People Detection in 3*d* Point Clouds using Local Surface Normals

Frederik Hegger, Nico Hochgeschwender,
Gerhard K. Kraetzschmar and Paul G. Ploeger

Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany
{frederik.hegger, nico.hochgeschwender, gerhard.kraetzschmar, paul.ploeger}@h-brs.de

Abstract. The ability to detect people in domestic and unconstrained environments is crucial for every service robot. The knowledge where people are is required to perform several tasks such as navigation with dynamic obstacle avoidance and human-robot-interaction. In this paper we propose a people detection approach based on 3*d* data provided by a RGB-D camera. We introduce a novel 3*d* feature descriptor based on Local Surface Normals (LSN) which is used to learn a classifier in a supervised machine learning manner. In order to increase the systems flexibility and to detect people even under partial occlusion we introduce a top-down/bottom-up segmentation. We deployed the people detection system on a real-world service robot operating at a reasonable frame rate of $5Hz$. The experimental results show that our approach is able to detect persons in various poses and motions such as sitting, walking, and running.

Keywords: Human-Robot Interaction, People Detection, RGB-D

1 Introduction

Domestic service robots such as the Care-O-bot 3 [5] and PR2 [3] are deployed more and more in realistic, unconstrained, and unknown environments such as offices and households. In contrast to artificial environments the real-world is populated with humans which are walking, sitting, and running. In order to interact in such environments with humans in a safe manner a service robot must be aware of their positions, movements, and actions. Therefore, a robust people detection system is crucial for every domestic service robot. An appropriate sensor type providing perceptual information about the environment is required to detect people in a robust and reliable manner. Quite recently, a new type of cameras has been become available, namely *RGB-D* cameras such as the *Microsoft Kinect*[1] and *Asus Xtion*[2]. Those cameras provide a 3*d* point cloud and additional RGB values at the same time. The low-cost and the high frequency of $30Hz$ makes the *Kinect* very attractive for the robotics community. Hence,

[1] www.xbox.com/kinect
[2] www.asus.com

the people detection approach described in this paper proposes to use and processes RGB-D data provided by a *Kinect* sensor. Previous contributions in the field of people detection are based on range data provided by laser range finders. In [1] Arras et al. proposed a system which segments a complete scan into smaller clusters, extracting geometric features, and then classifies the cluster in human or non-human. The classifier has been created in a supervised machine learning manner with methods such as AdaBoost and SVM. This principle has been extended in [7] by mounting two additional laser range finders on different height in order to retrieve a more sophisticated view on the scene. Spinello et al. [11] extended this idea by extracting a fixed set of $2d$ vertical scan lines from a full $3d$ point cloud. The detection is performed in each layer separately. The layers are later fused with a probabilistic voting scheme. Other approaches are based on vision as a primary modality. They apply well-known techniques such as implicit shape models [13], haar-like features [14], or histogram of oriented gradients [12] for feature extraction. However, all these approaches operate only in $2d$ space. First approaches operating in $3d$ are described by Satake et al. [9] where template matching (depth templates) is used to detect the upper body of humans. In [2] and [8] the $3d$ point cloud is first reduced to a $2.5d$ map in order to keep the computational effort low. The classification itself is again based on different $2d$ features and a machine learning classifier. The approach which comes close to our approach has been introduced by Spinello and Arras [10]. In a hybrid manner the detection is based on a combination of $3d$ depth and $2d$ image data. Inspired from a histogram of oriented gradients (HOG) detector Spinello proposes a novel histogram of oriented depths (HOD) for the detection in $3d$. Both information (HOG and HOD) are fused which yields in a robust and later GPU-optimized people detection system.

