A Fast and Robust 3D Person Detector and Posture Estimator for Mobile Robotic Applications

Benjamin Lewandowski, Jonathan Liebner, Tim Wengefeld, Steffen Müller and Horst-Michael Gross

***Abstract*— Although, due to recent deep learning techniques, person detection seems to be solved in the computer vision domain, it is still an issue in mobile robotics. On a robot only limited computing capacities are available. The challenge gets even more difficult when operating in an environment, with people in poses different from the standard upright ones. In this work the environment of a supermarket is considered. Unlike most scenarios targeted by the community, persons not only occur in standing postures, but also grasping into the shelves or squatting in front of them. Furthermore, people are heavily occluded, e.g. by shopping carts. In such a challenging environment, it is important to perceive people early enough and in real-time in order to enable a socially aware navigation. Classical person detectors often suffer from a high posture variance or do not achieve acceptable real-time detection rates. For this reason, different components from the 3D object detection domain have been used to create a new robust person detector for mobile application. Operating on 3D point clouds allows fast detections in real-time up to our goal distance of ten meters and above using the Kinect2 depth sensor. The detector can even differentiate between typical postures of customers who stand or squat in front of shelves.**

I. INTRODUCTION

In this paper we consider the scenario of a mobile robot operating in a supermarket. In continuation of our earlier research in the field of assistive shopping guides for do-it-yourself stores [1], our long term goal is the development of a robot for the autonomous detection of out-of-stocks in retail stores during opening hours. Perceiving humans like customers and employees early and having information about their posture, should enable our mobile robot to safely and politely navigate through the store.

The scenery of a supermarket waits with several challenges for a robot when detecting people and their postures to reason about further behaviors. The narrow, long corridors, as shown in Fig. 2a, require a detection of persons in distances of up to ten meters and above. Furthermore, the detection needs to be robust against different poses, like grasping, squatting and bending poses, as well as occlusions by shopping carts, goods on pallets, or other humans. Fig. 2 shows some situations from a typical supermarket. Besides these environmental constraints, there are also hardware limitations determining the choice of an appropriate person detector for the given scenario.

All Authors are with Neuroinformatics and Cognitive Robotics Lab, Technische Universität Ilmenau, 98694 Ilmenau, Germany. fg-nikr@tu-ilmenau.de

This work has received funding from the German Federal Ministry of Education and Research (BMBF) to the project ROTATOR (grant agreement no. 03ZZ0437D), and to the project 3D-PersA2 (grant agreement no. 03ZZ0460) in the program Zwanzig20 – Partnership for Innovation as part of the research alliance 3Dsensation.

Fig. 1: Example scene and output of our supermarket person detector (Section III). The green box indicates a standing, the red box a squatting person.

In this paper we present a person detection and pose estimation system for mobile robotic platforms which directly operates in the metrical 3D space and thus, enables high detection rates on a standard consumer CPU. More specifically, we combined a fast, depth based person candidate generation technique with components commonly used in the 3D object detection and map registration domain. Experiments showed that our system outperforms classical RGB and depth detection approaches commonly used in robotics and is also able to robustly estimate the posture of people.

The remainder of this paper is structured as follows. The next section briefly describes the State of the Art of vision-based person detection. Our own 3D person detector is proposed in detail in section III followed by a section on evaluation results.

II. RELATED WORK

Considering the task of person detection in color images, deep learning approaches, keep breaking records and are probably the first choice when it comes to vision tasks. Detectors like [2] perform incredibly good on upright pedestrians. In addition to simple bounding boxes, modern approaches can also estimate human skeletons [3], and it is even possible to do a per pixel segmentation [4]. With [5] and [6] deep learning has already arrived in the world of 3D data. There, the 3D information are converted into a voxel grid structure in order to apply convolutional neural networks. Unfortunately, deep learning approaches require

