# Tracking people within groups with RGB-D data

Matteo Munaro, Filippo Basso and Emanuele Menegatti
Department of Information Engineering
University of Padova
35131 - Padova, Italy
{matteo.munaro, filippo.basso, emg}@dei.unipd.it

***Abstract*— This paper proposes a very fast and robust multi-people tracking algorithm suitable for mobile platforms equipped with a RGB-D sensor. Our approach features a novel depth-based sub-clustering method explicitly designed for detecting people within groups or near the background and a three-term joint likelihood for limiting drifts and ID switches. Moreover, an online learned appearance classifier is proposed, that robustly specializes on a track while using the other detections as negative examples.**

**Tests have been performed with data acquired from a mobile robot in indoor environments and on a publicly available dataset acquired with three RGB-D sensors and results have been evaluated with the CLEAR MOT metrics. Our method reaches near state of the art performance and very high frame rates in our distributed ROS-based CPU implementation.**

## I. 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 ([9], [27], [6]) or 3D sensors only ([20], [25], [26], [7], [21]). 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 [3], 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 [18] 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.* [10], [11] 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 by the stereo pair allowed them to reach good results in challenging scenarios, but their algorithm reached real-time performance only 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], [23]), 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 or requiring GPU implementations [2].

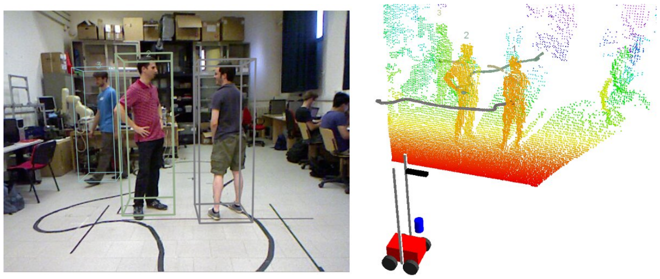


Fig. 1. Example of our system output: (left) a 3D bounding box is drawn for every tracked person on the RGB image, (right) the corresponding 3D point cloud is reported, together with people trajectories.

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. Recently, Samsung realized also a CMOS sensor capable of simultaneous color and range image capture [15], thus paving the way for a further diffusion of RGB-D sensors.

In [24] a people detection algorithm for RGB-D data is proposed, which exploits a combination of HOG and HOD descriptors. However, the whole frame is densely scanned to search for people, thus requiring a GPU implementation for being executed in real time. Also [8] relies on a dense GPU-based object detection, while [19] investigates how the usage of the people detector can be reduced using a depth-based ROI tracking. However, the obtained ROIs are again densely scanned by a GPU-based people detector.

In [17] a tracking algorithm on RGB-D data is proposed, which exploits the multi-cue people detection approach described in [24]. 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

*This research was partially supported by Telecom Italia S.p.A.*

with three RGB-D sensors.

In this work, we propose a multi-people tracking algorithm with RGB-D data for static or 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 26 frames per second with a standard CPU implementation. The main contributions are a 3D sub-clustering method that allows to efficiently detect people very close to each other or to the background, a three-term joint likelihood for limiting drifts and ID switches and an online learned appearance classifier that robustly specializes on a track while using other detections as negative examples.

The remainder of the paper is organized as follows: in Section II an overview of the two main blocks of our system is given. The detection phase is described in Section III, while Section IV details the tracking procedure and in Section V we describe the tests performed and we report the results evaluated with the CLEAR MOT metrics. Conclusions and future works are contained in Section VI.

## II. System Overview

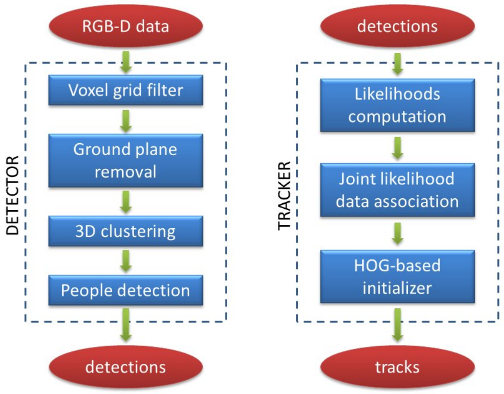


Fig. 2. Block diagram describing input/output data and the main operations performed by our detection and tracking modules.