In contrast to [10], our approach uses only the $3d$ point cloud provided by a Microsoft Kinect camera. The data is split into smaller clusters using a layered sub-division of the scene and a top-down/bottom-up segmentation technique. A random forest classifier is used to label the resulting $3d$ clusters either as human or non-human. Inspired from [6], we extended the idea of using local surface normals (LSN) and composed a new feature vector based on a histogram of local surface normals plus additional $2d$ and $3d$ statistical features. An overview of the complete processing pipeline is depicted in Figure 1. The major contribution of our approach is a novel feature descriptor based on local surface normals and the capability to robustly detect persons in various poses/motions, even if they are partially occluded like sitting behind a table or desk.

2 People Detection using Local Surface Normals

In this section we introduce our $3d$ people detection approach using Local Surface Normals (LSN). The approach consists of four phases as shown in Figure 1, namely *Preprocessing*, *Top-Down Segmentation*, *Classification*, and *Bottom-Up Segmentation*.

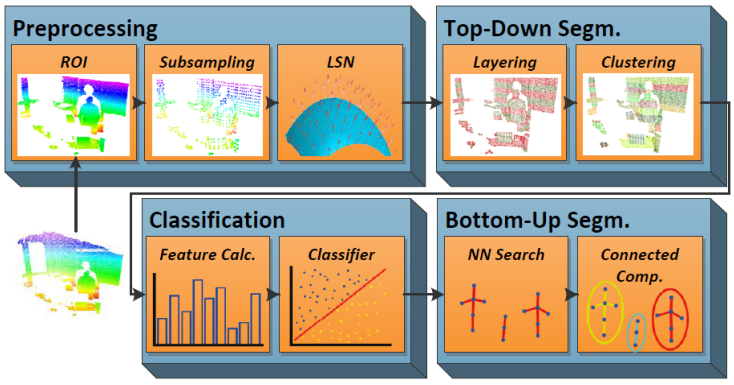


Fig. 1. The processing pipeline is divided into four phases (*blue boxes*). Each phase consists of several sub-components (*orange boxes*) which perform the actual computation of the input data.

2.1 Preprocessing

A single point cloud from the Kinect sensor consists of $\approx$ 300.000 points. In order to keep the overall processing time reasonable we carefully reduced the raw input data in the preprocessing phase.

Region of Interest (ROI). A major disadvantage of the Kinect camera is the increasing depth discretization error for large distances. Beyond $5m$ the depth values are very noisy and shaky. Therefore, the ROI is defined as $0.5m <= depth <= 5.0m$ and $0.0m <= height <= 2.0m$. The height has been choosen because people usually appear in this range. The ROI steps already reduces (depending on the actual scene) the point cloud to $\approx$ 110.000 points in average.

Subsampling. The remaining points provided by the ROI step are further reduced by a subsampling routine to make the point cloud more sparse, i.e. a $3d$ grid with a predefined cell size is overlayed over the full point cloud. The points inside each box are merged to a single new point. An increased cell size will yield to a sparse point cloud. We have used a cell size of $3cm$ x $3cm$ x $3cm$ which still maintains the desired accuracy for the normal estimation and simultaneously reduces the point cloud to $\approx$ 16.000 points in average.

Local Surface Normals (LSN). In the classification phase (see Section 2.3) we propose a feature vector which consists of a histogram of local surface normals. A local surface normal is computed through fitting a plane to the k-nearest neighbors of the target point. A more detailed description of the algorithm can be found in [6]. Before the preprocessed point cloud is forwarded to the segmentation phase, for all remaining points the local surface normals are computed. In case the normals would be calculated after the segmentation, the accuracy of the normal estimation for those points which lie on the border of a cluster would be significant lower. A reasonable part of the neighborhood might already belong to another cluster.

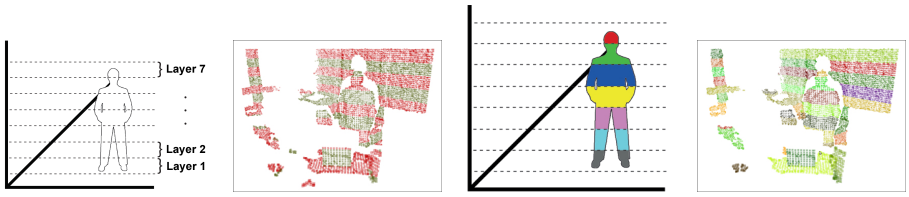
2.2 Top-Down Segmentation

The segmentation of large $3d$ point clouds is a (computational) costly and complex exercise. Segmentation approaches such as region growing or graph-based approaches are known to have a huge computational complexity. Therefore, such approaches are not feasible in robotics where reasonable performance is crucial.