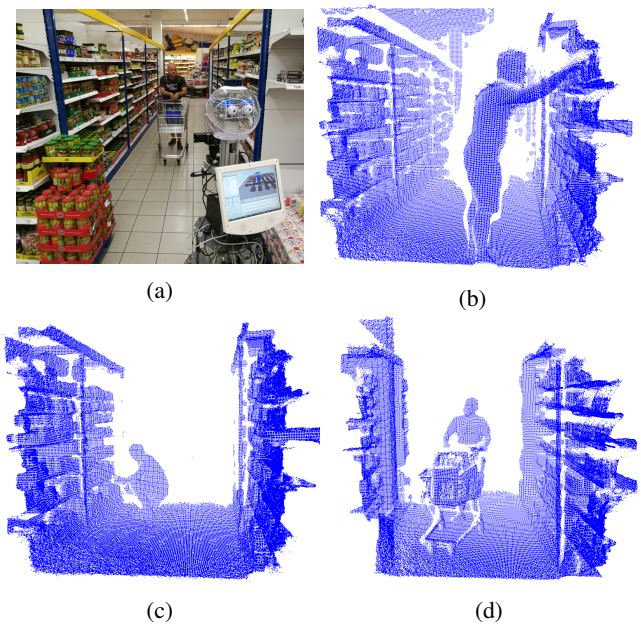

Fig. 2: A typical shelf corridor in a supermarket (a) and 3D point clouds of human poses in this environment (b - d). The poses differ greatly from pedestrians on the street. Here, people grab into a shelf (b), squat in front of a shelf (c), or push a shopping cart (d) which causes heavy occlusions.

specialized hardware with a high power consumption if real-time detection rates should be reached. This conflicts with the limited battery capacity of our mobile robot. Furthermore, in complex applications, like in [7], several modules for localization, navigation, person tracking, emergency handling, and other application specific services run in parallel and need to share hardware resources. Another drawback of deep learning is the need for a lot of data for training, which is hard to acquire in scenarios where no public data sets are available, like in our target supermarket environment.

More suitable for a mobile platform with limited resources are classical detection approaches. Here, the HOG-Detector [8] or its part-based extension [9] are famous representatives. Although, both approaches run on the CPU, they hardly reach real-time detection rates. This is because of the high number of image scales which need to be classified in order to detect persons of different sizes and distances. In [10] this problem is addressed by reducing the number of scales by means of scaling the classifiers instead.

By adding depth information, the detection can be accelerated even more. In [11] and [12] the number of candidates to be classified by an HOG-Detector are drastically reduced by exploiting depth or point cloud information. Depth images can also be used for classification directly. The HOD-Detector in [13] adapts the HOG idea on depth images and outperforms its RGB counterpart due to suboptimal illumination conditions in their data set. In [14] the HOD detector is applied after a graph based segmentation. In this way, frame rates of 30 Hz on a single CPU core could be reached. In [15] a segmentation based on 3D point clouds is combined with a depth template matching as detector and thus, reaches detection rates of more than 30 fps on a single CPU core. Those classical image detectors (color and depth) are mostly designed for the detection of upright persons. Therefore, they seem to be inflexible with respect to a high pose variance as given in our scenario. To overcome these restrictions, in [16] for example, eight separate classifiers had to be trained in order to detect lying people in all orientations with the drawback of an increased detection time.

While some approaches utilize metrical 3D representations, like 3D point clouds, only for a segmentation as a preprocessing step, the number of detectors which also do the detection task on this representation is rare. Nevertheless, metrical 3D data seem to offer more potential when it comes to a higher number of poses. In [17] for example, the task of lying person detection was solved by a single detector in real-time using metrical 3D data only. The detection of upright people in 3D point clouds was also done before [18], [19]. Both approaches employ simpler features and segmentation strategies compared to the proposed solution.

3D data seem to offer several advantages for a person detection system like in our scenario. Besides the possibility for easy and fast candidate generation based on geometric constraints, the variance of human shapes in 3D data on a parts scale appears to be drastically lower than variances in color images even under presence of unusual poses.

