Modeling, Identification and Control, Vol. 42, No. 2, 2021, pp. 37–46, ISSN 1890–1328

CNN-based People Detection in Voxel Space using Intensity Measurements and Point Cluster Flattening

J. Dybedal [1] G. Hovland [1]

[1] *Department of Engineering Sciences, University of Agder, N-4898 Grimstad, Norway*

Abstract

In this paper real-time people detection is demonstrated in a relatively large indoor industrial robot cell as well as in an outdoor environment. Six depth sensors mounted at the ceiling are used to generate a merged point cloud of the cell. The merged point cloud is segmented into clusters and flattened into gray-scale 2D images in the xy and xz planes. These images are then used as input to a classifier based on convolutional neural networks (CNNs). The final output is the 3D position (x, y, z) and bounding box representing the human. The system is able to detect and track multiple humans in real-time, both indoors and outdoors. The positional accuracy of the proposed method has been verified against several ground truth positions, and was found to be within the point-cloud voxel-size used, i.e. 0.04 m. Tests on outdoor datasets yielded a detection recall of 76.9 % and an F1 score of 0.87.

Keywords: Human detection, indoor, outdoor, point clouds, voxels, flattening, CNN

1 Introduction

The ability to detect the presence of people or other objects in three dimensional data is an important factor in enabling autonomy and automation in environments where machinery and humans are both present. A natural example is in the automotive industry, where autonomous vehicles must be able to accurately perceive their surroundings. Another example is in any industrial environment where robotic machinery must coexist with personnel, whether it is on a large offshore platform or in a small indoor robotic cell.

Several approaches for people detection exist, where the problem of detecting people and other objects in 2D images is well documented in the academic literature, especially methods composing different types of machine learning such as Histograms of Oriented Gradients (HOG) Dalal and Triggs (2005) and convolutional neural networks (CNNs). The problem of detecting people in 3D space, e.g. in point clouds, is also a hot topic. Much research has been done on methods for detection in data from 2D and 3D lidars typically found on autonomous vehicles and mobile robots. In Borgmann et al. (2017), an approach based on implicit shape models (ISM) was used to detect people in a 3D lidar scan, and in Kim et al. (2020) person detection and tracking was done using a laser scan of the person's hip area and a Sample-based Joint Probabilistic Data Association Filter (SJPDAF). A 3D lidar was also used in Spinello et al. (2011), where a bottom-up, top-down detector was used to select hypothetical candidates and perform validation on a tessellated volume, respectively.

For RGB-D images, the authors of Linder and Arras (2015) used a tessellation boosting approach for human feature classification, e.g. different types of clothes and hairstyles. This approach, combined with the feature-based detection method from Lewandowski et al. (2019), was used in Wengefeld et al. (2019) to estimate human poses based on performant features

doi:10.4173/mic.2021.2.1

extracted from colored point clouds. The detection method used a layer-based approach to calculate feature descriptors for each layer in a point cluster and concatenated the histograms to form feature vectors. In Linder and Arras (2016) both 2D laser scans and RGB-D images were used for human detection and tracking, including keeping track of multiple humans in groups.

Human pose estimation has also been heavily researched and CNN-based methods such as OpenPose Cao et al. (2017) for 2D and Tome and Russell (2017) for 3D pose estimations from 2D images. In Zimmermann et al. (2018), RGB-D data was used for 3D pose estimation, using depth information in addition to the color image. Vehicle detection in lidar point clouds using neural networks such as VoxelNet Zhou and Tuzel (2017), 3D YOLO Hakim (2018) and Simon et al. (2019) also show promising results.

A common denominator of the methods mentioned above is that they are either tailor-made for lidar scans or that they depend on RGB images as well as any depth data. While the availability of RGB images is a justifiable assumption, there are scenarios in which redundancy is crucial, such as in the red-zone of oil-rigs. Such areas would typically be monitored by both RGB and depth-sensors (see Velodyne Lidar (2020)), and while combining the measurements could yield the best detection results, in a scenario where the RGB information becomes unavailable a fall-back solution based on depth-information only should be available. In this paper, we therefore aim to remove the need for using RGB images when detecting people.

In addition, while most of the literature concentrates on single mobile sensors mounted on vehicles or robots, the method proposed in this paper utilizes a point cloud generated by several statically mounted 3D sensors as described by Dybedal et al. (2019), which is a just as likely scenario in industrial environments.

In Simon et al. (2018) Complex-YOLO is introduced, which is an extension of YOLOv2, a fast 2D standard object detector for RGB images, by a specific complex regression strategy to estimate multi-class 3D boxes in Cartesian space. A specific Euler-Region-Proposal Network (E-RPN) is proposed to estimate the pose of the object by adding an imaginary and a real fraction to the regression network. The result is a closed complex space which avoids singularities, which can occur by single angle estimations. In our work multiple depth sensors are used and the data is merged into a single point cloud used for detection. Hence, the single angle problem mentioned in Simon et al. (2018) is not a problem in the work presented here.