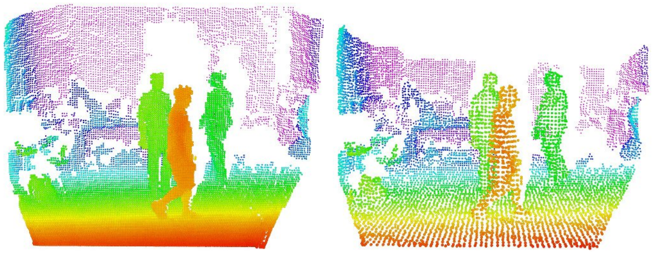
In this section, we briefly outline the two main software blocks of our people tracking system. As reported in Fig. 2, the RGB-D data are processed by a detection module that filters the point cloud data, removes the ground and performs a 3D clustering of the remaining points. Furthermore, we apply a HOG-based people detection algorithm to the RGB image of the resulting clusters in order to keep only those that are more likely to belong to the class of people. The resulting output is a set of detections that are then passed to the tracking module.

Our tracking algorithm performs detection-track association as a maximization of a joint likelihood composed by three terms: motion, color appearance and people detection confidence. For evaluating color appearance, a person classifier for every target is learned online by using features extracted from the color histogram of the target and choosing as negative examples also the other detections inside the image. The HOG confidence is also used for robustly initializing new tracks when no association with existing tracks is found.

## III. Detection

### A. Voxel grid filtering

The voxel grid filter consists of a smart down-sampling of the RGB-D point cloud. At each frame, the space is subdivided into a set of voxels (volumetric pixels) and all points inside each voxel are approximated with the coordinates of their centroid. By default, we chose the voxel size to be 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, this operation is also useful for obtaining point clouds with approximately constant density, where points density no longer depends on their distances from the sensor. In that condition the number of points of a cluster is directly related to its real size. As an example, in Fig. 3 we compare the raw point cloud of the Microsoft Kinect RGB-D sensor with the result of the voxel grid filtering when choosing the voxel size to be of 0.06m.



(a) Raw Kinect point cloud. (b) After voxel grid filtering.

Fig. 3. The effect of the voxel grid filter on Kinect 3D data.

### B. Sub-clustering groups of people

Since we make the assumption that people walk on a ground plane, our algorithm estimates and removes this plane from the point cloud provided by the voxel grid filter. We compute the plane coefficients with a RANSAC-based least square method and we remove all the inliers within a threshold distance. The ground plane equation is updated at every frame by considering as initial condition the estimation at the previous frame, thus allowing real time adaptation to small changes in the floor slope or camera oscillations typically caused by robot movements.

Once this operation has been performed, the different clusters are no longer connected through the floor, so they could be calculated by labeling neighboring 3D points on the basis of their Euclidean distances. However, this procedure can lead to two typical problems: (i) the points of a person could be subdivided into more clusters because of occlusions or some missing depth data; (ii) more persons could be merged into the same cluster because they are too close or they touch themselves or, for the same reason, a person could be clustered together with the background, such as a wall or a table.

For solving problem (i), after performing the Euclidean clustering, we merge clusters that are very near in ground plane coordinates[1], so that every person is likely to belong to only one cluster. For what concerns problem (ii), when more people are merged into one cluster, the more reliable way to detect individuals is to detect the heads, because there is a one to one person-head correspondence and heads are the body parts least likely to be occluded. Moreover, the head is usually the highest part of the human body. From these considerations we implemented the following algorithm, that detects the heads from a cluster of 3D points and segment it into sub-clusters according to the head positions:

1) for every cluster a height map[2] is created along the direction corresponding to the image $x$ axis;
2) local maxima are searched for within the height map;
3) only maxima farther than a threshold distance in ground plane coordinates are kept because people heads are not often nearer than the intimate distance [14], usually equal to 0.3m;
4) a sub-cluster is created for every remaining maximum and points nearer than the intimate distance in ground plane coordinates are associated to it;
5) sub-clusters with too few points or not enough high are discarded.