Hence, we decided to develop a 3D person detection system by ourselves in order to create a new person detector suitable to fulfill the constraints of our supermarket scenario.

III. PERSON DETECTION APPROACH

An overview of our detection system is shown in Fig. 3. During the preprocessing, a 3D point cloud is constructed from a single depth image. The aim of the following segmentation step is to extract clusters which consist either of all points belonging to a person, or all points belonging to other objects, i.e. negative samples. For each candidate cluster, a feature vector is computed and normalized afterwards. In the last step, a classifier assigns each cluster either to the positive human class, i.e. standing or non-standing person, or to the negative non-person class.

In the following the single steps are explained more in detail.

A. Preprocessing

Since we want to operate in the metrical 3D space, we have to convert the depth image, i.e. the input of our system, into a 3D point cloud at first. Converting 512×424 depth pixels results in a point cloud of more than 200,000 3D points. In order to handle this huge amount of data, a voxel grid filter as in [12] is used to create a down-sampled version of the input scene. As cell size of the filter we chose $6{\times}6{\times}6\ cm^3$, which is approximately the mean distance between points in the unfiltered cloud in a distance of 10 meters and thus, matches our specifications. In addition to the reduction of data, the filter also provides a consistent point density.

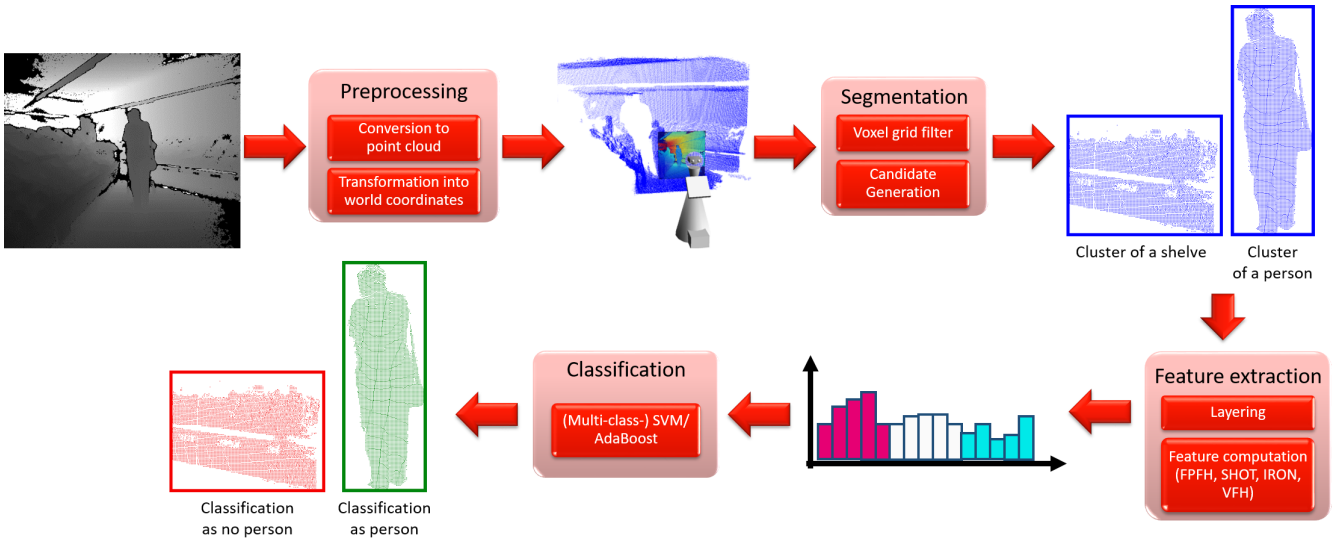

Fig. 3: Overview of our system architecture. After converting the input depth image into a point cloud in the right coordinate frame, we apply a segmentation step in order to generate candidate clusters. For each cluster a feature vector is calculated afterwards, which finally are classified successively.