In Munaro et al. (2016a) OpenPTrack is presented, which is an open source software for multi-camera calibration and people tracking in RGB-D camera networks. People detection is executed locally, in the machines connected to each sensor, while tracking is performed by a single node which takes into account detections from all over the network. For Kinect v1 and stereo cameras, which can produce color images, the HOG technique for people detection is applied to these images in correspondence of the clusters extracted from the point cloud. For the Kinect v2 the infrared images are used, since they are invariant to visible lighting.

In Munaro et al. (2016b) the proposed algorithms have been demonstrated further using aligned color and depth data in industrial environments. The algorithms have been released as open source as part of the ROS-Industrial project. The work presented in this paper is different from Munaro et al. (2016a,b) in that a CNN-based approached is used and that the people detection is performed on the merged point cloud on the central node using only depth measurements, as opposed to detection locally on each node using color or infrared images in addition to depth information. The authors believe that people detection and tracking on a central node will be more robust, since the central node has access to the full point cloud, merged from all the sensors in real-time, however this has not been demonstrated experimentally in this work.

In Hui et al. (2019) an object detection method based on 3D information extraction of laser point clouds is proposed. Similar to the approach taken in this paper, in Hui et al. (2019) the point cloud is flattened to 2D images which are used as a basis for learning using the AdaBoost algorithm.

In Tang et al. (2017) the proposed method maps the three-dimensional point cloud to the two-dimensional plane by a distance-aware expansion approach. The corresponding 2D contour and its associated 2D features are then extracted. A radial basis function (RBF) kernel support vector machine (SVM) is employed with the extracted features for classification. A selective binary and Gaussian filtering regularized level set (SBGFRLS) algorithm is utilized for contour detection in the 2D images. In addition, other popular feature descriptors from the projected 2D images, such as HOG, LBP and Haar-like features are extracted.

Yan et al. (2020) present recent work on human detection and a classification scheme based on 3D Lidar data and an algorithm using a standard Support Vector Machine (SVM). The 3D indoor data used in Yan et al. (2020) has been made publicly available. In the conclusions the authors write: *Future work should look at other classification methods such as deep neural networks.* The suggestion to use a CNN-approach is addressed in this paper, and the method developed here should fit such a system well, as it is designed to classify

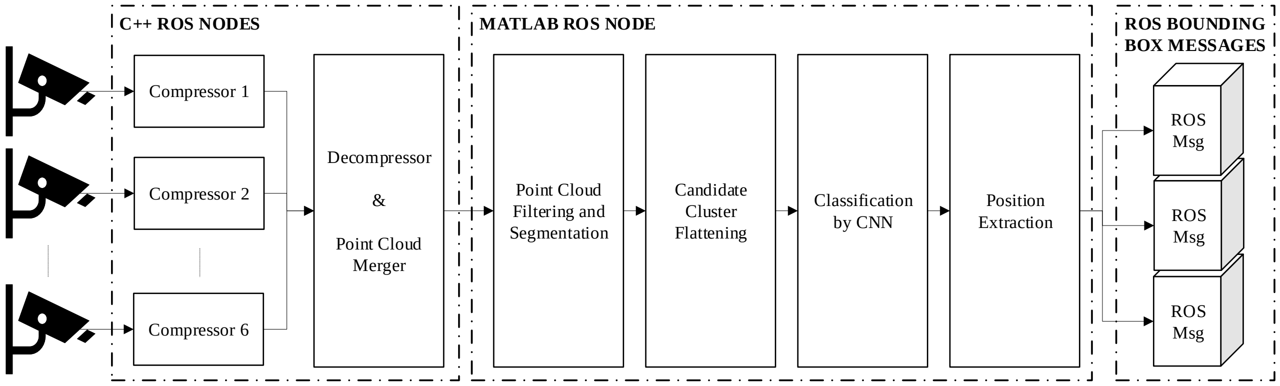


Figure 1: Flowchart of the people detection system. The point clouds from each sensor is compressed by an edge computer, and received by a centralized computer before being decompressed and processed.

humans in point clouds containing intensity measurements, similar to the ones generated by a Lidar-based system.

This paper is organized as follows: Section 2 presents the overall methodology and the different modules that were developed. Section 3 contains several experimental results, while discussion and conclusions are given in Section 4.

2 Methodology

The main inspiration for our work was the ‘3D YOLO’ type detectors where laser point clouds were used as inputs to image classifiers. To tackle the problem of human detection using only depth information, a scheme based on scene classification using convolutional neural networks (CNNs) was developed. To generate images for classification, a point cloud flattening approach was applied.

The input to the detector is a stream of compressed, voxelised point clouds, as described in Dybedal et al. (2019). These streams are published as ROS (Robot Operating System) topics. Six Microsoft Kinect V2 3D sensors were used to generate the depth measurements, corresponding to six point cloud streams. Although the Microsoft Kinect sensors can supply RGB images, the purpose of this study was to become independent from RGB sensors, thus only the depth measurements were used.

The methodology is demonstrated in this paper by real-time multiple people detection in both indoor and outdoor environments.

Figure 1 shows the structure of the developed system and the different steps will be further outlined in the following sections. The source code is available at Dybedal (2021a).