For the sub-clusters obtained we compute their HOG confidence, that is we apply a HOG people detector [9] to the part of the RGB image corresponding to the cluster theoretical bounding box, namely the bounding box that should contain the whole person, from the head to the ground. It is worth to notice that this procedure allows to obtain a more reliable HOG confidence, respect to applying the HOG detector directly to the cluster bounding box, also when a person is occluded. For the people detector, we used Dollár's implementation of HOG[3] and the same procedure and parameters described by Dalal and Triggs [9] for training the detector, thus reaching similar performance in terms of precision/recall.

In Fig. 4 we report an example of sub-clustering of a cluster that was composed by eight people very close to each other. In particular, we show: (a) the height map obtained from the original cluster as a black and white image, the final estimation of the head position as white points above the height map, the cluster segmentation into sub-clusters explained with colors and (b) the final output of the people detector on the whole image. Fig. 5 shows an example of how our sub-clustering method allows to correctly detect a person otherwise merged with the background.

## IV. Tracking

The tracking module receives as input the detections coming from one or more detection modules and solves the data association problem as the maximization of a joint likelihood encoding the probability of ground plane motion and color appearance, together with that of being a person.

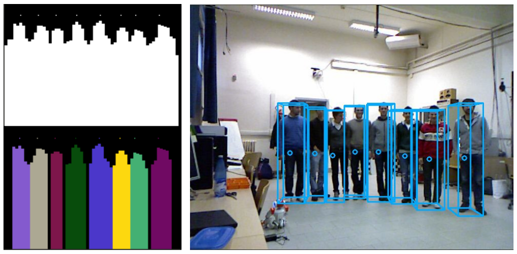

(a) Height map and segmentation. (b) People detection output.

Fig. 4. Sub-clustering of a cluster containing eight people standing very close to each other.

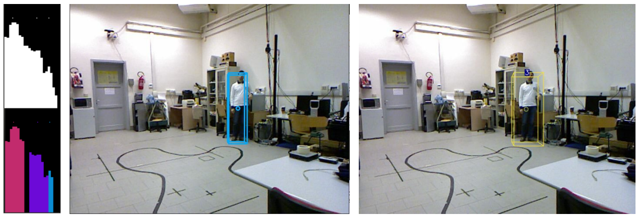

(a) (b) (c)

Fig. 5. Example of how the sub-clustering method allows to correctly detect a person otherwise merged with the background. a) Height maps, b) detector output, c) tracker output.

### A. Online classifier for learning color appearance

For every initialized track we maintain an online classifier based on Adaboost, like the one used in [13] or [17]. 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 compute the RGB color histogram of the points corresponding to the current detection associated to the track;
2) we select a set of randomized axis-aligned 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.

With this approach, the color histogram is computed only once for all the feature computations. In Fig. 6 we report the three most weighted features (parallelepipeds in the RGB color space) for each one of the three people of Fig. 1 at the initialization (first row) and after 150 frames (second row). It can be easily noticed how the most weighted features after 150 frames highly reflect the real targets colors.

For the training phase, we use as positive sample the color histogram of the target, but, instead of selecting negative examples only from randomly selected windows of the image as in [13], we consider also as negative examples the histograms calculated on the detections not associated to the current track. This approach has the advantage of selecting only the colors that really characterize the target and distinguish it from all the others. Fig. 7 clearly shows how this method increases the distance between the confidences of the correct track and the other tracks.

