Fast and Robust Multi-people Tracking from RGB-D Data for a Mobile Robot

Filippo Basso*, Matteo Munaro*, Stefano Michieletto,
Enrico Pagello, and Emanuele Menegatti

Department of Information Engineering of the University of Padova,
via Gradenigo 6B, 35131 - Padova, Italy
{bassofil,munaro,michieletto,epv,emg}@dei.unipd.it

Abstract. This paper proposes a fast and robust multi-people tracking algorithm for mobile platforms equipped with a RGB-D sensor. Our approach features an efficient point cloud depth-based clustering, an HOG-like classification to robustly initialize a person tracking and a person classifier with online learning to manage the person ID matching even after a full occlusion. For people detection, we make the assumption that people move on a ground plane. Tests are presented on a challenging real-world indoor environment and results have been evaluated with the CLEAR MOT metrics. Our algorithm proved to correctly track 96% of people with very limited ID switches and few false positives, with an average frame rate of 25 fps. Moreover, its applicability to robot-people following tasks have been tested and discussed.

Keywords: People tracking, real-time, RGB-D data, mobile robots.

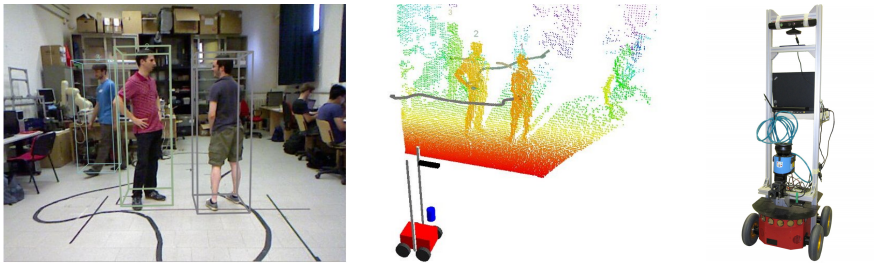
1 Introduction and Related Work

People detection and tracking are key abilities for a mobile robot acting in populated environments. Such a robot must be able to distinguish people from other obstacles, predict their future positions and plan its motion in a human-aware fashion, according to its tasks.

Many works exist about people detection and tracking by using RGB cameras only ([6], [21], [4]) or 3D sensors only ([14], [19], [20], [5], [15]). However, when dealing with mobile robots, the need for robustness and real time capabilities usually led researchers to tackle these problems by combining appearance and depth information. In [2], both a PTZ camera and a laser range finder are used in order to combine the observations coming from a face detector and a leg detector, while in [13] the authors propose a probabilistic aggregation scheme for fusing data coming from an omnidirectional camera, a laser range finder and a sonar system. Ess *et al.* [7],[8] describe a tracking-by-detection approach based on a multi-hypothesis framework for tracking multiple people in busy environments from data coming by a synchronized camera pair. The depth estimation provided

* These authors contributed equally to this work.

S. Lee et al. (Eds.): Intelligent Autonomous Systems 12, AISC 193, pp. 265–276.
springerlink.com



(a) (b) (c)

Fig. 1. Example of our system output: (a) a 3D bounding box is drawn for every tracked person on the RGB image, (b) the corresponding 3D point cloud is reported, together with people trajectories. In (c) the mobile platform we used for the experiments is shown. Note that in this paper we only employ RGB-D data from a Microsoft Kinect sensor and do not make use of other sensors such as Laser Range Finder or sonars.

by the stereo pair allowed them to reach good results in challenging scenarios, but their algorithm reached real time performance if one does not take into account the time needed by their people detection algorithm which needs 30s to process each image. Stereo cameras continue to be widely used in the robotics community ([1],[17]), but the computations needed for creating the disparity map always impose limitations to the maximum frame rate achievable, thus leaving less room for further algorithms operating in series with the tracking one.

With the advent of reliable and affordable RGB-D sensors a rapid boosting of robots capabilities can be envisioned. For example, the Microsoft Kinect sensor allows to natively capture RGB and depth information at good resolution (640x480 pixels) and frame rate (30 frames per second). Even though the depth estimation becomes very poor over eight meters of distance and this technology cannot be used outdoors because the sunlight can change the infrared pattern projected by the sensor, it constitutes a very rich source of information that can be simply used on a mobile platform. In [12] a tracking algorithm on RGB-D data is proposed that exploits the multi-cue people detection approach described in [18]. It adopts an on-line detector that learns individual target models and a multi-hypothesis decisional framework. No information is given about the computational time needed by the algorithm and results are reported for some sequences acquired from a static platform equipped with three RGB-D sensors.