Layering. We propose a basic top-down segmentation technique (see Figure 2). The general idea is decompose the point cloud into a fixed set of different $3d$ height layers and then start to segment each layer separately in smaller clusters. In detail, the layering and segmentation algorithm can be explained as follows: Let $\mathbf{P} = \{p_1, ..., p_N\}$ be a point cloud with $p_i = (x, y, z)$ and N which is equal to the number of points in the point cloud. Then $\mathbf{P}$ is split into a fixed number of $3d$ layers $\mathbf{L} = \{l_1, ..., l_M\}$ with

$$M = \frac{(Z_{max} - Z_{max})}{SH}$$

where Z_{min} and Z_{max} are the minimum and maximum height values of the predefined ROI and SH is the desired slice height. For each layer l_j the minimum and maximum height is calculated. For instance, assuming a predefined slice height of $20cm$ then the first layer l_1 contains only points with $0.0m <= p_i(z) <= 0.2m$. The remaining layers $l_2, ...l_M$ will be established according to this principle. As experimentally validated we consider a slice height of $25cm$ as optimal (see Section 3).



(a) schematic layering (b) layering result (c) schematic segmentation (d) segmentation result

Fig. 2. Images *(a)* and *(b)* show the layering process, where each point cloud is divided into a set of $3d$ layers according to a manually defined slice height. For the layering, we have applied a slice height of $25cm$. Each layer is segmented into clusters using a Euclidean Clustering approach (see Image *(c)* and *(d)*). The different colored points indicate either the different height layers or the segmented $3d$ clusters.

Clustering. The actual segmentation generates for each layer l_j a sequence of small clusters $\mathbf{C} = \{c_1, ..., c_O\}$, where each cluster $c_{j,k}$ contains a subset of points located in l_k. The segmentation applies an Euclidean clustering technique which is less parameterizable. Only a distance threshold $thres_{EuclDist}$ has to be

defined which defines whether a target point is added to the cluster or not. Furthermore, $thres_{EuclDist}$ also determines whether there are many small clusters ($thres_{EuclDist} \leftarrow 0$) or only a few large clusters ($thres_{EuclDist} \rightarrow \infty$). As mentioned, we have used a grid-size of $3cm$ for subsampling. According to this dimensions and a certain amount of noise, we set $thres_{EuclDist} = 2 \times grid\_size$ in order to ensure that two persons which stand close to each other are not merged to a single cluster. The proposed fine-grained clustering has the advantage over a clustering *without* prior layering when one object is partially occluded by another object. For instance, if a person is sitting at a table, our approach creates several smaller clusters for both objects. Instead, the pure Euclidean clustering would create a single cluster which consists of a table and the person, because the person is sitting very close to the table or has put the arms on it. Furthermore, the user-defined slice height plays also an important role for the performance of the segmentation. A reasonable small height ends up in really tiny clusters with few local surface normals which are not sufficient for a robust classification. On the other hand, a large slice height creates also large clusters (where two or more objects would get merged to a single cluster) which would alleviate the specific advantage of the proposed segmentation stage.

2.3 Classification of $3d$ Clusters

The previous segmentation phase produces a list of $3d$ clusters. In the classification phase we want to assign a label to each cluster (human or non-human). We approached the two-class classification problem with a supervised machine learning technique. We evaluated the performance of three popular machine learner on different datasets recorded in different environments, namely AdaBosst, SVM and Random Forests [4]. The results showed that for all datasets the Random Forest classifier outperforms both other machine learning techniques.