2.1 Point Cloud Pre-processing

Each of the six Kinect sensor nodes publishes compressed point cloud streams at up to 20 Hz. As described in Dybedal et al. (2019), the points in the compressed point clouds contain no colors, but an additional intensity value corresponding to the amount of measured points inside a single voxel, compensated for the distance to the sensor. The purpose of this addition was to add a measure of strength or confidence to each voxel as the point clouds were compressed, while accounting for the fact that objects close to the sensor will have a much higher point density than similar objects further away.

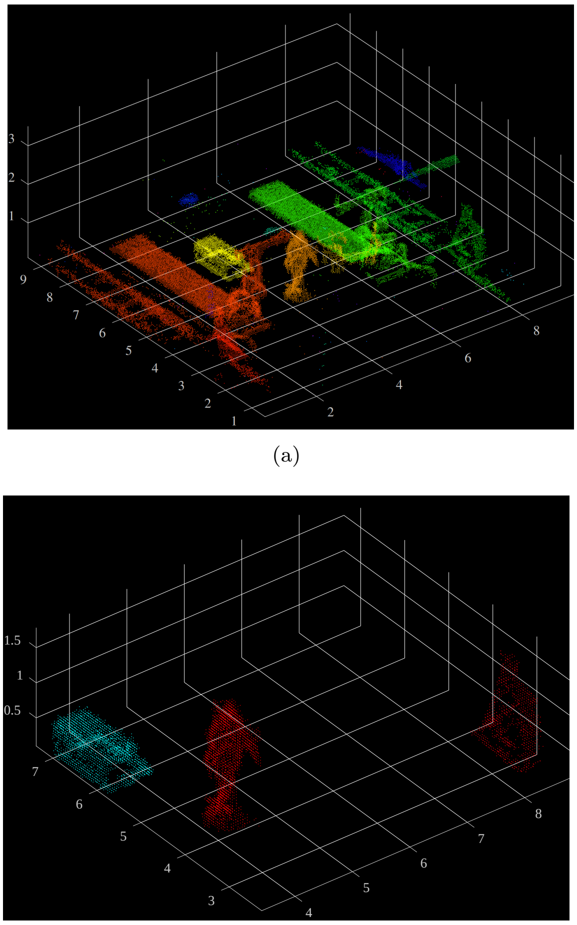
It should be noted that in addition to not including colors, the point clouds used in this paper are heavily compressed and down-sampled by voxelization. When using a voxel size of 4 cm × 4 cm × 4 cm and cropping to the volume of interest, Dybedal et al. (2019) found that each point cloud was typically reduced from 217 088 points to $17\,771 \pm 118$, with a reduction in required storage space of $1 : 40.5 \pm 0.5$.

In this paper, a ROS node was created which receives, synchronizes and merges the point clouds into a single cloud. The sensor nodes are time synchronized against a server, which ensures that the different point clouds are as close to each other as possible in the temporal space. Calibration to ensure optimized transfer functions between the sensors and the global coordinate system was performed according to the method described in Aalerud et al. (2019). When merging the point clouds, the intensity values are accumulated such that the points in the merged point cloud contain the sum of all intensity values corresponding to the same voxel. The result and output of the developed ROS node is thus a single voxelised point cloud stream, where each point contains an intensity value in addition to x, y and z coordinates.

When the single, merged point cloud was obtained, it was segmented into clusters based on a minimum Euclidean distance between points in the clusters. However, most of the resulting clusters could be discarded, and only clusters defined to be a human candidate were used further in the detection scheme. To select the candidate clusters, the following constraints based on normal human poses were applied:

- Min, Max height (z dimension): 0.5 m and 2.0 m
- Min, Max width (x,y dimensions): 0.2 m and 2.0 m
- Max. distance from floor: 0.2 m

As seen in Figure 2, this filtering resulted in a set of candidate clusters, which were used to generate grayscale images for CNN-based training and classification.



(a)

(b)

Figure 2: (**a**) Candidate clusters after segmentation, but before dimension constraints; (**b**) Candidate clusters after dimension constraints have been applied.

2.2 Point Cluster Flattening

Several methods for calculating the positions of humans in point clouds were considered, including slicing the entire point cloud into segments in the xz and yz planes and classify/detect humans in each slice. However, due to the fact that humans would normally be present in only a fraction of the points, this was deemed too computationally expensive. In addition to the inevitable trade-off between accuracy (slice thickness) and speed, the same human, or parts of it, could be detected in multiple slices. Hence, the method developed in this paper was to flatten each candidate point cluster in the xz and yz planes, resulting in just two small slices per candidate.

As the input point cloud has already passed through a compressor based on a voxel grid, where each point has a fixed coordinate in the voxel center, and where the voxels are aligned to the global coordinate system, the process of cluster flattening could be implemented by iterating over the voxels in each dimension. Eq. 1 shows the process of generating one pixel in the xz plane.