In this work, we propose a multi-people tracking algorithm with RGB-D data specifically designed for mobile platforms. By assuming that people are moving on a ground plane, our method is able to robustly track them with a medium frame rate of 25 frames per second by relying only on CPU computation. We tested our approach on sequences of increasing complexity, reaching very good results even when dealing with complex people trajectories and strong occlusions. The track initialization procedure, which relies on a HOG people detector, allows to minimize the number of false positives and the online learning person classifier

is used every time a person is lost, in order to recover the correct person ID even after a full occlusion.

The remainder of the paper is organized as follows: in Section 2 the various parts of our approach are introduced. Section 3 describes the downsampling scheme that we adopt for reaching real time performances, while the detection phase is described in Section 4. Section 5 details the tracking procedure and in Section 6 we describe the tests performed and we report the results evaluated with the CLEAR MOT metrics. Conclusions and future works are contained in Section 7.

2 System Overview

In this section, we outline the different software modules of our people tracking system. As reported in Fig. 2, the RGB-D data are processed by three main software blocks: filtering, detection and tracking. The filtering block consists in a smart down-sampling of the 3D point cloud. This is a crucial step for reducing the size of the big amount of data we deal with. The detection block performs the necessary operations for clustering the remaining points and selecting the clusters of points containing people, that are subsequently passed to the tracking module. In the last block, the clusters are associated to the correct track according to the predictions of an Unscented Kalman Filter and, whether the filter fails because of full occlusions, an online-learned person classifier is able to recover the correct association between the person and the track when the person return visible.

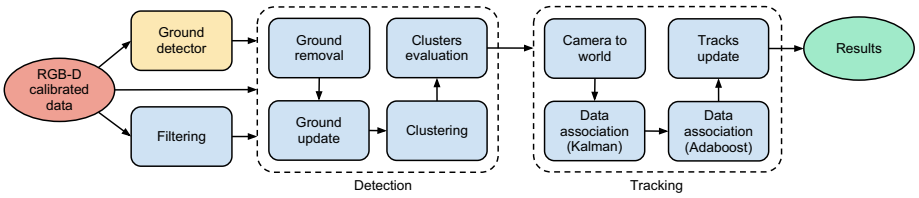


Fig. 2. System overview highlighting the main software blocks

3 Filtering

At each frame, the data provided by the RGB-D sensor are processed by a voxel grid filter, i.e. the space is subdivided in a set of tiny 3D boxes and all points inside each box are approximated with the coordinates of their centroid. By default, we chose the voxel size to be of 0.06m. This value allowed us to downsize the point cloud of our RGB-D sensor by an order of magnitude, thus reaching high real time performance and having enough data for performing the tracking procedure. Moreover, after this operation, points density no longer depends on their distances from the sensor.

4 Detection

4.1 Clustering

In order to divide the scene into different clusters, our algorithm remove all 3D points belonging to the floor from the output of the previous software block. In the first frame, the ground plane equation is estimated from an initial input from the user (i.e. click on three points of the ground plane in the image). Then, at every frame, this equation is updated by considering as initial condition the estimation at the previous frame. All points of the cloud within a threshold distance from the hypothetical plane are selected. Then, using a least squared method on these points, the plane coefficients are refined. This procedure permits to adapt to small changes in the floor slope and to the camera oscillations during robot movements. Once this operation has been performed, the different clusters are no longer connected through the floor, so they can be easily calculated by labeling neighboring 3D points on the basis of their Euclidean distances.

