USAGE OF MULTIPLE LIDAR SENSORS ON A MOBILE SYSTEM FOR THE DETECTION OF PERSONS WITH IMPLICIT SHAPE MODELS

Björn Borgmann[a,b*], Marcus Hebel[a], Michael Arens[a], Uwe Stilla[b]

[a] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Gutleuthausstr. 1, 76275 Ettlingen, Germany
(bjoern.borgmann, marcus.hebel, michael.arens)@iosb.fraunhofer.de
[b] Photogrammetry and Remote Sensing, Technische Universitaet Muenchen, Arcisstr. 21, 80333 Munich, Germany
stilla@tum.de

Commission II, WG 3

KEY WORDS: LiDAR, Mobile, Laser scanning, MLS, Person, Detection, Classification

ABSTRACT:

The focus of this paper is the processing of data from multiple LiDAR (light detection and ranging) sensors for the purpose of detecting persons in that data. Many LiDAR sensors (e.g., laser scanners) use a rotating scan head, which makes it difficult to properly time-synchronize multiple of such LiDAR sensors. An improper synchronization between LiDAR sensors causes temporal distortion effects if their data are directly merged. A merging of data is desired, since it could increase the data density and the perceived area. For the usage in person and object detection tasks, we present an alternative which circumvents the problem by performing the merging of multi-sensor data in the voting space of a method that is based on Implicit Shape Models (ISM). Our approach already assumes that there exist some uncertainties in the voting space. Therefore it is robust against additional uncertainties induced by temporal distortions. Unlike many existing approaches for object detection in 3D data, our approach does not rely on a segmentation step in the data preprocessing. We show that our merging of multi-sensor information in voting space has its advantages in comparison to a direct data merging, especially in situations with a lot of distortion effects.

1. INTRODUCTION

The detection of persons in the surroundings of a mobile system has several use cases. Such a functionality can be helpful for the safe operation of an autonomous system in the direct vicinity of humans. It is also useful for several kinds of assistance systems, supporting the operator or driver of such a system. In comparison to the general detection of moving objects or obstacles, the actual detection of persons makes it possible to pay particular attention to their safety and moving patterns. For example, a person can change the moving direction more abruptly than a car. There are also several use cases of person detection methods in the field of human-machine interaction.

Several kinds of sensors are appropriate for the realization of such a functionality, and a real-world system is often equipped with different kinds of sensors. This paper is focused on the usage of LiDAR sensors for such tasks. LiDAR sensors are able to directly evaluate the 3D features and 3D geometry of the recorded area, and they can operate independent of external light sources.

A mobile laser scanning (MLS) system can be equipped with multiple LiDAR sensors to increase the data density or to be able to record a larger part of the vehicle’s surroundings. This is especially helpful if a single LiDAR sensor is not able to cover the whole surroundings due to constructional limitations. However, the usage of multiple sensors requires some kind of data fusion between these sensors. To achieve this, a simple approach is to directly merge the data of the sensors. But given the scanning nature of each sensor, this results in temporal distortion effects at areas which are covered by more than one sensor. These distortion effects happen if moving objects occur in the scene, while the respective data are not recorded at the exact same time by the sensors. Figure 1 shows an example of such a distortion: the lower part is recorded by two sensors (green and red) and the person seems to have more than two legs. To prevent such effects, the multiple sensors have to record every area at the exact same time. This is hard to achieve due to the scanning nature of most of today’s LiDAR sensors.

Another kind of data fusion approach is to process the data of each sensor individually and to merge their processing results. However, this potentially discards some valuable information and gets difficult if the overlapping area between the sensors is small, meaning that a part of a person is only recorded by one sensor and another part only by a second sensor.

Figure 1. Example of distortion effects caused by the fusion of data from multiple sensors (green: sensor 1, red: sensor 2).