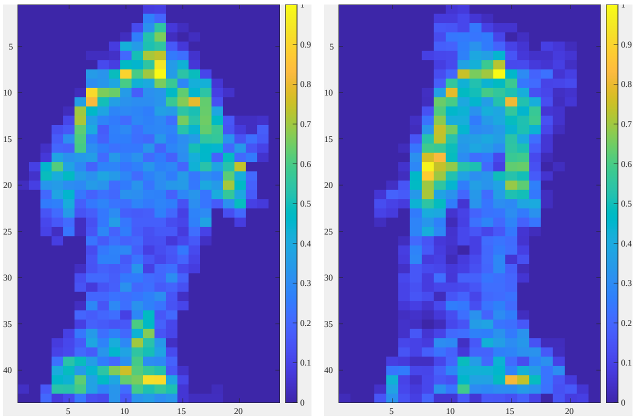
$$P(i,j) = \sum_{y=y_{min}}^{y_{max}} I(x_i, y, z_j), \tag{1}$$

where $P(i,j)$ is one pixel in the flattened image and $I(x_i, y, z_j)$ is the intensity value for the voxel at (x_i, y, z_j). The result is two gray-scaled images, where each pixel corresponds to the sum of the intensity values of the flattened points, as seen in Figure 3. Using the intensity values to create a gray-scale image instead of just counting points results in images with a larger dynamic range. As the human body is solid, points can only exist on the exterior, which greatly limits the amount of points that would result in a single pixel in the flattened image.

Due to the coarse resolution of the voxelised point cloud, a human is typically represented by an image of only 25 times 45 pixels when a voxel grid resolution of 4 cm is used.

2.3 Scene Classification

As the images are already cropped to the candidate point cluster, and the positions in the global coordinate system are known, there is no need for additional segmentation or bounding boxes. The classification of humans in the generated images could therefore be performed as a scene classification, i.e. the whole image was classified as a single class. As a proof of concept, a simple convolutional neural network was trained using images generated from captured datasets. The images corresponding to each candidate cluster in the training datasets were manually labeled as either 'Human' or



(a) (b)

Figure 3: (**a**) Candidate cluster after flattening in the y dimension (xz plane); (**b**) Candidate cluster after flattening in the x dimension (yz plane). Color represents the accumulated intensity values, where yellow is higher.

Layer	**Parameter**	**Value**
Input Layer	Size	$224 \times 224 \times 3$
Convolution Layer	Filter Size No. of filters Padding	3 16 1
Batch Norm. Layer		
ReLu Layer		
Max-pooling layer	Pool Size Stride	2 2
Convolution Layer	Filter Size No. of filters Padding	3 32 1
Batch Norm. Layer		
ReLu Layer		
Max-pooling layer	No. of pools Stride	2 2
Convolution Layer	Filter Size No. of filters Padding	3 64 1
Batch Norm. Layer		
ReLu Layer		
Fully-connected layer	Output Size	2
Softmax Layer		
2-class Classifier		

Table 1: CNN Structure and layer parameters.

’Not Human’. A total of 1105 and 1956 images were used for the ’Human’ and ’Not Human’ classes, respectively. The two classes were chosen as the purpose here was to detect the presence of humans in a robotic environment. It would be possible to extend the number of classes to be able to classify any other object, such as the robots, other machines, equipment, etc.

The labeled images were then randomly binned into training images (60 %), validation images (30 %) and testing images (10 %). In addition, the images were augmented with random rotation ($\pm 90deg$), reflection (around the z axis) and shear.

The structure of the neural network, including three convolution blocks, and the parameters used is shown in Table 1. The structure was inspired by a simple example by The MathWorks, Inc (2021), but tuned to maximize the validation accuracy. The input layer is three-dimensional (RGB) with an input size of 224 times 224 pixels. Thus, in addition to resizing, the images needed to be augmented such that the gray-scale content was replicated across all RGB channels. As many image classifiers, including pre-trained networks, are expecting color images, converting from gray-scale to color allows for a modular application where the classifier could be replaced by another without adding extra complexity. In contrast to many other scenarios, the images to be classified are very small, as described in the previous section. As a consequence, most images would need to be scaled up, not down, before being evaluated by the classifier.

Training of the network was performed in Matlab by the stochastic gradient descent with momentum (SGDM) optimizer. Using 100 epochs and an initial learning rate of 0.00001, the final validation accuracy was 95.75%.

2.4 Labeling and Position Extraction

The nature of the detection mechanism developed in this paper includes a coarse confidence measure, i.e. a human can be classified in either zero, one, or both of the flattened images. For the testing described in this paper, only the scenarios where both images are classified as humans were considered a detection.

In the event of a detection, the developed ROS program publishes a “marker” message that includes the position and extent of the detected human. The position used is the center of the bounding box surrounding the candidate point cluster, and the bounding box itself is used to describe the area where the person was detected.

3 Experimental Results

Four different test cases were used to evaluate the system. First, the scheme was tested in the Industrial

Robotics Lab (IRL) at the University of Agder Dybedal et al. (2019). In this environment, six 3D sensor nodes consisting of Microsoft Kinect V2s and NVIDIA Jetson TX2s are mounted on the walls around a robotic cell. Inside the area monitored by the sensors, there are three ABB robots, where two are track mounted and one is mounted on a Gdel gantry crane. The test served as a proof-of-concept, verifying that the detector worked on live data different from the datasets used for training and validation.