4.2 Clusters Evaluation

The objective of clusters evaluation is to preserve only those clusters containing a person and extract a series of features necessary for the subsequent tracking phase. For each cluster, we estimate: the height from the ground plane, the centroid, the distance from the sensor and the corresponding blob in the RGB image. Clusters with height out of a plausible range for an adult person[1] are discarded before computing the subsequent features and do not pass to the tracking phase. Moreover, we determine if a cluster is occluded by any other cluster or it is partially out of the camera view. When a cluster is not occluded, we compute also its HOG confidence. This consists in the confidence obtained by applying a HOG people detector [6] to the part of the RGB image corresponding to the cluster theoretical bounding box, that is the bounding box that should contain the whole person, from the head to the ground. For the people detector, we used Dollár's implementation of HOG[2] and the same procedure and parameters described by Dalal and Triggs [6] for training the detector.

5 Tracking

In this section we present the overall tracking approach we have designed. From now on, the positions relative to the detections (clusters) found in the previous phase are considered in world coordinates with respect to the starting position of the platform[3].

[1] In this work we considered clusters with height in the range 1.4 - 2.3m.
[2] Contained in his Matlab toolbox `http://vision.ucsd.edu/~pdollar/toolbox`
[3] Our platform estimates its position by means of odometry and a self localization system not described here.

5.1 Motion Model

Since we use a tracking-by-detection approach and our detector has been designed to minimize false negatives, we just choose to use an Unscented Kalman Filter to maintain people positions and velocities along the two world axes (x, y).

There are many methods to model human motion in tracking. However, a good way to manage full occlusions is to consider a constant velocity model, as described in [2]. For what concerns the observation model, and in particular the covariance matrix of the measurement noise, we expressed it as a sum of two different components, σ_v^2 and σ_d^2. σ_v^2 models the quantization error introduced by the voxel filter[4] while σ_d^2 models the measurement error introduced by the depth sensor. This error, for our particular sensor, is proportional to the distance squared[5]. Among the different extensions to the basic Kalman filter, we decided to use the Unscented version because it has prediction capabilities near those of a particle filter, but it is only slightly more computationally expensive than an Extended Kalman Filter [2].

5.2 Online Classifier

To overcome the limits imposed by the constant velocity model when dealing with full occlusions, we maintain for each track an online classifier based on Adaboost, like the one used in [10] or [12]. But, unlike these two approaches, that make use of features directly computed on the RGB (or depth) image, we calculate our features in the color histogram of the target, as following:

1. we select the image pixels belonging to the person by exploiting the blob mask given by the detector;
2. we compute the 3D color histogram of these points on the three RGB channels;
3. we select a set of randomized parallelepipeds (one for each weak classifier) inside the histogram. The feature value is given by the sum of histogram elements that fall inside a given parallelepiped.

For the training phase, since the approach introduced by [10] gave us poor performances, we use as positive sample the color histogram of the target, while as negative samples we consider the histograms calculated on the detections not associated to the current track. This approach has the advantage to select only the colors that really characterize the target and distinguish it from all the others.

5.3 Data Association

Once a new set of detections is available, we perform the detection-track association with two subsequent steps: at first only the prediction given by the Kalman

[4] This error is considered to be uniformly distributed between 0 and the voxel size multiplied by $\sqrt{3}$, namely the voxel diagonal.

[5] `http://www.ros.org/wiki/openni_kinect/kinect_accuracy`

filter is considered, while, as a second step, we consider only the results given by the classifier. We perform the data association based on the Kalman filter prediction with the Global Nearest Neighbor method, described in [11] and [2]. For each detection, we calculate the Mahalanobis distance from the predicted state of all the active tracks, we create the cost matrix based on this distance and then perform the association with the Munkres algorithm. Only at this point the remaining detections and tracks are considered for the association based on the classifier. If, after this step, there are some detections that are still unassociated, they are taken into account for the creation of new tracks.

We have not yet implemented more sophisticated methods, such as multiple hypothesis tracking or probabilistic data association filters, because in the considered scenarios the potential increase in performance would not compensate for the considerably higher computational load.

5.4 Tracks Management

The policies of creation/update/deletion of the tracks are really important to get good results from the whole tracking process. They can be summarized as following.

5.4.1 Initialization

A new track is created from an unassociated detection if the confidence value given by the HOG classifier is over a defined threshold for N frames. In our case N has been set to three. If this happen, the track is promoted to "validated", otherwise it is discarded.

5.4.2 Update

After the detection-track association, the Kalman filter is updated with the measurement of the theoretical centroid of the cluster. Then, if the cluster is not occluded also the classifier is updated. This is given by the fact that, when a person is occluded, some colors can be missing, updating the classifier would decrease the importance of these color features, thus distorting the results of successive classifications.