Feature Calculation. As a feature vector for the Random Forest we propose a histogram of local surface normals (HLSN). The use of such a feature vector can be motivated as follows: households and offices contain to a large extend walls, tables, desks, shelfs, and chairs. More precisely, a reasonable part of daily environments consists of horizontal and vertical planes. Whereas the human body has a more cylindrical appearance. With a histogram of LSNs we can express this property to distinguish between human and non-human clusters. We compute a fix-sized histogram over the normals for all points in a cluster which is the input for a feature vector. However, the Random Forests algorithm expects a one dimensional input vector. Therefore, a separate histogram for each normal axis (x, y and z) is established. In addition, the width and the depth of a cluster is added to the feature vector, which helps to decrease the false positive rate.

Classifier. Learning the Random Forest classifier requires a large-set of training samples. As in other fields the collection of positive and negative training samples is a time consuming task, especially when many samples (> 1000) are required and the annotation of each sample has to be done manually. Therefore, we integrated a procedure to capture positive and negative training samples au-

tomatically. Negative samples have been collected with a mobile service robot. We established a map of our University building which at least consisted of an office, laboratory, long corridor and an apartment. For each room a navigation goal has been manually annotated. An automatic procedure generated a random order in which the rooms should be visited. The robot started to navigate autonomously through all the environments and simultaneously segmenting each incoming point cloud. Each extracted cluster has been labeled as negative example. During the whole run we ensured that there has been no person in the field-of-view (FOV) of the robot. This process guarantees that the samples are indeed collected in a random manner. The positive samples have been collected with a static mounted Kinect camera. The camera was placed in a laboratory where people are frequently walking and sitting around. We defined a ROI which does not contain any object and consequently provides an empty point cloud. In case the person passed the ROI, the segmentation stage extracted the related clusters and labeled them as positive samples.

2.4 Bottom-Up Segmentation

In the last phase we obtain a sequence of $3d$ clusters which are classified as human. However, those "part-based" detections have to be assembled and associated to one respective person. A graph-based representation based on the cluster's center is created. The advantage is that not the whole data points of a cluster have to be processed which keeps the computational effort low. Each center point is then connected to its two nearest neighbors as long as the Euclidean distance between those points does not exceed a certain threshold. Each cluster has always a maximum height (equal to the predefined slice height) which allows us to derive the threshold, because the center points of two neighboring clusters can only have a maximum distance of $2 \times slice\_height$. When all the points in the queue have been processed the overall graph can be split in its connected components, which builds the actual person detection. Due to false positive detection when classifying the extracted $3d$ clusters, we consider a successful person detection only, if at least three clusters belong to one person (at least $\approx 45cm$ of the persons body must visible).

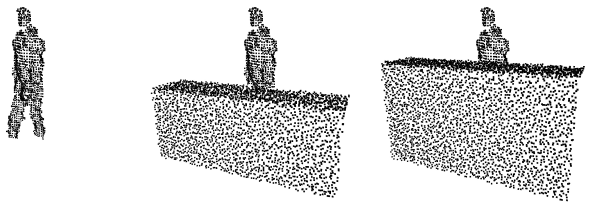
3 Experimental Evaluation

In order to evaluate the proposed people detection system we performed several experiments with different objectives as described below.

3.1 Experiment Objectives

Objective 1. Investigate the impact of the predefined slice height on the classification error.
The segmentation is based on separating the point cloud into several fix-sized

layers. The amount of layers depends on the chosen slice height. In this experiment we investigated the impact of the predefined slice height on the resulting classification error. The experiment was executed several times with different slice heights ranging from $10cm$ to $100cm$ (= half of the maximum perceivable height). Every range value below $10cm$ results in very few points which is not sufficient to represent a comprehensive distribution. Thus one requirement for the people detection approach is the ability to detect people even if they are partially occluded. In each experiment the slice height is constantly increased by $5cm$ (when starting at the minimum). A 10-fold cross-validation was applied. In order to evaluate the segmentation behavior against occlusion, synthetic generated occlusion (e.g., a cupboard) was added to the data. The experiment was repeated three times with different amount of occlusion, namely no occlusion, 50%, and 70% (see also Figure 3). Moreover, Gaussian noise was added to the synthetic data in order to achieve approximation to the Kinect data.