Second, the accuracy of the system was measured by comparing detected coordinates with ground truth coordinates measured by a Leica Laser Tracker. The same setup as in the first experiment was used.

Third, the scheme was tested using a datasets recorded at an outdoors test facility for offshore pipe handling equipment, without altering the algorithms. The datasets were recorded in August 2018, using the same sensor nodes as in the IRL lab, and includes recordings with multiple people in different weather conditions.

Lastly, the CNN was re-trained using outdoor data, and a detection capability test was performed, measuring possible detections versus actual detections.

All experiments were performed on a stationary computer running the detector in Matlab. The computer was running Ubuntu 16.04 with ROS Kinetic and contained an Intel Core i7 7820X 3,6 GHz processor, an NVIDIA GTX 1080Ti GPU and 32 GB of RAM.

The following subsections present the results from the three different test cases. The outdoor dataset is made publicly available at Dybedal (2021b).

3.1 Indoor Single Person Detection

After the detector had been trained, it was tested on a single person in the IRL lab. The aim was to verify that the detector would yield good results on data different to the training data.

Figure 4 shows the detected bounding boxes around a person in different poses. In (a), the detected position coordinates were x =4.42 m, y =5.22 m, z =0.94 m, where this point is the center of the displayed bounding box. The detection algorithm functioned as expected, marking the person with correctly sized bounding boxes for all the tested poses.

In addition, a small set of 160 'Human' images and 70 'Not Human', where the subject had different outfits than used in the training, was used to test the CNN. The accuracy in this test was 88.2%, which as expected is a little lower than the training accuracy, but still close to 90%.

3.2 Indoor Accuracy Validation

To verify the accuracy of the detector, the output X, Y coordinates were compared to coordinates that had been previously measured to sub-millimeter accuracy by a Leica laser tracker and marked on the floor. A person would walk around the monitored area, stopping at the marked coordinates, before moving on to the next. Performing the test in such a way would introduce some inaccuracy, as it is not possible to guarantee that the person is standing exactly above the ground truth coordinate. However, since the resolution of the input point cloud was as coarse as 0.04 m, the test would still yield useful results.

A total of 21 coordinates were pre-measured. During the test, a total of 94 detections were performed at the ground truth positions, and a subset is shown in Table 2. The result is shown in Table 3, and it can be seen that the mean absolute deviation of the detector was within the point cloud resolution of 0.04 m.

X	Y	X meas.	Y meas.	Z meas.
5.0	8.0	4.96	8.06	0.94
5.0	7.0	5.02	7.02	0.94
5.0	6.0	4.96	6.02	0.92
5.0	5.0	5.02	4.98	0.94
4.69	5.3	4.72	5.23	0.92
4.0	3.0	4.02	2.98	0.94
2.0	7.0	5.02	7.02	0.94
5.42	9.24	5.5	9.2	0.94

Table 2: A subset of results obtained during the accuracy validation.

	X	Y	Total
Mean absolute deviation (m)	0.035	0.039	0.037
Std.dev absolute deviation (m)	0.041	0.035	0.038
Mean deviation (m)	−0.001	−0.004	−0.003
Std.dev deviation (m)	0.054	0.053	0.053

Table 3: Results from accuracy test using 94 detections compared to ground truth coordinates.

The X and Y coordinates are the ones that have been used for verification, as these are the only ones with an accurate ground truth. However, the bounding box also contains the width, depth and height of the detected person. While the width and depth can vary greatly due to different poses, the measured height was compared to the height of the test subject. Using 56 of the detections performed while the subject was standing, the measured height was (1.76 ± 0.06) m, and the real height of the test subject was 1.75 m.

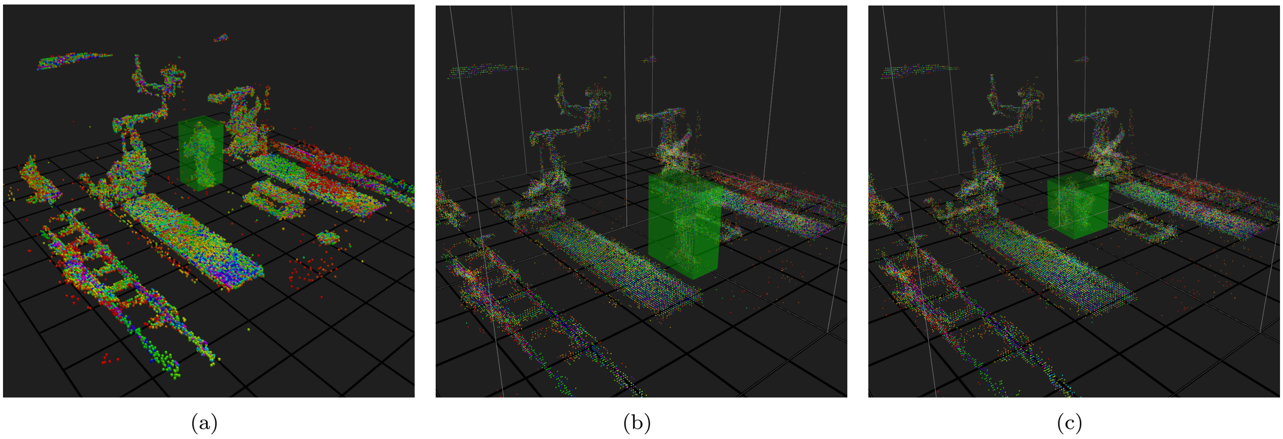


Figure 4: (**a**) Example detection, walking; (**b**) Example detection, arms out; (**c**) Example detection, crouching.

