose estimation transformer with multi-task network," 2024. (Cited on page 43.)

[100] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. (Cited on page 43.)

[101] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 6000–6010. (Cited on page 43.)

[102] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020. (Cited on page 43.)

[103] N. J. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International Conference on Machine Learning (ICML)*, 2018. (Cited on page 43.)

[104] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "Styleswin: Transformer-based gan for high-resolution image generation," 2022. (Cited on page 43.)

[105] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "Maskgit: Masked generative image transformer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. (Cited on page 43.)

[106] L. Stoffl, M. Vidal, and A. Mathis, "End-to-end trainable multi-instance pose estimation with transformers," *CoRR*, 2021. (Cited on page 43.)

[107] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660. (Cited on page 43.)

[108] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, pp. 472–487. (Cited on page 43.)

[109] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5686–5696. (Cited on page 43.)

[110] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, 2014, pp. 740–755. (Cited on pages 44 and 53.)

[111] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. (Cited on page 51.)

[112] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. (Cited on page 52.)

[113] *The OpenCV Reference Manual*, 4th ed., OpenCV, 2014. (Cited on page 52.)

[114] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *CoRR*, 2021. (Cited on page 52.)

[115] J. C. Eidson, M. Fischer, and J. White, "Ieee-1588™ standard for a precision clock synchronization protocol for networked measurement and control systems," in *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, 2002, pp. 243–254. (Cited on page 52.)

[116] K. Lao and G. Yan, "Implementation and analysis of ieee 1588 ptp daemon based on embedded system," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 4377–4382. (Cited on page 52.)

[117] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," https://github.com/ultralytics/ultralytics, 2023. (Cited on pages 53 and 73.)

[118] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "Vitpose: Simple vision transformer baselines for human pose estimation," https://github.com/ViTAE-Transformer/ViTPose, 2022. (Cited on page 53.)

[119] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (Cited on page 58.)

[120] Z. Wu, D. Hoang, S.-Y. Lin, Y. Xie, L. Chen, Y.-Y. Lin, Z. Wang, and W. Fan, "Mm-hand: 3d-aware multi-modal guided hand generative network for 3d hand pose synthesis," 2020. (Cited on page 59.)

[121] T. L. Munea, Y. Z. Jembre, H. T. Weldegebriel, L. Chen, C. Huang, and C. Yang, "The progress of human pose estimation: A survey and taxonomy of models applied in 2d human pose estimation," *IEEE Access*, vol. 8, pp. 133 330–133 348, 2020. (Cited on page 59.)

[122] M. Héder, E. Rigó, D. Medgyesi, R. Lovas, S. Tenczer, F. Török, A. Farkas, M. Emődi, J. Kadlecsik, G. Mező, Á. Pintér, and P. Kacsuk, "The past, present and future of the ELKH cloud," *Információs Társadalom*, vol. 22, no. 2, p. 128, 2022. (Cited on page 77.)

Appendix A

Supplementary materials

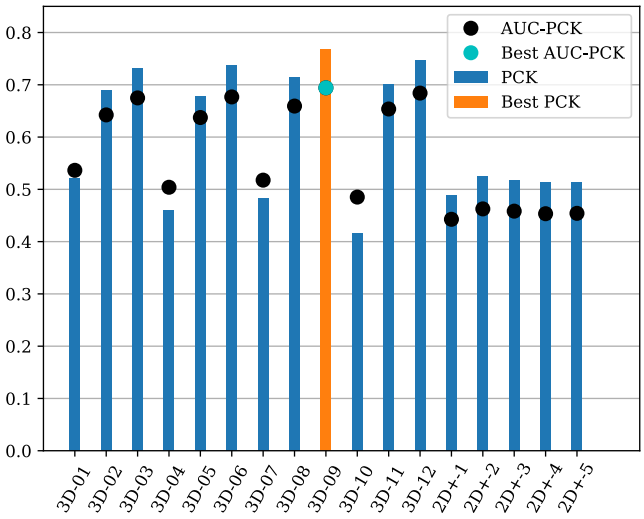


Figure A.1. PCK and AUC-PCK (both introduced in Section 4.4.1) values of the 3D predictions by LidPose–3D and LidPose–2D+ networks evaluated in 3D space with 3D metrics. The *AUC-PCK* was calculated on the $[0, 0.5]$ meter interval, as shown in Figure 4.12

Table A.1. Results of the *LidPose–3D* and *LidPose–2D+* networks with different input types and depth losses, evaluated in 3D space with 3D metrics.
The meaning of the *Input* values:
`D`: Lidar distance; `XYZ`: point 3D coordinates
`I`: Lidar intensity
Depth L. refers to the criterion used to calculate the depth loss during learning. *2D+* models do not have this parameter.
Percentage of Correct Keypoints (*PCK*) was calculated with the error being at most 0.2 meters. The *AUC-PCK* was calculated on the $[0, 0.5]$ meter interval, as shown in Figure 4.12