B. Candidate Generation

Since a point cloud is stored as an unordered list of points, neighborhood operations are more expensive than in 2D images where pixels are stored in a fix grid structure. Hence, for 3D point clouds we need a more complex approach for candidate generation than the sliding window approach on images. A sliding box, i.e. the 3D variant, would result in low frame rates otherwise. For this reason, we adopted the candidate generator of [15] for cluster extraction. In order to reduce the search space, a structure labeling is applied to the point cloud resulting in four different height bands. The ground plane band consists of all points on the ground which do not have to be considered for clustering. The elevated structures band at the other end contains all points at a height where no people are assumed. Given the assumption, that there will always be a free space above the heads of people the object band and a free space corridor are defined. If there is a high point density in the free space corridor, there will not be any person below this area. All remaining points of the object band, not filtered due to a high point density in the free space, are projected onto a 2D histogram in the ground plane. After smoothing with a Gaussian kernel, connected components are extracted in the histogram using the Quick Shift algorithm [20]. Each connected component corresponds to one candidate cluster. We expect that persons close to each other or close to shelves can be better separated by using the Quick Shift segmentation than by a simple Euclidean clustering. For our scenario, we limited the ground plane height to 0.05m, the object band to 2.05m and the free space corridor to 2.25m. The bin size of the ground plane histogram was set to 0.06m in both directions.
These parameters performed best for separating people from other objects and from each other in the given supermarket environment.

C. Feature Computation and used Features

After extracting candidate point clusters from the captured scene, each cluster is processed separately. The remaining task of the system is to decide which cluster represents a person. To do so, a feature vector for every cluster is calculated in the next step. We can utilize different features for a description. Since we are confronted with a high percentage of occluded people, we expected features covering only a small local area of a cluster to be well-suited in our scenario. In particular, the IRON features previously used for map matching [21] and lying person detection [17], the FPFH [22], and SHOT [23] features from the registration and object recognition domain have been considered. All these features compute a histogram as descriptor and describe the curvature of an object by exploiting local surface normals. As it can be seen in Fig. 4, normals of people differ from the irregular and confused surfaces of shelves containing the products. To improve occlusion handling further, we included a layering process similar as introduced in [19] into our system (see Fig. 5). The main idea is to divide the cluster into several sub clusters along the vertical dimension and calculate separate features per sub cluster. Concatenating all of them together results in a more specific feature representation of the complete cluster. In order to ensure a fixed size feature vector, we use a fixed number of layers from the ground to the top of the cluster.
Since a local feature is calculated for each point in a given layer, we have to combine all of them into a common representation. This is achieved by calculating the average feature histogram per layer, where empty layers due to occlusions are represented by empty histograms. The whole process is visualized in Fig. 5. We conduct experiments in Section IV-B in order to find the optimal number of layers for our supermarket scenario.

Fig. 4: The surface normals of a shelf (top) and the normals for a squatting and a standing person (bottom).

Additionally, we compared the VFH [24] as a global feature to the local features for people detection in our scenario.

D. Classification

To assign a class label (person/no person) to a point cluster, we need to classify the feature vectors computed in the previous step. We evaluated the performance of the popular machine learning techniques AdaBoost [25] and Support Vector Machines (SVM) [26] on our feature vectors. The achieved results are presented in the next section. In addition, we replaced the classifier with a 1-vs-1-multi-class SVM, which not only enables a detection of persons in different poses, but also to distinguish between their postures as presented in section IV-D.

IV. EXPERIMENTS

In this section, we evaluate the performance of our 3D point cloud person detector. Several parts of the system are exchangeable. For this reason, we want to show at first the best configuration for our scenario. Then, this configuration is compared to other available detectors, i.e. detectors based on HOG-features [27] and [10], the depth template detector from [15] and an adapted variant of the system in [17]. Furthermore, we include the approach of [3] as a deep learning representative. Finally, it is shown how the system can be used to differentiate between different postures.