3.3 Testing on Outdoor Datasets

To test the detection algorithm on more challenging data, datasets recorded on an outdoors facility were used. The datasets were recorded in August 2018, using the same sensors as used for the two first experiments, and contains a variety of weather conditions and multiple persons. Due to the lack of ARUCO codes used for the automatic calibration performed in the indoor experiments, the sensors' translation and rotation were calibrated manually using other features in the point clouds.

Figures 5 (a) and (b) show a snapshot of the first test, where four persons are present. As seen in Figure 5 (a), the weather conditions were challenging, with a low sun shining on wet concrete. The persons were all wearing different outfits, in addition to helmets, which were not part of the original training data. However, without altering the algorithm, i.e. only using training data from the indoor experiments, it was able to detect all four persons.

Another dataset was recorded in heavy rain. While the rain was causing an increased amount of noise in the point clouds, the system was still able to correctly classify humans, as seen in Figures 5 (c) and (d).

3.3.1 Human Detection Performance

To determine the detection capability of the system, the network was trained again, this time including data from the outdoor datasets. Six datasets, each consisting of approximately 1500 merged point cloud frames with 0-2 persons present, were added to the training and validation data. After image extraction and manual labeling, the final training and validation set now consisted of 4715 images for the ‘Human class’, and 6805 images for the ‘Not Human class’.

A seventh dataset was used for testing. This dataset contained a 300 s recording where up to four people were present at the same time. The persons were walking randomly, resulting in many different poses, as well as entering and leaving the area. From this set, a total of 949 merged point clouds were analyzed, where 21 contained one person, 83 contained two persons, 324 contained three persons and 521 contained four persons, i.e. there were a total of 3243 clusters that should be classified as humans. The detection accuracy (recall) was calculated by dividing the number of Humans actually detected by the total number of possible detections. This yielded a recall of 76.9 % (2495 out of 3243 humans were correctly detected). The calculated F1 score was 0.87, due to a high precision of 99.9 %.

During the testing, while the point clouds were streamed at around 5 Hz, the detector was outputting results at about 1.2 Hz. The relatively low rate is due to the fact that the detection system is run in Matlab and that temporary images are written to disk before they are classified.

4 Discussion and Conclusions

This paper has demonstrated a novel approach to people detection on very sparse point clouds using only depth information. The proof of concept and accuracy experiments show promising results, with an absolute error that is approximately the same as the resolution of the voxelised point cloud at ±4 cm.

During the accuracy test, the test subject was wearing different clothing with different colors than used in the training data. This implies that the use of intensity values and not colors in the point clouds allows the classifier to distinguish objects based primarily on shape, without being biased by certain outfits.