(a) No occlusion (b) 50% occlusion (c) 70% occlusion

Fig. 3. Different occlusion levels.

Objective 2. Investigate the actual people detection performance.
In order to assess the detection rates under different circumstances we defined two categories, namely poses and motions. For the pose category we evaluated the detection rate for persons sitting on a chair and for persons which where partially occluded (at least 30% of the whole body). For the motion category we evaluated three different natural motions: not moving, random walking, and random running. We executed the experiment with ten subjects in three different environments. In our RoboCup@Home laboratory, a real German living room, and the entrance of our University where people frequently enter and leave the building. The test procedure (or test cases) looked as follows:

1. **Standing pose:** the persons were asked to position themselves in various random positions and usual body postures.
2. **Sitting pose:** the persons were asked to sit down on a chair and position themselves in various random positions and usual sitting postures.
3. **Partially occluded pose:** the persons were asked to stand behind a cupboard of 80 cm height and to move up and down in a natural way.

4. **Not moving motion:** it is identical to the test for standing person and only mentioned for completeness.
5. **Random walking motion:** the test was execute at the entrance of our University. Many people were entering and leaving the building. Even sometimes in small groups.
6. **Random running motion:** the persons were asked to run in a jogging manner through the FOV of the camera in various paths.

For each of the ten persons and the corresponding posture/motion 200 frames have been evaluated. To avoid manual annotation a simplified change detection was applied. Initially the point cloud size (after ROI building) of ten subsequent frame has been averaged and stored. In the evaluation phase the size of the recent acquired point cloud is compared to the stored size. If the difference is above certain threshold, the person has entered the cameras FOV. This simplified evaluation was applied for the test cases 2, 3 and 6. In case of test case 1 and 4, we waited until the person reached a new position and then evaluated each time five frames. For test case 5, each frame had to be manually annotated since the number of persons in the FOV was varying between one and five during the whole test.

Objective 3. How does the people detection system behave in a scenario-like setting.
So far the people detection system has been evaluated stand-alone. However, we are interested in how the system behaves when it is integrated on a real-world domestic service robot. We have integrated the system on our Care-O-bot 3 robot and performed a more scenario-like evaluation, where an autonomous mobile service robot tries to find a predefined number of persons in the environment. The scenario is basically derived from the "Who is who?" test in the RoboCup@Home competition where five people are spread around in the apartment. As an initial knowledge, the robot has a map of the environment and a set of room poses for each part of the apartment (e.g., living room or kitchen). In our test implementation a script first generates random positions in the map for five persons (also defining whether the person should sit or stand). In case the proposed position is blocked (e.g., a wall or table) the person will be assigned to stand/sit next to the generated pose. When all persons are placed at the generated positions, the robot generates a random path through all available room poses. The rest of the experiment consists of executing a drive & search behavior which we have implemented for the RoboCup@Home competition.

3.2 Experiment Results

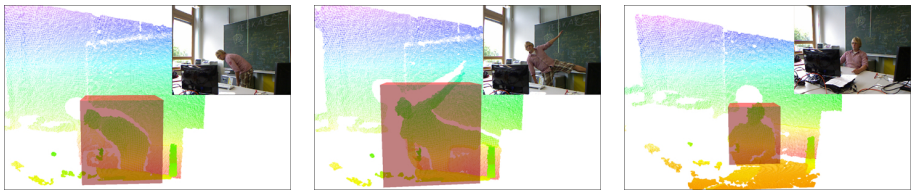
Objective 1. Figure 5 depicts the cross-validation error with respect to the actual slice height. In case of no occlusion of the actual person the classification decreases with an increasing slice height. Above $50cm$ the error converges to an error rate of $\approx 15\%$. However, occlusion causes a major increase of the error rate when applied to an increased slice height. The reason is that the segmentation with high slice height creates clusters which might contain parts of the human

and part of the object which causes the occlusion. We used the experiment to determine a good (minimized error rate) slice height. Thereby, we calculated the mean curvature for all three error curves and identified the global minima. A slice height of 25cm yielded in the minimum averaged error of 15.49%.