[1] Before doing this, we remove clusters too high with respect to the ground plane.

[2] For every bin, it contains the maximum height from the ground plane.

[3] Contained in his Matlab toolbox http://vision.ucsd.edu/~pdollar/toolbox.

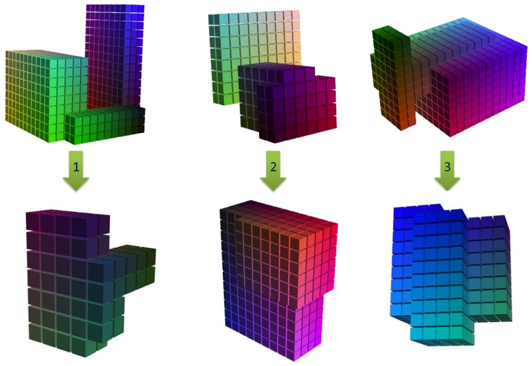
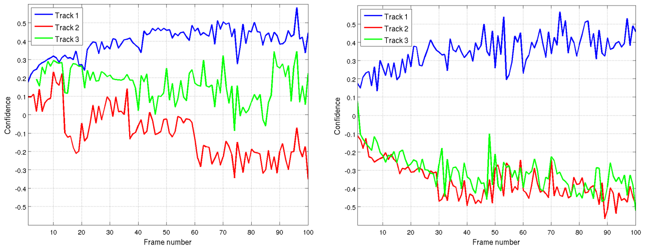


Fig. 6. From left to right: visualization of the features selected by Adaboost at the first frame (first row) and after 150 frames (second row) for the three people of Fig. 1.



(a) Random windows.

(b) Other detections and random windows.

Fig. 7. Confidence obtained by applying to the three people of Fig. 1 the color classifier trained on one of them (Track 1) for two different methods of choosing the negative examples.

### B. Three-term joint likelihood

For performing data association we use the Global Nearest Neighbor approach (solved with the Munkres algorithm), described in [16] and [3]. Our cost matrix derives from the evaluation of a three-term joint likelihood for every target-detection couple.

As motion term we compute the Mahalanobis distance of the detection position and estimated velocity from the predicted state of the track. Given that this distance is distributed as a chi-square, we use this distribution for defining a gating function for the possible associations.

An Unscented Kalman Filter is exploited to predict people positions and velocities along the two ground plane axes $(x, y)$, because it has prediction capabilities near those of a particle filter, but it is only slightly more computationally expensive than an Extended Kalman Filter [3]. As people motion model we chose a constant velocity model because it is good at managing full occlusions, as described in [3].

For modeling people appearance we add two more terms:

1) the color likelihood, that helps to distinguish between people when they are close to each other or when a person is near the background. It is provided by the online color classifier learned for every track;
2) the detector likelihood, that helps keeping the tracks on people, without drifting to walls or background objects when their colors look similar to those of the target. For this likelihood we use the confidence obtained with the HOG detector.

The joint likelihood to be maximized for every track $i$ and detection $j$ is then

$$L_{TOT}^{i,j} = L_{motion}^{i,j} \cdot L_{color}^{i,j} \cdot L_{detector}^{j}. \tag{1}$$

For simpler algebra we actually minimize the log-likelihood

$$l_{TOT}^{i,j} = -log\left(L_{TOT}^{i,j}\right) = \gamma \cdot D_{M}^{i,j} + \alpha \cdot c_{online}^{i,j} + \beta \cdot c_{HOG}^{j}, \tag{2}$$

where $D_M^{i,j}$ is the Mahalanobis distance between track $i$ and detection $j$, $c_{online}^{i,j}$ is the confidence of the online classifier of track $i$ evaluated with the histogram of detection $j$ and $c_{HOG}^{j}$ is the HOG confidence of detection $j$.

If there are unassociated detections with HOG confidence above a security threshold, new tracks are created.