5.4.3 Recovery

After a full occlusion, a person could be wrongly assigned to a new track instead of the old one. This happens if neither the Kalman filter, nor the classifier correctly associate the new detection to the previous track. But, while the trajectory error of the Kalman filter cannot be corrected, the classifier can manage to recover from this mistake. In particular, during the first frames after its creation, the new track histograms are evaluated by the classifiers of the missing tracks and, if the result is above a determined threshold, the new track is recognized to refer to an old one and associated to it. An example of this behavior is depicted in Fig. 3. From left to right it can be seen that the person associated to track #2 goes out of the field of view of the camera and, when it comes back in, it is

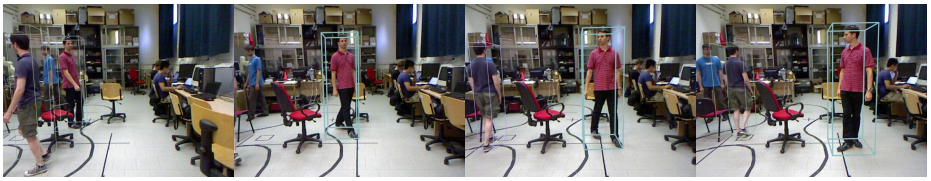


Fig. 3. From left to right, an example of track recovery. Track #2 becomes #4 after the full occlusion, but, in a few frames, it returns to its original ID.

assigned to a new track (#4), but, in a few frames, it returns to its original ID (#2).

5.4.4 Removal

After a person becomes occluded or goes out of the scene, the correspondent track is marked as missing. A track is deleted and no more considered if it remains in that state for a certain number of consecutive frames or, as described above, if it is not validated before time runs out.

6 Experiments

6.1 Experimental Setup

In this section we show the performance of our system on RGB-D video sequences collected in an indoor environment with the mobile robot shown in Fig. 1(c). It consists of a Pioneer P3-AT platform equipped with a Microsoft Kinect sensor, that is endowed with a standard RGB camera, an infrared camera and an infrared projector. This low cost hardware can provide RGB-D data with 640 x 480 pixel resolution at 30 frames per second. In these tests we acquired depth data at a reduced resolution, namely 160 x 120 pixel, for speed.

We performed tests both while keeping our platform as static and while moving it on a predefined path. For both cases we acquired videos in three different scenarios of increasing difficulty:

1. no obstacle is present, people move with simple (linear) trajectories;
2. no obstacle is present, people move with complex trajectories and interact with each other;
3. obstacles are present, people move with complex trajectories and interact with each other.

Every video sequence extends over about 750 frames, thus the total test set includes 4671 frames, 12272 instances of people and 26 tracks that have been manually annotated on the RGB image and that constitute the ground truth. The minimum distance between people is 0.2m while the minimum people-object distance is 0.05m.

Table 1. Frame rates for two test videos with our moving platform

	Kinect Data	**Detector**	**Detector+Tracker**
Complex traj.	29.971	26.268	26.215
With obstacles	29.967	25.243	25.227

6.2 Runtime Performance

The entire system is implemented in C++ within ROS, the Robot Operating System[6], making use of highly optimized libraries for 2D computer vision[7], 3D point cloud processing[8] [16] and bayesian estimation[9]. On a notebook with an Intel i5-520M 2.40GHz processor and 4GB of memory, we measured the frame rates reported in Table 1. They indicate the average rates reached by the Kinect data publication, the detection phase and our complete system (detection and tracking). These tests refer to two videos captured with our moving platform. A small frame rate reduction can be noticed when some obstacles are present in the scene. This happen because more clusters have to be evaluated by the detector. As it can be inferred from the data, the bottleneck of the system is the detector, while the tracking algorithm is very fast. Overall, the whole system can track multiple people in the scene at a medium frame rate of 25 frames per second.