Objective 2. As shown in Table 1 our system shows a quite robust performance at least for standing person. In Figure 4 some detections for person poses are shown. The performance is independent from the actual distance to the person and is only limited by the predefined maximum distance of 5 meters. However, we observed a degrading detection rate when the person is sitting. The detection rate is significant lower, namely 74.94%. This is due to the fact that the training was only performed with standing persons. Therefore, only the head and the upper body can be detected. The horizontal leg parts can not be detected.

Poses	Detection Rate	Motions	Detection Rate
standing	87.29%	not moving	87.29%
sitting	74.94%	rnd. walk	86.32%
part. occl.	82.35%	rnd. run	86.71%

Table 1. Detection rates for different human poses and motions.



(a) bending (b) random pose (c) partially occluded + sitting

Fig. 4. Detections for various person poses

In case the Random Forest would have trained also with sitting person, there would be clusters whose normal distribution would be similar to horizontal planes (because the upper leg is parallel aligned). Of course, this would cause a very high false positive rate. However, when a person is sitting, the upper body is still visible and sufficient for a quite robust detection with the model trained only with standing persons. Although, it is significant lower than detecting standing persons. Persons which were partially occluded, e.g. behind a table or a cupboard, can be detected similar robust to standing person, because only a minority of the lower body is occluded. For different motion speeds, only slightly different results could be observed. It does not matter in which speed the person is moving

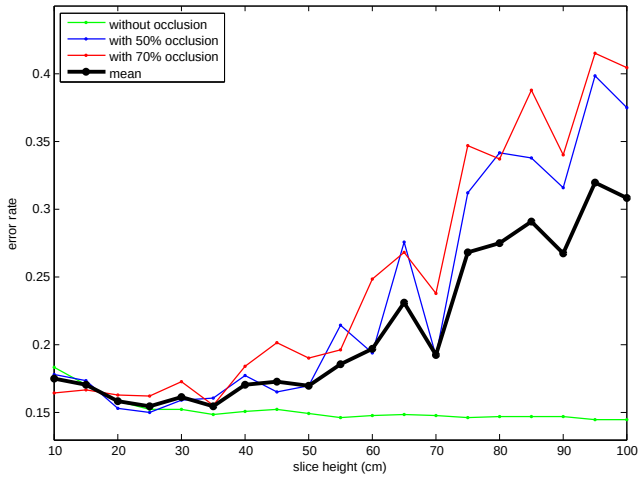


Fig. 5. Error rates of the segmentation in the presence of occlusion.

or even standing still, since the detection is done frame by frame. Only the pose configuration is different for the different motions. In general, the experiment showed that people are detected in various pose configurations and speeds with an average detection rate of 84.15%. A short video showing the people detection can be found on: `http://www.youtube.com/watch?v=d0O4nQE8Qko`.

Objective 3. In total ten runs of the described experiment were executed (see Table 2). In all cases the robot was able to find at least the two standing persons and always one sitting person. The missed detections where caused by a occlusion through another person or when the person was sitting in an arm chair and only a small part of the shoulder and head was visible. Beside the successful and missing detections, there were quite a lot false positive detections. In each run at least one false positive detection occurred. Due to the fact that a detected person (in this cases a false detection) is approached only once and then stored, the false detections do not effect the overall performance so much. Only the time for approaching the false detection for the first time is gone. However, in other scenarios this effect could result in a worst performance. Nevertheless, the integration of the people detection component into a higher level behavior was able to successfully detect the majority of people in the environment. Standing people could be detected with a rate of 86.67% and sitting person with 75.00% in this experiment. Astonishingly, the detection rates from this experiment almost reflect the results acquired in the experiment for the second objective.