*Corresponding author

2. RELATED WORK

There are different groups of approaches which are commonly used to detect persons or, more general, objects in 3D data. The problem is often separated into a segmentation step and a classification of the segments. For the classification a multitude of classifiers can be used. One group of classifiers are support vector machines (SVM). They are trained by determining a hyperplane in the feature space of the training data. This hyperplane separates the different classes from each other and can be used for the classification of new data later on. Premebida et al. (2014) presented an approach which uses an SVM to detect pedestrians in depth images. These depth images are generated by a LiDAR sensor and an RGB camera. The LiDAR sensor is used to generate depth values for the pixels of the RGB images. Another approach uses two consecutive SVMs to detect persons in LiDAR point clouds. The first one uses several geometric features of the processed cloud segments. The second one uses the output of the first and a number of tracking features to generate the output of the approach (Navarro-Serment et al., 2010).

Bag-of-words approaches are also widely used to solve classification problems. They use a dictionary of words which vote for a certain class. These words are usually represented by feature descriptors. The dictionary is the result of a training process. Features are extracted for the processed data and then these features are matched to words in feature space. The matched words are used to classify the data based on their votes. Behley et al. (2013) use a bag-of-words approach to classify point cloud segments. Instead of only using one bag-of-word classifier, they utilize several of them with differently parameterized features. This allows them to deal better with the characteristics of each point cloud segment. For example, the data density might vary between the different segments. It is conceivable to deal with distorted segments in such an approach by using yet another group of classifiers, which are specially trained and parameterized for the use with distorted data.

Another way to solve the problem of person detection is not to classify segments of the data but each individual element of it. Shotton et al. (2011, 2013) use random decision forests to classify each pixel of depth images and to detect persons in the data. They also track several body parts of detected persons. Random decision forests utilize several decision trees and use them together to classify data. Each tree is trained with a certain random element, meaning that the resulting trees are not completely identical. This prevents the problem of overfitting, which otherwise often occurs.

In recent years, deep learning with convolutional neural networks has successfully been used for object recognition tasks. At first, these approaches were considered for the processing of 2D images. But they have later been adopted for either depth images (Socher et al., 2012) or volumetric representations of 3D data (Maturana and Scherer, 2015; Garcia-Garcia et al., 2016).

Implicit shape models (ISM) are a modification of the classical bag-of-words approach. They modify this approach in a sense that the words not only vote for a class but also for a position of the classified object. They then look for positions in voting space at which multiple votes converge. This allows them to consider not only the existence of certain features but also their relative position in the data. Especially in case of 3D data, a lot of information lies in the geometrical structure of the recorded area. Therefore, considering the relative position of features is an obvious improvement of the detection method. ISM were first used by Leibe et al. (2008) for the object detection in 2D images. Later they were modified several times to be used for 3D data. Knopp et al. (2010) use a 3D ISM approach for general object recognition tasks, which uses 3D SURF features calculated for well-chosen interest points. Velizhev et al. (2012) use ISM to detect parked cars and light poles in point clouds of an urban environment. These clouds have been created by merging several separate scans. Their approach does not deal with moving objects and considers them as noise. Our own approach uses ISM to detect persons in single scans (rotations of the scan head) of a 360° LiDAR sensor (Borgmann et al., 2017). Although most ISM approaches for 3D data utilize some kind of segmentation, the basic idea of ISM does not rely on this.

3. OUR APPROACH

In this section, we describe our approach for the detection of persons in 3D data of multiple LiDAR sensors. We assume that the data of each sensor are provided as streams of general 3D point clouds, and we assume that the sensor setup has already been calibrated geometrically. We also assume that there is some kind of time-synchronization between the multiple sensors, in a sense that we are able to match together data which have been acquired at roughly the same time. But due to the effects mentioned in the introduction of this paper (e.g., scanning sensors), this time-synchronization does not prevent all temporal distortion effects between the separate sensors.