## V. Experiments

### A. Indoor tracking from mobile robot

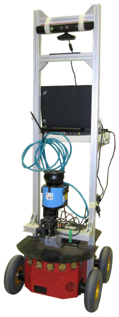


Fig. 8. The mobile platform we used for the experiments. Note that for this work we only exploit RGB-D data from a Microsoft Kinect sensor and do not make use of other sensors such as Laser Range Finder or sonars.

We present here some results obtained with our tracking system on RGB-D video sequences collected in an indoor environment with the mobile robot shown in Fig. 8. It consists of a Pioneer P3-AT platform equipped with a Microsoft Kinect sensor, which 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 pixels, for speed. It is worth to notice that this choice does not affect the accuracy achievable by our system, because of the voxel dimension we chose for the voxel grid filter we apply.

We performed tests while the robot was moving along one direction 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 (chairs, a whiteboard) 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.

For the purpose of evaluating the tracking performance we adopted the CLEAR MOT metrics [4], 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 number 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 [12] (intersection over union of bounding boxes) of the associations between ground truth and tracker results by setting the validation threshold to 0.5. We computed the MOTA index with the following formula

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + ID_t^{sw})}{\sum_t g_t} \qquad (3)$$

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.

In Table I we report, for every test sequence, the MOTP and MOTA indexes, the percentage of false positives and false negatives and the number of ID switches. The results are very good for these tests: the only ID switches are due to people who change motion direction when occluded by other people or outside the camera field of view. In Fig. 9 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.

At the Italian robotics fair *Robotica 2011*, held in Milan on the 16-19th November 2011, we tested the real time capabilities of our tracking system in crowded environments. Our robot successfully managed to detect and track people within groups and to follow a particular person within a crowded environment by means of only the data provided by the tracking algorithm. Some images collected from those days are shown in Fig. 10.

### *B. Test with the RGB-D People Dataset*

For the purpose of comparing with other state of the art algorithms we tested our tracking system with the *RGB-D People Dataset*[4] ([24], [17]), that contains about 4500 RGB-D frames acquired from three vertically mounted Kinect sensors. Even if the exact position between the sensors is not provided in the dataset webpage, we deduced it to be as shown in Fig. 11. For this test, we used three independent people detection modules (one for each Kinect), then detections have been fused at the tracking stage.

TABLE I

TRACKING RESULTS FOR TESTS WITH A MOVING ROBOT.

| | **MOTP** | **MOTA** | **FP** | **FN** | **ID Sw.** |
|---|---|---|---|---|---|
| Simple traj. | 82.2% | 95.8% | 2.5% | 1.6% | 3 |
| Complex traj. | 83.5% | 90.9% | 4.7% | 4.4% | 1 |
| With obstacles | 83.3% | 94.3% | 4.7% | 0.9% | 3 |

TABLE II

TRACKING EVALUATION WITH RGB-D PEOPLE DATASET.

| | **MOTP** | **MOTA** | **FP** | **FN** | **ID Sw.** |
|---|---|---|---|---|---|
| Ours | 73.7% | 71.8% | 7.7% | 20.0% | 19 |
| [17] | N/A | 78% | 4.5% | 16.8% | 32 |

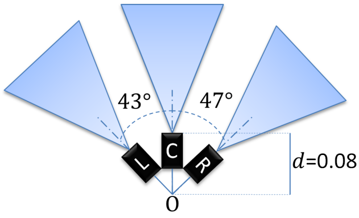


Fig. 11. Sensory setup configuration for the *RGB-D People Dataset*.