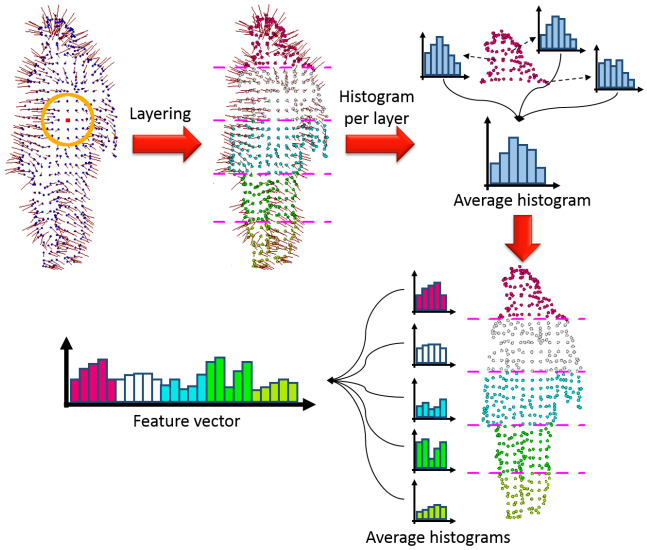

Fig. 5: Feature extraction process. A point cloud cluster is grouped into a fixed set of layers. Per layer, an average local feature descriptor is calculated. Concatenating the histograms results in the feature vector of the cluster.

A. Supermarket Data Set

For training and test we need a data set which contains people in different postures, including squatting and grasping, as well as people with shopping carts. Furthermore, the data must be available as RGB- and depth images in order to compare our point cloud detector against other types of detectors. Since to the best of our knowledge such a scenario was never treated before we recorded our own dataset[1].
For training, data in a local mid-sized German supermarket and in our lab has been recorded using a Kinect2 sensor. We manually annotated the ground truth in each frame into three categories: standing people, squatting people, and others, which contains e.g. all postures between sitting and standing. All samples are in a distance up to 18 meters. Besides the positive samples, about 50,000 negative examples have been sampled from images containing no persons.
For testing, we recorded data in the supermarket only, but on a seperate day to prove generalization capability. Our test set also contains people up to a distance of 18 meters and the same categories as the training set.
The distribution of both data sets is given in Table I. Note,

	standing	squatting	other	#persons	#images
Training	6511	9041	375	15927	15309
Test	3702	707	638	5047	8201

TABLE I: Number of persons and total number of images in our supermarket data sets. The persons are additionally subdivided into postures.

[1]Unfortunately, due to German data protection legislation we are not authorized do make the dataset available.

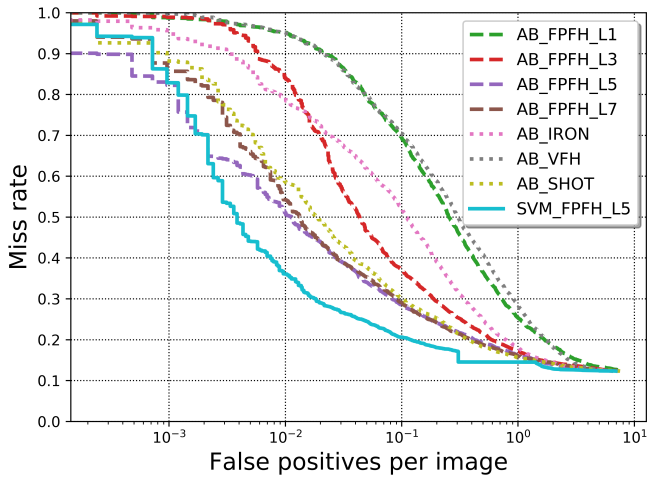

Fig. 6: Comparison of different module combinations of our person detector on a subset of our data set. *AB*: AdaBoost, *L*: Number of layers

that both contain persons with different levels of occlusions, labeled as in [28].