Although the focus of this paper lies on the detection of persons in the data of multiple LiDAR sensors, our approach can also be used for the detection of other object classes, or for the exploitation of data of a single LiDAR sensor. It is based on our existing approach (Borgmann et al., 2017) and uses implicit shape models (ISM). It performs the merging of data between the sensors in the voting space of the ISM. Our approach already assumes that the votes in the voting space are not completely exact and searches for maxima in this space. Therefore, additional uncertainties caused by temporal distortion effects should not have a large influence on the performance of our method. In comparison, the processing of directly merged point clouds affects the determined features and is likely to reduce the detection performance.

Our approach consists of three main processing steps: preprocessing, casting of votes and search for maxima in the voting space. While the first two steps are performed separately for every point cloud of each sensor, the last one is performed only once for each set of simultaneously acquired point clouds. Figure 2 shows a schematic diagram of our approach, considering data from two sensors. In the following subsections, we describe the main parts of our approach in more detail.

3.1 Feature descriptor

Our approach uses feature descriptors to describe the local shape of the processed data. ISM-based approaches assume that an object can be classified by such local shape descriptions, and that they are sufficient to detect objects of certain classes. With regard to the feature descriptors, two general strategies are commonly used. One is to consider a smaller amount of highly descriptive features, which are determined for well-chosen interest points (Knopp et al., 2010). Another strategy is to compute a larger amount of less descriptive features for a larger part of the

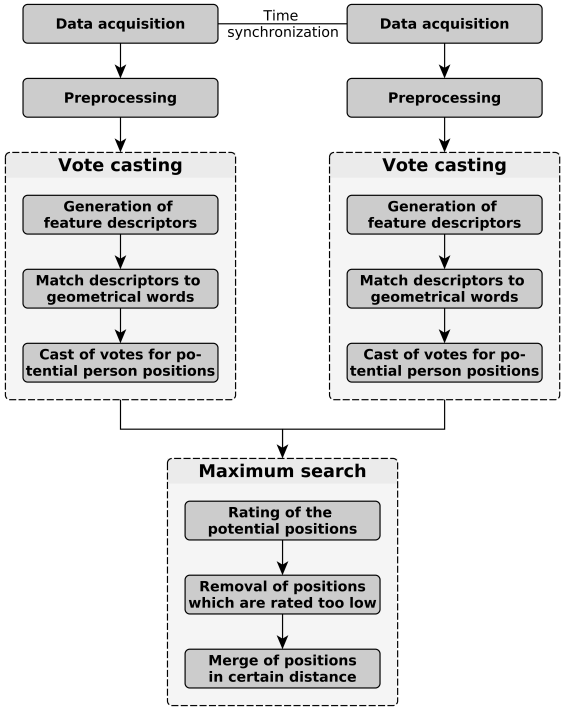


Figure 2. Schematic diagram of our approach for the processing of 3D data of two LiDAR sensors

data or the complete data set. This strategy seems to deal better with noise and occlusions (Velizhev et al., 2012). Since we have to deal with many occlusions in our use case, we choose the second strategy and evaluate fast point feature histograms (Rusu et al., 2009) as descriptor, which we calculate for every 3D point.

3.2 Dictionary

The dictionary is the result of a training process and it contains geometrical words, which later vote for the potential position of a person. The structure of our dictionary is shown in Figure 3. Each word is described by a feature descriptor and casts at least one vote. Each vote is cast for an object with a certain class, the vote has a position vector for that class and it has a weight between 0 and 1. For the purpose of this paper, we only consider the classes ”person” and ”not a person”. (Manually) pre-classified point cloud segments are used during the training process. These training segments either only contain a single person or no person at all. The actual training process is explained in greater detail in our previous work (cf. Borgmann et al. (2017)). At first, feature descriptors are determined for each point of the processed training data. These descriptors are used to initialize new words. In a second step of the training, words which are similar in feature space are clustered together. This clustering reduces the size of the resulting dictionary. After the clustering, a word might have more than one vote. Similar votes of such words are also clustered together and the total weight of each word is normalized to 1. This means that votes of descriptive words, which only cast a few different votes, have a higher weight than votes of less descriptive words, which cast a multitude of votes.