4 Conclusion

We presented an approach to detect the $3d$ position of people in $3d$ point clouds using a feature vector which is composed of a histogram of local surface nor-

Run	TP standing	TP sitting	FN standing	FN sitting	FP
1	3	2	0	0	2
2	3	1	1	1	2
3	2	2	1	0	1
4	3	2	0	0	2
5	2	1	1	1	2
6	2	2	1	0	1
7	2	1	1	1	1
8	3	2	0	0	2
9	3	1	0	1	2
10	3	1	0	1	1

Table 2. Result of 10 executed runs with auto-generated person positions (three standing and two sitting). TP = true positives, FN = false negatives, FP = false positives.

mals. The preliminary segmentation is based on a top-down/bottom-up technique which supports the detection of partially occluded persons, e.g. standing behind a desk or cupboard. The information gained from the local surface normals enables our system to detect a person in various poses and motions, e.g., sitting on other objects, bended to the front or side, walking fast/slow. With the presented approach we are able to detect even multiple people up to a distance of 5m with a detection rate of 84%. Future improvements will cover a reduction of false positive detections by extending the existing feature set with additional geometrical and statistical features. The proposed approach covered only the detection of people in $3d$, a $3d$ tracking system would also enhance the overall system performance. We further aim an implementation on GPU, in order to improve the processing performance towards a real-time system. Another step would be the integration of color information into the detection process, which is provided simultaneously with the point cloud data by the Kinect sensor.

Acknowledgement. We gratefully acknowledge the continued support by the b-it Bonn-Aachen International Center for Information Technology.

References

1. Kai Oliver Arras, Oscar Martinez Mozos, and Wolfram Burgard. Using Boosted Features for the Detection of People in 2D Range Data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3402–3407, Rome, Italy, 2007.
2. M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. H. Matthies. A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle. *The International Journal of Robotics Research*, 28(11-12):1466–1485, July 2009.

3. J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
4. Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.
5. B. Graf, U. Reiser, M. Hagele, K. Mauz, and P. Klein. Robotic home assistant care-o-bot 3 - product vision and innovation platform. In *Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 2009.
6. Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-Time Plane Segmentation using RGB-D Cameras. In *Proceedings of the 15th RoboCup International Symposium*, Istanbul, Turkey, 2011.
7. Oscar Martinez Mozos, Ryo Kurazume, and Tsutomu Hasegawa. Multi-Layer People Detection using 2D Range Data. In *Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*, Kobe, Japan, 2009.
8. L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian Detection and Tracking Using Three-dimensional LADAR Data. *The International Journal of Robotics Research*, 29(12):1516–1528, May 2010.
9. Junji Satake and Jun Miura. Robust Stereo-Based Person Detection and Tracking for a Person Following Robot. In *Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*, Kobe, Japan, 2009.
10. Luciano Spinello and Kai Oliver Arras. People Detection in RGB-D Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*, San Francisco, USA, 2011.
11. Luciano Spinello, Kai Oliver Arras, R. Triebel, Roland Siegwart, M. Luber, G.D. Tipaldi, B. Lau, Wolfram Burgard, and Others. A Layered Approach to People Detection in 3D Range Data. In *IEEE International Conference on Robotics and Automation*, volume 55, pages 30–38, Anchorage, Alaska, 2010.
12. Luciano Spinello and Roland Siegwart. Human Detection using Multimodal and Multidimensional Features. In *Proceedings of the International Conference in Robotics and Automation (ICRA)*, Pasadena, USA, 2008.
13. Luciano Spinello, Roland Siegwart, and Rudolph Triebel. Multimodal People Detection and Tracking in Crowded Scenes. In *Proc. the AAAI Conf. on Artificial Intelligence: Physically Grounded AI Track*, pages 1409–1414, Chicago, USA, 2008.
14. Zoran Zivkovic and Ben Kroese. Part based People Detection using 2D Range Data and Images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 214–219, San Diego, USA, 2007.