In Table II we reports the results obtained with our default system against those obtained in [17]. A video with our tracking results can be found at this link: `http://youtu.be/b70vLKFsriM` and in Fig. 12 all the estimated trajectories are shown from a top view. Our MOTA index is 71.8%, while for [17] is 78%, but the number of ID switches we obtained is considerably lower (19 instead of 32). Furthermore, the following considerations must be taken into account:

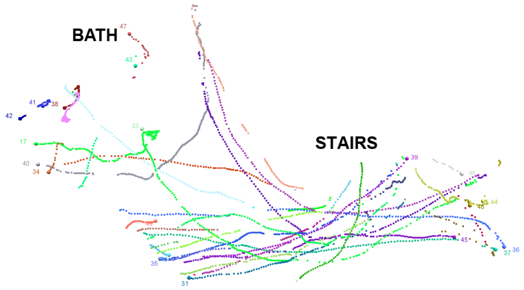


Fig. 12. Top view of the resulting estimated trajectories for the *RGB-D People Dataset*.

- 10% of people instances of this dataset appear on the stairs, but tracking people who do not walk on a ground plane was out of our scope. It is then worth to notice that half of our false negatives refer to those people;

[4] http://www.informatik.uni-freiburg.de/ spinello/RGBD-dataset.html.

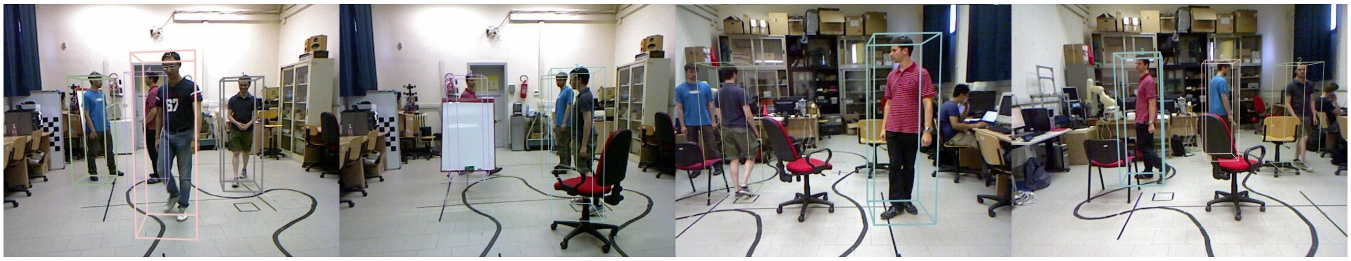


Fig. 9. Tracking output on some frames extracted from our test set collected from a moving robot.

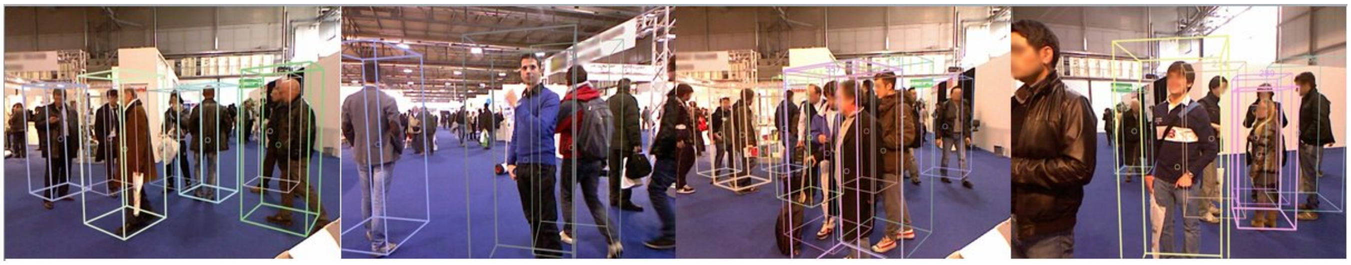
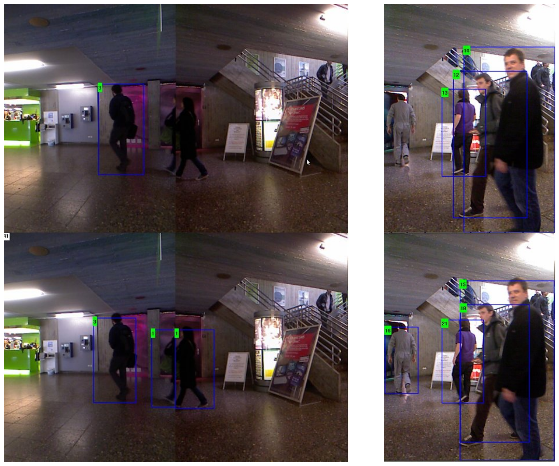


Fig. 10. Examples of tracking results from our mobile robot moving in a crowded environment.