B. Finding the Best Detector Combination

As mentioned at the beginning, our system has several exchangeable components comprising the number of layers, the kind of features, and the type of classifier. We compared all combinations by using AdaBoost as classifier, because it is trained much faster than a nonlinear SVM. The best combinations were than trained with the SVM, in order to compare both classifiers. The most interesting results are visualized in Fig. 6 as detection error tradeoff curves. Note, that, due to large training times, we used only a subset of our data in order to find the best detector combination. In Fig. 6, the dashed lines show the influence of dividing a candidate point cloud cluster into different numbers of layers when using the FPFH as descriptor exemplarily. As expected, increasing the number of layers provides an explicit performance boost because the detector can handle occlusions in a better way. But the improvement is limited to five layers for the FPFH, since more subdivisions result in very thin layers which in turn seem to reduce descriptiveness, especially for squatting postures. We repeated this layering approach for all tested features. The best result per feature is presented in Fig. 6 with dotted lines. As it can be seen, FPFH in five layers stands as the best feature-layer-combination deploying AdaBoost. Replacing the classifier with a non-linear SVM using the RBF kernel function afterwards, lead to a remarkable increase of the detection performance.
In conclusion, our best detector combination divides each point cloud cluster into five layers, calculates FPFH-features and uses a nonlinear SVM. As we will show in Section IV-D, switching from two-class SVM to multi-class SVM additionally enables our detector to differentiate between standing and squatting postures.

C. Results

We compared our supermarket person detector with the RGB approaches [27] and [10], which we call PartHOG and FPDW respectively. Furthermore, we tested the skeleton estimation framework OpenPose [3] as representative for deep learning approaches, because it is already trained on a lot of different human poses. OpenPose is meant to run on a graphics card, but also offers a much slower implementation which runs on the CPU only. As a reminder, on our platform we have limited battery capacity which does not allow us to use high energy consuming GPUs for such an approach. But for completeness, we also benchmarked our detector against both variants of this modern deep learning system.
As 3D competitors, we employed the upper body depth template [15] and the NDT-Detector from [17].
All detectors were retrained on our data set which is described in section IV-A, except for the depth template and OpenPose. The first calculates a mean upper body depth patch. Due to the diverse poses in our training data, retraining in this case resulted in a worse performance than the original model available online. OpenPose on the other hand requires skeleton annotations, which our data set does not provide. However, OpenPose was trained with about 1.5 million samples and hence, it should outperform all the other competitors.
The results of all detectors can be found in Fig. 7. For evaluation, we oriented ourselves to [28] and use detection error tradeoff curves. Since the depth and the RGB sensor have different fields of view, we only included and evaluated samples visible in both images.
As Fig. 7 shows, except for the deep learning approach our 3D supermarket person detector works best in the presented scenario. By doing just a single false detection every ten images, we reach a detection rate of over 70%. The best competitor suitable for our platform performs 10 percentage points worse. Note that the resolution of the RGB image of the Kinect2 is much higher, compared to the depth image (1920×1080 vs. 512×424). The fact that the FPDW achieves better detection results than the same detector that was retrained on our data shows that this approach has problems with the high pose variance. The bad performance of the NDT-Detector is most probably caused by the NDT-map representation, which is restricted up to a certain distance. Considering samples up to 4.5m only, the performance of the NDT-Detector increases drastically. In this case, its performance is between the retrained PartHOG and FPDW at 0.1 false positives per image. Overall the results indicate that depth information are competitive to color images, even if the resolution of the depth images is much smaller.
As expected, OpenPose as deep learning representative has a higher detection performance on our data set than all the other approaches. But OpenPose was trained on about 1.5 million positive training samples, which is over 94 times more compared to our 15,927 person point clouds. Suprisingly, there is a relatively large performance difference between the CPU and GPU version. OpenPose runs almost