3.3 Preprocessing

The preprocessing part of our approach serves the purpose to reduce the amount of data which has to be processed later on. This is done to increase the runtime performance. The first step of

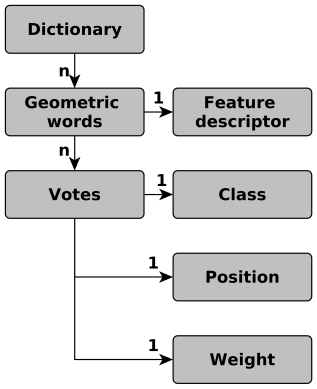


Figure 3. Structure of our dictionary

the preprocessing is the estimation of a ground grid. This grid is then used to perform a removal of data points at ground level. We have shown that such a ground removal can typically reduce the amount of data by about $45\,\%$ (cf. Borgmann et al. (2017)). The usage of a ground grid allows us to deal with uneven ground, which would not be achievable by a simple ground plane estimation. The ground grid is generated as follows: at first, each grid cell is initialized by determining its ground level based on the height values of the points which lie in that cell. To achieve this, it is assumed that every point, besides outliers, either belongs to the ground level or lies above the ground. If a cell does not contain any ground points, this method will give incorrect results. To avoid this, the grid cells are validated in a second step. We traverse them starting from a well-chosen starting cell which contains ground. For the traversal, a criterion is used for the maximum steepness of the ground, and hence for the allowed height difference between neighboring cells. Grid cells which cannot be reached by the traversal without violating the steepness criterion are considered as cells which do not contain any ground and are subsequently removed from the ground grid.

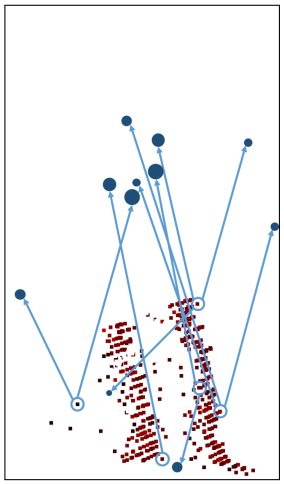
In our previous work, we performed a segmentation of the remaining data based on a region-growing method. Following that segmentation, we filtered out segments for which we could assume that they do not represent a person. This filtering step was done by evaluating simple geometrical features like the aspect ratio or size of the segments. Although these segmentation and filtering steps allowed a further data reduction and a better runtime performance, they also have their disadvantages: Segmentation errors may occur, which subsequently cause wrong filtering results. In addition, such segmentation errors are problematic if the further processing is at least partly dependent on a correct segmentation. Therefore, we modified our previous approach and no longer use a segmentation and filtering of the data.

3.4 Casting of votes

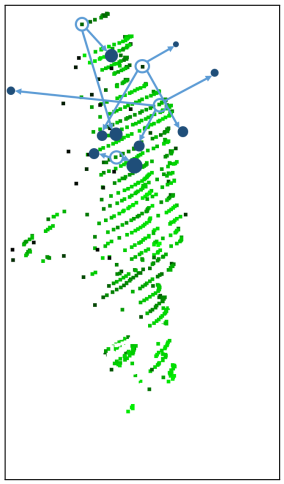
The casting of votes consists of three steps. First, a feature descriptor is calculated for each point in the processed point clouds. Then a search in the dictionary is performed to find the best matching word for each of the calculated feature descriptors. These words are used for the actual casting of the votes. Figure 4a and Figure 4b exemplary show the result of such a vote casting for data from two LiDAR sensors.