- in the annotation provided with the dataset some people are missing even if they are visible and, when people are visible in two images they are annotated only in one of these. Our algorithm, however, correctly detects people in every image they are visible. Examples of these kinds of annotation errors are reported in Fig. 13. Actually, 90% of our false positives are due to these annotation errors, rather than to false tracks. Without these errors, the FP and MOTA values would be 0.7% and 78.9%;
- half of our ID switches are due to tracks re-initialization just after they are created because of a poor initial estimation for track velocity. If we do not use the velocity in the Mahalanobis distance for motion likelihood computation the ID switches decrease to 9, while obtaining a MOTA of 70.5%.



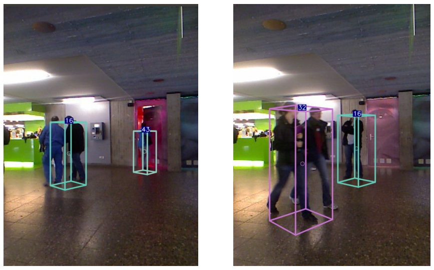
(a) Track 1

(b) Track 16

Fig. 13. Examples of people missing in the ground truth (first row) while detected by our algorithm (second row).

Given these issues, it seems that our tracking algorithm could achieve state of the art performance if evaluated in a proper way.

If we do not use the sub-clustering method described in Section III-B the MOTA index decreases by 10%, while the ID switches increase by 17. In Fig. 14 we report two examples of people merged together when not using the sub-clustering technique.



(a)

(b)

Fig. 14. Examples of people merged together when not using the sub-clustering method for the *RGB-D People Dataset*.

For this particular dataset the online classifier has not been very useful because most of the people are dressed with the same colors and the Kinect auto-brightness function makes the brightness to considerably and suddenly change among the images.

### C. Runtime perfomance

The entire system is implemented in C++ within ROS, the Robot Operating System[5], making use of highly optimized libraries for 2D computer vision [5], 3D point cloud processing [22] and bayesian estimation[6]. Our implementation

[5] http://ros.org.

[6] Bayes++ - http://bayesclasses.sourceforge.net.

TABLE III

FRAME RATES FOR PROCESSING ONE KINECT STREAM (FPS).

| CPU | Detector | Detector+Tracker |
|---|---|---|
| Intel Xeon E31225 3.10GHz | 28 | 26 |
| Intel i5-520M 2.40 GHz | 23 | 19 |

does not rely on GPU processing, so that it can be used with robots with limited computational resources.

In Table III we report the frame rates we measured for the detection algorithm and for our complete system (detection and tracking) with two computers we used for the tests: a workstation with an Intel Xeon E31225 3.10 GHz processor and a laptop with an Intel i5-520M 2.40 GHz[7]. These frame rates are achieved using Kinect QQVGA depth resolution, while they halve at Kinect VGA resolution. Our implementation exploits ROS multi-threading capabilities and is explicitly designed for real time operation and for correctly handling data coming from multiple sensors, delays and lost frames. These results suggest that a robot could use the same computer for people tracking and other tasks like navigation and self localization. As a further test, we forced our system to process in real time the 3x30Hz stream of the *RGB-D People Dataset*. Even if only 32% of the images could be processed (68% of frames was lost) it produced good results, in fact the MOTA index computed for those images was 69.3% with 24 ID switches.