6.3 Tracking Evaluation

For the purpose of evaluating the tracking performance we adopted the CLEAR MOT metrics [3], that consists of two indexes: MOTP and MOTA. The MOTP indicator measures how well exact positions of people are estimated, while the MOTA index gives an idea of the amount of errors that are made by the tracking algorithm in terms of false negatives, false positives and mismatches. In particular, given that our ground truth does not consist of the metric positions of all persons, but of their positions inside the image, we computed the MOTP index as the average PASCAL index [9] (intersection over union of bounding boxes) of the associations between ground truth and tracker results by setting the validation threshold to 0.5. Instead, we computed the MOTA index with the following formula

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + ID_t^{sw})}{\sum_t g_t} \tag{1}$$

where fn_t is the number of ground truth people instances (for every frame) not found by the tracker, fp_t is the number of output tracks instances that do not have correspondences with the ground truth, ID_t^{sw} represents the number of times a track corresponding to the same person changes ID over time and g_t is the total number of ground truth instances present in all frames.

[6] `http://ros.org`
[7] OpenCV - `http://opencv.willowgarage.com`
[8] PCL - `http://pointclouds.org`
[9] Bayes++ - `http://bayesclasses.sourceforge.net`

In Table 2 and 3 we report, for every test sequence, the MOTP and MOTA indexes, the percentage of false positives and false negatives, the number of ID switches and the number of recoveries. With recovery we mean when a person that was lost by the Kalman-based tracking is re-identified by means of the classifier. The average precision of our tracker (MOTP) is always around 80%, while its accuracy (MOTA) slightly decreases from 93% to 89% as we increase the scenario complexity. The minimum accuracy is reached for the video that we captured while moving our platform inside an environment with obstacles that can significantly occlude people. These occlusions mainly have the effect to delay tracks initialization because we compute the HOG confidence only for not occluded clusters. Thus the false negative percentage increases and the MOTA index decreases. Nevertheless, the track initialization based on the HOG confidence makes our algorithm to totally avoid false tracks. The percentage of false positives shown in the tables are due to bounding boxes generated by real persons that are not perfectly centered on the ground truth data, thus they do not pass the PASCAL test.

Table 2. Tracking evaluation results for static tests

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**	**Rec.**
Simple traj.	79.0%	93.2%	3.8%	2.9%	0	0
Complex traj.	79.2%	96.1%	3.1%	0.7%	0	0
With obstacles	82.6%	89.7%	2.9%	7.5%	1	0

Table 3. Tracking evaluation results for moving tests

	MOTP	**MOTA**	**FP**	**FN**	**ID Sw.**	**Rec.**
Simple traj.	81.9%	92.8%	2.8%	4.4%	0	1
Complex traj.	83.9%	89.2%	4.8%	6.1%	0	1
With obstacles	84.1%	89.2%	5.3%	5.5%	0	0

As we said in Section 5, the method we use for selecting which clusters are passed to the data association algorithm is based on a simple check of clusters height from the ground plane. This approach proved to be very effective for people tracking. In fact, if only the clusters that have an HOG confidence above a threshold (e.g. -3) were used to update the tracks, the MOTA index would drop of about 30%. This is due to the fact that the HOG people detector score decreases very quickly in case of occlusion.

In general, our algorithm showed to be very robust even for tracking partially occluded people moving with complex trajectories. In fact, we correctly track 96% of every ground truth track. The only ID switch happened when a person was lost due to a sudden movement and after then the Kinect automatically adjusted its image brightness. This episode led the appearance classifier to fail, thus a new track was initialized. Apart from this case, the on-line person-specific classifier correctly managed the most part of track recoveries after a full person

occlusion. In particular, for the most complicated scenario we tested (moving platform, obstacles and complex trajectories), it allowed to track 20% more of ground truth tracks without ID switches.

In Fig. 4 we report some examples of correctly tracked frames from our test set. Different IDs are represented by different colors and the bounding box is drawn with a thick line if the algorithm estimates a person to be completely visible, while a thin line is used if a person is considered occluded. It can be noticed that our tracking system obtains very good performance even in presence of very strong occlusion (>70%). Furthermore, people are correctly classified as visible or occluded.

Fig. 4. Tracking output on some frames extracted from the test set

Fig. 5. People following test. First row: examples of tracked frames while a person is robustly followed along a narrow corridor with many lighting changes. Second row: other examples of correctly tracked frames when other people are present in the scene.