As described earlier, these steps are performed for each sensor individually. Since the determination of a feature descriptor for a point is based on its neighboring points, it is likely to be affected

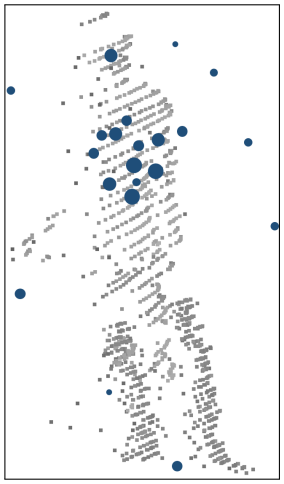
by distortion effects. Therefore, we avoid a merging of the multi-sensor data before the feature extraction. In doing so, we cannot benefit from the higher data density available in areas covered by multiple sensors. This constitutes a disadvantage in comparison to approaches that directly merge the point clouds, especially if the data density is low. However, given our use case, this disadvantage is more than compensated by the advantages in dealing with moving objects in these areas.



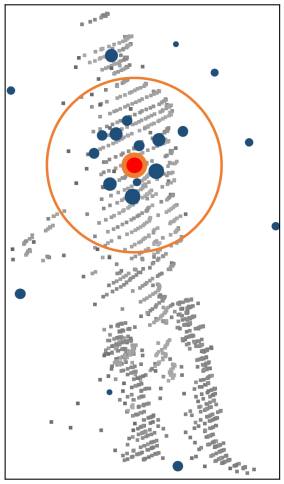
(a) Examples of votes cast for a point cloud from sensor 1



(b) Examples of votes cast for a cloud from sensor 2



(c) Merged votes of both processed clouds. The size represents their weight



(d) Rating of a potential object position using votes in surrounding interest area

Figure 4. Illustration of our approach for person detection in multiple point clouds

3.5 Detection of person positions in voting space

After votes for each of the point clouds have been cast, all votes are put together and their individual sensor source is ignored further on. The next step is the search for maxima in the voting space. To find maxima, a weighting of votes for potential object positions is used. Figure 4c exemplary shows the voting space at the beginning of this processing step: each potential position has the weight of its original vote in the dictionary. We assume that a high amount of weight indicates the actual position of an object. Therefore we evaluate the neighborhood of each potential position to rate this position. We use the following equation to determine the rated weight of each potential object position:

$$R_p = \sum_{k \in K} W_k \cdot W_{\text{norm}} \cdot e^{-\frac{{D_{pk}}^2}{2\sigma^2}} \tag{1}$$

where R_p = Rated weight of position p
K = All potential positions with same class as p
W_k = Weight of position k
W_{norm} = Weight normalization factor
D_{pk} = Euclidean distance between positions p and k
σ = Determines the width of the normal distribution

The above equation adds a fraction of the original vote weight of neighboring positions to the rated weight of the currently examined position. This fraction is calculated based on the distance between the two positions using the Gaussian normal distribution. W_{norm} is a normalization factor. In previous work, we calculated this normalization factor based on the total weight of all potential positions in the processed point cloud segments. This works well, as long as only one or only a small amount of objects end up in a segment. Since we no longer use a segmentation and since we process multiple point clouds from different sensors, we changed the calculation of the normalization factor: Now we use an interest area around the currently processed position for the determination of its weight. Then Equation 1 only considers positions inside the interest area for K. The normalization factor is calculated for this interest area as follows:

$$IW_{\text{norm}} = \frac{1}{N(P_I)} \tag{2}$$

where IW_{norm} = Weight normalization factor for interest area I
$N(P_I)$ = Number of potential positions in interest area I

This factor normalizes the weight based on the number of potential positions in the interest area. The interest area is defined by a certain radius around the examined position. This normally works well, but it is susceptible to very isolated potential positions which only have a small amount or no other positions in their respective interest area. Therefore we also evaluate a criterion for the minimum amount of other positions with the same class in this area. If this criterion is violated, the position will not be processed further. This accords to the basic assumption of our approach that a correct detection is indicated by a great amount of votes for positions close to the actual position of the person.