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have presented a very fast and robust algorithm for multi-people tracking from static or mobile platforms equipped with a RGB-D sensor. The assumption that people move on a ground plane and a novel sub-clustering method allow to detect people even when very close to each other or to the background. A three-term joint likelihood is exploited for the data association process for limiting drifts and ID switches and an online learned appearance classifier is proposed, that robustly specializes on a track while using the other detections as negative examples.

Tests have been performed with data acquired from a mobile robot in indoor environments and on a publicly available dataset acquired with three RGB-D sensors. Our method reached near state of the art performance and our ROS implementation has been tested to track people at 26fps on a standard computer.

As a future work, we plan to improve the estimation of tracks velocity, in order to avoid some ID switches. Moreover, as our tracking stage has been explicitly designed for fusing data from multiple distributed detection modules, we plan to test it also with data coming from a RGB-D camera network or a team of robots.

## REFERENCES

[1] 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. In *International Journal of Robotics Research*, volume 28, pages 1466–1485, 2009.

[2] M. Bansal, S. Jung, B. Matei, J. Eledath, and H. Sawhney. A real-time pedestrian detection system based on structure and appearance classification. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 903–909. IEEE, 2010.

[3] N. Bellotto and H. Hu. Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters. *Auton. Robots*, 28:425–438, May 2010.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10, January 2008.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *International Conference on Computer Vision*, 2009.

[7] A. Carballo, A. Ohya, and S. Yuta. People detection using range and intensity data from multi-layered laser range finders. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5849 –5854, 2010.

[8] W. Choi, C. Pantofaru, and S. Savarese. Detecting and tracking people using an rgb-d camera via multiple detector fusion. In *IEEE ICCV Workshops*, pages 1076–1083, 2011.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 886–893, June 2005.

[10] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition 2008.*, pages 1–8, 2008.

[11] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Moving obstacle detection in highly dynamic scenes. In *Proc. of ICRA 2009*, ICRA'09, pages 4451–4458, Piscataway, NJ, USA, 2009.

[12] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88:303–338, June 2010.

[13] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–267, Washington, DC, USA, 2006.

[14] E. Hall. *The Hidden Dimension*. 1966.

[15] W. Kim, W. Yibing, I. Ovsiannikov, S. Lee, Y. Park, C. Chung, and E. Fossum. A 1.5Mpixel RGBZ CMOS Image Sensor for Simultaneous Color and Range Image Capture. In *IEEE International Solid-State Circuits Conference*, San Francisco, USA, February 2012.

[16] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proc. of the 4th international conference on Computer systems and technologies: e-Learning*, pages 290–295, New York, USA, 2003.

[17] M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[18] C. Martin, E. Schaffernicht, A. Scheidig, and H.-M. Gross. Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking. *Robotics and Autonomous Systems*, 54(9):721–728, 2006.

[19] D. Mitzel and B. Leibe. Real-time multi-person tracking with detector assisted structure propagation. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011.

[20] O. Mozos, R. Kurazume, and T. Hasegawa. Multi-part people detection using 2d range data. *Int. Journal of Social Robotics*, 2:31–40, 2010.

[21] L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian detection and tracking using three-dimensional ladar data. In *FSR*, pages 103–112, 2009.

[22] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. of ICRA 2011*, Shanghai, China, May 9-13 2011.

[23] J. Satake and J. Miura. Robust stereo-based person detection and tracking for a person following robot. In *Workshop on People Detection and Tracking IEEE ICRA*, 2009.

[24] L. Spinello and K. O. Arras. People detection in rgb-d data. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[25] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart. A layered approach to people detection in 3d range data. In *Proc. 24th AAAI Conference on Artificial Intelligence, PGAI Track*, Atlanta, USA, 2010.

[26] L. Spinello, M. Luber, and K. O. Arras. Tracking people in 3d using a bottom-up top-down people detector. In *IEEE International Conference on Robotics and Automation (ICRA'11)*, Shanghai, 2011.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR 2001*, volume 1, pages 511–518, 2001.

[7] Both computers had 4GB DDR3 memory.