As a further test for proving the robustness and the real time capabilities of our tracking method we asked the robot to follow a particular person along a narrow corridor with many lighting changes and in a hall where many people are present. In Fig. 5 the output of the tracking algorithm is visualized for some frames collected during this experiment. In the first row it can be seen that the person is correctly tracked and followed by the platform along the corridor, while the lighting conditions considerably change. This is allowed by a good prediction of people position in ground plane coordinates, obtained with the use of the Unscented Kalman Filter that can also benefit of many available measures

thanks to the high frame rate. In the second row the tracking (and following) robustness is shown when other people walk next to the followed person or between him and the robot.

7 Conclusions and Future Works

In this paper we have presented a very fast algorithm for multi-people tracking from mobile platforms equipped with a RGB-D sensor. A robust track initialization is obtained by checking the confidence consistence of a HOG people detector and tracks are recovered after a full occlusion thanks to an online learned classifier based on features extracted from RGB histograms. A proper downsampling is applied to the raw sensor data in order to make them tractable in real time.

Tests were performed in scenarios of increasing complexity and our tracking system proved to be very effective even when dealing with strong occlusions and complex trajectories. Some limits of our system are due to the limited field of view of the RGB-D sensor we are using and the poor depth estimates over eight meters of distance.

As future works, we plan to improve our clustering method in order to correctly handle situations when people are too close or touching each other and mount more Kinect sensors for extending the field of view of our platform. Moreover, given the high real time potential of our tracking algorithm, we envision to study motion planning algorithms that can allow robots to move in a human-aware fashion and fully integrate in a populated environment.

References

1. Bajracharya, M., Moghaddam, B., Howard, A., Brennan, S., Matthies, L.H.: A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. International Journal of Robotics Research 28, 1466–1485 (2009)
2. Bellotto, N., Hu, H.: Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. Auton. Robots 28, 425–438 (2010)
3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. J. Image Video Process. 2008, 1:1–1:10 (2008)
4. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: IEEE International Conference on Computer Vision (October 2009)
5. Carballo, A., Ohya, A., Yuta, S.: People detection using range and intensity data from multi-layered laser range finders. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5849–5854 (2010)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (June 2005)
7. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: IEEE Conference on Computer Vision and Pattern Recognition 2008, pp. 1–8 (2008)

8. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Moving obstacle detection in highly dynamic scenes. In: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Piscataway, NJ, USA, pp. 4451–4458 (2009)
9. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vision 88, 303–338 (2010)
10. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, Washington, DC, USA, pp. 260–267 (2006)
11. Konstantinova, P., Udvarev, A., Semerdjiev, T.: A study of a target tracking algorithm using global nearest neighbor approach. In: Proceedings of the 4th International Conference Conference on Computer Systems and Technologies: e-Learning, New York, NY, USA, pp. 290–295 (2003)
12. Luber, M., Spinello, L., Arras, K.O.: People tracking in rgb-d data with online boosted target models. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011 (2011)
13. Martin, C., Schaffernicht, E., Scheidig, A., Gross, H.-M.: Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking. Robotics and Autonomous Systems 54(9), 721–728 (2006)
14. Mozos, O., Kurazume, R., Hasegawa, T.: Multi-part people detection using 2d range data. International Journal of Social Robotics 2, 31–40 (2010)
15. Navarro-Serment, L.E., Mertz, C., Hebert, M.: Pedestrian detection and tracking using three-dimensional ladar data. In: FSR, pp. 103–112 (2009)
16. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 (2011)
17. Satake, J., Miura, J.: Robust stereo-based person detection and tracking for a person following robot. In: Workshop on People Detection and Tracking IEEE ICRA (2009)
18. Spinello, L., Arras, K.O.: People detection in rgb-d data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011 (2011)
19. Spinello, L., Arras, K.O., Triebel, R., Siegwart, R.: A layered approach to people detection in 3d range data. In: Proc. 24th AAAI Conference on Artificial Intelligence, PGAI Track (AAAI 2010), Atlanta, USA (2010)
20. Spinello, L., Luber, M., Arras, K.O.: Tracking people in 3d using a bottom-up top-down people detector. In: IEEE International Conference on Robotics and Automation (ICRA 2011), Shanghai (2011)
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR 2001, vol. 1, pp. 511–518 (2001)