After the weighting, positions with a rating below a certain threshold are removed. The remaining set of positions are then assumed to represent correctly detected persons. If several of these positions end up in close proximity of each other, we assume they represent the same person and merge them together.

4. EXPERIMENTS

We performed several experiments to determine the improvement that is achievable by merging multi-sensor information in the voting space of our ISM method for LiDAR-based person detection. The improvement and performance are evaluated in comparison to a direct merging of the data. This direct merging is achieved by combining the 3D point clouds of the separate sensors. In this section, we first describe the experiments and then present and discuss their results.

4.1 Experimental setup

For our experiments we analyzed data which contain persons captured by multiple LiDAR sensors. In order to generate such data we used our multi-sensor vehicle MODISSA, which, among other sensors, is equipped with several LiDAR sensors. The vehicle is shown in Figure 5. For our experiments we used the two Velodyne HDL-64E mounted in a tilt-angle at the front of the vehicle. The individual sensors are able to perform 1.3 million measurements in distances up to 120 m. They utilize a rotating scan head, giving them a 360° horizontal field of view. Vertically, the field of view is 26.9°, which is divided into 64 scan lines. In the setup used for the experiments, the scan heads of both sensors rotated at 10 Hz. Due to the mounting angle of the sensors, their fields of view form an overlapping coverage area in front of the vehicle. Mainly this area has been used for the experiments. We considered the data recorded in 0.10 s as single point clouds, which corresponds to one rotation of a scan head. Both sensors are geometrically calibrated and time-synchronized. The vehicle is equipped with an inertial measurement unit and GNSS receivers, allowing us to compensate for the vehicle’s movement while recording 3D LiDAR data (direct georeferencing).

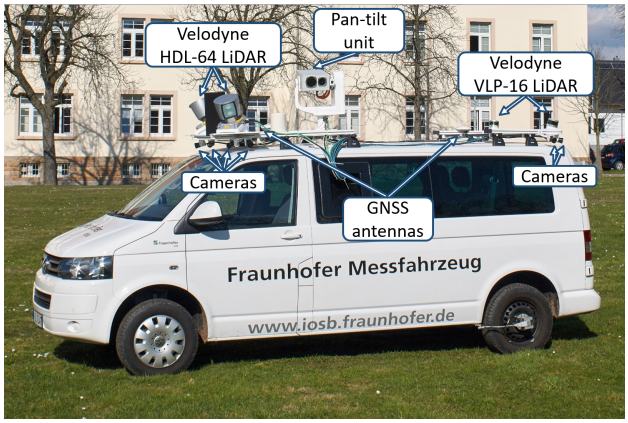


Figure 5. Multi-sensor vehicle MODISSA equipped with multiple LiDAR sensors

We recorded two sequences in which a person moves within the coverage area of the sensors, going-in and coming-out of the overlapping part. We manually annotated these sequences to generate a ground truth to evaluate our results. The first sequence is considered to be the ”easy” sequence, in which the person is only perceived by one sensor most of the time and only crosses the coverage area of both sensors a few times. In the overlapping coverage area, the person moved slowly, causing only negligible distortion effects. The second sequence is considered to be the ”difficult” sequence, containing more movements of the person between fields of view of both sensors. In addition, the person moves faster and stays longer in the overlapping coverage area. Due to these differences, this sequence contains much more distortion effects. Both sequences were processed by merging the information in voting space and by a direct data merging, in both cases using a processing chain that kept unchanged otherwise. Additional manually annotated point clouds from previous measurement campaigns were used to train the detector. Positive as well as negative examples were used for the training.