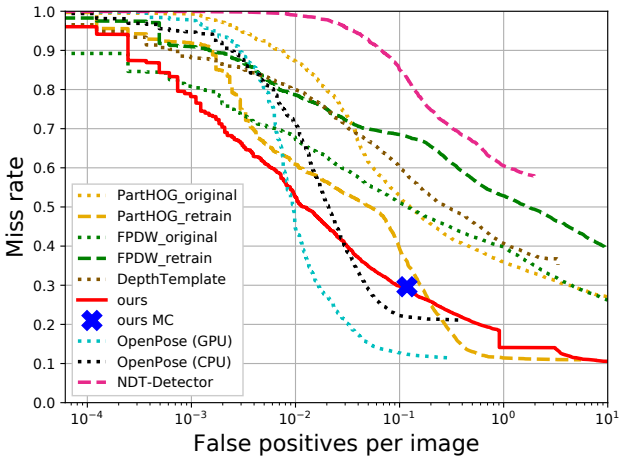

Fig. 7: Evaluation results on our supermarket test set. Dashed lines stand for approaches, which were retrained on our data, dotted lines indicate models available online. The blue cross represents our detector using the multi-class-SVM where no threshold variation is possible.

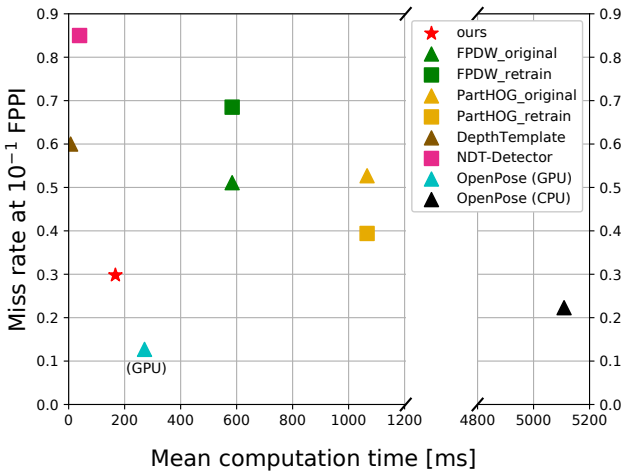

Fig. 8: Mean computation times and miss rates at 0.1 false positives per image. Note: While all approaches run on the CPU, OpenPose is meant to run on a GPU.

10 percentage points better on the graphics card. Nevertheless, we are not able to use it on our robot due to the hardware and battery limitations as already mentioned.
Aside from the pure detection performance, detections in real-time are mandatory for robotic applications. For this reason, we measured the mean computation time of all detectors. As hardware we used an Intel Core i7-7700H CPU for all approaches, except for the GPU implementation of OpenPose. There, we used a GeForce GTX1060 graphics card. All CPU detectors run on a single CPU core. This is important for our application because our robot has to carry out further high level tasks which require CPU computing capacities too. As Fig. 8 indicates, our supermarket person detector provides a good trade-off between detection performance and computation time without the need of a high energy consuming graphics card. Fig. 8 also illustrates that OpenPose on CPU is more than 30 times slower than our approach and hence, does not meet our real-time requirements.

D. Differentiating Standing and Squatting Postures

Since our system is able to detect persons in different poses reliably, we came up with the idea to also differentiate between standing and squatting postures. Therefore, we substituted the two-class SVM in our system by a 1-vs-1-multi-class SVM. This allows the system to directly detect people as standing, squatting or other. The posture could be used as indicator whether a person will remain at the same place for a while or not, which is a useful information when doing social navigation.
For evaluation of the posture estimation, we decided to leave out the *other* class in our test set, because it is hard to define rules when a certain posture counts as *standing* or as *squatting* and thus, the transitions between all classes are fluent. In addition, we only consider samples with an occlusion ratio up to 35% (i.e. the reasonable class according to [28]) in order to have sufficient information available for the task at hand.
As Table II shows, our system is able to correctly classify nearly 97% of the detected standing persons. In the *squatting* class, the system even makes no mistake. Fig. 1 shows an example, how the detector can differentiate between the postures.
As it can be seen in Fig. 7, our system is still able to reliably detect more than 70% of all the persons in the test set by doing just one false positve every ten images, when using a multi-class-SVM.