Model	Input	Depth L.	ADE↓	PCK↑	AUC-PCK↑	LAE↓	LLE↓
3D–01	`XYZ`	L1	0.3372	0.5222	0.5364	22.5130	0.9247
3D–02	`XYZ`	L2	0.1848	0.6904	0.6424	21.9994	0.0966
3D–03	`XYZ`	SSIM	0.1683	0.7322	0.6749	20.7884	**0.0903**
3D–04	`XYZ+D`	L1	0.2679	0.4599	0.5040	24.8060	0.1868
3D–05	`XYZ+D`	L2	0.1873	0.6784	0.6374	22.3703	0.0964
3D–06	`XYZ+D`	SSIM	0.1676	0.7374	0.6768	**20.6084**	0.0908
3D–07	`XYZ+I`	L1	0.2576	0.4822	0.5176	25.0920	0.1769
3D–08	`XYZ+I`	L2	0.1762	0.7147	0.6593	21.6107	0.1047
3D–09	`XYZ+I`	SSIM	**0.1583**	**0.7678**	**0.6942**	20.6737	0.0952
3D–10	`XYZ+D+I`	L1	0.2764	0.4164	0.4852	31.0437	0.2274
3D–11	`XYZ+D+I`	L2	0.1794	0.7014	0.6537	21.9404	0.1064
3D–12	`XYZ+D+I`	SSIM	0.1633	0.7466	0.6841	21.1505	0.0983
2D+–1	`D`	-	**2.4477**	0.4887	0.4427	**32.4529**	1.4299
2D+–2	`XYZ`	-	2.4758	**0.5242**	**0.4626**	32.5635	1.4419
2D+–3	`XYZ+D`	-	2.4910	0.5165	0.4583	33.3569	1.4723
2D+–4	`XYZ+I`	-	2.5901	0.5141	0.4534	34.0922	1.5538
2D+–5	`XYZ+D+I`	-	2.5671	0.5133	0.4541	33.5011	**1.3705**

Table A.2. Mean Per-Joint Position Error (described in Section 4.4.1) results of the LidPose–3D networks for different joint types.

Model	head	shoulders	elbows	wrists	hips	knees	ankles	ADE↓
3D–01	0.1693	0.2252	1.1299	0.3677	0.1741	0.2045	0.3413	0.3372
3D–02	0.1368	0.1537	0.2038	0.2252	0.1374	0.1768	0.3320	0.1848
3D–03	0.1375	0.1404	0.1694	0.1933	0.1368	0.1611	0.2860	0.1683
3D–04	0.1817	0.2512	0.3898	0.3914	0.2009	0.2334	0.3560	0.2679
3D–05	0.1410	0.1576	0.2063	0.2297	0.1386	0.1767	0.3305	0.1873
3D–06	0.1386	0.1391	0.1654	0.1898	0.1347	0.1591	0.2899	0.1676
3D–07	0.1698	0.2214	0.3676	0.4060	0.1999	0.2239	0.3466	0.2576
3D–08	0.1323	0.1442	0.1762	0.2040	0.1369	0.1705	0.3349	0.1762
3D–09	**0.1290**	**0.1272**	**0.1509**	**0.1734**	**0.1303**	**0.1585**	**0.2827**	**0.1583**
3D–10	0.1853	0.2282	0.3865	0.4642	0.2308	0.2387	0.3372	0.2764
3D–11	0.1332	0.1486	0.1819	0.2085	0.1430	0.1771	0.3330	0.1794
3D–12	0.1309	0.1347	0.1568	0.1813	0.1399	0.1625	0.2855	0.1633
2D+–1	**2.4256**	1.6951	2.2149	3.2206	**1.9731**	2.4557	3.1819	**2.4477**
2D+–2	2.5820	**1.6506**	**2.1371**	3.1362	2.0028	2.4956	3.1668	2.4758
2D+–3	2.5793	1.6879	2.2053	3.1573	1.9953	2.4664	3.2130	2.4910
2D+–4	2.8987	1.7478	2.2795	3.1611	2.0405	2.4757	**3.0646**	2.5901
2D+–5	2.9546	1.7358	2.1397	**3.0070**	2.0248	**2.4215**	3.1053	2.5671

Appendix B

List of Abbreviations

ADE	Average Distance Error
AUC	Area Under Curve
COCO	Microsoft COCO: Common Objects in Context dataset
FoV	Field of View
FPS	Frames Per Second
GAN	Generative Adversarial Network
GT	Ground Truth
IoU	Intersection over Union
IMU	Inertial Measurement Unit
HUN-REN	Hungarian Research Network
LAE	Limb Angle Error
LLE	Limb Length Error
MEMS	Micro-electromechanical system
MHSA	Multi-Head Self-Attention
MRF	Markov Random Fields
MoG	Mixture of Gaussians
MPJPE	Mean Per-Joint Position Error
MSE	Mean Squared Error
NIR	near-infrared
NRCS	Non-repetitive Circular Scanning

PCK	Percentage of Correct Keypoints
PTPd	Precision Time Protocol daemon
ReLU	Rectified Linear Unit
ROI	Region of Interest
RMB	Rotating multi-beam
SLAM	Simultaneous Localization and Mapping
SSIM	Structural Similarity Index Measure
STN	Spatial Transformer Network
ToF	Time-of-Flight
ViT	Vision Transformer

List of Figures

LIST OF FIGURES

List of Tables