For the evaluation of our results, we use the indicators *precision* and *recall*. The precision shows how many of the detections are correct and the recall shows how many of the persons in the scene have been detected. The indicators are defined as follows:

$$Precision = \frac{tp}{tp + fp} \tag{3}$$

$$Recall = \frac{tp}{tp + fn} \tag{4}$$

where tp = True positive detections
fp = False positive detections
fn = False negative detections

4.2 Results and discussion

Figure 6 shows our results as precision-recall curves. We compare both methods for the two sequences. As shown in Figure 6a, the direct merging of data outperforms the merging of information in the ISM voting space in case of the ”easy” sequence. We assume that this results from the low amount of distortion effects in this sequence. In such cases, only small negative influences on the performance are to be expected if the data are directly merged. In addition, this direct data merging results in a higher data density, which benefits the feature extraction and makes it more accurate. In contrast, the merging of information in the ISM voting space is based on an individual feature extraction for the low-density data of each sensor.

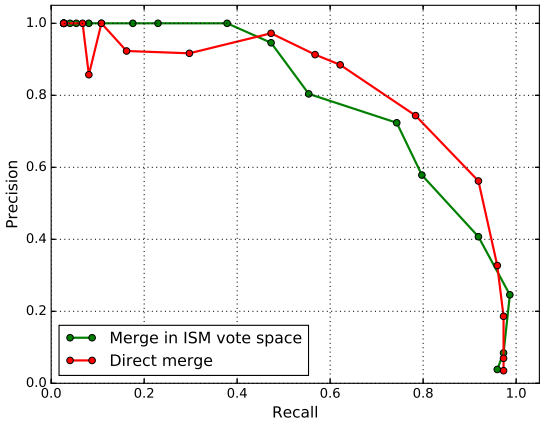
In case of the ”difficult” sequence, the performance of the merging in ISM voting space increases in comparison to the direct merging of data (cf. Figure 6b). We assume that this becomes even more obvious in cases with more than two LiDAR sensors, more overlapping fields of view, moving persons, faster movements, and so on.

As a result, our approach to perform the merging of information in ISM voting space is only slightly influenced by distortion effects. On the other hand, the direct merging of data performs well in cases without distortion effects that affect the data. We expect that the limits of our merging approach in voting space are reached as soon as the distortion effects cause the distorted votes to leave the interest area of the rating process (cf. Section 3.5). However, this would only be possible in cases where objects move much faster than persons.

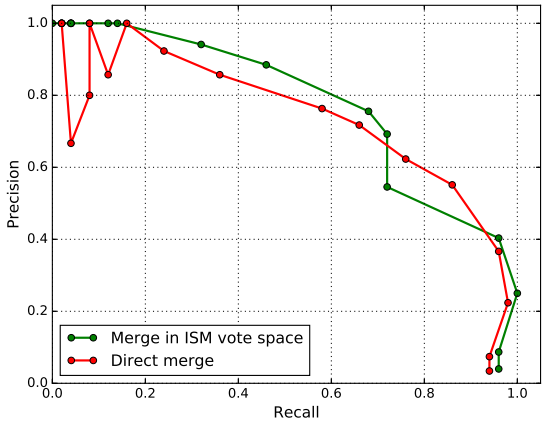
Figure 7 exemplary shows some results of our approach. Although we merged the point clouds of both sensors for the visualization, the actual detection resulted from the merging of information in voting space. Besides being able to deal with the described distortion effects, our method can detect persons and determine their correct position even if the person is only partly visible.

5. CONCLUSION AND FUTURE WORK

We presented an approach for ISM-based object detection that is robust against distortion effects caused by the usage of data from multiple scanning LiDAR sensors. To achieve this, we extended our existing ISM approach and perform the merging of multi-sensor information in the voting space of the ISM method, instead of merging the 3D data directly. This circumvents the distortion effects when extracting 3D features, while still utilizing some of the advantages of using multiple sensors in the actual