V. CONCLUSIONS

In this work, we had the objective of detecting persons in a supermarket environment with a mobile robot. Due to several advantages of 3D data, we experimented with different components from the 3D object detection and map registration domain in order to achieve this goal. Thereby, we were able to create a system for detecting people in typical poses of that scenario. Humans can be detected correctly in real-time up to ten meters and above on a standard customer CPU. Furthermore, our robust detector is able to differentiate between postures and thus, offers a good starting point for a socially aware navigation strategy based on our approach.

		ground truth	
		standing	**squatting**
pred.	**standing**	1521 (96.9%)	0 (0%)
	squatting	48 (3.1%)	267 (100%)

TABLE II: Confusion matrix including per class accuracies for discriminating standing and squatting people by using a multi-class-SVM.

REFERENCES

[1] H.-M. Gross, H.-J. Boehme, Ch. Schroeter, St. Mueller, A. Koenig, E. Einhorn, Ch. Martin, M. Merten, and A. Bley, "TOOMAS: Interactive Shopping Guide Robots in Everyday Use – Final Implementation and Experiences from Long-Term Field Trials," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2009, pp. 2005–2012.

[2] M. Eisenbach, D. Seichter, T. Wengefeld, and H.-M. Gross, "Cooperative multi-scale convolutional neural networks for person detection," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 267–276.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 7291–7299.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[5] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 922–928.

[6] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, "PointNet: A 3D Convolutional Neural Network for real-time object class recognition," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 1578–1584.

[7] H.-M. Gross, St. Mueller, Ch. Schroeter, M. Volkhardt, A. Scheidig, K. Debes, K. Richter, and N. Doering, "Robot Companion for Domestic Health Assistance: Implementation, Test and Case Study under Everyday Conditions in Private Apartments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5992–5999.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[10] P. Dollar, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," in *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2010, pp. 68.1–68.11.

[11] Y. Sun, L. Sun, and J. Liu, "Real-time and fast RGB-D based people detection and tracking for service robots," in *12th World Congress on Intelligent Control and Automation (WCICA)*, 2016, pp. 1514–1519.

[12] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with RGB-D data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 2101–2107.

[13] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3838–3843.

[14] B. Choi, C. Meriçli, J. Biswas, and M. Veloso, "Fast human detection for indoor mobile robots using depth images," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 1108–1113.

[15] O. H. Jafari, D. Mitzel, and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5636–5643.

[16] S. Wang, S. Zabir, and B. Leibe, "Lying Pose Recognition for Elderly Fall Detection," in *Robotics: Science and Systems VII*, 2011, pp. 345–353.

[17] B. Lewandowski, T. Wengefeld, T. Schmiedel, and H. M. Gross, "I see you lying on the ground - Can I help you? Fast fallen person detection in 3D with a mobile robot," in *26th International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2017, pp. 74–80.

[18] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A Layered Approach to People Detection in 3D Range Data." in *Conference on Artificial Intelligence*, vol. 10. AAAI Press, 2010, pp. 1625–1630.

[19] F. Hegger, N. Hochgeschwender, G. K. Kraetzschmar, and P. G. Ploeger, "People Detection in 3d Point Clouds Using Local Surface Normals," in *RoboCup 2012: Robot Soccer World Cup XVI*. Springer Berlin Heidelberg, 2013, pp. 154–165.

[20] A. Vedaldi and S. Soatto, "Quick Shift and Kernel Methods for Mode Seeking," in *European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2008, pp. 705–718.

[21] T. Schmiedel, E. Einhorn, and H. M. Gross, "IRON: A fast interest point descriptor for robust NDT-map matching and its application to robot localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3144–3151.

[22] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3212–3217.

[23] F. Tombari, S. Salti, and L. Di Stefano, "Unique Signatures of Histograms for Local Surface Description," in *European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2010, pp. 356–369.

[24] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 2155–2162.

[25] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[26] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[27] C. Dubout and F. Fleuret, "Exact Acceleration of Linear Object Detectors," in *European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2012, pp. 301–311.

[28] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, April 2012.