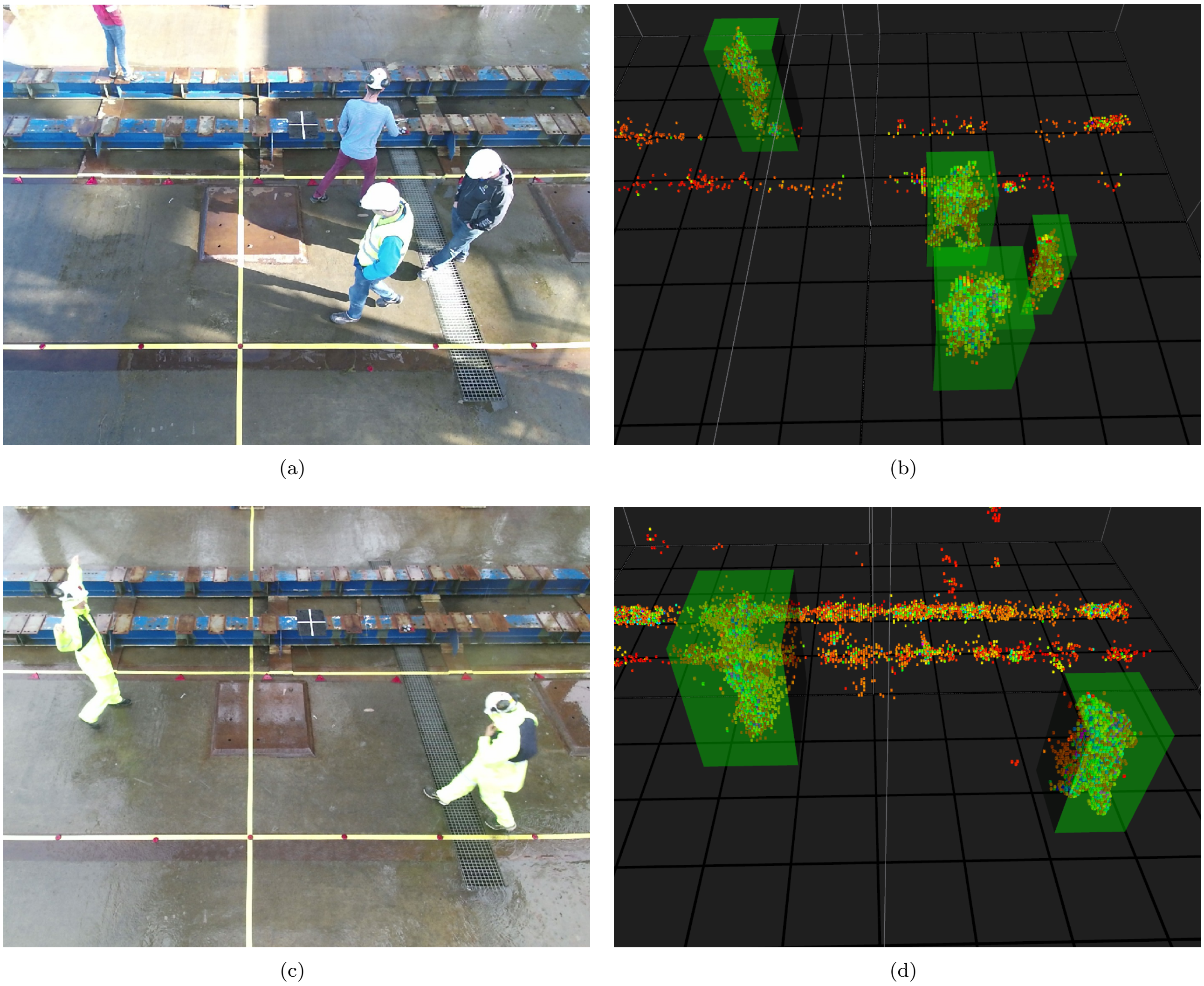


Figure 5: Detection test on Outdoor Data - **(a)** and **(b)** Four people present and detected at the test site, in low sun on wet ground; **(c)** and **(d)** Two people present and detected, in heavy rain.

The tests on outdoor datasets yielded a detection recall of 76.9 % and an F1 score of 0.87. While this is not optimal, further investigation has led to the conclusion that this is mostly due to the filtering and segmentation process, and not the classification. Firstly, work should be done to optimize the segmentation to better distinguish persons from the environment. Typical scenarios are where persons are standing on or close to constructions, or carrying large items. In these scenarios, the Euclidean distance segmentation approach struggles to create the correct candidate clusters, and the clusters are discarded immediately and never passed to the classifier.

Another improvement would be to classify parts of persons instead of whole persons. In some scenarios, e.g. when wearing very dark clothes or when occluded by objects, parts of the body may not be visible in the point clouds. In an example where only the torso is visible, the current candidate cluster processing would discard these clusters as they would not be within the constraints. The same problem occurs when a person is entering or exiting the monitored area, leading to partial point clouds of the body.

As our approach is using a scene classifier, the approach could be extended to detect any class of items by adding more classes during training or by training on other classes entirely. This makes it possible augment the system to detect human body parts or any other objects such as robots and equipment. If needed, the system could also be extended by including the (x,y) plane in the flattening process, thus generating a third image for classification at the cost of increased computational load.

Future work should also include testing the classifier on other, published datasets, to compare the performance to other people detection techniques. Specifically, the precision of the presented solution was found to be 99.9 %, however there were few foreign objects present in the data sets which led to very few false positives. In addition, work should be done to port the classifier and detector from MATLAB to a more efficient environment, e.g. a native ROS node, to achieve a higher maximum detection rate than the 1.2 Hz achieved in this paper.

Acknowledgments

The research presented in this paper has received funding from the Norwegian Research Council, SFI Offshore Mechatronics, project number 237896.

References

Aalerud, A., Dybedal, J., and Hovland, G. Automatic Calibration of an Industrial RGB-D Camera Network Using Retroreflective Fiducial Markers. *Sensors*, 2019. 19(7):1561. doi:10.3390/s19071561.

Borgmann, B., Hebel, M., Arens, M., and Stilla, U. Detection of Persons in MLS Point Clouds using Implicit Shape Models. *pf.bgu.tum.de*, 2017. doi:10.5194/isprs-archives-XLII-2-W7-203-2017.

Cao, Z., Simon, T., Wei, S. E., and Sheikh, Y. Realtime multi-person 2D pose estimation using part affinity fields. In *Proc. 30th IEEE Conf. Comp. Vision and Pattern Rec., CVPR 2017*. pages 1302–1310, 2017. doi:10.1109/CVPR.2017.143.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Proc. 2005 IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recognition, CVPR 2005*, volume I. pages 886–893, 2005. doi:10.1109/CVPR.2005.177.