(a) Precision-recall curves for the ”easy” sequence



(b) Precision-recall curves for the ”difficult” sequence

Figure 6. Results for both sequences

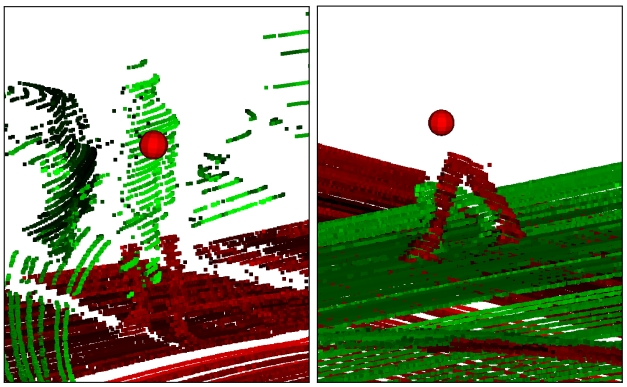


Figure 7. Exemplary results: Our approach is robust against distortion effects and detects partially visible persons

detection step. In an evaluation of our approach, we showed that our method provides results that are more stable than those resulting from a direct merging of the 3D data. When detecting objects by an ISM approach in combination with a multi-sensor setup, information merging in voting space should be the preferred way of sensor fusion, especially if there are many distortion effects induced by object movements.

In addition, our approach does not rely on a segmentation of the original data. This prevents segmentation-induced errors caused by an over- or under-segmentation of the data. This is an improvement in comparison to our previous work and many similar object detection approaches.

In future works we plan to improve the performance of our approach by utilizing some meta-knowledge about persons. For example, a person normally has some contact to the ground. Since we already determine the ground level, we could use that knowledge to deal with several false-positive detections. In addition, we plan to add a tracking component to our approach, which allows us to use knowledge about previously detected persons to improve the performance for the currently processed data. To achieve this, we plan to include the tracking information to the voting space of the ISM approach.

References

Behley, J., Steinhage, V. and Cremers, A. B., 2013. Laser-based segment classification using a mixture of bag-of-words. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4195–4200.

Borgmann, B., Hebel, M., Arens, M. and Stilla, U., 2017. Detection of persons in MLS point clouds using implicit shape models. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W7, pp. 203–210.

Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M. and Azorin-Lopez, J., 2016. Pointnet: A 3D convolutional neural network for real-time object class recognition. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1584.

Knopp, J., Prasad, M., Willems, G., Timofte, R. and Van Gool, L., 2010. Hough transform and 3D SURF for robust three dimensional classification. In: *Proceedings of the 11th European Conference on Computer Vision: Part VI*, ECCV’10, Springer-Verlag, Berlin, Heidelberg, pp. 589–602.

Leibe, B., Leonardis, A. and Schiele, B., 2008. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision* 77(1), pp. 259–289.

Maturana, D. and Scherer, S., 2015. Voxnet: A 3D convolutional neural network for real-time object recognition. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928.

Navarro-Serment, L. E., Mertz, C. and Hebert, M., 2010. Pedestrian detection and tracking using three-dimensional LaDAR data. *The International Journal of Robotics Research* 29(12), pp. 1516–1528.

Premebida, C., Carreira, J., Batista, J. and Nunes, U., 2014. Pedestrian detection combining RGB and dense LiDAR data. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4112–4117.

Rusu, R. B., Blodow, N. and Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ICRA’09, IEEE Press, Piscataway, NJ, USA, pp. 1848–1853.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. and Blake, A., 2011. Real-time human pose recognition in parts from a single depth image. In: *CVPR*, IEEE.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A. and Blake, A., 2013. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12), pp. 2821–2840.

Socher, R., Huval, B., Bath, B., Manning, C. D. and Ng, A. Y., 2012. Convolutional-recursive deep learning for 3D object classification. In: *Advances in Neural Information Processing Systems*, pp. 656–664.

Velizhev, A., Shapovalov, R. and Schindler, K., 2012. Implicit shape models for object detection in 3D point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3, pp. 179–184.