Dybedal, J. Human detector for point clouds using point cloud flattening an cnn scene classifier. `https://github.com/dybedal/wp3-human-voxel-detector`, 2021a.

Dybedal, J. Replication Data for: CNN-based People Detection in Voxel Space using Intensity Measurements and Point Cluster Flattening. *DataverseNO*. 2021b. doi:10.18710/HMJVFM.

Dybedal, J., Aalerud, A., and Hovland, G. Embedded Processing and Compression of 3D Sensor Data for Large Scale Industrial Environments. *Sensors*, 2019. 19(3):636. doi:10.3390/s19030636.

Hakim, E. A. 3D YOLO: End-to-End 3D Object Detection Using Point Clouds. Technical report, 2018.

Hui, L., Yun, L., Meiyi, Q., and Shujuan, P. Object detection method based on three-dimension information extraction of laser point cloud. In *ACM Intl. Conf. Proc. Series*. Association for Comp. Mach., New York, USA, pages 208–213, 2019. doi:10.1145/3307363.3307366.

Kim, J. M., Kim, Y.-J., and Moon, C.-B. Human Target Tracking using a 3D Laser Range Finder based on SJPDAF by Filtering the Laser Scanned Point Clouds. *Intl. J. Control, Automation and Systems*, 2020. 18(X):1–11. doi:10.1007/s12555-019-0603-6.

Lewandowski, B., Liebner, J., Wengefeld, T., Muller, S., and Gross, H. M. Fast and robust 3D person detector and posture estimator for mobile robotic applications. In *Proc. IEEE Intl. Conf. Robotics and Automation*. pages 4869–4875, 2019. doi:10.1109/ICRA.2019.8793712.

Linder, T. and Arras, K. O. Real-time full-body human attribute classification in RGB-D using a tessellation boosting approach. In *IEEE Intl. Conf. Intelligent Robots and Systems*. pages 1335–1341, 2015. doi:10.1109/IROS.2015.7353541.

Linder, T. and Arras, K. O. People detection, tracking and visualization using ROS on a mobile service robot. *Studies in Computational Intelligence*, 2016. 625:187–213. doi:10.1007/978-3-319-26054-9_8.

Munaro, M., Basso, F., and Menegatti, E. OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks. *Robotics and Autonomous Systems*, 2016a. 75:525–538. doi:10.1016/j.robot.2015.10.004.

Munaro, M., Lewis, C., Chambers, D., Hvass, P., and Menegatti, E. RGB-D human detection and tracking for industrial environments. In *Adv. Intell. Systems and Computing*. pages 1655–1668, 2016b. doi:10.1007/978-3-319-08338-4_119.

Simon, M., Amende, K., Kraus, A., Honer, J., Sämann, T., Kaulbersch, H., Milz, S., and Gross, H. M. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds. Technical report, 2019.

Simon, M., Milz, S., Amende, K., and Gross, H.-M. Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds. Technical report, 2018.

Spinello, L., Luber, M., and Arras, K. O. Tracking people in 3D using a bottom-up top-down detector. In *Proc. IEEE Intl. Conf. Robotics and Automation*. pages 1304–1310, 2011. doi:10.1109/ICRA.2011.5980085.

Tang, H. L., Chien, S. C., Cheng, W. H., Chen, Y. Y., and Hua, K. L. Multi-cue pedestrian detection from 3D point cloud data. In *Proc. IEEE Intl. Conf. Multimedia and Expo*. pages 1279–1284, 2017. doi:10.1109/ICME.2017.8019455.

The MathWorks, Inc. Scene Classification Using Deep Learning. https://blogs.mathworks.com/deep-learning/2019/11/25/scene-classification-using-deep-learning/, 2021. Accessed: 2021-04-28.

Tome, D. and Russell, C. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. Technical report, 2017. URL http://visual.cs.ucl.ac.uk/pubs/liftingFromTheDeep.

Velodyne Lidar. Automated with Velodyne — The Marsden Group — Velodyne Lidar. 2020. URL https://velodynelidar.com/automated-with-velodyne/the-marsden-group/.

Wengefeld, T., Lewandowski, B., Seichter, D., Pfennig, L., and Gross, H. M. Real-time person orientation estimation using colored pointclouds. In *Proc. 2019 European Conf. Mobile Robots*. 2019. doi:10.1109/ECMR.2019.8870914.

Yan, Z., Duckett, T., and Bellotto, N. Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods. *Autonomous Robots*, 2020. 44:147–164. doi:10.1007/s10514-019-09883-y.

Zhou, Y. and Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *Proc. IEEE Comp. Soc. Conf. Comp. Vision and Pattern Rec.*, 2017. pages 4490–4499. doi:10.1109/CVPR.2018.00472.

Zimmermann, C., Welschehold, T., Dornhege, C., Burgard, W., and Brox, T. 3D Human Pose Estimation in RGBD Images for Robotic Task Learning. In *Proc. IEEE Intl. Conf. Robotics and Automation*. pages 1986–1992, 2018. doi:10.1109/ICRA.2018.8462833.